AFIN 2014

The Sixth International Conference on Advances in Future Internet

ISBN: 978-1-61208-377-3

November 16 - 20, 2014

Lisbon, Portugal

AFIN 2014 Editors

Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania

Rao Mikkilineni, C3DNA Inc., USA

AFIN 2014

Foreword

The Sixth International Conference on Advances in Future Internet (AFIN 2014), held between November 16-20, 2014 in Lisbon, Portugal, continued a series of events dealing with advances on future Internet mechanisms and services.

We are in the early stage of a revolution on what we call Internet now. Most of the design principles and deployments, as well as originally intended services, reached some technical limits and we can see a tremendous effort to correct this. Routing must be more intelligent, with quality of service consideration and 'on-demand' flavor, while the access control schemes should allow multiple technologies yet guarantying the privacy and integrity of the data. In a heavily distributed network resources, handling asset and resource for distributing computing (autonomic, cloud, on-demand) and addressing management in the next IPv6/IPv4 mixed networks require special effort for designers, equipment vendors, developers, and service providers.

The diversity of the Internet-based offered services requires a fair handling of transactions for financial applications, scalability for smart homes and ehealth/telemedicine, openness for web-based services, and protection of the private life. Different services have been developed and are going to grow based on future Internet mechanisms. Identifying the key issues and major challenges, as well as the potential solutions and the current results paves the way for future research.

We take here the opportunity to warmly thank all the members of the AFIN 2014 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to AFIN 2014. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the AFIN 2014 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that AFIN 2014 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Future Internet.

We are convinced that the participants found the event useful and communications very open. We hope Lisbon provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

AFIN 2014 Chairs:
Jun Bi, Tsinghua University, China

Eugen Borcoci, University Politehnica of Bucharest, Romania
Petre Dini, Concordia University - Montreal, Canada / China Space Agency Center - Beijing,China

AFIN 2014

Committee

AFIN Advisory Chairs

Petre Dini, Concordia University - Montreal, Canada / China Space Agency Center - Beijing, China
Eugen Borcoci, University Politehnica of Bucharest, Romania
Jun Bi, Tsinghua University, China

AFIN 2014 Technical Program Committee

Rocío Abascal Mena, Universidad Autónoma Metropolitana - Cuajimalpa, México
Marie-Hélène Abel, University of Technology of Compiègne, France
Alessandro Aldini, University of Urbino "Carlo Bo", Italy
Javier A. Barria, Imperial College London, UK
Khalid Benali, LORIA - Université de Lorraine, France
Jun Bi, Tsinghua University, China
Alessandro Bogliolo, University of Urbino, Italy
Eugen Borcoci, University Politehnica of Bucharest, Romania
Christos Bouras, University of Patras and Research Academic Computer Technology Institute, Greece
Tharrenos Bratitsis, University of Western Macedonia, Greece
Chin-Chen Chang, Feng Chia University, Taiwan
Grzegorz Chmaj, University of Nevada - Las Vegas, USA
Hauke Coltzau, Fernuniversität in Hagen, Germany
Maurizio D'Arienzo, Seconda Università di Napoli, Italy
Jiangbo Dang, Siemens Corporation | Corporate Technology Princeton, USA
Guglielmo De Angelis, CNR - ISTI, Italy
Sagarmay Deb, Central Queensland University, Australia
Gayo Diallo, University of Bordeaux Segalen, France
Daniel Díaz-Sánchez, University Carlos III - Madrid, Spain
Sudhir Dixit, HP Labs India - Bangalore, India
Jonas Etzold, Fulda University of Applied Sciences, Germany
Florian Fankhauser, TU-Wien, Austria
Wu-Chang Feng, Portland State University, USA
Liers Florian, TU Ilmenau, Germany
Alex Galis, University College London, UK
Ivan Ganchev, University of Limerick, Ireland
Rosario G. Garroppo, Università di Pisa, Italy
Christos K. Georgiadis, University of Macedonia, Greece
Apostolos Gkamas, Higher Eccelesiastic Academy Vellas of Ioannina, Greece
George Gkotsis, University of Warwick, U.K.
William I. Grosky, University of Michigan-Dearborn, USA
Vic Grout, Glyndwr University, U.K.
Adam Grzech, Wrocław University of Technology, Poland
Puneet Gupta, Infosys Labs, India
Dongsoo Han, Korea Advanced Institute of Science and Technology(KAIST), Korea

Torsten Reiners, Curtin University, Australia
Jelena Revzina, Transport and Telecommunication Institute (TTI), Latvia
Simon Pietro Romano, University of Napoli 'Federico II', Italy
Gustavo Rossi, La Plata National University, Argentina
Cristian Rusu, Pontificia Universidad Católica de Valparaíso, Chile
Michele Ruta, Politecnico di Bari, Italy
Kouichi Sakurai, Kyushu Unversity, Japan
Demetrios G Sampson, University of Piraeus, Greece
Abdolhossein Sarrafzadeh, Unitec Institute of Technology - Auckland, New Zealand
Hira Sathu, Unitec Institute of Technology, New Zealand
Hiroyuki Sato, University of Tokyo, Japan
Hans D. Schotten, University of Kaiserslautern, Germany
Bernd Schuller, Jülich Supercomputing Centre, Germany
Omair Shafiq, University of Calgary, Canada
Asadullah Shaikh, Najran University, Saudi Arabia
Dimitrios Serpanos, University of Patras and ISI, Greece
Nikolay Shilov, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), Russia
Dorgham Sisalem, Iptelorg/Tekelek, Germany
Michael Sheng, The University of Adelaide, Australia
Vasco Soares, Instituto de Telecomunicações / Polytechnic Institute of Castelo Branco, Portugal
José Soler, Technical University of Denmark, Denmark
Kostas Stamos, University of Patras, Greece
Tim Strayer, BBN Technologies, USA
Maciej Szostak, Wroclaw University of Technology, Poland
Alessandro Testa, National Research Council (CNR) & University of Naples "Federico II", Italy
Brigitte Trousse, INRIA Sophia Antipolis, France
Steve Uhlig, Queen Mary, University of London, UK
J.L. van den Berg, University of Twente, The Netherlands
Rob van der Mei, CWI Centrum Wiskunde en Informatica, The Netherlands
Costas Vassilakis, University of Peloponnese, Greece
Sodel Vázquez Reyes, Universidad Autónoma de Zacatecas, Mexico
Massimo Villari, University of Messina, Italy
Maurizio Vincini, Universita' di Modena e Reggio Emilia, Italy
Jozef Wozniak, Gdańsk University of Technology, Poland
Toshihiro Yamauchi, Okayama University, Japan
Zhixian Yan, Samsung Research, USA
Chai Kiat Yeo, Nanyang Technological University, Singapore
Fan Zhao, Florida Gulf Coast University, U.S.A.

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Semantic Network Organization Based on Distributed Intelligent Managed Elements

Improving efficiency and resiliency of computational processes

Mark Burgin
UCLA
Los Angeles, USA
mburgin@math.ucla.edu

Rao Mikkilineni
C^3 DNA Inc.
Santa Clara, USA
rao@c3dna.com

***Abstract* — A new network architecture based on increasing intelligence of the computing nodes is suggested for building the semantic grid. In its simplest form, the distributed intelligent managed element (DIME) network architecture extends the conventional computational model of information processing networks, allowing improvement of the efficiency and resiliency of computational processes. This approach is based on organizing the process dynamics under the supervision of intelligent agents. The DIME network architecture utilizes the DIME computing model with non-von Neumann parallel implementation of a managed Turing machine with a signaling network overlay and adds cognitive elements to evolve super recursive information processing, for which it is proved that they improve efficiency and power of computational processes. The main aim of this paper is modeling the DIME network architecture with grid automata. A grid automaton provides a universal model for computer networks, sensor networks and many kinds of other networks.**

Keywords - semantic network; DIME network architecture; grid automaton; structural operation; connectivity; modularity; Turing O-Machine; cloud computing.

I. INTRODUCTION

Information processing networks play more and more important role in society. For instance, close to a billion hosts are connected to the Internet. The rapid rise in popularity of the Internet is due to the World Wide Web (WWW), search engines, e-mail, social networking and instant communication systems, which enable high-speed and resourceful exchanges and transformation of information, as well as provide unlimited access to a huge amount of information [1].

Recently, cloud and grid computing have been regarded as the most promising paradigms to interconnect heterogeneous commodity computing environments. To make it more efficient, the concept of the semantic web or semantic grid was introduced as a new level of the Internet and the World Wide Web. This new level is based on establishing a new form of Web content that is meaningful to computers. The Semantic Web proposes to help computers in obtaining information from the Web and using it for achieving various goals. The first step is to add metadata to Web pages making the existing World Wide Web machine comprehensible and providing machine tools to find, exchange, and to a limited extent, interpret information. Being an extension of, but not a replacement for, the World Wide Web, this approach will unleash a revolution of new possibilities.

In this paper, the distributed intelligent managed element (DIME) network architecture [2, 3, 4, 5, 6] previously discussed at the Turing Centenary Conference [7] in Manchester, is aimed at the development of semantic networks extending the conventional computational model of the network architecture. It is aimed at improving efficiency and resiliency of computational processes by organizing their evolution to model process dynamics under the supervision of intelligent agents. The computing hardware resources are combined with software functions to arrange processes and their dynamics using a network of DIMEs where each end node can be either a DIME unit or a sub-network of DIME units executing a workflow. The hardware resources are characterized by their parameters such as the required CPU, memory, network bandwidth, latency, storage throughput, IOPs and capacity. The efficiency of computation is determined by the required resources, while the expressiveness of the computational process dynamics is established by the structure of the DIME units and connecting hardware units, such as servers or routers, along with its interactions within and with the external world.

The suggested approach to the semantic web lies in provisioning of resource descriptions and ontologies to DIME agents. The agent would search through metadata that clearly identify and define what the agent needs to know. Metadata are machine-readable data that describe other data. In the Semantic Web, metadata are invisible as people read the page, but they are clearly visible to DIME agents. Metadata can also allow more complex, focused Web searches with more accurate results and interpreting these data for controlling DIME basic processors.

To achieve all these goals, it is necessary to base the entire design of the whole network of applications, as well as of the components that build the network, on a system technology with flexibility to interconnect different applications and devices from different vendors. Rigid standards may be suitable to meet a short term requirement, but in the long run, they will limit choices as it will inhibit innovation. System technology, in turn, provides efficient design methods and results in creation of better networks, which satisfy necessary requirements. All these requirements demand a new approach to application and device network design, upgrading, and maintenance.

 ISBN: 978-1-61208-377-3

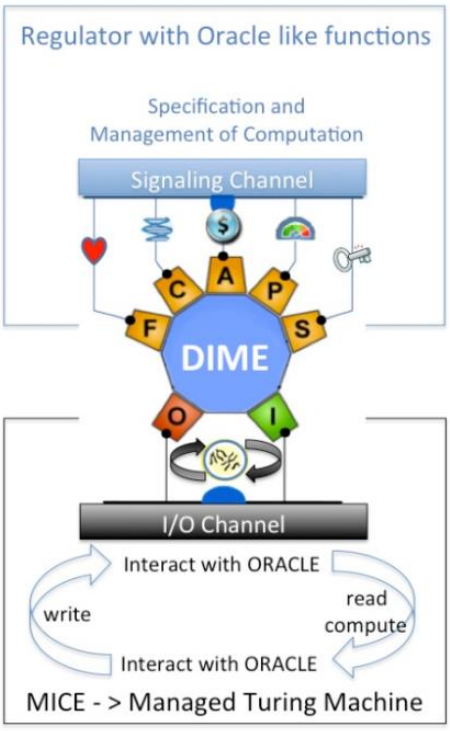


Figure 1: A Distributed Intelligent Managed Element is a managed Turing Oracle Machine endowed with a signaling network overlay for policy based DIME network management

Here, we develop tools for such a systemic network design, upgrading, and maintenance based on three principles: 1) modularity; 2) system representation of each module by grid automata; and 3) utilization of modular operations with networks, which are introduced in this paper. Modular approach means division of a complex system into smaller, manageable ones, making implementation much easier to handle.

Section II, reviews the DIME network architecture and current state of the art. Section III presents a review of the theory of Grid Automata and Grid Arrays. Section IV describes modeling DIME networks with Grid Automata, while in Section V, some conclusions are considered and directions for future work are suggested.

II. DIME NETWORK ARCHITECTURE

The DIME network architecture introduces three key functional constructs to enable process design, execution and management to improve both resiliency and efficiency of computer networks [2, 3, 5].

1) Machines with an Oracle

Executing an algorithm, the DIME basic processor *P* performs the {read -> compute -> write} instruction cycle or its modified version the {interact with a network agent -> read -> compute -> interact with a network agent -> write} instruction cycle. This allows the different network agents to influence the further evolution of computation, while the computation is still in progress. We consider three types of network agents:

(a) A DIME agent.
(b) A human agent.
(c) An external computing agent.

It is assumed that a DIME agent knows the goal and intent of the algorithm (along with the context, constraints, communications and control of the algorithm) the DIME basic processor is executing and has the visibility of available resources and the needs of the basic processor as it executes its tasks. In addition, the DIME agent also has the knowledge about alternate courses of action available to facilitate the evolution of the computation to achieve its goal and realize its intent. Thus, every algorithm is associated with a blueprint (analogous to a genetic specification in biology), which provides the knowledge required by the DIME agent to manage the process evolution. An external computing agent is any computing node in the network with which the DIME unit interacts.

2) Blue-print or policy managed fault, configuration, accounting, performance and security monitoring and control

The DIME agent, which uses the blueprint to configure, instantiate, and manage the DIME basic processor executing the algorithm uses concurrent DIME basic processors with their own blueprints specifying their evolution to monitor the vital signs of the DIME basic processor and implements various policies to assure non-functional requirements such as availability, performance, security and cost management while the managed DIME basic processor is executing its intent. Figure 1 shows the DIME basic processor and its DIME agent, which manages it using the knowledge provided by the blueprint [3, 7].

3) DIME network management control overlay over the managed Turing oracle machines

In addition to read/write communication of the DIME basic processor (the data channel), other DIME basic processors communicate with each other using a parallel signaling channel. This allows the external DIME agents to influence the computation of any managed DIME basic processor in progress based on the context and constraints. The external DIME agents are DIMEs themselves. As a result, changes in one computing element could influence the evolution of another computing element at run time without halting its Turing machine executing the algorithm. The signaling channel and the network of DIME agents can be programmed to execute a process, the intent of which can be specified in a blueprint. Each DIME basic processor can have its own oracle managing its intent, and groups of managed DIME basic processors can have their own domain managers implementing the domain's intent to execute a process. The management DIME agents specify, configure, and manage the sub-network of DIME units by monitoring and executing policies to optimize the resources while delivering the intent.

 ISBN: 978-1-61208-377-3

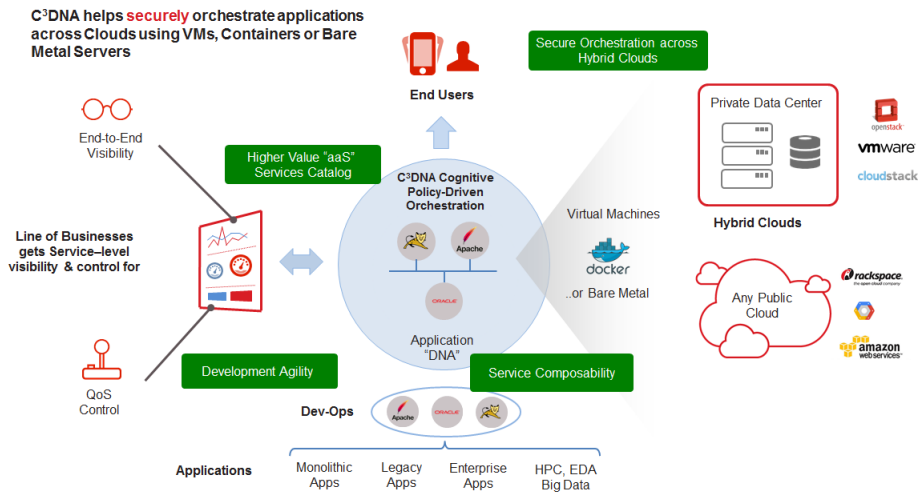


Figure 2: Implementation architecture of a Web application workflow using a physical server network and a cloud using Virtual Machines

Figure 2 shows the DIME network implementation architecture for a process with different hardware, functions and an evolving structure used to attaining the intent of the process.

This architecture has following benefits from current architectures deploying virtual machines to provide cloud services such as self-provisioning, self-repair, auto-scaling and live-migration:

1. Using DNA, same cloud services can be provided at application and workflow group level across physical or virtual servers. The mobility of applications comes from utilization of the policies implemented to manage the intent through the signaling network overlay over the managed computer network. Applications are moved into static Virtual Machines with given service levels provisioned.

2. Scheduling, monitoring, and managing distributed components and groups with policies at various levels de-couple the application/workflow management from underlying distributed infrastructure management systems. The vital signs (cpu, memory, bandwidth, latency, storage IOPs, throughput and capacity) are monitored and managed by DIMEs, which are functioning similar to the Turing o-machines.

While implementing the monitoring and management of the DIME agent, the DIME network monitors and manages its own vital signs and executing various policies to assure availability, performance and security. At each level in the hierarchy, a domain specific task or workflow is executed to implement a distributed process with a specific intent. In figure 2, each web component has its own policies and the group has the service level policies that define its availability, performance and security. Based on policies, the elements are replicated or reconfigured to meet the resource requirements based on monitored behavior.

In essence, the DIME computing model infuses sensors and actuators connecting the DIME basic processor with the DIME agent to manage the DIME basic processor and its resources based on the intent, interactions and available resources. Policy managers are used to configure, monitor and manage the basic processor's intent. The DIME network architecture has been successfully implemented using 1) a Linux operating system and 2) a new native operating system called parallax [2, 3]. More recently, a product based on DIME network architecture was used to implement auto-failover, auto-scaling, and live-migration of a web based application deployed on distributed servers with or without virtualization [8]. In this paper we model the DIME network architecture with grid automata. A grid automaton [9] is shown to be more efficient and expressive than the von Neumann implementation of the Turing Machine. In the next section, we review the Grid Automata and Grid Arrays.

III. GRID AUTOMATA AND GRID ARRAYS

All computer and embedded system networks, as well as their software, are grid arrays in the sense of [9, 10, 11]. The Internet is a grid array. The Grid [12, 13] is also a grid array. Computing grid arrays consist of computing devices connected by some ties, e.g., channels. Grid automata provide theoretical models for grid arrays and thus, for computer software, hardware, networks and many other systems. At first, we give an informal definition.

Definition 1. A grid automaton is a system of abstract automata and their networks, which are situated in a grid, are called nodes, are optionally connected and interact with one another.

The difference is that a grid array consists of real (physical) information processing systems and connections between them, while a grid automaton consists of abstract automata as its nodes. Nodes in a grid automaton can be finite automata, Turing machines, vector machines, array machines, random access machines, inductive Turing machines, and so on. Even more, some of the nodes can be also grid automata.

 ISBN: 978-1-61208-377-3

Translating this definition into the mathematical language, two types of grid automata - *basic grid automata* and *grid automata with ports* – are considered.

The basic idea of interacting processes is for a transmitting process to send a message using a port and for the receiving process to get the message from another port. To formalize this structure, we assume, as it is often true in reality, that connections are attached to automata by means of ports. Ports are specific automaton elements through which data come into (*input ports* or *inlets*) and send outside the automaton (*output ports* or *outlets*). Thus, any system P of ports is the union of its two (possibly) disjoint subsets $P = P_{in} \cup P_{out}$ where P_{in} consists of all inlets from P and P_{out} consists of all outlets from P. If in the real system, there are ports that are both inlets and outlets, in the model, we split them, i.e., represented such ports as pairs consisting of an input port and an output port. There are different other types of ports. For instance, contemporary computers have parallel and serial ports. Ports can have inner structure, but in the first approximation, it is possible to consider them as elementary units.

We also assume that each connection is directed, i.e., it has the beginning and end. It is possible to build bidirectional connections from directed connections.

Let us consider a class of automata **B** with ports of types **T** and a class of connections/links **L** that can be connected to automata from **B**.

Definition 2. A (*port*) *grid automaton* $\boldsymbol{G}$ over the collection (**B**, **P**, **L**), which is called *accessible hardware*, is the system

$\boldsymbol{G} = (A_G, P_G, C_G, \boldsymbol{p}_{IG}, \boldsymbol{c}_G, \boldsymbol{p}_{EG})$

that consists of three sets and three mappings:

- A_G is the set of all automata from $\boldsymbol{G}$, assuming $A_G \subseteq \mathbf{B}$;
- C_G is the set of all links from $\boldsymbol{G}$, assuming $C_G \subseteq \mathbf{L}$;
- $P_G = P_{IG} \cup P_{EG}$ (with $P_{IG} \cap P_{EG} = \varnothing$) is the set of all ports of G, assuming $P_G \subseteq \mathbf{P}$, where P_{IG} is the set of all ports (called *internal ports*) of the automata from A_G , and P_{EG} is the set of *external ports* of G, which are used for interaction of G with different external systems;
- $\boldsymbol{p}_{IG} : P_{IG} \to A_G$ is a total function, called the *internal port assignment function*, that assigns ports to automata;
- $\boldsymbol{c}_G : C_G \to (P_{IGout} \times P_{IGin}) \cup P'_{IGin} \cup P''_{IGout}$ is a (eventually, partial) function, called the *port-link adjacency function*, that assigns connections to ports where P'_{IGin} and P''_{IGout} are disjunctive copies of P_{IGin} ;
- $\boldsymbol{p}_{EG} : P_{EG} \to A_G \cup P_{IG} \cup C_G$ is a function, called the *external port assignment function*, that assigns ports to different elements from G.

To have meaningful assignments of ports, the port assignment functions $\boldsymbol{p}_{IG}$ and $\boldsymbol{p}_{EG}$ have to satisfy some additional conditions.

Examples:

1. The screen of a computer monitor is an output port. Such a screen can be also treated as a system of output ports (pixels).
2. The mouse of a computer is an input port. It can be also treated as a system of input ports.
3. The touch screen of a computer is an input port. Such a screen can be also treated as a system of input ports.

Definition 3. A *basic grid automaton* $\boldsymbol{A}$ over the collection (**B**, **L**), which is called *accessible hardware*, is a system $\boldsymbol{A} = (A_{\boldsymbol{A}}, C_{\boldsymbol{A}}, \boldsymbol{c}_{\boldsymbol{A}})$ that consists of two sets and one mapping:

- the set A_A is the set of all automata from $\boldsymbol{A}$, assuming $A_{\boldsymbol{A}} \subseteq \mathbf{B}$;
- the set C_A is the set of all connections/links from $\boldsymbol{A}$, assuming $C_{\boldsymbol{A}} \subseteq \mathbf{L}$;
- the mapping $\boldsymbol{c}_A$: $C_A \to A_A \times A_A \cup A'_A \cup A''_A$, which is a (variable) function, called the *node-link adjacency function*, which assigns connections to nodes where A'_A and A''_A are disjunctive copies of A_A .

There are different types of connections. For instance, computer networks links or connections are implemented on a variety of different physical media, including twisted pairs, coaxial cable, optical fiber, and space.

It is possible to group connections in grid arrays and grid automata into three main types:

1. *Simple connections* that are not changing deliberately transmitted data and themselves when the automaton or array is functioning.
2. *Transformable connections* that may be changed when the automaton or array is functioning.
3. *Processing connections* that can transform transmitted data.

A grid automaton G is described by three grid characteristics and three node characteristics.

The grid characteristics are:

1. The *space organization* or *structure* of the grid automaton G.

This space structure may be in the physical space, reflecting where the corresponding information processing systems (nodes) are situated, it may be the system structure defined by physical connections between the nodes, or it may be a mathematical structure defined by the geometry of node relations. System structure is so important in grid arrays that in contemporary computers connections between the main components are organized as a specific device, which is called the computer bus. In a computer or on a network, a bus is a transmission path in form of a device or system of devices on which signals are dropped off or picked up at every device attached to the line.

There are three kinds of space organization of a grid automaton: *static structure* that is always, the same; *persistent dynamic structure* that eventually changes between different cycles of computation; and *flexible dynamic structure* that eventually changes at any time of computation. Persistent Turing machines [14] have persistent dynamic structure, while reflexive Turing machines [15]

 ISBN: 978-1-61208-377-3

have flexible dynamic structure and perform emergent computations [16].

2. The *topology* of *G* is determined by the type of the node neighborhood and is usually dependent on the system structure of *G*.

A natural way to define a neighborhood of a node is to take the set of those nodes with which this node directly interacts. In a grid, these are often, but not always, the nodes that are physically the closest to the node in question.

For example, if each node has only two neighbors (right and left), it defines linear topology in *G*. When there are four nodes (upper, below, right, and left), the *G* has two-dimensional rectangular topology.

However, it is possible to have other neighborhoods. For instance, consider linear cellular automata in which the neighborhood of each cell has the radius $r > 1$ [9]. It means that *r* cells from each side of a given cell directly influence functioning of this cell.

3. The *dynamics* of *G* determines by what rules its nodes exchange information with each other and with the environment of *G*.

For example, when the interaction of Turing machines in a grid automaton *G* is determined by a Turing machine, then *G* is equivalent to a Turing machine. At the same time, when the interaction of Turing machines in a grid automaton *G* is random, then *G* is much more powerful than any Turing machine.

The node characteristics are:

1. The *structure* of the node. For example, one structure determines a finite automaton, while another structure is a Turing machine.
2. The *external dynamics* of the node determines interactions of this node.

According to this characteristic there are three types of nodes: *accepting nodes* that only accept or reject their input; *generating nodes* that only produce some input; and *transducing nodes* that both accept some input and produce some input. Note that nodes with the same external dynamics can work in grids with various dynamics.

3. The *internal dynamics* of the node determines what processes go inside this node.

For example, the internal dynamics of a finite automaton is defined by its transition function, while the internal dynamics of a Turing machine is defined by its rules. Differences in internal dynamics of nodes are very important because a change in producing the output allows us to go from conventional Turing machines to much more powerful inductive Turing machines of the first order [17].

Representation of grid automata without ports called basic grid automata is the first approximation to a general network model [9, 1], while representation of grid automata with ports is the second (more exact) approximation. In some cases, it is sufficient to use grid automata without ports, while in other situations, to build an adequate, flexible and efficient model of a network, we need automata with ports. Usually, basic grid automata are used when the modeling scale is big, i.e., at the coarse-grain level, while port grid automata are used when the modeling scale is small and we need a fine-grain model.

Neural networks, cellular automata, systolic arrays, and Petri nets are special kinds of grid automata [9]. However, grid automata provide computer science with much more flexibility, expressive power and correlation with real computational and communication systems than any of these models. In comparison with cellular automata, a grid automaton can contain different kinds of automata as its nodes. For example, finite automata, Turing machines and inductive Turing machines can belong to one and the same grid. In comparison with systolic arrays, connections between different nodes in a grid automaton can be arbitrary like connections in neural networks. In comparison with neural networks and Petri nets, a grid automaton contains, as its nodes, more powerful machines than finite automata. An important property of grid automata is a possibility to realize hierarchical structures, that is, a node can be also a grid automaton. In grid automata, interaction and communication becomes as important as computation. This peculiarity results in a variety of types of automata, their functioning modes, and space organization.

Internal ports of a port grid automaton *B* to which no links are attached are called *open*. External ports of a port grid automaton *B* to which no links or automata are attached are called *free*. External ports of a port grid automaton *B*, being always open, are used for connecting *B* to some external systems.

All ports of a grid automaton are divided into three classes: *input ports*, which can only accept information; *output ports*, which can only transmit information; and *mixed ports*, which can accept and transmit information (in the form of signals or symbols).

This typology of ports, as is used in the general case of information processing systems [9], induces the following classification of grid automata:

1. Grid automata without input and output (called *closed grid automata*).
2. Grid automata with input (called *closed from the right* or *open from the left grid automata*).
3. Grid automata with output (called *closed from the left* or *open from the right grid automata*).

Grid automata with both input and output (called *open grid automata*).

IV. Modeling DIME Networks with Grid Automata

In the context of grid automata, a DIME network is represented by a grid automaton with such nodes as DIME units, servers, routers, etc.

Each DIME unit is modeled by a basic automaton A with an Oracle O. The automaton A models the DIME basic processor P, while the Oracle O models the DIME agent DA. Turing machines with Oracles, inductive Turing machines with Oracles, limit Turing machines with Oracles [15], and evolutionary Turing machines with Oracles [19] are examples of such an automaton A with an Oracle O.

 ISBN: 978-1-61208-377-3

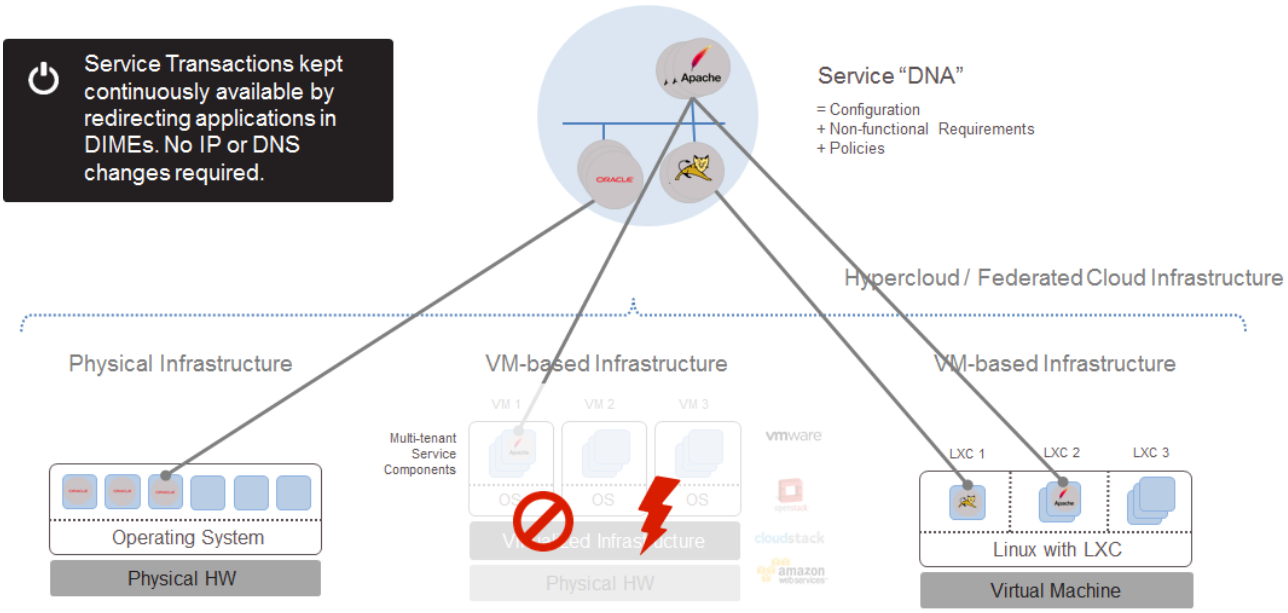


Figure 3: A Web Service Workflow Deployed in a physical server and providing mobility a Virtual server

The Oracle O in a DIME unit knows the intent of the algorithm (along with the context, constraints, communications and control of the algorithm) the basic automaton A is executing under its influence and has the visibility of available resources and the needs of the automaton A as it executes its function. In addition, the Oracle also has the knowledge about alternate courses of action available to facilitate the evolution of the computation to achieve its intent. Thus, every algorithm is associated with a blueprint (analogous to a genetic specification in biology), which can provide the knowledge required by an Oracle to manage its evolution.

In addition to read/write communication of the basic automaton (the data channel), the Oracles manage different basic automata communicating with each other using a parallel signaling channel. This allows the external Oracles to influence the computation of any managed basic automaton in progress based on the context and constraints just as a Turing Oracle is expected to do.

The Oracle uses the blueprint to configure, instantiate, and manage the automaton A executing the algorithm. Utilization of concurrent automata in the network with their own blueprints specifying their evolution to monitor the vital signs of the DIME basic automaton and to implement various policies allows the Oracle to assure non-functional requirements such as availability, performance, security and cost management, while the managed DIME basic automaton is executing its task to achieve its goal and realize its intent.

The external Oracles represent DIME agents, allowing changes in one computing element influence the evolution of another computing element at run time without stopping its basic automaton executing the algorithm. The signaling channel and the network of the Oracles can be programmed to execute a process whose intent itself can be specified in a blueprint. Each basic automaton can have its own Oracle managing its intent, and groups of managed basic automata can have their own domain managers implementing the domain's intent to execute a process. The management Oracles specify, configure and manage the sub-network of DIMEs by monitoring and executing policies to optimize the resources while delivering the intent. The DIME network implementing the Oracles is itself managed by monitoring its own vital signs and executing various FCAPS policies to assure availability, performance and security.

An Oracle is modeled by an abstract automaton that has higher computational power and/or lower computational complexity than the basic automaton it manages. For instance, the Oracle can be an inductive Turing machine, while the basic automaton is a conventional Turing machine. It is proved that inductive Turing machines have much higher computational power and lower complexity than conventional Turing machine [9].

DIME agents possess a possibility to infer new data and knowledge from the given information. Inference is one of the driving principles of the Semantic Web, because it will allow us to create software applications quite easily. For the Semantic Web applications, DIME agents need high expressive power to help users in a wide range of situations. To achieve this, they employ powerful logical tools for making inferences. Inference abilities of DIME agents are developed based on mathematical models of these agents in the form of inductive Turing machines, limit Turing machines [9] and evolutionary Turing machines [18, 19].

Figure 3 shows a workflow DNA of a web application running on a physical infrastructure that has policies to manage auto-failover by moving the components when the vital signs being monitored at various levels are affected. For

 ISBN: 978-1-61208-377-3

example if the virtual machine in the middle server fails, the service manager at higher level detects it and replicates the components in another server on the right and synchronizes the states of the components based on consistency policies.

V. CONCLUSION

Three innovations are introduced, namely, the parallel monitoring of vital signs (cpu, memory, bandwidth, latency, storage IOPs, throughput and capacity) in the DIME, signaling network overlay to provide run-time service management and machines with Oracles in the form of DIME agents. This allows interruption for policy management at read/write in a file/device allow self-repair, auto-scaling, live-migration and end-to-end service transaction security with private key mechanism independent of infrastructure management systems controlling the resources and thus, provide freedom from infrastructure and architecture lock-in. The DIME network architecture puts the safety and survival of applications and groups of applications delivering a service transaction first using secure mobility across physical or virtual servers. It provides information for sectionalizing, isolating, diagnosing and fixing the infrastructure at leisure. The DIME network architecture therefore makes possible reliable services to be delivered on even not-so-reliable infrastructure. Modeling this architecture by grid automata allows researchers to study properties and critical parameters of semantic networks and provides means for optimizing these parameters. Future work will investigate specific predictions that can be made from the theory for a specific DIME network execution and compare the resiliency and efficiency using both recursive and super-recursive implementations.

ACKNOWLEDGMENT

Rao Mikkilineni thanks Giovanni Morana and Ian Seylor for implementing the DIME network architecture and Vijay Sarathy, Nishan Sathyanarayan and Paul Camacho from C^3 DNA Inc., for making it an enterprise-class product.

REFERENCES

[1] N. Olifer and V. Olifer, Computer networks: Principles, technologies and protocols for network design, New York:Wiley, 2006.

[2] R. Mikkilineni, Designing a new class of distributed systems. New York: Springer, 2011.

[3] R. Mikkilineni, G. Morana and I. Seyler, "Implementing distributed, self-Managing computing services infrastructure using a scalable, parallel and network-centric computing model." In Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice, ed. M. Villari, I. Brandic and F. Tusa, 57-78, 2012.

Accessed September 05, 2014. doi:10.4018/978-1-4666-1631-8.ch004.

[4] R. Mikkilineni, "Architectural resiliency in distributed computing," International Journal of Grid and High Performance Computing (IJGHPC) 4, 2012.

Accessed (September 05, 2014), doi:10.4018/jghpc.2012100103.

[5] R. Mikkilineni, G. Morana, D. Zito, and M. Di Sano, "Service virtualization using a non-von Neumann parallel, distributed, and scalable computing model," Journal of Computer Networks and Communications, vol. 2012, Article ID 604018, 10 pages, 2012. doi:10.1155/2012/604018.

[6] R. Mikkilineni, "Going beyond computation and its limits: Injecting cognition into computing." Applied Mathematics 3, pp. 1826-1835, 2012.

[7] R. Mikkilineni, A. Comparini and G. Morana, "The Turing O-Machine and the DIME network architecture: Injecting the architectural resiliency into distributed computing, In Turing-100. The Alan Turing Centenary, (Ed.) Andrei Voronkov, EasyChair Proceedings in Computing, Volume 10, pp. 239-251, 2012.

http://dx.doi.org/10.1155/2012/604018

[8] R. Mikkilineni and G. Morana, "Infusing cognition into distributed computing: A new approach to distributed datacenters with self-managing services" Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2014 23rd IEEE International Conference, June 2011.

[9] M. Burgin, Super-recursive Algorithms, New York: Springer, 2005.

[10] M. Burgin, From Neural networks to Grid automata, in Proceedings of the IASTED International Conference "Modeling and Simulation", Palm Springs, California, 2003

[11] M. Burgin, Cluster computers and Grid automata, in Proceedings of the ISCA 17th International Conference "Computers and their applications", International Society for Computers and their Applications, Honolulu, Hawaii, pp. 106-109, 2003.

[12] I. Foster, "Computational Grids," in The Grid: Blueprint for a future computing infrastructure, San Francisco, CA,: Morgan Kauffman, pp. 15-52, 1998.

[13] M. L. Bote-Lorenzo, Y.A.Dimitriadis and E. Gómez-Sánchez, Grid characteristics and uses: a grid definition, in First European Across Grids conference, LNCS 2970, pp. 291–298, 2004.

[14] D. Goldin and P. Wegner, Persistent Turing Machines, Brown University Technical Report, 1988.

[15] M. Burgin, "Reflexive Calculi and Logic of Expert Systems", in *Creative processes modeling by means of knowledge bases*, Sofia, pp. 139-160, 1992.

[16] J. P. Crutchfield and M. Mitchell, "Evolution of Emergent Computation" Computer Science Faculty Publications and Presentations. Paper 3, 1995.

http://pdxscholar.library.pdx.edu/compsci_fac/3

[17] M. Burgin, "Inductive Turing machines," Notices of the Academy of Sciences of the USSR, 270 N6 pp. 1289-1293, 1983. (translated from Russian).

[18] M. Burgin and E. Eberbach, "On foundations of evolutionary computation: an evolutionary automata approach," in *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies* (Hongwei Mo, Ed.), IGI Global, Hershey, Pennsylvania, pp. 342-360, 2009.

[19] M. Burgin and E. Eberbach, "Evolutionary Automata: Expressiveness and Convergence of Evolutionary Computation," Computer Journal, v. 55, No. 9 pp. 1023-1029, 2012.

 ISBN: 978-1-61208-377-3

Interconnected Multiple Software-Defined Network Domains with Loop Topology

Jen-Wei Hu
National Center for High-performance Computing &
Institute of Computer and Communication Engineering
NARLabs & NCKU
Tainan, Taiwan
hujw@narlabs.org.tw

Chu-Sing Yang
Institute of Computer and Communication Engineering
NCKU
Tainan, Taiwan
csyang@mail.ee.ncku.edu

Te-Lung Liu
National Center for High-performance Computing
NARLabs
Tainan, Taiwan
tlliu@narlabs.org.tw

***Abstract*—With the trends of software-defined networking (SDN) deployment, all network devices rely on a single controller will create a scalability issue. There are several novel approaches proposed in control plane to achieve scalability by dividing the whole networks into multiple SDN domains. However, in order to prevent broadcast storm, it is important to avoid loops in connections with OpenFlow devices or traditional equipments. Therefore, one SDN domain can only have exactly one connection to any other domains, which will cause limitation when deploying SDN networks. Motivated by this problem, we propose a mechanism which is able to work properly even the loops occurred between any two controller domains. Furthermore, this mechanism can also manage link resources more efficiently to improve the transfer performance. Our evaluation shows that the transmissions between hosts from different areas are guaranteed even if the network topology contains loops among multiple SDN domains. Moreover, the proposed mechanism outperforms current method in transferring bandwidth.**

***Keywords-Software-defined networking; OpenFlow; multiple domains; loop topology*.**

I. INTRODUCTION

During the last decades, numbers of innovative protocols are proposed by researchers in network area. However, it is hard to speed up the innovation because network devices are non-programmable. The software defined networking (SDN) approach is a new paradigm that separates the high-level routing decisions (control plane) from the fast packet forwarding (data plane). Making high-speed data plane still resides on network devices while high-level routing decisions are moved to a separate controller, typically an external controller. OpenFlow [1] is the leading protocol in SDN, which is an initiative by a group of people at Stanford University as part of their clean-slate program to redefine the Internet architecture. When an OpenFlow switch receives a packet it has never seen before, for which it has no matching flow entries, it sends this packet to the controller. The controller then makes a decision on how to handle this packet. It can drop the packet, or it can add a flow entry directing the switch on how to forward similar packets in the future.

Moving local control functionalities to remote controllers brings numerous advantages, such as device independency, high flexibility, network programmability, and the possibility of realizing a centralized network view [2]. However, with the number and size of production networks deploying OpenFlow equipments increases, there have been increasing concern about the performance issues, especially scalability [3].

The benchmarks on NOX [7] showed it could only handle 30,000 flow installs per second. However, in [2][4][5][6], authors mention fully physically centralized control is inadequate because relying on a single controller for the entire network might not be feasible. In order to alleviate the load of controller and achieve more scalability, there are several literatures proposed their solutions. DevoFlow [8] which addresses this problem by proposing mechanisms in data plane (e.g., switch) with the objective of reducing the workload towards the controller [6]. In contrast to request reducing in data plane, the other way is to propose a distributed mechanism in control plane. A large-scale network should be divided into multiple SDN domains, where each domain manages a relatively small portion of the whole network, such like that many data centers may be located on different areas for improving network latency. However, if separating to multiple SDN domains, we will lose the consistent centralized control. Currently, there is no protocol for solving this issue [9]. Thus, there are some proposed frameworks [4][5][6] in which create a specific controller to collect information (e.g., states, events, etc.) from multiple domain controllers. They all focus on solving controller scalability issues and facilitating a consistent centralized control among multiple controller domains.

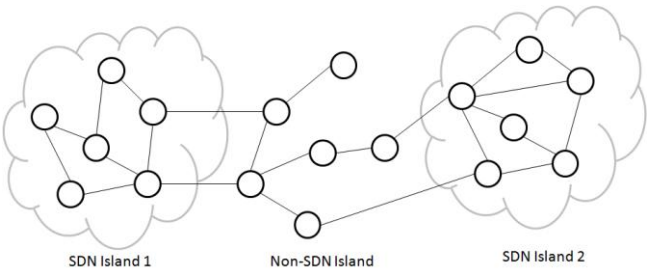


Figure 1. Looped example of a topology with SDN devices and traditional equipments.

 ISBN: 978-1-61208-377-3

For increasing reliability and transmission rate, multiple links are often deployed between nodes in ordinary network design, which practically create loops in topology. The loop leads to broadcast storms as broadcasts are forwarded by switches out of every port, the switches will repeatedly rebroadcast the broadcast packets flooding the network. Since the Layer 2 header does not support a time to live (TTL) value, if a packet is sent into a looped topology, it can loop forever and bring down the entire network. Border gateway protocol (BGP) can handle the loop topology in current Internet, but it is designed based on Layer 3. Thus, BGP still cannot prevent broadcast storm. In order to tackle broadcast storm issue, the spanning tree protocol (STP) is usually used to only allow broadcast packets to be delivered through the STP tree. This design avoids broadcast storm but reduces the overall utilization of the links as a consequence. When in a single SDN domain, controller has the complete knowledge over the entire network topology, thus the tree can be easily built. However, when the SDN network has been split into SDN islands, with the traditional network in between, the controller no longer knows the complete topology, resulting in the inability to build an effective tree, as shown in Figure 1. In this case, the existence of the loop may block the communication between two islands because the broadcast packets transmitted in the topology would mislead the controller to believe that the two hosts in communication are from the same island, thus wrong flow entries are incorporated. The situation will become even more complex in multiple SDN domains. In this paper, we focus on Layer 2 and propose a mechanism which can work properly even the loops occurred between any two SDN domains. Furthermore, this mechanism can also more efficiently in using link resources to improve the transfer performance.

The remainder of the paper is organized as follows. Section 2 presents a brief review of relevant research works focus on solving scalability issue in control plane. In Section 3, we define the preliminaries which will be used in proposed mechanism. Then, we briefly describe proposed mechanism for solving the loop limitation between multiple SDN domains and traditional networks in Section 4. In Section 5, we evaluate our mechanism in a real environment among multiple SDN domains and experiments are reported and compared with the current approach. Finally, the paper is concluded.

II. RELATED WORKS

Onix [5] is a control plane platform, which was designed to enable scalable control applications. It is a distributed instance with several control applications installed on top of control plane. In addition, it implements a general API set to facilitate access the Network Information Base (NIB) data structure from each domain. However, authors mention this platform does not consider the inter-domain network control due to the control logic designer needs to adapt the design again when changed requirements.

HyperFlow [4] is a distributed event-based control plane for OpenFlow. It facilitates cross-controller communication by passively synchronizing network-wide view among OpenFlow controllers. They develop a HyperFlow controller application and use an event propagation system in each controller. Therefore, each HyperFlow controller acts as if it is controlling the whole network. In addition, each HyperFlow controller processes and exchanges these self-defined events and the performance gets poor when the number of controllers grows [4].

HyperFlow and Onix assume that all applications require the network-wide view; hence, they cannot be of much help when it comes to local control applications. In Kandoo [6], authors design and implement a distributed approach to offload control applications over available resources in the network with minimal developer intervention without violating any requirements of control applications. Kandoo creates a two-level hierarchy for control planes. One is local controller which executes local applications as close as possible to switches in order to process frequent requests, and the other is a logically centralized root controller runs non-local control applications. It enables to replicate local controllers on demand and relieve the load on the top layer, which is the only potential bottleneck in terms of scalability.

The above proposals focus on network control, which proactively create the inter-domain links. Thus, these approaches are able to provision cross-domain paths but the complexity of maintaining global proactive flow rules can be minimized. However, loops may form between controller domains and need to be dealt with carefully. To meet this requirement, we develop a mechanism which dynamically forwards packets in the reactive way and solves the loop limitation between multiple controller domains and traditional networks.

III. PRELIMINARIES

We assume the network topology as an undirected graph $G = (V, E)$, where $V = \{\nu_1, \dots, \nu_n\}$ is a finite set, the elements of which are called vertices, and $E = \{e_1, \dots, e_n\} \subset V \times V$ is a finite set, the elements of which are called edges, where each edge e_k can be represented by (ν_i, ν_j), with $\nu_i \neq \nu_j$.

In order to limit the propagation of edge information that is only relevant in certain portions of the network and to improve scalability, we group vertices into structures called areas, denoted as A. Each vertex $\nu \in V$ is assigned a label $L(\nu) = l_k$ that is taken from a set L, describing the area that ν belongs to. Each area has a controller which is charged to coordinate vertices to exchange edge information with its connected areas. As shown in Figure 2, vertices ν_1, ν_2, and ν_3 are on the same area $A_{(0\,1)}$. $C_{(0\,1)}$ is the controller of area $A_{(0\,1)}$. φ is a special label used to represent the area is not belong to any controller domains (e.g., legacy network). That is, area $A_{(\varphi)}$ do not have a controller which can communicate with other areas. There is another special area, called Root Area (RA), which is the root of all areas whose starting item of label match to the label of RA. The controller of RA, called Rooter, collects edge information from its area controllers. Therefore, the Rooter owns a global relationship among its controlled areas.

 ISBN: 978-1-61208-377-3

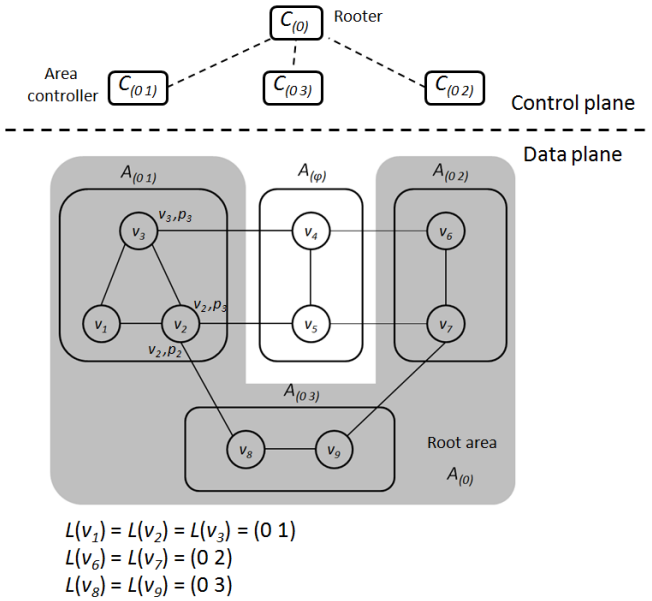


Figure 2. A sample network where vertices have been assigned to areas, represented by rounded boxes.

A vertex $u \in A_{L(u)}$ incident on an edge (u, v), such that $v \notin A_{L(u)}$ is called a border vertex for $A_{L(u)}$, and is in charge of exchanging the edge information to other connected area $A_{L(v)}$where $L(v) \neq L(u)$. If a vertex $u \in A_{L(u)}$ has an edge (u, v), such that $v \in A_{L(u)}$, we call u and v as inner vertices. An inner vertex only exchanges edge messages to other inner vertices in the same area.

There are two different edge information messages. one is exchanged among border vertices in network $G = (V, E)$, called $\Pi =< v, p(v), L(v), \delta >$ where $v \in V$ is the vertex which fires this edge information message; $p(v)$ is the port of vertex v that sends the outgoing edge information; $L(v)$ is the label of v to represent the area that v belongs to; δ (possible empty) is the set of parameters of the port (e.g., priority), which can realize some simple link utilization functions. Each area controller disseminates the other edge message, called $M =< \sigma(v), \sigma(w), \delta >$, to its Rooter. As illustrated in Figure 2, $C_{(0)}$ will receive M from $C_{(0\,1)}$, $C_{(0\,2)}$, and $C_{(0\,3)}$ respectively. There are three fields in M, $\sigma(v)$ is a composition field which includes vertex v, its label $L(v)$, and port $p(v)$; the definition of $\sigma(w)$ is same as $\sigma(v)$ but the vertex $w \in V$ such that $v \neq w$ is the end vertex; δ is the set of parameters of the port.

Each area controller owns full edge information of neighbor areas via all its border vertices. If there exists a port on one border vertex, it receives the edge information of a specific area which is the same as its area controller, we call this port of the vertex is a Representative Port (*RP*) for this neighbor area. As illustrated in Figure 2, (v_2, p_3) and (v_3, p_3) are two *RP*s for $A_{(0\,2)}$ in area $A_{(0\,1)}$.

IV. PROPOSED MECHANISM

In this section, we describe the design philosophy and implementation of our approach. In general SDN network environment, all vertices in the same area will exchange edge information to each other. This is well defined in OpenFlow specification and all popular controllers have already implemented it. However, extended edge information that beyond this area will not be exchanged. In order to compose edge information across different areas, the border vertex which resides in one area will exchange the edge information to its neighbor border vertices that may be directly connected or through one or more area $A_{(\varphi)}$. These operations are formalized in Figure 3. First of all, area controller $C_{L(u)}$ calls sub-procedure to update its data structures according to the received message Π_{new}, such as its neighbor area list and connected edges list for areas. Then, it creates and disseminates edge information that are newly appeared in Π_{new}. After processing all new edge information, $C_{L(u)}$ disseminates edge information that updated the set of parameters. Finally, the procedure updates the list of *RP*, that is used when area controller determines which ports allowed to forward broadcast packets to all connected areas. Note that, in each area controller we have a background process to check the validity of edge information. If it exceeds the timeout $T_{L(u)}(\pi)$, all related information of edge will be removed.

01: **procedure** UpdateBorderVertexEdges (u, Π_{new})
02: BVP is a set stores all border vertices and ports in $C_{L(u)}$
03: A is the set of all connected areas in $C_{L(u)}$
04: $ABVP$ is the map of $area \rightarrow (vertex, port, \delta)$ that stores the information of all discovered border vertices according to different connected areas.
05: BVP^2 is the map of $(u, p(u)) \rightarrow (v, p(v), \delta)$, where u and v are resided in two different controller areas respectively.
06: RP is the map of $area \rightarrow list\ of\ (vertex, port, \delta)$ which represents if sending packets to specific area, we can choose one vertex-port pair in the list.
07: **for each** $\pi_{new} =< v, p(v), L(v), \delta > \in \Pi_{new} \backslash \Pi_{L(u)}$
08: UpdateElements (π_{new})
09: Initialize $T_{L(u)}(\pi_{new})$ to a configured edge timeout
10: Create a new M
11: $M.\sigma(v) = (u, p(u), L(u))$
12: $M.\sigma(w) = (v, p(v), L(v))$; $M.\delta = \delta$
13: Send M to the Rooter
14: UpdateRepresentativePorts $(BVP, ABVP, BVP^2)$
15: **end for**
16: **for each** $\pi =< v, p(v), L(v), \delta_{old} > \in \Pi_{L(u)} \backslash \Pi_{new}$
17: **if** $\exists\ \pi_{new} =< v, p(v), L(v), \delta_{new} > \in \Pi_{new}$
18: π_{new} is an updated instance of π in $\Pi_{L(u)}$
19: $\pi.\delta_{old} = \delta_{new}$; $M.\delta = \delta_{new}$; renew $T_{L(u)}(\pi)$
20: Send M to the Rooter
21: **end if**
22: **end for**
23: **end procedure**
24: **subprocedure** UpdateElements (π)
25: $\Pi_{L(u)} \cup \{\pi\}$
26: **if** $L(v) \notin A$
27: $A \cup \{L(v)\}$

```
28:  if (v, p(v), δ) ∉ BVP²(u, p(u))
29:    BVP²(u, p(u)) ∪ {v, p(v), δ}
30:  if (v, p(v), δ) ∉ ABVP(L(v))
31:    ABVP(L(v)) ∪ {v, p(v), δ}
32: end subprocedure
33: function UpdateRepresentativePorts (BVP, ABVP, BVP²)
34:  for each bvp ∈ BVP
35:   for each a ∈ A
36:    if BVP²(bvp) = ABVP(a)
37:     RP(a) ∪ {BVP²(bvp), δ}
38:   end for
39:  end for
40: end function
```

Figure 3. Algorithm used for updating the set $\Pi_{L(u)}$ of known edge information in area controller $C_{L(u)}$ when the new edge information Π_{new} arrived at a border vertex u.

As we described in Section 3, each area controller will send edge information M to its Rooter periodically. In our proposal, this Rooter is responsible for providing the network-wide topology. Although our mechanism works well even eliminating the Rooter, we still keep this element for preserving the control flexibility to network management system in the upper layer, such as altering the original flow path and so on. We now illustrate the operations undertaken by Rooter to update its controlled area topology according to received M. These operations are formalized as the procedure UpdateAreaTopology (M_{new}) in Figure 4. First of all, Rooter composes any newly edge information from area controllers. If received M is already existed in edge information of Rooter, this process only updates the new parameter and the timeout of this edge. Last, the Rooter will refresh the area topology according to updated M_{Rooter} and keep this data structure for computing the area path in the future.

Both area $A_{(0\,1)}$ and $A_{(0\,2)}$ in Figure 2 have two border vertices. Take vertex v_3 in area $A_{(0\,1)}$ as an example, it receives edge information from area $A_{(0\,2)}$ by two paths, $(v_7 - v_5 - v_4 - v_3)$ and $(v_6 - v_4 - v_3)$. Similar to vertex v_2, we also can discover two paths. Therefore, four edges are discovered between the area $A_{(0\,1)}$ and $A_{(0\,2)}$ in Rooter. After processing the algorithm of Figure 4, the area topology is delivered, as shown in Figure 5.

```
01: procedure UpdateAreaTopology (M_new)
02:  m is a new or an updated edge from controlled areas
03:  for each m =< σ(v), σ(w), δ > ∈ M_new \ M_Rooter
04:   Update M_Rooter by m and initialize T_Rooter(m)
05:  end for
06:  for each m =< σ(v), σ(w), δ_old > ∈ M_Rooter \ M_new
07:   if ∃ m_new =< σ(v), σ(w), δ_new > ∈ M_new
08:    m.δ_old = δ_new; renew T_Rooter(m)
09:  end for
10:  Refresh area topology according to M_Rooter
11: end procedure
```

Figure 4. Algorithm to update area topology at Rooter according to the received edge information from controller areas.

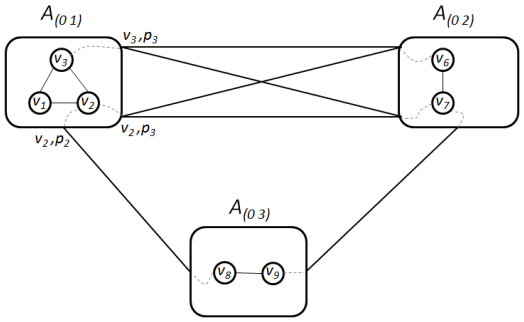


Figure 5. The logical network reduced from the sample network in Figure 2.

We illustrate the operation undertaken by area controller $C_{L(u)}$ when it receives a broadcast packet from any vertex in the controlled domain. These operations are formalized in Figure 6. The input parameters are composed of all connected areas (A), the representative port list generated in Figure 3, and the broadcast packet (*packet*). This procedure goes through the set A and checks how many vertex-port pairs in RP of A. If there contains two or more pairs, it uses the function PickForwardingVertexPorts (RP) to determine the ports of border vertices for forwarding this broadcast packet.

```
01: procedure HandleBroadcastPacket (A, RP, packet)
02:  FT(u) is the flow tables of vertex u
03:  A is the set of all connected areas in C_L(u)
04:  RP is the map of area → list of (vertex, port, δ)
05:  MP is the map of vertex → (mac, port) in C_L(u)
06:  for each a ∈ A
07:   if size of RP(a) > 1
08:    (vertex, port) = PickForwardingVertexPorts (RP(a))
09:   else // only one item in RP(a)
10:    (vertex, port) = RP(a)
11:    mac = packet.src.mac
12:    FT(vertex) ∪ {mac, port}
13:   end if
14:  end for
15: end procedure
16: function PickForwardingVertexPorts (RP)
17:  ε = ϕ
18:  for each rp ∈ RP
19:   if ε = ϕ or rp.δ.prio ≥ ε.δ.prio
20:    ε ← rp
21:  end for
22:  ε.δ.prio = ε.δ.prio − 1
23:  return ε
24: end function
```

Figure 6. Algorithm used for determining which ports on its all border vertices will be used to forward broadcast packets to other areas.

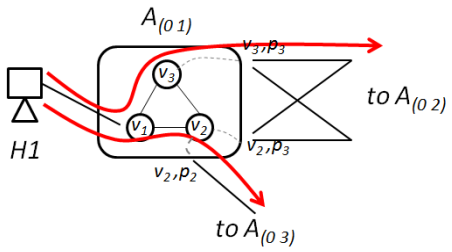


Figure 7. An example of forwarding packets from a host to different areas.

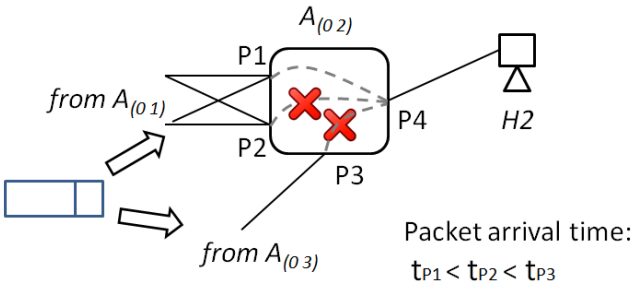


Figure 8. An example of selecting one port in the receiver according to arriving time of the same request packet.

To show an example of process in Figure 6, we consider the area $A_{(0\,1)}$ in Figure 5 and let one host H1 connect to vertex v_1 in this area, as shown in Figure 7. There are three vertex-port pairs on all border vertices in $A_{(0\,1)}$, two of them can reach to area $A_{(0\,2)}$ and the other is for area $A_{(0\,3)}$. Assume that H1 sends a broadcast packet (e.g., an ARP request). In the meantime, the area controller triggers the procedure HandleBroadcastPacket and determines the forwarding ports on its border vertices to transmit the packet to other areas. Note that we only consider *RP*s on all border vertices, if there exists other type of ports, such as access ports (e.g., connecting to hosts) or intra-switch ports (e.g., the port on v_3 connecting to v_2), our mechanism also forwards the broadcast packet to these ports.

In Figure 7, both (v_2, p_3) and (v_3, p_3) are *RP*s for area $A_{(0\,2)}$, according to the algorithm in Figure 6 we will select only one *RP* and forward the broadcast packet. As shown in Figure 7, we choose the port 3 of vertex v_3 to forward the broadcast packet of host H1 to area $A_{(0\,2)}$ while using the port 2 of vertex v_2 to transmit the same packet to area $A_{(0\,3)}$. In this way, we can decrease the number of packets in network to offload area controllers. Moreover, selecting the forwarding edge from one single area can ensure the effective usage of the links without conflicts with other hosts which target the same area. We choose the *RP* to forward packets according to the parameter on the port (e.g., δ). If this port is selected this time, it will adjust the priority to ensure we can choose another ports next time. This priority value will be restored when the flow is released.

In addition, to ensure that the receiver H2 only replies via one of the ports, the arrival time of the request is recorded and used to determine the returning port of H2's reply. In Figure 8, area $A_{(0\,2)}$ receives the same broadcast packet from three different ports, P1, P2, and P3. Assume the arrival times are t_{P1}, t_{P2}, and t_{P3} respectively and t_{P1} is the minimum of them. Thus, host H2 chooses the P1 to send its reply packet.

V. EVALUATION

In this section, we describe the performance evaluation of our mechanism. We simulate physical network connection between TWAREN and Internet2 as our experiment topology. There are 2 physical servers equipped with 64G of RAM and 2 Intel Xeon(R) L5640 CPUs. Each of them runs a Mininet [10] to emulate OpenFlow network topology in TWAREN (e.g., $A_{(0\,1)}$) and Internet2 (e.g., $A_{(0\,2)}$). In addition, we create another domain, called $A_{(0\,3)}$, with 2 physical OpenFlow switches and 1 physical host. There are 4 controllers, one of them represents the Rooter controller and the others install Floodlight (version 0.9) and manage their own areas. The topology of our experiment is shown in Figure 9.

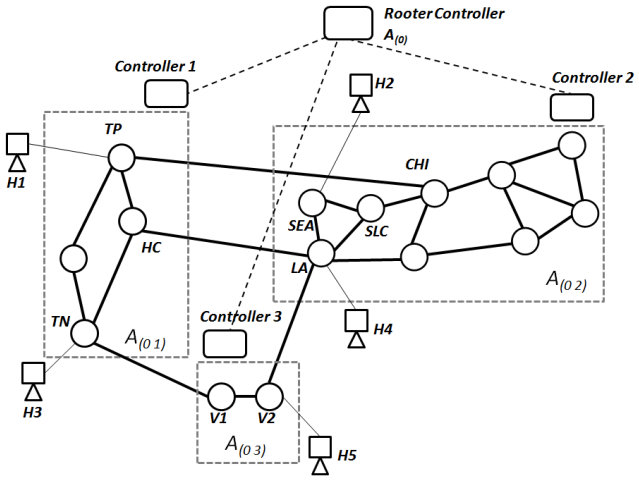


Figure 9. Experiment topology.

As we described in Section 1, controller can handle loop topology in a single domain but not multiple domains. Therefore, we only consider the loops across two or more domains. A loop is represented as a series of edge nodes (e.g., TP-CHI-LA-HC-TP).

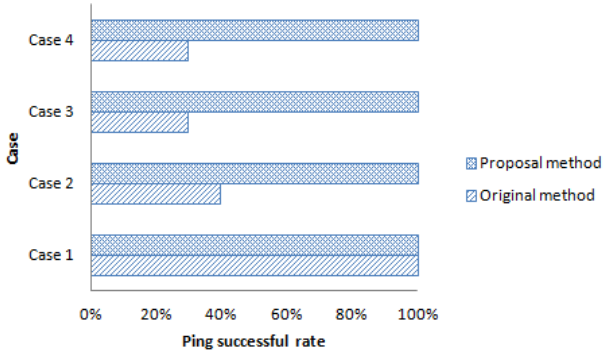


Figure 10. Ping successful rate with four different cases.

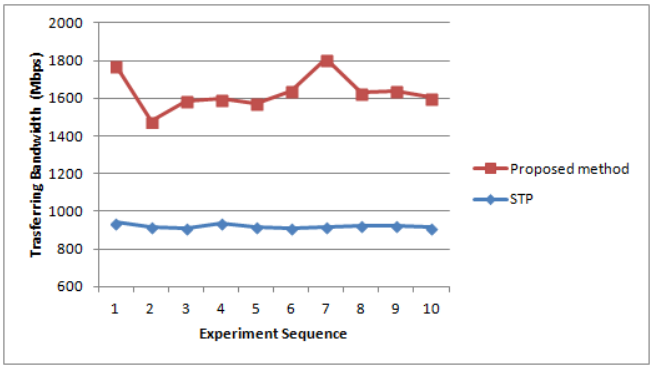


Figure 11. Comparison of transferring bandwidth in two methods when there are two pairs of hosts transferring packets at the same time.

In the first experiment, we evaluate the ping successful rate of our method by comparing with the original forwarding method. There are four cases in this experiment. In each case, we run 10 times on host pairs (e.g., H1-H2, H1-H3, H1-H4, and H1-H5) and compute the ping successful rate. In Case 1, we remove two inter-domain links (e.g., HC-LA and V2-LA) to construct the topology with no loops. As depicted in Figure 10, both our proposed method and the original method have 100% ping successful rate. In the second case, we add the link HC-LA to form one loop (e.g., TP-CHI-LA-HC-TP) between domain $A_{(0\,1)}$ and $A_{(0\,2)}$. We note that there are 2 paths from CHI to LA (e.g., CHI-SEA-LA and CHI-SLC-LA). But in our experiment, we only require that there is at least one path between these 2 edge nodes to assure the connectivity in a single domain. Thus, this is not regarded as a failure. The Case 2 in Figure 10 shows our method reaching 100% ping successful rate while the original method is only 40%. We add link V2-LA in the third case to create the topology with 3 loops (e.g., TP-CHI-LA-HC-TP, HC-LA-V2-V1-TN-HC, and TP-CHI-LA-V2-V1-TN-HC-TP). Our proposed method still reaches 100% but the original forwarding method is lower than 30%, shown as Figure 10. In the final case, we add an extra link between HC and LA. The result is illustrated in Case 4 of Figure 10. This experiment shows that our method is working regardless the number of loops in the topology. We can guarantee any host can successfully communicate with hosts in other areas.

Next, we compare our proposal with STP protocol. We use two pairs of hosts (e.g., H1-H2 and H3-H4) to measure the performance when these hosts transferring packets at the same time. Host H2 and H4 are selected as the iperf servers while the others are the clients of iperf. In order to simulate STP protocol, we remove two inter-domain links (e.g., HC-LA and V2-LA) to build a tree structure in a looped topology. Figure 11 demonstrates the throughput results from STP and our method. We observe our proposed is performing considerably better than STP, offering 77% increasing throughput compared to STP. The reason is STP has only one path between domain $A_{(0\,1)}$ and $A_{(0\,2)}$. Thus, two pairs of hosts share this inter-domain link TP-CHI. But in our proposed method, if there exists another link between these two domains, they will all be used to improve the transfer performance.

VI. CONCLUSION

In this paper, we have proposed a mechanism for solving transmission problem among SDN domains with loops. The proposed algorithms select one port for each connected area to forward broadcast packets. It decreases the number of packets in network to offload area controllers. In addition, the area controller uses our method to filter the repeated broadcast packets at a border vertex and do not forward these packets to avoid broadcast storm. Besides, as compared with original forwarding method, our method can efficiently use multiple edges in loops topology to improve the transferring bandwidth.

Our future work is to improve path compute by defining more granular parameters across multiple SDN domains. In addition, our proposal use the Advanced Message Queuing Protocol (AMQP) to exchange edge information between the Rooter controller and area controllers. We can integrate our approach with other event-based frameworks (e.g., Kandoo) for resources management among multiple domains.

REFERENCES

[1] N. McKeown et al., “Openflow: Enabling Innovation in Campus Networks,” ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69-74, Apr. 2008.

[2] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, “On Scalability of Software-Defined Networking,” IEEE Commun. Mag., vol. 51, no. 2, pp. 136-141, Feb. 2013.

[3] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, “Logically Centralized? State Distribution Trade-offs in Software Defined Networks,” Proc. ACM Workshop Hot Topics in Software Defined Networks (HotSDN '12), Aug. 2012, pp. 1-6, ISBN: 978-1-4503-1477-0.

[4] A. Tootoonchian and Y. Ganjali, “HyperFlow: A Distributed Control Plane for OpenFlow,” Proc. Internet Network Management Workshop/Workshop on Research on Enterprise Networking (INM/WREN '10), USENIX Association, Apr. 2010, pp. 3-3.

[5] T. Koponen et al., “Onix: A Distributed Control Platform for Large-scale Production Networks, ” Proc. USENIX Symp. on Operating Systems Design and Implementation (OSDI '10), Oct. 2010, pp. 351-364, ISBN: 978-1-931971-79-9.

[6] S. H. Yeganeh and Y. Ganjali, “Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications,” Proc. ACM Workshop Hot Topics in Software Defined Networks (HotSDN '12), Aug. 2012, pp. 19-24, ISBN:978-1-4503-1477-0.

[7] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, “Applying NOX to the Datacenter,” Proc. ACM Workshop on Hot Topics in Networks (HotNets '09), Oct. 2009, pp. 1-6.

[8] A. R. Curtis et al., “DevoFlow: Scaling Flow Management for High-Performance Networks,” Proc. ACM SIGCOMM 2011 Conference (SIGCOMM '11), Aug. 2011, pp. 254-265, ISBN: 978-1-4503-0797-0.

[9] H. Yin et al., “SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains,” IETF Internet-Draft, draft-yin-sdn-sdni-00, Jun. 2012.

[10] B. Lantz, B. Heller, and N. McKeown, “A network in a Laptop: Rapid Prototyping for Software-Defined Networks,” Proc. ACM Workshop on Hot Topics in Networks (HotNets '10), Oct. 2010, pp. 1-6.

 ISBN: 978-1-61208-377-3

Sentiment Analysis on Online Social Network Using Probability Model

Hyeoncheol Lee, Youngsub Han
Department of Computer and Information Sciences
Towson University
Towson, USA
hlee23, yhan3@students.towson.edu

Kwangmi Ko Kim, Ph.D.
Department of Mass Communication
Towson University
Towson, USA
kkim@towson.edu

***Abstract*—Sentiment analysis is to extract people's opinion and knowledge from text messages. Recently, demands on automated sentiment analysis tool for text messages generated from web have dramatically increased and the literature on this topic has been growing. In this paper, we propose a semi-automated sentiment analysis method on online social network using probability model. The proposed method reads sample text messages in a train set and builds a sentiment lexicon that contains the list of words that appeared in the text messages and probability that a text message is positive opinion if it includes those words. Then, it computes the positivity score of text messages in a test set using the list of words in a message and sentiment lexicon. Each message is categorized as either positive or negative, depending on threshold value calculated using a train set. To check the accuracy, we compared the sentiments of the proposed method with sentiments of human coders. This research is unique and novel in that it guarantees high accuracy rates and does not require additional information, such as users' profile and network relationship.**

Keywords-sentiment analysis; social network.

I. INTRODUCTION

In recent years, online social network sites, such as Facebook, Twitter, Blogger, LinkedIn, YouTube and MySpace, have changed the way people communicate with each other. People share information, report news, express opinions and update their real-time status on the online social network sites. With the increasing popularity of the online social network sites, a huge amount of data is being generated from them in real time. Analyzing the data in social media can yield interesting perspectives to understanding individuals and human behavior, detecting hot topics, and identifying influential people, or discovering a group or community [10][11].

Several user-generated text messages contain users' emotional state and mood about topics, such as events, products, and services. Sentiment analysis is to extract the users' opinion and knowledge from the text messages [12][13]. Recently, automatic sentiment analysis on online social network has received a lot of attention from researchers. Most approaches focus on identifying whether a text expresses positive or negative opinion about a topic [13][14]. The high volume of such data has called for automated tools that assign positive or negative for much easier and quicker analysis.

In spite of high demands for automatic sentiment analysis on text messages in online social network data, the development of the automatic sentiment analysis faces some challenges as the text messages in online social networks are unstructured, unlabeled, dynamic and noisy [2][15]. Due to the characteristics of the messages, accuracy of previous automatic sentiment analysis approaches remains around 80%, which should be further improved for more accurate analysis. In addition, some existing approaches require additional information, such as user's tendency or relationship, which is not always available on online social networks. For these reasons, we propose a sentiment analysis algorithm that guarantees higher accuracy than existing approaches and can be used broadly in any social network sites without requiring additional information.

The rest of the sections are organized as follows: In Section 2, the related works on sentiment analysis are summarized. Section 3 outlines main methodology of sentiment analysis we propose, and Section 4 presents experiment results. Section 5 concludes our works and gives direction to future research.

II. RELATED WORKS

There are two main approaches to extracting sentiment from text messages. The first approach is lexicon-based sentiment analysis which is found on pattern matching with pre-built lexicon. Many researches tried to extract sentiment or opinion from text messages using this approach [1][17][18][19]. O'Connor et al. [1] analyzed political opinion using a sentiment analysis algorithm. They collected text messages related to political opinion from Twitter from 2008 to 2009. Also, they built a lexicon where each word was categorized as either positive or negative keywords based on OpinionFinder [3]. The number of positive and negative keywords was counted for every message. A message is defined as positive if it contains any positive word, and negative if it contains any negative word. As a result, the ratio of positive messages versus negative messages was compared with survey results and it showed

 ISBN: 978-1-61208-377-3

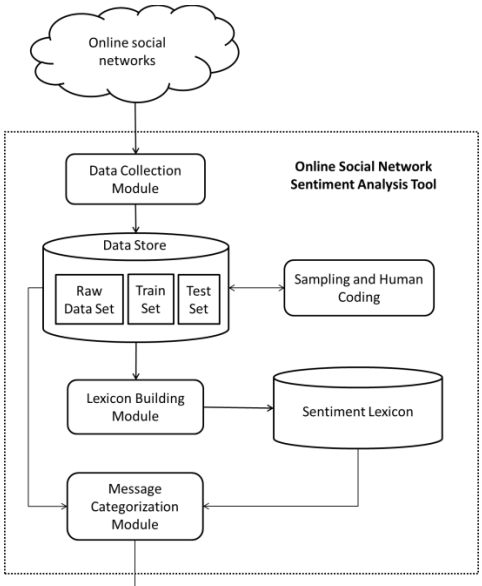
data correlation between results of sentiments analysis and survey is as high as 80%. The results indicate that the method can be used as a supplement for traditional survey. However, this lexicon based approach has weakness in that a message including positive keywords does not necessarily yield positive opinion. For instance, a word *like i*s categorized as a positive word in the lexicon, meaning if a message includes the word *like*, it is categorized as a positive message. Nevertheless, if the message includes the word *don't* right before *like*, the actual opinion of message should be categorized as negative. In this sense, such lexicon-based approach should be improved regarding the nature of language. The second approach is classification-based sentiment analysis, also known as supervised classification. It builds a sentiment classifier using a train set that contains labeled texts or sentences and test new texts using the classifier. Statistical and machine learning techniques can be used in this approach. Bayesian modeling approach has proven to be a capable method for multi-class sentiment classification and multi-dimensional sentiment distribution predictions [5]. Machine learning techniques, such as Naïve Bayes (NB), Support Vector Machines (SVM), Maximum Entropy, Decision Tree and K-Nearest Neighbor Classifier have been shown to be effective methods for sentiment analysis of messages [6][16].

Some of sentiment analysis approaches examine message author's information or behavior. Guerra et al. [2] proposed a sentiment analysis algorithm using bias of social media users toward a topic. They posit users tend to express their opinion multiple times and a user's bias tends to be more consistent over time as a basic property of human behavior. Thus, they measured bias of social media users toward a topic and analyzed sentiment by transferring users biases into textual features. Kucuktunc et al. [7] also proposed a method of analyzing sentiment based on characteristics of users, such as gender, age and education level. However, these methods cannot be broadly used because it requires relationship data among users and previous messages that the users have posted, which are not always provided by social networks due to the privacy laws.

Speriosu et al. [4] applied label propagation (LPROP) approach based on graph representation to analyze sentiment of messages in Twitter. Their assumption is that each tweet written by a user is linked to other tweets written by the same user, and each author is influenced by the tweets written by users whom he or she follows. They represented such a relationship using a graph where the features of the message, such as words, emoticon and authors, are inter-related to each other. Those features affect positivity or negativity of the message in the graph. They tested the accuracy of the LPROP approach with messages in four different topics and compared it with the accuracies of other approaches. The results show that accuracy of the proposed LPROP approach is the highest among other sentiment analysis approaches as it reached 65.7% to 84.7%, depending on the topics. However, there is a room for improving the accuracy of the LPROP because its average accuracy is still 72.08%.

III. METHODOLOGY

In this section, we describe the methodology of sentiment analysis for text messages generated from web. Figure 1 shows the overall process of sentiment analysis on text messages.



Messages	Positivity Score	Sentiment
I love the commercial	0.9	Positive
Well done	0.72	Positive
This is gross	0.40	Negative
...	...	...

Figure 1. Overall Process of Sentiment Analysis Tool

First, data collection module collects text messages from online social networks, such as Twitter and YouTube, which will be saved as a raw data set in data store. Then, it generates *sample* text messages from the data store and human coders categorize the messages into positive or negative opinions. The categorized *sample* messages are saved into a train set in the data store. After that, lexicon building module scans all categorized *sample* messages in the train set and calculates the weighted probability that the message is positive opinion if the word is included in a message. The list of words and the probabilities for each of them are saved in sentiment lexicon. Finally, the message categorization module calculates positivity scores for every message and categorizes whether the messages are positive or negative. To check the accuracy of the proposed method, we generated a test set which is also categorized in the same way the train set is made. Details of methodologies are explained in later on this section.

A. Data Collection Module and Datasets

Several social networks allow us to collect data with Application Programming Interface (API) [8][9]. For example, YouTube provides us with API to collect the data

in YouTube. The main purpose of the YouTube API is to integrate the functionalities of YouTube into software applications. In addition to the main functionalities, the API allows developers to collect every kind of YouTube data, such as video information, user profile, and comments. Twitter also provides us with API for data collection.

In this research, we have developed a data collecting tool that automatically collects comments posted on YouTube videos. We have selected 3 commercial videos: *Prom* (for Audi), *Farmer* (for Ram) and *Perfect match* (for Go Daddy) that aired during the Super Bowl Game in 2013 which created a lot of buzz on online social networks. Then, we collected all comments that were posted on the videos using the tool. The comments are saved into a raw data set in data store.

B. Sampling and Human Coding

Among all comments, we randomly selected a total of 3,000 comments, 1,000 comments for each video. The comments were categorized as positive or negative by human coders. Two graduate students were involved in the coding process. We built a data *sample* using the messages that both human coders categorized into the same sentiment. In this process, we excluded messages that have neutral or mixed opinions that have both positive and negative opinions in the sample message. The categorized messages are saved into a train set in data store.

C. Building Sentiment Lexicon

Once sample messages are categorized by human coders and saved into the train set in data store, lexicon building module generates sentiment lexicon. It consists of word, the number of occurrence in positive messages, and the number of occurrence in negative messages and probability that a message is positive opinion if it contains the word, which will be used as base resource to categorize sentiment of messages in message categorization module.

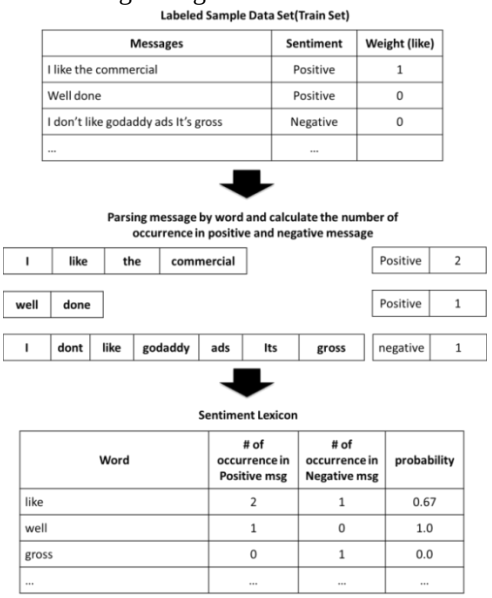


Figure 2. Example of Building Sentiment Lexicon using Labeled Sample Data Set(Train Set)

The process of building sentiment lexicon is as follows. First, it reads a message in the train set. Then it parses the message by word and checks the labeled sentiment and weight. In the comments of YouTube, a user can add a *like* or *dislike* tag, indicating the degree of user's agreement on the message. We use the tags as a weight point. The number of occurrence for every word in positive and negative messages are counted and saved into sentiment lexicon. Finally, the probability that the message is positive opinion if it includes the word is computed for every word and saved into sentiment lexicon.

Figure 2 shows the overall process and example of building sentiment lexicon using labeled sample data set (train set). Assume there are 3 messages in a train set and each message is labeled as shown in figure 2. If the word *like* appears in a message labeled positive opinion, the number of occurrence in positive opinion for the word is increased by one. If the labeled message has a like tag, the number of occurrence in positive opinion for the word is increased by two. If the word *like* appear in positive opinion twice and negative opinion once, the probability that a message is positive will be 0.67 if the message includes the word *like*.

D. Categorize the comments

Once sentiment lexicon is built completely, message categorization module classifies a text message into a positive or negative opinion. The comment sentence is represented with vector space model (VSM), where each word in the message and its probability in sentiment lexicon are shown together. Then the positivity score of a document (comment) is computed as follows.

$$\text{Positivity Score (d)} = \frac{\sum_{i=1}^{n} P(wi)}{n} \quad (1)$$

In (1), *w* is each word in a document *d* and *n* is the number of words in the document. *P* is probability of the word which is saved in sentiment lexicon with the word. Example of computing positivity score for a comment is visualized in Figure 3.

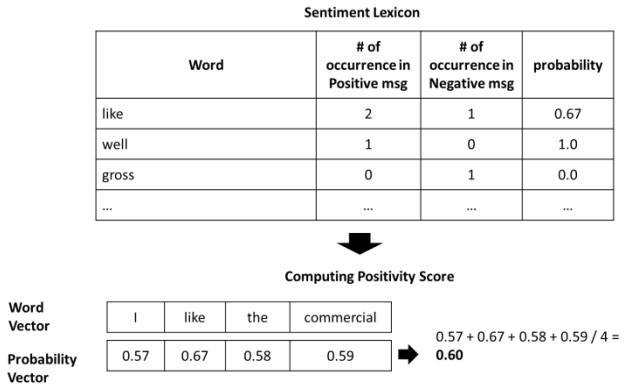


Figure 3. Example of computing positivity score

Once the positivity scores of all comments in train set are computed, message categorization module reads them again and computes the threshold of positivity score to classify the

comment as either a positive or negative message. The threshold value is derived by computing mean value of positivity scores for all positive and negative messages in the train set. The example of computing threshold value is visualized in Figure 4.

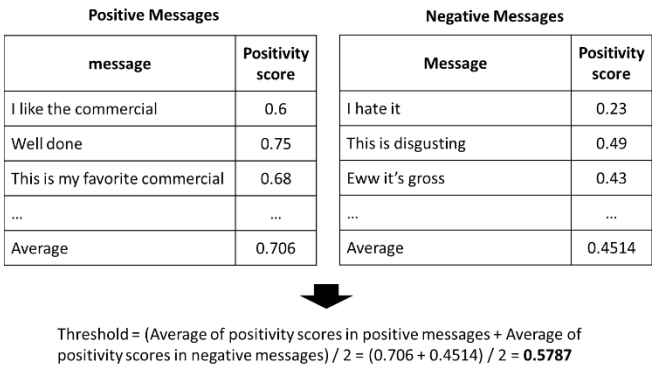


Figure 4. Computing Threshold using positivity scores of positive and negative messages

The last step of sentiment analysis is to categorize messages in the test set using the threshold. The positivity score of each comment in the test set is computed in the same way as the previous step in the message categorization module. Then, it classifies the comment as either a positive or negative message. If the positivity score is greater than the threshold, it is categorized as a positive message. Similarly, if the positivity score is less than the threshold, it is categorized as a negative message. The example of classifying sentiment of comments is visualized in Figure 5.

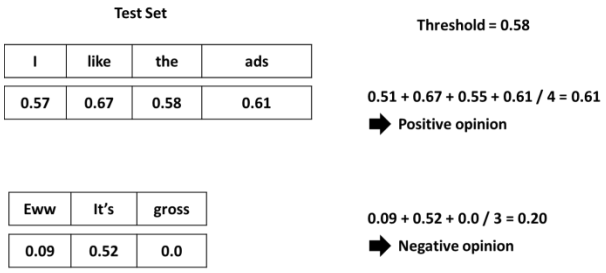


Figure 5. Example of Classifying Sentiment of Comments

Suppose a message "I like the ads" is given as shown in the Figure 5. Each word in the message is represented with VSM and the probabilities are assigned to each word (I:0.57, like:0.67, the:0.58 and ads:0.61). Then, the positivity score is computed according to the (1) and compared with the threshold value. Since the positivity score 0.61 is greater than 0.58, the message is classified as positive opinion. In the similar way, the positivity score of the second message "Eww it's gross" is computed, compared with the threshold, and classified as negative opinion.

IV. EXPERIMENTS

This section presents the experiment results of the sentiment analysis method we proposed.

A. Data Collection

Table 1 shows data collection results. We collected the video information and comments posted under the video on May 26, 2014 using the data collection tool introduced in the previous section. Video ID is an identification key generated by YouTube. We collected a total of 25,003 comments for the videos.

TABLE I. DATA COLLECTION RESULTS

Video Title	Comments Count
Official Ram Trucks Super Bowl Commercial "Farmer"	16683
Audi 2013 Big Game Commercial - "Prom"	2977
Go Daddy Bar Refaeli Kiss Super Bowl Commercial 2013 - FULL	5343

For each video, 1,000 comments are selected and used for building the sentiment lexicon and pre-processing the train data as described in the previous sections.

B. Sentiment Lexicon

Table 2 is part of sentiment lexicon. Every word appeared in the comments is saved in the first column of sentiment lexicon. The number of word occurrence in positive and negative messages is recorded in the second and third column with the words. The probability that a message is positive if it contains the word is computed using the words occurrence in positive and negative messages and is saved in the last column. As a result, sentiment lexicon was built with total of 739 words with the probability.

TABLE II. SENTIMENT LEXICON

Word	The number of occurrence in positive message	The number of occurrence in negative message	Probability
love	41	0	1
great	35	5	0.87
car	23	4	0.85
pretty	9	2	0.81
all	33	8	0.8
good	17	6	0.73
dad	8	3	0.72
prom	13	5	0.72
my	50	34	0.59
make	11	9	0.55
me	25	22	0.53
not	26	29	0.47
stupid	5	6	0.45
never	6	8	0.42
kiss	5	9	0.35
why	4	8	0.33
fuck	2	10	0.16
disgusting	1	13	0.07
awkward	1	13	0.07
gross	0	20	0

C. Sentiment Categorization Results

To show the accuracy of the proposed algorithm, we labeled a test set in the same way as the train set is built.

 ISBN: 978-1-61208-377-3

Then, the sentiments of comments derived by the proposed method are compared with the sentiments labeled by human coders as shown in Table 3. If human coders and the proposed method categorized a message into the same sentiment, the result is classified as *correct*. Otherwise, the result is classified as *incorrect.* The accuracy of the proposed method is computed as shown in the Figure 6. It shows that the accuracy of the proposed method is at 86%. However, the accuracy for the negative messages is relatively lower than the accuracy for positive messages, which needs to be considered and improved in the future research.

TABLE III. CLASSYFYING SENTIMENT OF COMMENTS AND COMPARING THE SENTIMENT BY THE PROPOSED METHOD WITH SENTIMET BY HUMAN CODERS

Text(Comment)	Positivity Score	Sentiment by the proposed method	Sentiment by human coders	Results
This was the best commercial! It was so powerful.....	0.67	Positive	Positive	Correct
Whoever at Dodge decided to go with this ad is a Goddamn genius!	0.64	Positive	Positive	Correct
love love love!!!!!!	1.00	Positive	Positive	Correct
Just so touching and I loved this.	0.70	Positive	Positive	Correct
Ok so I think that just made me cry a little bit. That was beautiful	0.69	Positive	Positive	Correct
VERY uncomfortable and retarded	0.41	Negative	Negative	Correct
The sound effects though.. oh goshh eww (/.)	0.36	Negative	Negative	Correct
No. I hate it	0.47	Negative	Negative	Correct
AH!! MY EYES	0.59	Positive	Negative	Incorrect
This is DISGUSTING!	0.49	Negative	Negative	Correct

Test Set

Category	Positive Message		Negative Messages	
Topic	Correct	Incorrect	Correct	Incorrect
Audi	84	7	7	2
Dodge	80	6	7	7
Go Daddy	7	3	73	17
Total	171	16	87	26

Correct Message / Total Message = 258 / 300 = **86%**

Figure 6. Sentiment Analysis Results and Accuracy of the Proposed Method

To compare performance of the proposed method with other approaches, we applied F-measure that can be used to compute test's accuracy [13]. F-measure uses two measurement degrees; precision *p* and recall *r*. *p* is the number of correct results divided by the number of all returned results. *R* is the number of correct results divided by the number of results. The F1 score is calculated as shown in (2).

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \quad (2)$$

TABLE IV. COMPARION OF F-SOCRE RESULTS

Method	F1 score
PANAS-t	0.737
Emoticons	0.948
SASA	0.754
SenticNet	0.810
SentiWordNet	0.789
SentiStrength	0.894
Happiness Index	0.821
LIWC	0.731
Proposed Approach	0.890

Table 4 shows results of F-measures. F1 score of our approach is 0.890 which is relatively higher than other approaches. However, it is lower than F1 score of Emoticons and SentiStrengh. Improving the accuracy needs to be considered in the future research.

V. CONCLUSION AND FUTURE RESEARCH

This research developed and proposed a sentiment analysis method using probability model that guarantees relatively higher accuracy than existing approaches with broader application. The result shows that it outperforms most existing sentiment analysis approaches in terms of accuracy. In addition, the proposed approach can be implemented only using text information without requiring any additional information. This proposed approach, however, has a limitation that requires preprocessing of sample text messages by human coders. We will investigate a fully automated sentiment analysis method in the next research, and continue to work on improving the accuracy rate of a proposed method.

REFERENCES

[1] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From Tweets to Polls:Linking Text Sentiment to Public Opinion Time Series", Proceedings of the International AAAI Conference on Weblogs and Social Media, May 2010, pp. 122-129.

[2] P. H. Guerra, A. Veloso, W. Meira, and V. Almeida, "From Bias to Opinion: A Transfer-Learning Apporach to Real-Time Sentiment Analysis", Proceddings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining(KDD 11), August 2011, pp. 150-158, ISBN:978-7-4503-0813-7.

[3] T. Winson et al., "OpinionFinder:A System for Subjectivity Analysis", Proceedings of HLT/EMNLP 2005 Interactive Demonstrations, October 2005, pp. 34-35, doi:10.3155/1225733.1225751.

[4] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter Polarity Classification with Label Propagation over

Lexical Links and the Follower Graph", Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing, July 2011, pp. 53-64, ISBN: 978-1-937284-13-8.

[5] Y. He, "A Bayesian Modeling Approach to Multi-Demensional Sentiment Distributions Predictions", Proceedings of the Frist International Workshop on Issues of Sentiment Discovery and Opinion Mining, August 2012, Article No.1, ISBN:978-1-4503-1543-2.

[6] A. Sharma and S. Dey, "A Boosted SVM based Sentiment Analysis Approache for Online Opinionated Text", Proceedings of the 2013 Research in Adaptive and Convergent Systems, October 2013, pp. 28-34, ISBN: 978-1-4503-2348-2.

[7] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A Larege Scale Sentiment Analysis for Yahoo! Answers", Proceedings of the fifth ACM international conference on Web search and data mining, February 2012, pp. 633-642, ISBN: 978-1-4503-0747-5.

[8] Google Developers, accessed on May 2014, <https://developers.google.com/youtube/>.

[9] Twitter Developers, accessed on May 2014, <https://dev.twitter,com>.

[10] I. King, J. Li, and K. T. Chan, "A brief survey of computational approaches in social computing". In IJCNN'09: Proceedings of the 2009 international joint conference on Neural Networks, pp. 2699–2706, Piscataway, NJ, USA, 2009. IEEE Press. ISBN:978-1-4244-3549-4.

[11] C. Byun, H. Lee, J. You, and Y. Kim, "Dynamic Seed Analysis in a Social Network for Maximizing Efficiency of Data Collection", Proceedings of the Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference, July 2013, pp. 132-136.

[12] A. Sharma and S. Dey, "A comparative study of feature selection and machine learing techniques for sentiment anlaysis", Proceedings of the 2012 ACM Research in Applied Computation Symposium, October 2012, pp. 1-7, ISBN:978-1-4503-1492-3.

[13] P. Goncalves, M. Araújo, F. Benevenuto, and M. Cha, "Comparing and combining sentiment analysis methods", Proceedings of the first ACM conference on Online social networks, October 2013, pp. 27-38, ISBN: 978-1-4503-2084-9.

[14] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification", Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, June 2009, pp. 1275-1284, ISBN: 978-1-60558-495-9.

[15] C. Byun, Y. Kim, H. Lee, and K. K. Kim, "Automated Twitter data collecting tool and case study with rule-based analysis", Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services, December 2012, pp. 196-204, ISBN: 978-1-4503-1306-3.

[16] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis", Proceedings of the 2012 Research in Adaptive and Convergent Systems, October 2012, pp. 1-7, ISBN:978-1-4503-1492-3.

[17] R. Feldman, "Techniques and applications for sentiment analysis", Communications of ACM, vol 56, Issue 4, April 2013, pp. 82-89, doi: 10.1145/2436256.2436274

[18] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-Based Methods for Sentiment Analysis", Computational Linguistics, vol 37, Issue 2, June 2011, pp. 267-307, doi: 10.1162/COLI_a_00049

[19] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", Foundations and Trends Information Retrieval, vol 2, Issue 1-2, January 2008, pp.1-135, doi: 10.1561/1500000011

 ISBN: 978-1-61208-377-3

Keyword-Based Breadcrumbs: A Scalable Keyword-Based Search Feature in Breadcrumbs-Based Content-Oriented Network

Kévin Pognart, Yosuke Tanigawa, Hideki Tode
Dept. of Computer Science and Intelligent Systems
Osaka Prefecture University
Osaka, Japan
{pognart@com., tanigawa@, tode@}cs.osakafu-u.ac.jp

***Abstract*—The Internet shows limited performances for users' needs especially on content sharing and video streaming. Content-Oriented Networks (CONs) are efficient approaches for such uses. They abandon the location-based routing of the Internet (IP routing) for a content identifier-based routing. In CONs, users must know the exact content identifier to request it. To give users an easier use of CONs, we propose the Keyword-based Breadcrumbs (KBC), a scalable keyword-based retrieval function for CONs based on Breadcrumbs (BC). Our work focuses on BC because of its simplicity, scalability, particularity and because it can be deployed in today's network, even partially. KBC uses stored information about contents in routers to retrieve several possible answers to a keyword-based request while keeping original behavior of BC for content identifier-based request. We present in this paper the working scheme of KBC, some KBC request managing rules to collect answers, and simulations results to show its performances.**

Keywords-Breadcrumbs; Content-Oriented Network; search; keyword; cache.

I. INTRODUCTION

The current Internet has a host-to-host architecture for allowing an easy communication between two machines. But in the recent years, people use the Internet not for direct communication but principally for sharing contents with a lot of people and viewing video streams. As a result, the current Internet has limited performances. Nowadays, some systems such as peer-to-peer (BitTorrent) can improve content sharing performances by coordinating several users by the contents they have.

Inspired by peer-to-peer systems, Content-Oriented Network (CON) is an alternative to the current Internet with some special features. Content identifier is used instead of location identifier for routing messages. Also, content identifiers are unique. The main characteristic of CON is that contents can be copied and cached in the network while keeping the same content identifier. It allows answering a request by one of the content copies located over the network.

As for CON, several routing methods have been proposed to realize it [1][2]. In our work, we particularly focus on Breadcrumbs (BC) [3][11] due to its attractive features described in Section II. This BC-based CON has simple content caching, location and routing systems. In BC, we assume that users and possibly routers have a content cache. Routers have also a BC table used to route requests. When content passes through a router, a BC entry is created in its BC table to indicate the direction of the cached content. If the content goes through a node having a content cache, the content is cached. Requests are firstly sent to a server to download contents by using IP routing. When a request arrives at a router where a BC entry for the same content identifier exists, the request is redirected to follow the direction shown in the BC entry. Each next node will redirect the request according to the direction in BC entries until finding the content in a content cache. If an issue occurs during the redirection, the BC entries are invalidated and the request is forwarded again to the server by IP routing.

To perform the routing, content identifiers must be unique. This uniqueness makes the requests difficult from a user's point of view. This problem also exists in the current Internet with URLs, and it leads to the need to use web search engines. Current web search engines are not an efficient solution because they use location and they cannot use cached content information. Hence, we propose Keyword-based Breadcrumbs (KBC). We extensively designed the BC framework to complement it with a keyword-based search feature while keeping the way of working of BC and its advantages. Also, we have implemented different KBC request behaviors to retrieve answers and we compare their performances.

In this paper, we present CONs and some unknown content search features. Then, we describe principle, specifics, and settings of KBC, and we compare KBC and the unknown content search features presented. We continue by evaluating KBC performances with some simulation scenario. To conclude, we summarize the main points about KBC and we talk about our future work.

II. RELATED WORK

A. Related CON schemes

To create a CON, several schemes have been proposed. The Data oriented Network Architecture (DONA) [5], the Network of Information (NetInf) [6], the Publish-Subscribe Internet Routing Paradigm (PSIRP) [7], and the Content-Centric Networking (CCN) [4] are the main approaches. In DONA, sources publish contents into the network and their information is spread to the nodes called resolution handlers. A request goes to a resolution handler to be routed to the

 ISBN: 978-1-61208-377-3

content. Then, the content is sent back to the requester by the reverse path or by a shortest route. NetInf can retrieve contents by name resolution and by name-based routing. Depending on the model used, the publication of a content uses a Name Resolution Service (NRS) by registering the link between the name and the locator, or it uses a routing protocol to announce the routing information. A node having a content copy can register it with NRS and by adding a new name/location binding. If an NRS is available, the requester can first resolve a content name into several available locators and find a copy from the best source. Alternatively, the requester can send a request with the content name for finding a content copy by name-based routing. Then, content found is sent back to the requester. In PSIRP, contents are published into the network but publications receive a particular Name Scope. Users can subscribe to contents. Publications and subscriptions are linked by a rendezvous system. The scope identifier requested and the rendezvous identifier form the name of the content. By a matching procedure, the corresponding forwarding identifier is sent to the content source. Then, the content is sent to the requester. In CCN, contents are published at servers and nodes, and routing protocols are used to distribute the content location information. Requests are forwarded toward a publisher location. CCN router maintains a Pending Interest Table (PIT) for outstanding requests. PIT maintains this state for all requests and maps them to the requester network interfaces. Contents are then sent to the requester interfaces. CCN can perform on-path caching: when a content arrives at a router, this router can cache a content copy. It allows subsequent received requests for that content to be answered from that cache. While the namespace of DONA, NetInf and PSIRP are flat and names are not human-readable, the CCN namespace is hierarchical and the names can be human-readable. Flat namespace allows persistent names while the hierarchical one is IP compatible. With flat namespace, the routing is structured and the control overhead is low. With hierarchical namespace, the routing is unstructured based on flooding and the control overhead is high.

B. Breadcrumbs-based CON

We particularly focus on Breadcrumbs [3][11] which has been designed to reduce server loads and to form an autonomous CON in cooperation with cached contents. The network is a cache network where routers can cache contents and manage a table of BC entries which are guidance information to a node holding the corresponding content. Note that in our research, actually, not core nodes but edge nodes including STBs or terminals only have content caches for higher feasibility, though this limitation can be removed easily. When a content passes through a router, this router creates in its BC table a BC entry corresponding to the content as shown in Figure 1. A BC (BC entry) is data containing the content ID, the next node and the previous node on the content path, and the most recent time at which the content was requested and was forwarded via this router. BC is used for in-network guiding of request. Nodes information in BC is used to route requests. Time information is used to manage BCs in BC table and delete

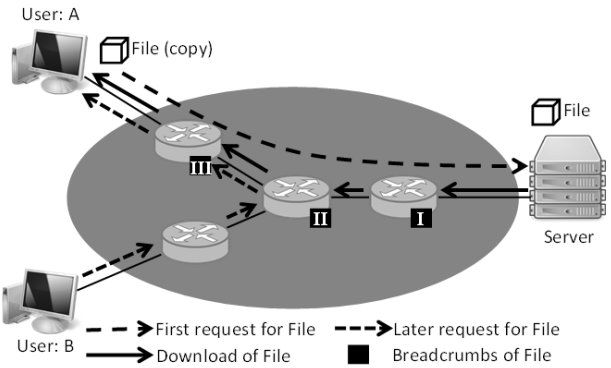


Figure 1. BC system overview

the oldest ones (since the last time update if any). When a request is created at a user node, its destination is set to a server containing the desired content in an ideal case. On its path, if the request encounters a router where a BC corresponding to the desired content exists, the router will redirect the request to the direction of the next node indicated by the BC entry, and the subsequent BC trail, series of BC entries, will guide the request until it finds the content in a cache. If a problem occurs during this redirection (content not cached at BC trail destination, lack of BC entry in the BC trail), the request is redirected to its initial server by IP routing while invalidating the whole corresponding BC entries. Namely, through tracing a series of BC entries, a request can follow the content downloaded previously. Some advantages are that the server loads are reduced and that there is no need to implement coordination protocol for cached contents. Also, it combines IP-routing for the first destination of request and BC trail routing when a right BC is found on path by requests. In terms of feasibility and scalability, BC is very interesting. It combines location-based routing and content name-based routing. Moreover, since location-based routing is the default routing system, BC can work in a partial deployment scenario allowing incremental deployment in the network. It has been demonstrated that this partial deployment is feasible but the performances highly depend on the deployment proportion [8]. Nevertheless, it has been shown that overlay can be used to improve these performances too.

C. Unknown contents search feature in CON

Regarding the keyword-based search feature for CON, some approaches have been proposed. It is important to add this feature in CON because the current web search engines use centralized data centers, and they cannot access caches. Hence, some advantages due to basic concepts of CON are not used. One approach to provide such a feature is to implement a system similar to typical multimedia search engines into CCN [9]. This system searches the contents similar to the content the user includes inside its request. When a search by content name is performed, the search interest is flooded over the network. Each node sharing searchable content performs a feature extraction task: a feature vector containing information about the content characteristics is extracted for each content, and an index is

 ISBN: 978-1-61208-377-3

formed for each content type. When a request for similar contents is received by a node, it performs similarity search by comparing the request descriptors (the feature vector of the content requested) against the index to find a set of the most similar contents. When a node has similar contents, a new content is created: it is a collection of corresponding CCN links constituted by a label name for the similar object descriptors and a target name for the CCN name. An interest is sent to the requester to inform him about its availability. He requests the collection objects from each interest received. Then, data packets carrying the collections of names of similar objects are sent to the requester. This approach seems to be extendable to a keyword-based search feature but it does not seem feasible for a network such as the current Internet because of the flooding of messages all over the network. Another approach uses keywords as a secondary content identifier by converting them to IDs. Independent Search and Merge (ISM) and Integrated Keywords Search (IKS) are two solutions based on the same general settings [10]. Content is identified by its ID, its location and its list of keywords. Each keyword has an ID by using a pre-defined hash function. The whole content IDs and keyword IDs form the general IDs. Another hash function maps general IDs to an IP address. Each node has a content search table which stores the mapping information between content ID and content location. When a user requests a content by its ID, the hash function generates the destination IP address. Once a node received a content request, it adds the new mapping entry in its content search table. Also, each node has a keyword search table which contains keyword IDs (the ones which generate the node address by the hash function) and the list of the corresponding content IDs. With ISM, for each keyword requested, it is converted into keyword ID and then into IP address. The answers are the list of content IDs for each keyword ID. With IKS, the keywords requested are first sorted and then each subset of keywords is converted into keyword ID and then into IP address. In these two methods, keywords are at the same level as content IDs. It improves the search efficiency but either results are too numerous and irrelevant (if each keyword is handled independently) or the number of keyword IDs is too large (if each subset of keyword lists has one keyword ID).

III. KEYWORD-BASED BREADCRUMBS

In order to have a feasible and scalable keyword-based search feature for CON, we introduce Keyword-based Breadcrumbs (KBC). Our goal is to add an intrinsic keyword-based search feature to BC system while preserving the BC advantages in terms of simplicity, scalability, feasibility and working. For this purpose, we add elements to BC system to allow two ways of working: the standard working using content name-based request and sending back of content, and a new one using keyword-based request, where KBC entries are used to find other contents in other location than server, and where answers are information about content and not the content itself. To distinguish BC system and KBC system, BC entry will be renamed to KBC entry from now when it concerns KBC system.

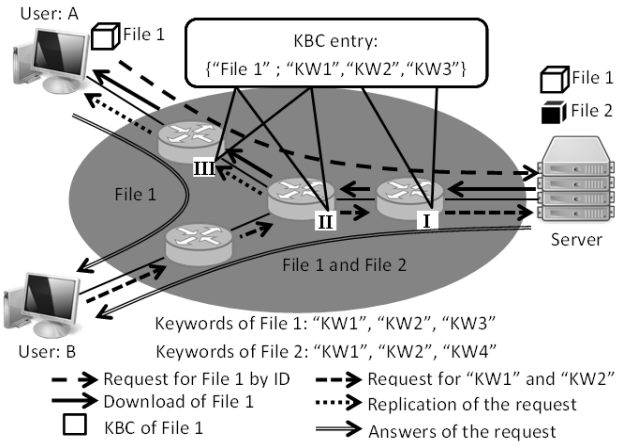


Figure 2. Keyword-based Breadcrumbs system overview

A. Principle of the Keyword-Based Search Feature

The basic idea is to use KBC to find closest corresponding contents. In the initial state, there are no cached contents and no guidance information. When a content is downloaded, KBCs are created on-path like in the BC system. The difference appears for a keywords-based request. For the KBC request, the first destination remains a server. If the request reaches a node with one or more KBC entries whose keywords correspond to the requested ones, the request will be replicated as shown in Figure 2. Replicated requests follow their KBC trail while the original request continues its path to the server. Then, when a right content is found, an answer containing the content ID, its list of keywords and its location is sent back to the requester. By this method, the requester can get a large number of answers with information for choosing the one he wants and if there are several identical contents, he can select the closest one. Also, IP-routing is used for downloading a content found by such a request because the answer gives the content ID and the location, and so performing another BC request for this content ID is unnecessary.

B. Specificities of KBC

In the proposed KBC system, we created new messages type: requests by keywords (KBC request, in opposition to BC request for a request by content ID) and answer (to KBC request because for BC request the answer is the content itself). We set rules for managing the behavior of KBC request. Also, some additions have been done to nodes to allow the use of keywords. As described previously, content has its list of keywords in addition to its ID for the creation of KBCs. Each server contains contents and a server table which contains some of its closest other servers. This information is used to redirect KBC request for having enough answers. KBC entry contains the content ID, the content keywords, the next node and the previous node on the content path, and the most recent time the content was requested by its ID and was seen at this node. Time information is used to manage KBC in KBC table. If a KBC timer reaches the time out limit because of inactivity, it is deleted. Routers have a KBC table and a KBC request table

 ISBN: 978-1-61208-377-3

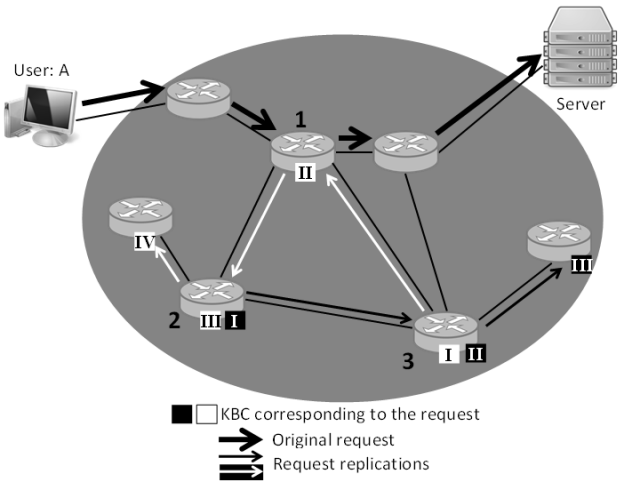


Figure 3. KBC request replications loop

containing the request IDs of recent KBC requests that went through them. This table exists to avoid an issue of KBC request replications loop. This issue happens when a triangle of routers is as follows: one KBC trail follows two edges and another one with the same keywords list follows the last edge in the same way. Figure 3 presents such a situation. The two nodes containing these two KBC entries (nodes 2 and 3) create KBC request replication whenever KBC request for the corresponding keywords list goes in. In node 2, a replication is made for the black KBC trail, and in node 3 a replication is made for the white KBC trail which will go again in node 2 via node 1, and so on. A KBC request contains the list of keywords set by the requester, a request ID for managing answers and for avoiding the previous issue, and the last node ID on its path. This information is used to optimize some request replications. When a request follows even partially a KBC trail on its reverse path, each router will replicate the request to follow this KBC trail. Then, a lot of replications are useless. Only the first one is enough, others are flooding the network. Hence, if the next node shown in KBC is equal to the previous node on the request path, the request is not replicated. An answer contains the content ID and its list of keywords for allowing the user to know if this answer corresponds to the content he wants. Also, it contains the request ID for linking the answer to its request. And it contains the location of the content for allowing the requester to select the closest one he wants between several identical contents. By this addition, the request for a content found by a KBC search will not perform another search in the network (BC request) because this work was already done with the KBC search. Note that an answer must not contain the content itself. The goal is to search corresponding contents, but the user has to select the content(s) he wants from the answers list before the download. Thus, answers to KBC requests are only information about contents and not the contents themselves.

C. KBC request settings

An important challenge for keyword-based search feature in CON is to be efficient while not overloading the network and limiting the messages flooding. We propose here two KBC request settings to manage their behavior. As explained previously, a request needs a server destination at its creation. In "1 Server", the KBC request is sent to one server only, but we set a threshold of minimum number of answers found in server. If this threshold is not reached, the request is redirected to another server which was not reached yet by the request, thanks to the servers table. In "1 Server Extended", we keep the settings from "1 Server" but we propose to add new information in contents and KBCs, the origin server location of the content. If a KBC request finds a KBC entry whose origin server is not one of the destination servers, a request replication is created to go to the new server. We add also in KBC request a list of destination servers which is updated at each replication for a server to avoid useless replications. We add to server a request ID table to avoid several answers for the same KBC request ID (the server sends answers for a KBC request only one time for each KBC request ID).

D. Comparison between KBC and other unknown contents search features

KBC has the fact that keyword is at the same level as content name in common with ISM, IKS, and similarity content search. ISM looks like KBC with one notable exception, the entry used to search contents contains one keyword and a list of contents associated, while in KBC only the KBC table is used and so KBC contains one content name and its keywords list. This difference makes more complicated KBC search in routers but it avoids the large number of irrelevant answers in ISM. IKS being close to ISM, the same comparison can be made. However, IKS solves the irrelevant answers issue by giving to each subset of keywords list a unique ID. It causes a feasibility problem because the number of keyword IDs is too large to be implemented. The similarity content search does not use keywords but descriptors to find similar contents. It is why we focus on comparable elements in the search mechanisms. Requests flood the network to find right contents. Unlike in KBC, only cache nodes have the indexes used for the search. Hence, the search is less structured than in KBC.

IV. EVALUATION

A. Simulation Scenario

To evaluate KBC, we use a modified version of Breadcrumbs+ (BC+) simulator for implementing KBC. Hence, BC+ with adaptive invalidation is used instead of BC [11]. It is an improvement of BC to avoid the issue in which some requests cannot reach the intended content in a particular situation. The differences with BC are that a BC+ entry has a list of the previous nodes on the content path instead of the previous one only, and if at the end of a BC trail, content is replaced or cannot be cached, an invalidation message is sent to all the nodes in this previous nodes list.

- Network Topology: To evaluate the proposed KBC, we use a flat router-network based on the Waxman model on a lattice points of 1000x1000, α=0.1 and β=0.05 [12]. There are 1000 routers, 5000 users and

 ISBN: 978-1-61208-377-3

50 servers. Each router is connected to five users and the server locations are chosen according to uniformly random distribution. Regarding caches, only edge nodes including STBs or terminals have content caches for higher feasibility, though this limitation can be removed easily. Each cache can have a maximum of two contents.

- Keywords: For evaluating KBC, we set three types of keywords (KW1, KW2 and KW3) which are hierarchically linked. All contents and requests contain one of each previous type of keywords (1 KW1, 1 KW2 and 1 KW3). In KBC system, a KBC request is initially routed toward a server. Keyword types are hierarchical for practicability of the initial routing. KW1 represents the main characteristic of the content (video, audio, etc.). Only keywords belonging to a single KW1 are used. KW2 represents a sub-domain of KW1 (if KW1 is "Video", KW2 can be "Action", "News", "Sports"…). There are 25 different keywords for KW2. KW3 is a more specific keyword describing more precisely the content. For each KW2, there are four keywords possible for KW3. In total, 100 keywords combinations are possible.
- Contents: Servers contain in total 10,000 contents which are all unique by their content ID and which are all defined by three random keywords (one of each keyword type). Hence, each keyword combination corresponds to around 100 contents. Also, servers have the same contents during all the simulation time and for each simulation.
- Servers: Each server has a set of its three nearest server neighbors to redirect the requests in the situation where the threshold number of answers from servers is not reached. Servers have also a list of request IDs of requests went to them. We did not set a size for this set. However, it can be easily done by setting a time out to entries.
- KBC table: It does not have limitation about its size but information about its size is collected during simulations.
- Requests: The two types of user request (by content ID and by keywords) are generated at an independent, identical and exponentially-distributed random interval. In a first time, 50,000 BC requests by content ID are made for initializing the network and for spreading KBCs. Then, we study KBC system for 55,000 BC requests and a variable number of KBC requests depending on the wanted ratio between these two request types.
- Answers: When answers for KBC requests are received, one of them is selected to download the content by IP routing.

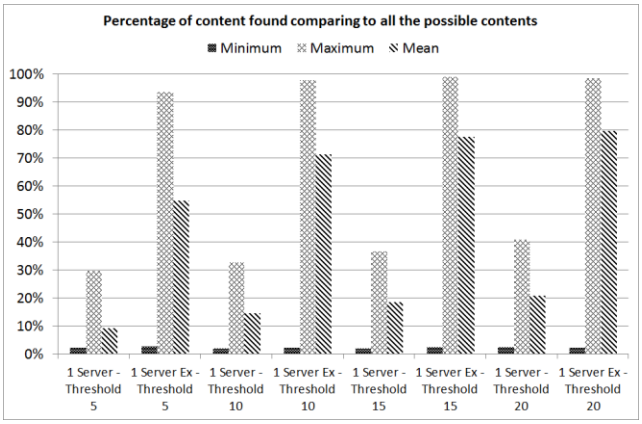


Figure 4. Content retrieval efficiency for KBC requests

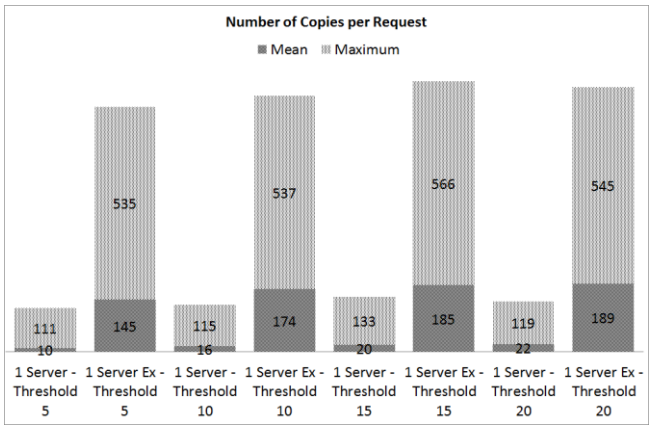


Figure 5. KBC request replications

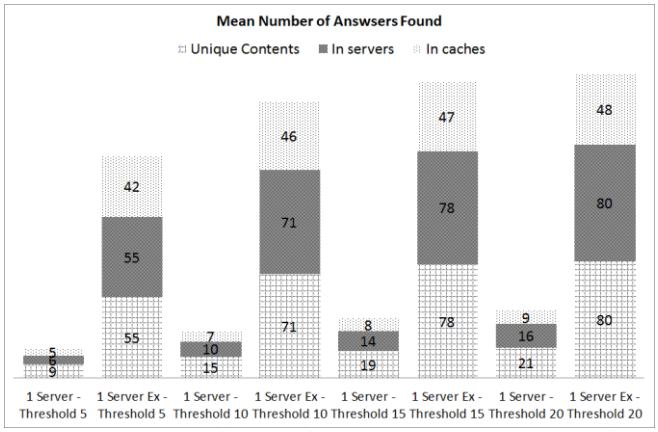


Figure 6. Repartition of answers from server or cache, and number of unique contents found (without taking into account identical contents)

We have four requests patterns to switch between BC requests and KBC requests. For 1 KBC request, 2 BC requests are performed (2 BC), 4 BC requests are performed (4 BC), 10 BC requests are performed (10 BC) or 15 BC requests are performed (15 BC). Also, we set four thresholds for the minimum number of answers found in servers: 5, 10, 15 and 20. For each threshold value, we take the mean of the results of each different requests pattern to focus on the threshold values.

B. *Performances*

Figure 4 presents the efficiency of KBC system for retrieving right contents. The setting 1 Server has limited performances because requests are restrained to a close area

TABLE I. KBC TABLE SIZE

Mean size of KBC table	Unbiased variance of the BC table size	Standard deviation of the BC table size
71	712	28

of the first destination server. 1 Server Extended shows high efficiency for finding different but right contents.

Figure 5 shows the number of KBC request replications. With 1 Server, requests are replicated only few times in mean, which means that it does not degrade the network performances. With 1 Server Extended, replications are numerous and can interfere with a good network working.

The repartition of answers between caches and servers shown in Figure 6 is also interesting because it indicates how many KBC trails are successfully followed. Once again in 1 Server, the results are low. On the other hand, 1 Server Extended can find a lot of corresponding KBC trails even if the threshold in server is low. In a small network area, KBC requests can easily find a KBC about a content from outside of this area. Hence with 1 Server Extended, KBC requests can go all over the network. It is confirmed by the equality between the number of different contents found (Unique Contents) and the number of contents found in servers.

Regarding the KBC tables, no limit was set because the needed size is important to know. Table I presents the mean size of KBC table with its unbiased variance and its standard deviation. Viewing these results, we can propose to have a KBC table of 100 entries, which is 1/100 of all contents in our simulated network.

V. CONCLUSION

We presented in this paper a keyword-based search feature using Breadcrumbs and some keyword-based request settings to control their behavior. The purpose is to make CON easier by a feature similar to web search engines from users' point of view. Our system is different from other approaches because it does not flood the network at each request. It is scalable not only in CON but also in partially deployed Breadcrumbs because keyword-based search is close in its working to content name-based one, and thanks to Breadcrumbs characteristics. We showed that the setting 1 Server Extended has a good potential even if there is a trade-off between the network flooding and the search efficiency.

In our future work, we want to use other keyword settings, and change our network for having non unique contents. Also, we will take into account the content popularity, and we want to implement an indicator of users' satisfaction (if a content is downloaded thanks to a keyword-based search, it means that for the keyword list used, the user is satisfied of this content).

ACKNOWLEDGMENT

This research was supported in part by National Institute of Information and Communication Technology (NICT), Japan.

REFERENCES

[1] J. Choi, J. Han, E. Cho, K. Kwon, and Y. Choi, "A Survey on Content-Oriented Networkinf for Efficient Content Delivery," IEEE Communications Magazine, vol.49, no.3, Mar. 2011, pp.121-127.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-Centric Networking," IEEE Communications Magazine, vol.50, no.7, Jul. 2012, pp.26-36.

[3] E. Rosensweig and J. Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," in Proc. IEEE INFOCOM 2009, Apr. 2009, pp.2631-2635.

[4] V. Jacobson, et al., "Networking named content," in Proc. ACM CoNEXT 2009, Dec. 2009, pp.1-12.

[5] T. Koponen, et al., "A Data-Oriented (and Beyond) Network Architecture," in Proc. SIGCOMM '07, Aug. 2007, pp. 181-192.

[6] B. Ahlgren, et al., "Second NetInf Architecture Description," Deliverable D-6.2 in The Network of the Future - FP7-ICT-2007-1-216041, Apr. 2010.

[7] M. Ain, et al., "Architecture Definition, Component Descriptions, and Requirements," Deliverable D2.3 in Publish-Subscribe Internet Routing Paradigm - FP7-INFSO-IST-216173, Feb. 2009.

[8] T. Tsutsui, H. Urabayashi, M. Yamamoto, E . Rosenweig, and J. Kurose, "Performance Evaluation of Partial Deployment of Breadcrumbs in Content Oriented Networks," in Proc. IEEE ICC FutureNet 2012, Jun. 2012, pp.5828-5832.

[9] P. Daras, T. Semertzidis, L. Makris, and M. Strintzis. "Similarity Content Search in Content Centric Networks," in Proc. ACM MM '10, Oct. 2010, pp.775-778.

[10] Y. Mao, B. Sheng, and M. Chuah, "Scalable Keyword-Based Data Retrievals in Future Content-Centric Networks," in Proc. 2012 Eighth International Conference on Mobile Ad-hoc and Sensor Networks (MSN) , Dec. 2012, pp.116-123.

[11] M. Kakida, Y. Tanigawa, and H. Tode, "Breadcrumbs+: Some Extensions of Breadcrumbs for In-network Guidance for Inter-AS Content-Oriented Network Topology," in Proc. WTC 2012, Mar. 2012, pp. 1283-1292.

[12] B. M. Waxman, "Routing of Multipoint Connections," IEEE J. Sel. Areas Comms (Special Issue on Broadband Packet Communication), vol.6, no.9, Dec. 1998, pp.1617-1622.

 ISBN: 978-1-61208-377-3

Low-Power and Shorter-Delay Sensor Data Transmission Protocol in Mobile Wireless Sensor Networks

Sho Kumagai and Hiroaki Higaki

Department of Robotics and Mechatronics, Tokyo Denki University, Japan
Email: {kuma,hig}@higlab.net

***Abstract*—In a sensor network, sensor data messages reach the nearest stationary sink node connected to the Internet by wireless multihop transmissions. Recently, various mobile sensors are available due to advances of robotics technologies and communication technologies. A location based message-by-message routing protocol, such as Geographic Distance Routing (GEDIR) is suitable for such mobile wireless networks; however, it is required for each mobile wireless sensor node to know the current locations of all its neighbor nodes. On the other hand, various intermittent communication methods for a low power consumption requirement have been proposed for wireless sensor networks. Intermittent Receiver-driven Data Transmission (IRDT) is one of the most efficient methods; however, it is difficult to combine the location based routing and the intermittent communication. In order to solve this problem, this paper proposes a probabilistic approach with the help of one of the solutions of the secretaries problem. Here, each time a neighbor sensor node wakes up from its sleep mode, an intermediate sensor node determines whether it forwards its buffered sensor data messages to it or not based on an estimation of achieved pseudo speed of the messages. Simulation experiments show that the proposed probabilistic method achieves shorter transmission delay than the two naive combinations of IRDT and GEDIR in sensor networks with mobile sensor nodes and a stationary sink node.**

***Keywords*–Wireless Sensor Networks, Routing Protocol, Intermittent Communication, Low Power Consumption, Mobile Sensor Nodes, Probabilistic Approach.**

I. INTRODUCTION

A sensor network is anticipated to play an important role of a fundamental infrastructure for Internet of Things (IoT) and the big data support. A sensor network consists of multiple wireless sensor nodes and a stationary sink node connected to the Internet. Sensor data messages are transmitted along a wireless multihop transmission route which is a sequence of wireless sensor nodes to the sink node. Then, the sensor data messages reach a dedicated server computer through the Internet [1]. Since only limited battery capacity is available in each sensor node, it is not reasonable for each sensor node to transmit sensor data messages directly to the sink node. Hence, each sensor node transmits sensor data messages to one of its neighbor nodes within its wireless signal transmission range. In order for the sensor data messages to reach the sink node, intermediate sensor nodes forward the received sensor data messages. For such wireless multihop transmissions, various ad-hoc routing protocols have been proposed [9]. In most of such routing protocols, it is assumed that all wireless nodes are always active; i.e., the wireless nodes can send and receive data messages anytime. However, in wireless sensor networks, due to limitation of battery capacity and difficulty for continuous power supply, low-power communication is required. Especially, for support of mobile wireless sensor networks, such as mobile robot networks with various sensors, human centric sensor networks and vehicle-mounted sensor networks for Intelligent Transport Systems (ITS), the low-power consumption requirement is serious.

Intermittent communication technique is widely introduced in sensor networks for reduction of power consumption. In each wireless sensor node, its wireless communication module should be active when it observes objects and creates sensor data messages as a source sensor node and when it forwards sensor data messages in transmission as an intermediate sensor node. Otherwise, i.e., while the wireless sensor node is not engaged in any sensor data transmissions, it gets in its sleep mode to reduce its battery consumption for longer lifetime. In order to realize the intermittent communication, it is difficult for each intermediate sensor node to synchronize with its previous- and next-hop sensor nodes. In a source sensor node, its wireless communication module is required to be active only after the sensor node observes certain objects and achieves its sensor data. Hence, it simply enters its active mode. On the other hand, in an intermediate wireless sensor node, it is required to be active before it receives sensor data messages from one of its neighbor sensor nodes. Hence, it is difficult for the intermediate wireless node to determine when it gets in its active mode.

Intermittent Receiver-driven Data Transmission (IRDT) is an asynchronous intermittent communication protocol for sensor networks [4]. In IRDT, an intermediate wireless sensor node with sensor data messages in transmission waits for its next-hop neighbor wireless sensor node to be active without continuous transmissions of control messages which is required in various Low Power Listening (LPL) [6] protocols. Though it is a power-efficient communication method, it is difficult for conventional ad-hoc routing protocols to be applied since the protocols are designed to support only wireless networks consisting of always-on stationary wireless sensor nodes. In order to realize power-efficient routing with intermittent communication in wireless sensor networks, this paper proposes IRDT-GEDIR under an assumption that a location acquisition device, such as a GPS module is in each sensor node. IRDT-GEDIR is a combination of IRDT and a well-known location-based greedy ad-hoc routing protocol Geographic Distance Routing (GEDIR) [8]. GEDIR is based on the message-by-message routing, which is suitable for various sensor networks where short sensor data messages are usually transmitted and especially for dynamic sensor networks whose topology is not stable due to mobility of sensor nodes

 ISBN: 978-1-61208-377-3

and their removal caused by battery consumption and failure. An asynchronous intermittent communication reduces power consumption; however, the transmission delay of sensor data messages usually gets longer by synchronization overhead in each intermediate sensor node with its previous- and next-hop sensor nodes. In addition, for combination of IRDT and GEDIR, location acquisition overhead for next-hop selection is not negligible in mobile wireless sensor networks. In IRDT-GEDIR, introduction of a novel probabilistic next-hop selection method reduces the transmission delay of data messages.

This paper is organized as follows: Section II shows the related works for intemittent sensor data transmission protocols. In Section III, we propose IRDT-GEDIR which combines intermittent sensor data transmissions and a geographical ad-hoc routing protocol. Section IV evaluates the performance of IRDT-GEDIR. Section V concludes this paper and shows the future works.

II. Related Works

Battery capacity in sensor nodes consisting of wireless sensor networks is limited and usually there is no continuous power supply to them. Hence, intermittent communication is introduced where sensor nodes switch between their active and sleep modes [11]. Their communication module works only in the active modes. In order for sensor data messages to be transmitted to the sink node along a wireless multihop transmission route, each intermediate sensor node should be in the active mode when its previous-hop node forwards a sensor data message. Such intermittent communication methods are classified into synchronous and asynchronous. In the synchronous methods, all the sensor nodes are closely synchronized and each sensor node transmits sensor data messages according to a predetermined schedule as in Traffic-Adaptive Medium Access Protocol (TRAMA) [10] and Lightweight Medium Access Protocol (LMAC) [5]. However, they are based on the close synchronization among sensor nodes which requires frequent exchange of control messages as the distributed clock synchronization protocols [3]. Even though the required clock synchronization overhead is acceptable, additional control messages are required to be transmitted to update their sleep-wakeup schedules consistently to follow the unstable network topology due to the mobility of the wireless sensor nodes.

On the other hand, in the asynchronous methods, synchronization among neighbor nodes is required only when a sensor node forwards a sensor data message to its next-hop sensor node. In LPL [6], when a sensor node requests to transmit a sensor data message to its next-hop sensor node, it continues transmissions of a preamble message during a mode switching interval and all its neighbor nodes receiving the preamble message should be in an active mode even if they are not the next-hop sensor node as shown in Figure 1. In IRDT [4], a current-hop sensor node N_c waits for receipt of a polling message from its next-hop sensor node N_n as in Figure 2. Every sensor node switches between its active and sleep modes in the same interval and broadcasts a polling message with its ID each time when it changes its mode active. Then, it waits for a transmission request message $Sreq$ from its previous-hop node in its active mode. If it does not receive $Sreq$, it goes into its sleep mode. Otherwise, i.e., if N_c receives a polling message from N_n which enters its active mode and transmits $Sreq$ to N_n with its ID, N_n transmits an acknowledgement message $Rack$ back to N_c and a virtual connection is established between them. Then, data messages are transmitted from N_c to N_n. Different from LPL, a current-hop node N_c does not transmit a preamble message continuously but only waits for receipt of a polling message in IRDT. Therefore, low-overhead, i.e., low battery consuming intermittent communication among wireless sensor nodes is realized.

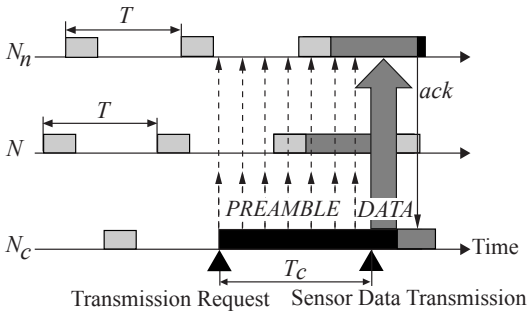


Figure 1. LPL Intermittent Communication.

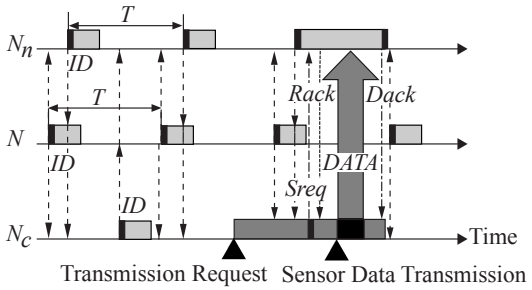


Figure 2. IRDT Intermittent Communication.

In [7], a wireless multihop routing protocol for IRDT-based sensor networks has been proposed. It is a proactive routing protocol where each sensor node keeps its routing table for the shortest transmission route to a sink node up-to-date. In order for the sensor nodes to determine their next-hop neighbor sensor node, a flooding of a control message initiated by the sink node is applied. Though it works well in usual ad-hoc networks consisting of always-on mobile nodes, it is difficult for sensor networks with intermittent communication since a control message is not always received by all the neighbor sensor nodes due to their sleep mode. Thus, the control message is required to be retransmitted. Hence, in the worst case, a sensor node unicasts the control message to all its neighbor nodes one by one. In addition, in order to support mobile wireless sensor networks, it is difficult for proactive routing protocols to keep the routing tables consistent to the current network topology especially with the intermittent communication among the mobile sensor nodes.

III. Proposal

A. Next-Hop Selection

As discussed in the previous section, for wireless multihop transmissions of sensor data messages to reach a stationary sink node with the intermittent communication in mobile

wireless sensor nodes, a novel routing protocol is required to be developed. In order to reduce the communication overhead and transmission delay for sensor data message transmissions with intermittent communication, this paper proposes a combination IRDT-GEDIR of IRDT and GEDIR [8] which is one of the well-known location-based ad-hoc routing protocols with low communication overhead for synchronization among sensor nodes. GEDIR is a message-by-message based routing protocol. That is, an intermediate node determines its next-hop node for each data message according to the most up-to-date locations of itself, its neighbor nodes and the destination node. Each sensor node with a GPS-like location acquisition device broadcasts its current location information in a certain interval and thus it achieves location information of its neighbor nodes. The original GEDIR is designed for always-on wireless nodes and the broadcasted location information is surely received by all the neighbor nodes. Only the localized information, i.e., location information of not all but only neighbor nodes, is required to determine its next-hop node according to the following method.

[Next-Hop Selection in GEDIR]
An intermediate wireless sensor node N_c selects one of its neighbor sensor node N_n as its next-hop node where the distance $d_n = |N_nS|$ to the sink node S is the shortest among all its neighbor sensor nodes as shown in Figure 3. □

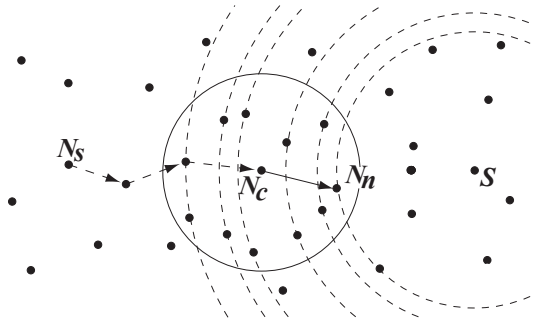


Figure 3. GEDIR Overview.

In IRDT, each sensor node transmits a polling message each time it enters its active mode. Thus, by piggybacking its location information to the polling message as in Figure 4, its location information is broadcasted without additional communication overhead and notified to its possible previous-hop nodes. However, the polling message is not surely received by all its neighbor sensor nodes since they might be in their sleep mode where their network interfaces do not work. If the sensor nodes are stationary, a neighbor node which receives the polling message by chance holds the location information and uses it for its next-hop determination. However, in a mobile sensor network, the achieved location information gets stale and the most up-to-date location information is required for the next-hop selection.

An intermediate sensor node N_c requires location information of its neighbor nodes only when it has a sensor data message to be transmitted to the sink node through its next-hop sensor node. Thus, in our proposal, based on the location information piggybacked to the received polling messages, N_c determines its next-hop sensor node. Here, since a neighbor sensor node N waits for receiving an $Sreq$ message only for a

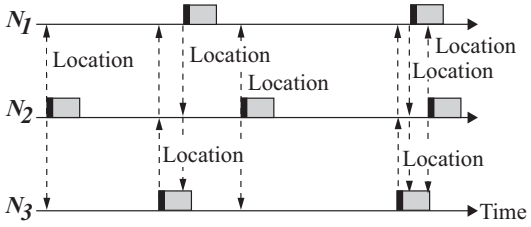


Figure 4. Location Information Propagation by Polling Messages.

predetermined interval after transmission of a polling message from N, N_c should determine during this interval whether it selects N as its next-hop node or not.

In order to solve this problem, according to a certain criterion, N_c evaluates N and compares the evaluation result and an expected evaluation where one of the later activating neighbor sensor nodes are selected as its next-hop node. In GEDIR, the distance to the destination sink node is applied as the criterion for selection of its next-hop node for achieving shorter transmission route to the sink node. On the other hand in IRDT-GEDIR, since wireless sensor nodes communicate intermittently, forwarding to the neighbor sensor node nearest to the destination sink node does not always reduce the transmission delay. Even when a sensor node N is not the nearest to the sink node, shorter transmission delay might be achieved by forwarding it to N being active currently. Thus, this paper introduces a novel criterion *pseudo speed* of sensor data message transmission which is achieved by division of difference of distance to the sink node S, i.e., $|N_cS|-|NS|$, by the time duration between the transmission request and receipt of the polling message as shown in Figure 5. It is a reasonable criterion for selection of a next-hop sensor node in intermittent communication environments for shorter transmission delay to the sink node.

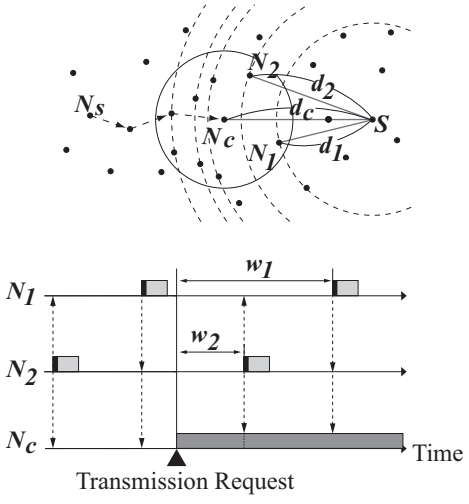


Figure 5. Next-Hop Selection based on Pseudo Speed.

Due to IRDT intermittent communication, an intermediate

sensor node N_c should determine whether it selects a neighbor sensor node N as its next-hop node soon after it receives a polling message from N since N_c should transmits an $Sreq$ message to N while N is in its active mode. That is, N_c cannot compare all pseudo speed sv_i each of which is achieved in case that N_c forwards a sensor data message to a neighbor node N_i since each sv_i is only achieved when N_i wakes up and broadcasts its polling message containing its current location information. This is almost the same setting as in the secretaries problem [2].

The secretaries problem is one of the famous problems of the optimal stopping theory. It has been studied extensively in the fields of applied probability, statistics and decision theory. The basic form of the problem is as follows:

- An administrator is willing to hire the best secretary out of n rankable candidates.
- The candidates are interviewed one by one in an random order.
- A decision about each particular candidate is to be taken immediately after the interview.
- Once rejected, a candidate cannot be recalled.
- During the interview, the administrator can rank the candidate among all candidates interviewed so far; however, it cannot rank the candidate among unseen forthcoming candidates.
- The problem is about the optimal strategy to maximize the expectation of the rank of the selected candidate.

In our next-hop selection, neighbor nodes get active one by one and an intermediate sensor node with sensor data messages in transmission can evaluate the pseudo speed of data messages to them at that time. It should immediately determine whether it selects the currently active neighbor node as its next-hop node or not even though it cannot evaluate the pseudo speed of data messages to the forthcoming active neighbor nodes. Thus, the solution of our next-hop selection problem is expected to be achieved based on the secretaries problem.

N_c evaluates the pseudo speed sv where it forwards a sensor data message to N from which N_c receives a polling message and the expected pseudo speed $\overline{sv}$ where it forwards it not to N but to one of the later activating sensor nodes. If $sv > \overline{sv}$, N_c transmits an $Sreq$ message to N; i.e., it selects N as its next-hop node. Otherwise, i.e., $sv < \overline{sv}$, N_c does not transmit an $Sreq$.

B. Expectation of Pseudo Speed

In the proposed method in the previous subsection, an intermediate sensor node determines whether it forwards a sensor data message to a currently active neighbor sensor node from which it receives a polling message by comparison of pseudo speed of transmission of a data message. For the comparison, this subsection discusses the method to evaluate the expected pseudo speed of transmission of a data message in case that the intermediate node forwards the message not to the currently active neighbor node but to one of the later activating nodes. Here, let T be the constant interval of activations in sensor nodes, i.e., the interval of consecutive transmissions of polling messages and n be the number of neighbor sensor nodes of an intermediate sensor node N_c with a sensor data message in transmission.

First, we investigate the distribution of distances $|NS|$ from neighbor nodes N of N_c to the destination sink node S. As shown in Figure 6, let r, d_c and d be a wireless transmission range of N_c, the distance from N_c to S ($d_c > r$) and the distance from N to S ($d_c - r \leq d \leq d_c + r$). Under an assumption that sensor nodes are distributed with the same density, the probability $DP(d)$ where the distance $|NS|$ is shorter than d is as follows:

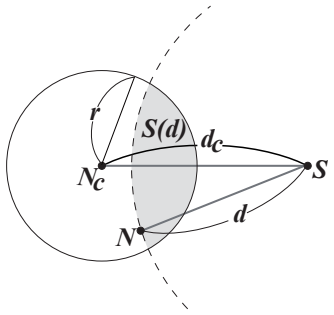


Figure 6. Area of Candidates of Next-Hop Node.

$$\begin{aligned} DP(d) &= \frac{S(d)}{\pi r^2} \\ &= \frac{2}{\pi r^2}\Big(\int_{d_c-d}^{x'} \sqrt{d^2-(x-d_c)^2}dx \\ &\qquad + \int_{x'}^{r} \sqrt{r^2-x^2}dx\Big) \end{aligned} \quad (1)$$

(where x'=$(d_c^2 + r^2 - d^2)/2d_c$)

Since $DP(d)$ is the distribution function of d, the probability density function $dp(d)$ where $|NS|$ equals to d is as follows:

$$\begin{aligned} dp(d) &= \frac{d}{dd}DP(d) \\ &= \frac{2}{\pi r^2}\frac{d}{dd}\Big(\int_{d_c-d}^{x'} \sqrt{d^2-(x-d_c)^2}dx \\ &\qquad + \int_{x'}^{r} \sqrt{r^2-x^2}dx\Big) \end{aligned} \quad (2)$$

The probability density function $p(l)$ of the reduction of distance $l = d_c - d$ to S achieved by forwarding a sensor data message from N_c to N is as follows:

$$\begin{aligned} p(l) &= dp(d_c - l) \\ &= -\frac{2}{\pi r^2}\frac{d}{dl}\Big(\int_{l}^{x''} \sqrt{(x-l)(2d_c-l-x)}dx \\ &\qquad + \int_{x''}^{r} \sqrt{r^2-x^2}dx\Big) \end{aligned} \quad (3)$$

(where x''=$((2d_c - l)l + r^2)/2d_c$)

Next, we examine the distribution of time duration from the transmission request of a sensor data message in N_c to the receipt of a polling message from N. Here, the transmission is

supposed to be requested at $t = 0$. Let t_i be the time when the ith polling message is transmitted from one of the neighbor nodes of N_c. Thus, $i-1$ neighbor sensor nodes transmit polling messages in an interval $[0, t_i)$ and the rest $n - i$ neighbor sensor nodes transmit polling messages in an interval (t_i, T). Under an assumption that the transmission time t of the polling messages from the $n-i$ neighbor sensor nodes are distributed in the interval (t_i, T) according to the unique distribution, the probability density function $pp(i, j, t)$ where jth $(i < j \leq n)$ polling message is transmitted from one of the neighbor sensor nodes of N_c at time $t \in (t_i, T)$ is as follows:

$$\begin{aligned} pp(i,j,t) &= {}_{n-i}C_{j-i-1}\left(\frac{t-t_i}{T-t_i}\right)^{j-i-1} \\ &\quad \times {}_{n-j+1}C_1 \frac{1}{T-t_i} \times \left(\frac{T-t}{T-t_i}\right)^{n-j} \\ &= {}_{n-i-1}C_{j-i-1}\frac{(n-i)(t-t_i)^{j-i-1}(T-t)^{n-j}}{(T-t_i)^{n-i}} \quad (4) \end{aligned}$$

Since the location of a neighbor sensor node and the time when it transmits a polling message are independent each other, the probability density function $g(i, j, t, l)$ where N_c transmits a sensor data message to a neighbor sensor node N which transmits the jth $(i < j \leq n)$ polling message at time t $(t_i < t < T)$ and the distance to the sink node S is reduced l by this forwarding is induced by (3) and (4) as follows:

$$g(i, j, t, l) = pp(i, j, t) \cdot p(l) \quad (5)$$

Here, the pseudo speed sv of transmissions of sensor data messages is l/t.

In case that N_c does not select a neighbor sensor node which transmits the ith polling message at t_i as its next-hop node, N_c selects another sensor node which transmits the jth $(i < j \leq n)$ polling message at t_j $(t_i < t_j < T)$ or a sensor node transmitting its second polling message after $t = T$. In the latter case, kth $(1 \leq k \leq i)$ polling messages are transmitted at t_k $(0 \leq t_k \leq t_i)$ and the distance reduction by forwarding to the neighbor node is l_k. Thus, the pseudo speed achieved by forwarding on receipt of the second polling message is $sv_k = l_k/(t_k + T)$. Since N_c has already achieved both t_k and l_k $(1 \leq k \leq i)$, the expected pseudo speed where N_c forwards a sensor data message at $t \geq T$ is as follows:

$$\overline{sv}_n = \max_{1 \leq k \leq i} sv_k = \max_{1 \leq k \leq i} \frac{l_k}{t_k + T} \quad (6)$$

This is an expected pseudo speed in case that N_c does not forward a sensor data message to a neighbor node transmitting the nth polling message. Based on (6), we evaluate the expected pseudo speed $\overline{sv}_j$ when N_c does not forward a sensor data message to a neighbor node transmitting the jth $(i \leq j \leq n)$ polling message.

In case of $j = n$, $p(l)$ and $pp(i, n, t_n)$ are defined in an area $(-r \leq l \leq r$ and $t_i < t_n < T)$ as shown in Figure 7 and $g(i, n, t_n, l)$=$pp(i, n, t_n) \cdot p(l)$. Here, the area is divided into S and S' by a line l=$\overline{sv}_n t_n$. In S, since the pseudo speed l/t_n is higher than $\overline{sv}_n$, N_c forwards a sensor data message to a neighbor node transmitting the nth polling message. On the other hand, since the pseudo speed l/t_n is lower than $\overline{sv}_n$ in S', N_c forwards a sensor data message to the node transmitting not nth but kth polling message which gives the maximum $l_k/(t_k + T)$ in (6). Therefore, $\overline{sv}_{n-1}$ is evaluated by the following formula:

$$\overline{sv}_{n-1} = \int_S \frac{l}{t_n} g(i, n, t_n, l) dS + \int_{S'} \overline{sv}_n g(i, n, t_n, l) dS' \quad (7)$$

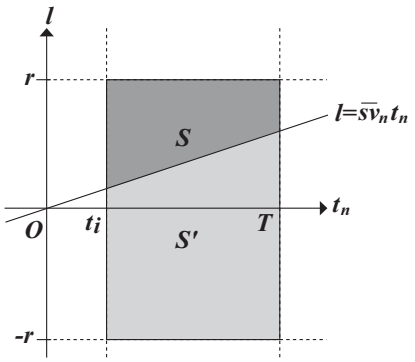


Figure 7. Expected Pseudo Speed where Transmitter of $n-1$th Polling Message is not Selected as Next-Hop Node.

Generally, the expected pseudo speed when N_c does not forward a sensor data message to a neighbor node transmitting the jth $(i \leq j < n)$ polling message is also evaluated as in the same way. That is, the area $(-r \leq l \leq r$ and $t_i < t_{j+1} < T)$ in which $g(i, j+1, t_{j+1}, l)$ is defined is divided into sub-areas S and S' by a line $l = \overline{sv}_{j+1} t_{j+1}$ as in Figure 8. In S, since

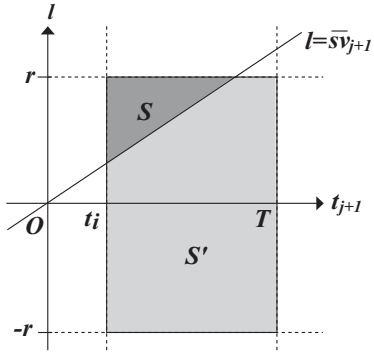


Figure 8. Expected Pseudo Speed where Transmitter of jth Polling Message is not Selected as Next-Hop Node.

the pseudo speed l/t_{j+1} is higher than $\overline{sv}_{j+1}$, N_c forwards a sensor data message to a neighbor node transmitting the $j+1$th polling message. On the other hand, since the pseudo speed l/t_{j+1} is lower than $\overline{sv}_{j+1}$ in S', N_c forwards a sensor data message to the transmitting node of not $j+1$th polling message but a later transmitted polling message. Therefore, $\overline{sv}_j$ is evaluated by the following formula:

$$\begin{aligned} \overline{sv}_j &= \int_S \frac{l}{t_{j+1}} g(i, j+1, t_{j+1}, l) dS \\ &\quad + \int_{S'} \overline{sv}_{j+1} g(i, j+1, t_{j+1}, l) dS' \quad (8) \end{aligned}$$

According to (6) and (8), N_c calculates $\overline{sv}_i$. Thus, if a neighbor sensor node N which is l_i nearer to the sink node S than N_c transmits the ith polling message at time t_i, N_c determines whether it selects N as its next-hop node as follows:

- If $l_i/t_i \geq \overline{sv}_i$, N_c forwards a sensor data message to N.
- Otherwise, i.e., if $l_i/t_i < \overline{sv}_i$, N_c does not forward a sensor data message to N.

In our proposed protocol, only ID and location information of mobile sensor nodes are piggybacked. In a wireless sensor network with stationary sensor nodes, it is enough for precisely estimate the pseudo speed of its neighbor nodes. However, in a mobile wireless sensor network, since no mobility information is piggybacked, it is impossible for an intermediate node to estimate future locations of its neighbor nodes. Thus, it may possible that the achieved locations are changed when the next polling messages are transmitted. That is, l_k might be changed and in the worst case the neighbor node goes out of the wireless transmission range of the intermediate node when it transmits the next polling message. The effect is later discussed in the performance evaluation and the conclusion sections.

IV. Evaluation

First, we evaluate the 1-hop transmission performance achieved by the proposed IRDT-GEDIR next-hop selection method. Here, pseudo speed is evaluated in IRDT-GEDIR and two conventional naive methods. A wireless transmission range of a wireless sensor node is assumed 10m and the distance from an intermediate node N_c currently holding a sensor data message to the sink node is 100m. 5–20 neighbor sensor nodes are randomly distributed in a wireless signal transmission range according to the unique distribution randomness. All sensor nodes are assumed stationary. The interval of activations in each sensor node is 1s and the initial activation time is also randomly determined. The proposed IRDT-GEDIR is compared with the following two conventional methods and an unrealistic locally optimum method;

- N_c forwards a sensor data message to the neighbor node which transmits the first polling message after the transmission request in N_c. (Greedy Conventional)
- N_c forwards a sensor data message to the neighbor node which provides the highest pseudo speed determined after receiving polling messages from all the neighbor nodes of N_c. (Conservative Conventional)
- N_c forwards a sensor data message to the neighbor node which provides the highest pseudo speed determined by the information of locations and activation times in all the neighbor nodes. (Locally Optimum)

Locally Optimum is evaluated only for comparison since it is impossible for N_c to achieve location information of its neighbor nodes without any overhead. If N_c is a dead-end node which cannot select its next-hop node, the pseudo speed is evaluated as 0m/s.

Figures 9–12 show the results of simulation experiments. Here, the value of the distribution function $f(sv) = p(sv' < sv)$ of probability where pseudo speed sv' is lower than sv. In all the results, higher pseudo speed is achieved in the order IRDT-GEDIR, Greedy Conventional and Conservative Conventional. Locally Optimum provides the ideal pseudo speed, since N_c achieves all the required information to determine its next-hop node in advance. The performance of Conservative Conventional is low since the overhead to receive all the polling messages is too high. Though the performance of Greedy Conventional and IRDT-GEDIR is almost the same in low density environments, higher pseudo speed is achieved by IRDT-GEDIR in more dense environments. In IRDT-GEDIR, no additional control messages are required to determine its next-hop nodes as discussed in the previous section. Therefore, IRDT-GEDIR is expected to realizes low-power shorter-delay transmissions of sensor data messages in intermittent wireless sensor networks.

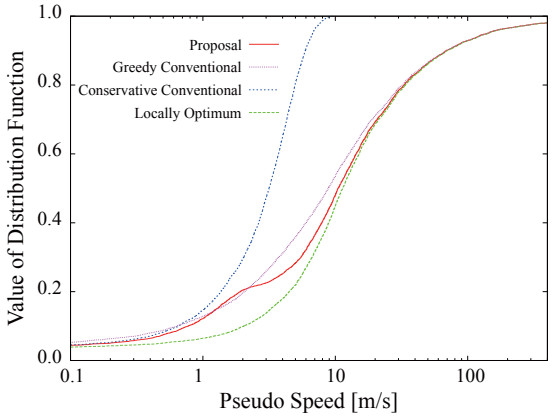


Figure 9. 1-Hop Transmission Performance (5 Neighbor Nodes).

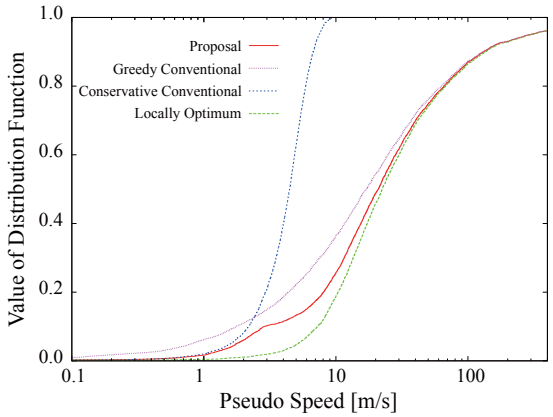


Figure 10. 1-Hop Transmission Performance (10 Neighbor Nodes).

Next, we evaluate the multihop transmission performance in mobile wireless sensor networks. In a 100m $\times$ 100m square simulation field, 1,000 mobile wireless sensor nodes with 10m wireless signal transmission range are randomly distributed according to the unique distribution randomness. It is assumed that the interval of activations in each sensor node is 1.0s, communication overhead for 1-hop transmission is 0.1s and the activation time offset is also randomly determined in each sensor node according to the unique distribution in [0s, 1s). The speed of mobile wireless nodes is 0.1–2.0m/s and their mobility is according to the Random-Way-Point model. A location of a stationary sink node is also randomly determined, which is assumed to be advertised to all the mobile sensor nodes in advance. In IRDT-GEDIR, for

 ISBN: 978-1-61208-377-3

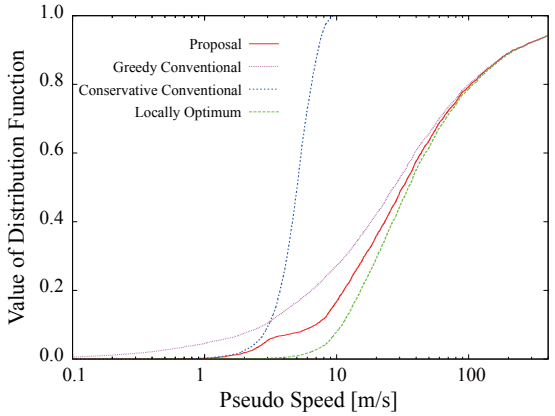


Figure 11. 1-Hop Transmission Performance (15 Neighbor Nodes).

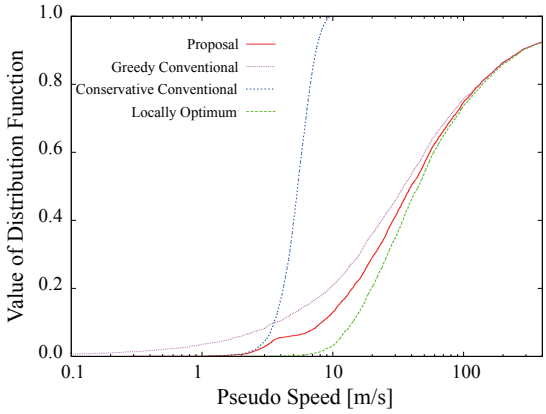


Figure 12. 1-Hop Transmission Performance (20 Neighbor Nodes).

calculation of expectation of pseudo speed, the number of neighbor nodes n is needed; however, it is difficult for an intermediate sensor nodes to determine n in an intermittent communication environment. Hence, the average number of mobile sensor nodes in its wireless transmission range is applied as n in the simulation experiments. Thus, in this experiment, $n = 1,000 \div (100 \times 100) \times (10 \times 10 \times \pi) = 31$. End-to-end transmission delay and hop counts of a sensor data message is evaluated in IRDT-GEDIR, Greedy Conventional, Conservative Conventional and Locally Optimum. Figures 13–17 and Figures 18–22 show the simulation results of 1,000 trials of end-to-end transmission delay and hop counts, respectively. The x-axis represents distances between a source mobile sensor node and the stationary sink node when the multihop transmission is initiated.

Though an intermediate sensor node transmits a sensor data message soon after it receives a polling message from one of its neighbor sensor nodes in Greedy Conventional and Locally Optimum. However, it determines its next-hop sensor node after receipt of all the polling message always in Conservative Conventional and sometimes in IRDT-GEDIR. In such cases, due to the interval between the receipt of the polling message and the transmission of a sensor data message and mobility of the sensor nodes, it may fail to forward the sensor data message if the neighbor node moves out of the wireless transmission range. In our simulation results, only Conservative Conventional fails to forward as shown in Table 1. Thus, it is not suitable especially for high speed mobility.

TABLE I. RATIO OF FORWARDING FAILURE IN CONSERVATIVE CONVENTIONAL.

Mobility Speed [m/s]	0.1	0.2	0.5	1.0	2.0
Failure Ratio [%]	15.9	26.1	64.6	74.0	88.3

As shown in Figures 13–22, independently of the mobility speed of wireless sensor nodes, all the simulation results, i.e., both end-to-end transmission delay and hop counts are proportional to the distance between a source sensor node to the destination sink node. The order of transmission delay is Locally Optimum, IRDT-GEDIR, Greedy Conventional and Conservative Conventional and the order of hop counts is Conservative Conventional, Locally Optimum, IRDT-GEDIR and Greedy Conventional. Though Conservative Conventional achieves the smallest hop counts, which means the lowest power consumption transmissions are realized, it requires too long transmission delay and suffers too high transmission failure ratio. The relation among Locally Optimum, IRDT-GEDIR and Greedy Conventional is almost the same in all the results. In IRDT-GEDIR and Greedy Conventional, 18.56% and 23.06% additional transmission delay and 21.70% and 35.64% additional hop counts are required to those of Locally Optimum. Hence, IRDT-GEDIR achieves improvement in both power consumption and end-to-end transmission delay.

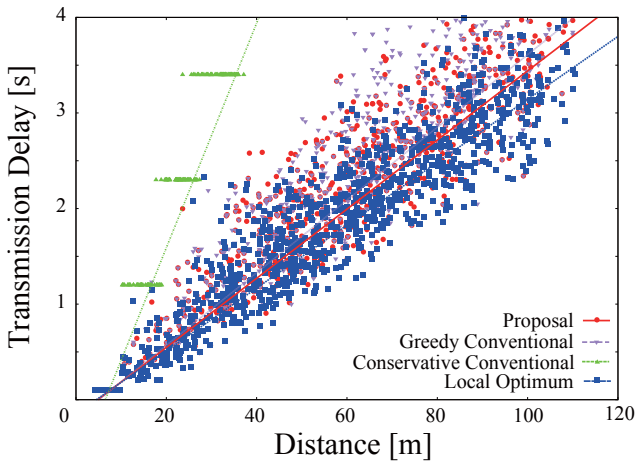


Figure 13. End-to-End Delay in Wireless Multihop Transmissions (0.1 m/s).

 ISBN: 978-1-61208-377-3

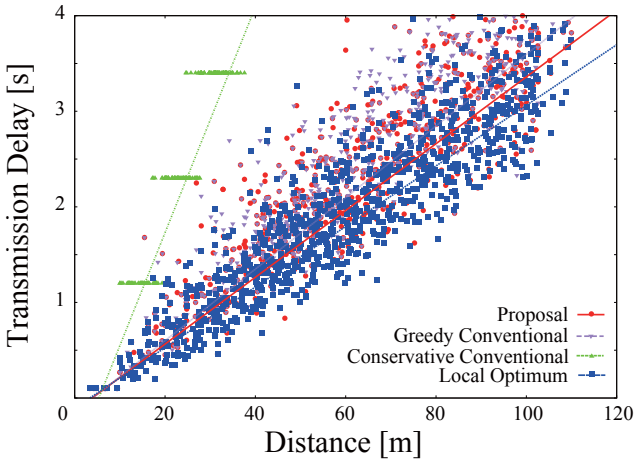


Figure 14. End-to-End Delay in Wireless Multihop Transmissions (0.2 m/s).

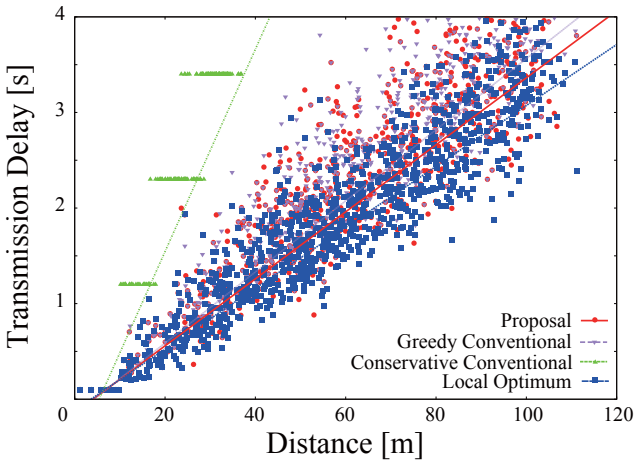


Figure 15. End-to-End Delay in Wireless Multihop Transmissions (0.5 m/s).

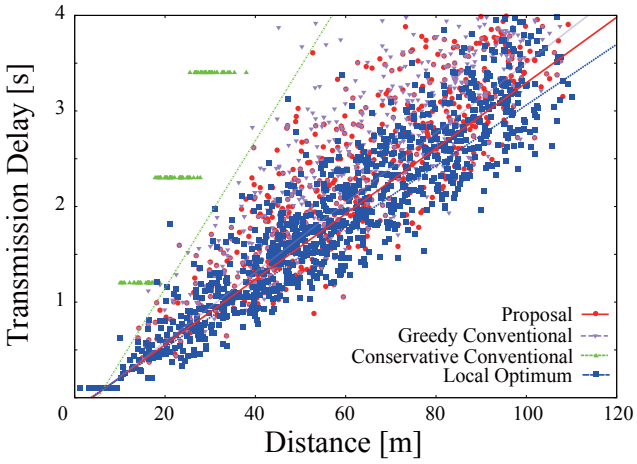


Figure 16. End-to-End Delay in Wireless Multihop Transmissions (1.0 m/s).

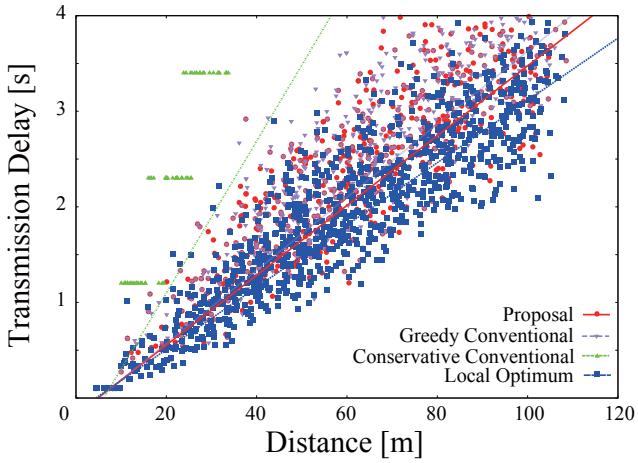


Figure 17. End-to-End Delay in Wireless Multihop Transmissions (2.0 m/s).

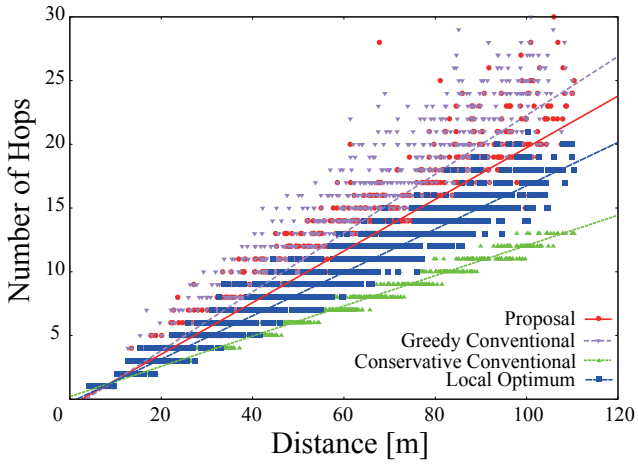


Figure 18. Hop Counts of Data Message Transmissions (0.1 m/s).

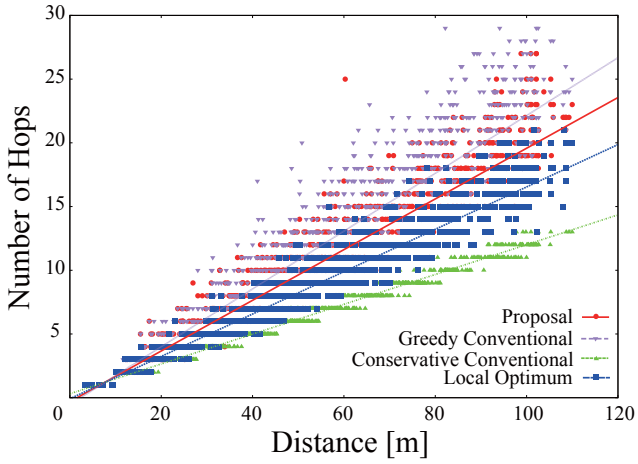


Figure 19. Hop Counts of Data Message Transmissions (0.2 m/s).

 ISBN: 978-1-61208-377-3

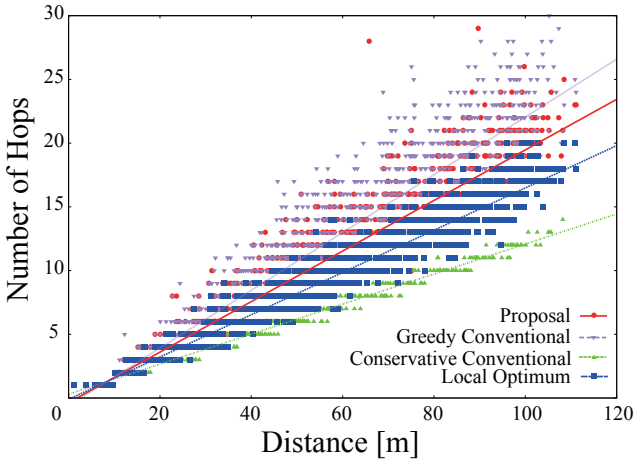


Figure 20. Hop Counts of Data Message Transmissions (0.5 m/s).

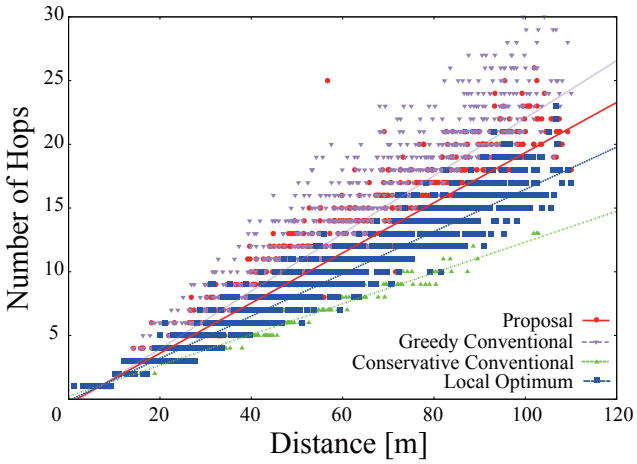


Figure 21. Hop Counts of Data Message Transmissions (1.0 m/s).

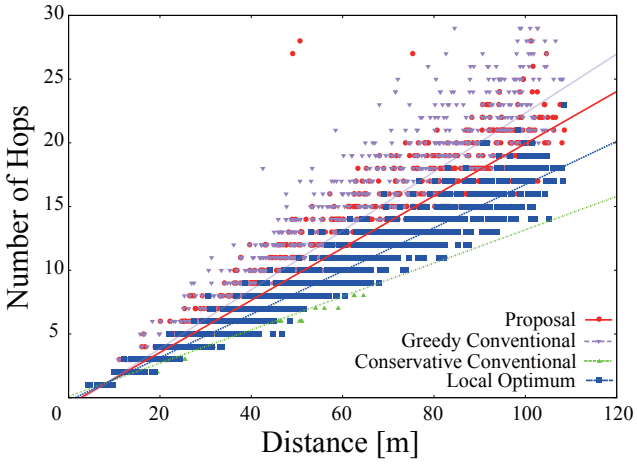


Figure 22. Hop Counts of Data Message Transmissions (2.0 m/s).

V. Conclusion

This paper proposes IRDT-GEDIR which is combination of IRDT intermittent communication protocol with lower power consumption and GEDIR location-based message-by-message ad-hoc routing protocol. In intermittent communication, it is difficult for an intermediate node to select its next-hop node due to difficulty to achieve location and activation time information from neighbor nodes. By introduction of a solution of the secretaries problem and a pseudo speed criterion, a novel next-hop selection method is induced. The 1-hop simulation experiments in a stationary sensor network show that the proposed method achieves better next-hop selection with higher pseudo speed. In addition, the wireless multihop transmission experiments in a mobile sensor network show that it is expected for IRDT-GEDIR to achieve shorter end-to-end transmission delay and smaller hop counts of sensor data messages even with the sleep mode in intermediate sensor nodes due to the intermittent communication. Here, no forwarding failure occurs even without mobility information of neighbor nodes. Therefore, IRDT-GEDIR improves the performance of mobile sensor networks.

In this paper, all the mobile sensor nodes assume to have the same activation interval. However, it is required for mobile sensor nodes to have different activation intervals, e.g., depending on the battery capacity. In our future work, the next-hop selection method is extended to support variation of the activation interval in sensor nodes.

References

[1] D. E. Culler and W. Hong, “Wireless Sensor Networks,” Communications of the ACM, Vol. 47, No. 6, 2004, pp. 30–33.

[2] J. Gilbert and F. Mosteller, “Recognizing the Maximum of a Sequence,” Journal of the American Statistical Association, Vol. 61, 1966, pp. 35–73.

[3] A. Giridhar and P. R. Kumar, “Distributed Clock Synchronization over Wireless Networks: Algorithms and Analysis,” Proceedings of the 45th IEEE Conference on Decision and Control, 2006, pp. 4915–4920.

[4] T. Hatauchi, Y.Fukuyama, M. Ishii, and T. Shikura, “A Power Efficient Access Method by Polling for Wireless Mesh Network,” Transactions of IEEJ, Vol. C-128, No. 12, 2008, pp. 1761–1766.

[5] L. F. W. Hoesel and P. J. M. Havinga, “A Lightweight Medium Access Protocol for Wireless Sensor Networks,” Proceedings of the 1st International Conference on Networked Sensing Systems, 2004, pp. 205–208.

[6] R. Jurdak, P. Baldi, and C. V. Lopes, “Adaptive Low Power Listening for Wireless Sensor Networks,” IEEE Transaction on Mobile Computing, Vol. 6, No. 8, 2007, pp. 988–1004.

[7] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, and Y. Fukuyama, “Performance Evaluation of Intermittent Receiver-Driven Data Transmission on Wireless Sensor Networks,” Proceedings of the 6th International Symposium on Wireless Communication Systems, 2009, pp. 141–145.

[8] X. Lin and I. Stojmenovic, “Geographic Distance Routing in Ad Hoc Wireless Networks,” Technical Report in University Ottawa, TR-98-10, 1998.

[9] C. E. Perkins, “Ad Hoc Networking,” Addison-Wesley, 2001.

[10] V. Rajendran, K. Obraczka, and J. J. Garacia-Luna-Aceves, “Energy-Efficient Collision-Free Medium Access Control for Wireless Sensor Networks,” Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems, 2003, pp. 181–192.

[11] S. Methley, “Essentials of Wireless Mesh Networking,” Cambridge University Press, 2009.

 ISBN: 978-1-61208-377-3

High-Performance Computing on the Web: Extending UNICORE with RESTful Interfaces

Bernd Schuller, Jedrzej Rybicki

Jülich Supercomputing Centre
Forschungszentrum Jülich GmbH
Jülich, Germany
Email: {b.schuller,j.rybicki}@fz-juelich.de

Krzysztof Benedyczak

Interdisciplinary Center for
Mathematical and Computational Modelling
Warsaw, Poland
Email: golbi@icm.edu.pl

***Abstract*—UNICORE (UNiform Interface to COmputing REsources) is a Grid middleware for accessing high-performance computing capabilities and storage resources in a secure and seamless fashion. In its current version (7.0), it offers web services using SOAP (Simple Object Access Protocol) in conjunction with a security stack based on the Security Assertions Markup Language (SAML) and the WS-Security specification. To accommodate recent integration use cases, the need for more lightweight ways to access resources through UNICORE has arisen. This work describes the architecture, design, and first implementation results of an interface to UNICORE services based on the REST (Representational State Transfer) architectural style. Crucial boundary conditions included a lightweight security layer, and full interoperability with the existing SOAP-based interfaces. This RESTful interface will greatly simplify access to and interaction with the UNICORE services and enable new use cases. It will allow integrating high-performance computing and data management services into web-based and mobile applications.**

Keywords–UNICORE; REST; Security; High-Performance Computing

I. INTRODUCTION

UNICORE was developed in the course of several German and European projects since 1997 [1]. It is a mature software suite for building federated systems and Grids. It is deployed and used in a variety of settings, from small projects to large (multi-site) infrastructures involving high-performance computing (HPC) resources. UNICORE can be characterized as a vertically integrated Grid system, that comprises the full software stack from clients to various server components down to the components for accessing the actual compute or data resources. Its basic principles are abstraction of resource-specific details, openness, interoperability, operating system independence, security, and autonomy of resource providers. In addition, the software is easy to install, configure and administrate. UNICORE software is available as open source from the SourceForge repository [2] under a permissive, commercially friendly license.

The UNICORE services can be accessed through a SOAP web service stack, realising stateful services through the Web Service Resource Framework (WSRF) specification [3]. The security layer is based on Transport Layer Security (TLS), SAML, and XML digital signatures. All these are open, well-documented standards, and in principle it is possible to implement clients to access the services in any language, and Web Service tooling exists for many programming languages. However, practical experience has shown that due to the high complexity of WSRF and SAML only Java and C# have been used. In fact, UNICORE had to provide an implementation of the WSRF specification, since none of the Java web service toolkits offers one.

Consequently, it can be difficult or even impossible to use the current Web Service APIs offered by UNICORE. For example, this may occur when integrating UNICORE services into existing applications or community workflows using different technologies than the above mentioned Java and C#. Thus, simpler, more easily accessible APIs are required. To this end we are working on two concrete use cases. The first one originates from the European Human Brain Project [4]. UNICORE will form the basis for the project's HPC Platform. It comprises of four major HPC sites, cloud storage and other resources. Here, developers want to write Python applications for accessing services of the HPC Platform, such as job management or data transfer. Lightweight mechanisms, such as OpenID Connect (OIDC) [5], should be used for authentication. The second use case is a standalone client for the UNICORE file transfer protocol (UFTP) [6], which allows users to access their data without requiring the use of a full UNICORE client. Common for the both use cases is the requirement for strong authentication and delegation of rights. Subsequently, it is important to stay compatible with the usual access through UNICORE despite the introduction of the new interfaces.

A popular alternative to SOAP are RESTful services [7]. These are usually more lightweight and more easily accessible for a number of reasons. Typically, they make use of JSON [8] instead of XML for resource representations and exploit HTTP semantics for the resource manipulation instead of defining their own. To make implementation of clients easier, one would like to avoid having to handle digital signatures on the client, so more lightweight mechanisms, such as OpenID-Connect, are of interest.

The remainder of the paper is organized as follows. Section II describes the UNICORE services and service container and how RESTful services have been realized within this context. A review of the UNICORE security solution as it applies to this work is given in Section III. The initial APIs and some first performance results are given in Section IV. The paper concludes with an outlook and the next steps.

 ISBN: 978-1-61208-377-3

II. SERVICES, INTERFACES AND TECHNOLOGIES

UNICORE is a four-tiered system, consisting of the client, gateway, services and target system tiers. All components with the exception of the target system tier are implemented in Java.

The *Gateway* is essentially a HTTPS reverse proxy, that serves as a firewall transversal point to avoid having to configure many open firewall ports. The Gateway forwards information about the connecting client (such as the IP address or the client's SSL certificates) to the servers behind it.

The *UNICORE/X* server is the central component of a UNICORE installation. It is built around the XNJS execution engine [9], which provides the execution backend and communicates with the target system tier, and a set of service interfaces. The basic services include a Registry service that provides information about available services, job submission and management services as well as file access and file transfer.

Finally, the target system tier consists of the interface to the local operating system, file system and resource management (batch) system. This *target system interface* (TSI) is responsible for submitting jobs, performing file I/O, and checking job status. The TSI is implemented in Perl, as a server running on the resource (e. g., the login node in case of a compute cluster).

The WSRF specification introduces the concept of service instances, which can be individually addressed, and are comparable to objects in an object-oriented system. The conceptually most important UNICORE services are listed in the following, where many instances of each service will usually exist in a UNICORE container:

- *Sites* are abstracted compute resources. They have a set of properties (e. g., number of cores), have a set of storages attached, and accept job submissions.
- *Storages* are abstracted file system like data resources, which offer typical operations, such as listing files. To give access to files, storages act as factory services for file transfer resources.
- *Jobs* represent actual compute jobs on the underlying batch system. Jobs always have a working directory, which is accessible through a Storage resource. To create a new job, a job description and optionally some input data is required. The job description details what is to be executed, gives the required resources (e. g. number of CPUs), and a list of date files to be staged in and result files to be staged out. Jobs are submitted to a Site resource.
- *File transfers* are used to read or write to a physical remote file. UNICORE supports both client-server data transfers and server-server transfers, with several available data transport protocols.
- *Site factories* support virtualization technologies, since a Site is always created through a Site factory.
- *Storage factories* allow the creation of storage service instances, and can support multiple backends (e. g. plain file systems or the Hadoop file system).

The services are hosted in a container called UNICORE Services Environment (USE) that is built from well-established open source components, such as Apache CXF, Jetty and many others. USE provides the web server, security layer, service configurations, a persistence subsystem and the base classes on which the actual services are built.

Care has been taken to decouple the front-end service implementations (e. g. SOAP WSRF) from the internal state and from the representations that are sent to the clients, aiming to implement the model-view-controller pattern.

Building upon Apache CXF, RESTful services following the JAX-RS standard [10] can be deployed in the USE. The RESTful services can access all USE subsystems (e. g. for persistence) and can thus access the same resources as the WSRF services. One positive side-effect of this approach is that the same state (e. g. storage) can be accessed consistently through different interfaces.

III. SECURITY

The flexible security system in UNICORE is one of its main assets. In this section, we briefly describe how *authentication*, *authorization* and *delegation of rights* work in UNICORE.

The goal of the *authentication* process is that the UNICORE/X server knows the X.500 name (distinguished name, DN) of the user and has verified that it is correct. Traditionally, authentication required that each entity in the system (users and servers) required an X.509 end-entity certificate. This was used for establishing SSL connections where both parties (client and server) could check that the other party was trusted. In UNICORE 7, the security architecture has been made much more flexible through the new Unity service [11], which can authenticate users using some other means (e. g., username and password). Client certificates are no longer necessary, though they can still be used. Server certificates are still required, for instance for securing the communication channel.

The *authorization* process takes the DN established by authentication and maps it to a set of user attributes, which are used for two purposes. First, the server's access control policies (written in XACML) are evaluated to decide whether the user is allowed to perform the current operation. Second, the attributes are used later by the services' business logic. For example, two typical attributes are the local Unix username and groups, which are required, e. g., for job submission.

All of the authentication and authorization process is configurable by the UNICORE administrator, in accordance with the principle of autonomy of the resource provider.

Finally, *delegation* is needed, for example when a user submits a job that requires data to be downloaded from another UNICORE server. In such a case it is not acceptable to impersonate the user by passing along her credentials. This would pose a potential security risk. Instead, a delegation token is required that cryptographically asserts the original user's identity and asserts that the original user delegates rights to the server (for performing particular action on her behalf). Delegation is implemented in UNICORE via signed SAML assertions, where the user delegates her trust explicitly to another party (which can be a server or another user), which is identified via a DN. The trusted party can then work on the user's behalf, even delegate trust again, forming a trust delegation chain. For more details we refer the reader to the more extensive description in [12].

When the user does not have a X.509 private key, she cannot sign any assertions. Thus, in UNICORE 7 the Unity

 ISBN: 978-1-61208-377-3

server can be used as the source of the "bootstrap" trust delegation that asserts that the user trusts the first server. As the trust delegation mechanism is based on SAML and requires heavy weight XML processing, it does not readily lend itself to a RESTful architecture, especially it has to be avoided on the client side.

In delegation, two use cases need to be considered:

1) REST-to-REST : the user invokes a RESTful service, which needs a delegated call to another RESTful service
2) REST-to-SOAP : user invokes a RESTful service, which needs a delegated call to a SOAP service

IV. First results

We have implemented a number of extensions to link the JAX-RS implementation provided by Apache CXF to the UNICORE resources framework provided by USE. These include

- an authentication handler uses the HTTP basic authentication header (i. e., username and password) and maps them to a X.500 DN using a configurable chain of authentication components. Available authentication options are a local username/password file or the admin can delegate the authentication to Unity;
- mechanisms are used to inject the requested resources and other required information into the JAX-RS service class;
- access control checks using the XACML policy decision point.

As a baseline, we have started the implementation of RESTful services for the fundamental UNICORE entities listed in Section II.

Both JSON and HTML representations of individual resources can be served, as well as lists of all the resources available to the current user. Table I shows the current state of the API.

TABLE I. Initial REST API for UNICORE services.

HTTP method on resource	*Description*	*Media type*
GET /jobs	Lists user's jobs	JSON, HTML
POST /jobs	Submits a new job	JSON
GET /jobs/{id}	Get job properties	JSON, HTML
DELETE /jobs/{id}	Remove a job	
GET /storages	Lists user's storages	JSON, HTML
GET /storages/{id}	Get storage properties	JSON, HTML
DELETE /storages/{id}	Remove a storage	
GET /storages/{id}/files/{path}	Get file properties	JSON, HTML
GET /storages/{id}/files/{path}	Download a file	Binary
PUT /storages/{id}/files/{path}	Upload a file	Binary
POST /storages/{id}/imports	Create a new file import	JSON
POST /storages/{id}/exports	Create a new file export	JSON
GET /sites	Lists user's sites	JSON, HTML
GET /sites/{id}	Get site properties	JSON, HTML
DELETE /sites/{id}	Remove a site	

It is possible to consistently access these resources through both the WSRF layer (using standard UNICORE clients) and through the REST layer (using HTTP clients, such as *curl*).

Data can be downloaded and uploaded through the web server using the HTTP protocol using GET and PUT requests. In addition, other file transfer protocols (e.g., UFTP) are supported as well by explicitely creating new file import/export resources.

The services API is currently under discussion and further development. The UNICORE resources form a tree with many interconnections. For example, a job has a working directory, which is a storage resource. Thus, the working directory resource should be accessible via both `/jobs/{j_id}/wd` and `storages/{s_id}`. In the WSRF API, these links between resources are discovered by the client, and in RESTful designs this dynamic discovery of resource links is considered the most elegant (according to the "HATEOAS" principle in [7]). On the other hand, having to dynamically discover everything can lead to increased network traffic and latencies, and clients may want to leverage certain knowledge of the REST API.

The delegation issue is solved partly: when authenticating a user using Unity, the REST authentication handler also receives a SAML assertion, which can be used later to make invoke services on behalf of the user. However, this only works when invoking SOAP/WSRF services. A solution for delegated access to resources through the REST API still needs to be agreed upon. Several possible solutions are conceivable. For example, a JSON rendering of the SAML assertions used by UNICORE is possible. Since these would be handled entirely on the server, the clients would not be made more complex.

A. Job submission example

For job submission, a simple JSON job description is used, which consists of the executable, arguments, environment settings as well as data stage-in and stage-out and required consumable resources such as wall time. This is currently translated into UNICORE's internal XML format before being submitted to the internal execution engine. As a trivial example,

```
{
  Executable: "/bin/echo",
  Arguments: ["Hello World"],
}
```

would be a valid job. This JSON job description syntax is already in use in the UNICORE commandline client [13], and thus well known to UNICORE users. Using curl as a simple HTTP client, the submission of a job in file "job.u" can be done by (ignoring security for the moment):

```
curl -X POST <base_url>/jobs
  -H "Content-type: application/json"
 --data-binary @job.u -i
```

The server will reply with a "201 Created" status and the location of the new job:

```
HTTP/1.1 201 Created
Location:  <base_url>/jobs/<id>
```

B. Initial performance tests

Since the new REST interface shares the back-end and business logic with the WSRF interface, any performance improvements are due to the smaller overhead of the REST interface. To quantify these improvements, we have run a number of simple performance tests, comparing a "GET" operation

 ISBN: 978-1-61208-377-3

on a resource via both the WSRF and the REST interfaces. We have used a production-like setup, where the REST service is accessed via SSL and via a UNICORE Gateway. All servers and the client code was run on "localhost". The test machine was a quad-core Intel i7 at 2.8GHz, with 8 GBs of RAM running Java 7 (OpenJDK 1.7.0_65).

TABLE II. THROUGHPUT FOR GET REQUESTS VIA WSRF AND REST.

Client threads	*Interface*	*Requests/sec*
1	WSRF	27
	REST	79
2	WSRF	57
	REST	193
4	WSRF	80
	REST	286
8	WSRF	76
	REST	332

We sent 1000 requests each using 1,2,4 or 8 client threads. As table II shows, using the REST interface has much higher throughput, and scales better to higher numbers of concurrent client threads.

As a second example, we have evaluated job submission, using simple 'hello world' jobs as shown above. Here we submitted 400 jobs. Table III shows the results. Again the REST interface is onsistently better in terms of throughput and scalability.

TABLE III. THROUGHPUT FOR JOB SUBMISSION VIA WSRF AND REST.

Client threads	*Interface*	*Jobs/sec*
1	WSRF	5
	REST	34
2	WSRF	11
	REST	54
4	WSRF	12
	REST	75

These initial tests already show that significant performance improvements can be expected from the REST interface.

V. SUMMARY AND OUTLOOK

We have extended UNICORE to allow building RESTful services that are fully consistent with the existing SOAP/WSRF based services. This includes the security stack used for RESTful services, which is fully compatible and consistent with the rest of the UNICORE world.

One fundamental issue remains to be fully solved: delegation to allow a server to make delegated calls to other RESTful services. One option is to use Unity to provide SAML assertions once the user has authenticated, and translate the SAML delegation assertions to a JSON rendering.

The new RESTful APIs will open up the world of HPC and access to large-scale scientific data to a much wider audience, by allowing applications to use simple authentication mechanisms, submit compute tasks to HPC machines, manage results, move data and much more.

The basic REST support and initial service implementations will be released with UNICORE 7.1, and will be evolved further towards a major release, UNICORE 8.

Next steps will focus on finalizing the security architecture and implementing OpenID-Connect support, i. e. validating OIDC tokens and if required creating SAML trust delegation assertions from them using Unity. Furthermore, the service APIs will be developed further, aiming at basic job submission and management for the first release, and adding full capabilities in the UNICORE 8 release.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 604102 (Human Brain Project)

REFERENCES

[1] "UNICORE Website," http://www.unicore.eu/, [accessed: 2014-07-10].

[2] "UNICORE Open Source project page," http://sourceforge.net/projects/unicore/, [accessed: 2014-07-10].

[3] "Web Services Resource Framework," http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf, [accessed: 2014-07-10].

[4] "Human Brain Project," http://www.humanbrainproject.eu/, [accessed: 2014-07-10].

[5] "OpenID Connect," http://openid.net/connect, [accessed: 2014-07-10].

[6] B. Schuller and T. Pohlmann, "UFTP: High-Performance Data Transfer for UNICORE," in Proceedings of 7th UNICORE Summit 2011, ser. IAS Series, no. 9. Forschungszentrum Jülich GmbH, 2011, pp. 135–142.

[7] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[8] D. Crockford, "The application/json media type for javascript object notation (JSON)," RFC 4627, Jul. 2006.

[9] B. Schuller, R. Menday, and A. Streit, "A Versatile Execution Management System for Next-Generation UNICORE Grids," in Proceedings of 2nd UNICORE Summit 2006 in conjunction with EuroPar 2006, ser. LNCS, no. 4375. Springer, 2006, pp. 195–204.

[10] "Java API for RESTful Services (JAX-RS)," https://jax-rs-spec.java.net/, [accessed: 2014-07-10].

[11] "Unity Identity Management Solution," http://www.unity-idm.eu/, [accessed: 2014-07-10].

[12] K. Benedyczak, P. Bała, S. van den Berghe, R. Menday, and B. Schuller, "Key aspects of the UNICORE 6 security model," Future Generation Computer Systems, vol. 27, 2011, pp. 195–201.

[13] "UNICORE commandline client job description format," http://unicore.eu/documentation/manuals/unicore6/files/ucc/ucc-manual.html#ucc_jobdescription, [accessed: 2014-07-10].

 ISBN: 978-1-61208-377-3

Constraint-Based Distribution Method of In-Network Guidance Information in Content-Oriented Network

Masayuki Kakida Yosuke Tanigawa Hideki Tode
Dept. of Computer Science and Intelligent Systems
Osaka Prefecture University, Osaka, Japan
Email: {kakida@com., tanigawa@, tode@} cs.osakafu-u.ac.jp

***Abstract*—Lately, network usage is dominated by content distribution, and hence, Content Oriented Network (CON) has attracted attentions. CON has been actively studied at many research organizations, but it is still immature and difficult to establish clean-slate network. Therefore, our research goal is to establish a scalable and feasible architecture, which integrates the conventional IP network and a new content-oriented network framework. We use Breadcrumbs (BC) architecture as the base of integrated network but there still remains large overhead on managing and looking up forwarding table at each router. In this paper, we propose BC-Scoping on Popularity, which distributes the guidance information adaptively based on content popularity. Popularity is one of the criteria for the proposed BC-Scoping framework, which manages several BC's distribution scopes. The proposed method enables the network to use different forwarding policies based on content popularity, and brings the benefits of smaller overhead and better system performance. Finally, we demonstrate the effectiveness of the proposed method by extensive computer simulation, including the comparison with the currently operated Content Delivery Networks (CDN) approach, using state-of-the-art popularity model newly defined.**

***Index Terms*—Breadcrumbs, cache, content oriented network, Breadcrumbs+, BC-Scoping, popularity**

I. INTRODUCTION

For several years, access loads on servers and network traffic are greatly increasing due to larger content-size and higher request-frequency in content distribution networks. To solve this problem, web caching approaches [1] [2] have been proposed, where network nodes cache copies of contents. On the other hand, the idea of content oriented network (CON) [3] [4] has attracted attention as a framework of new-generation network. This concept is derived from the viewpoint that users are interested not in where the content is but in what the content is; content distribution and retrieval dominate network usage now but network works on host-address oriented architecture designed several decades ago. Therefore, there are some researches that combine the ideas of "caching" with "content oriented network" [5] - [7]. Content-Centric Networking (CCN) proposed by Van Jacobson et al. [8] is also based on the similar concept, where both request and its' request are routed by a name of the content.

In CON, however, we encounter a serious scalability problem on content retrieval especially in a large-scale network like the Internet, where extremely wide variety of contents are distributed. In CON, packets are forwarded based not on address of nodes but on name of contents in network. A much larger number of contents than the one of network nodes increase costs of exchanging routing information and of looking up a routing table at each router.

CON has been actively studied at many research organizations, but it is still immature and difficult to establish clean-slate network. Therefore, our research goal is to establish a scalable and feasible architecture, which integrates the conventional IP network and a new content-oriented network. In terms of scalable routing on CON itself, not an active approach but a passive approach should be adopted for simplicity. Therefore, we established the routing method based on Breadcrumbs [7] [9] among researches on CON routing. Breadcrumbs is an architecture with guidance information (routing information) leading to a node holding a cached content, and log information to follow each content is created or updated only when the content is downloaded at routers on the download path but not at any other routers. BC's passive and simple approach allows us to make scalable and feasible CON autonomously in cooperation with cached contents in network. Although this routing approach has higher scalability than other ones like Forwarding Information Base (FIB) in CCN, BC-based routing still has large overhead on managing and looking up forwarding table at each router. One of the reasons is that most of routing information is not used effectively and searching a required information from the routing table is a costly task.

In the target integration architecture of IP and CON, we tackle to reduce various cost for networking (e.g., exchanging routing information, transferring data, frequency of forwarding operation, etc.) as much as possible. This contributes to enhance the comprehensive network system performance (e.g., content download time, robustness, etc.). Specifically, we propose BC-Scoping on Popularity, which distributes the guidance information of only popular contents. Popularity is one of the criteria for BC-Scoping framework which manages several BC's distribution scopes. As for unpopular contents, IP routing is definitely applied, which is desirable from the viewpoint of routing overhead. Such contents are requested less frequently, hence most of the guidance information for the contents are not used effectively. BC-Scoping on Popularity brings the following benefits.

1) It reduces the amount of BC entries in network, and then diminishes control overhead, including looking up

 ISBN: 978-1-61208-377-3

a table, creating and deleting BC entries, etc.
2) It prevents the cache replacement of popular content caches by unpopular content ones, and then we can get better performance.

Though we already proposed BC-Scoping on Domain [10], this scoping is based on hierarchical network structure, and its goal is to localize content retrieval within a network domain and to reduce inter-domain communication. On the other hand, the goal of BC-Scoping on Popularity proposed in this paper is to reduce various cost for networking. Thus, it has completely different control policy compared with [10].

Finally, we demonstrate the effectiveness of the proposed method by the extensive computer simulation using state-of-the-art popularity model newly defined.

The main contributions of this paper are the following.

- We extend a specified criterion, which is the network domain in the earlier proposal, for BC-Scoping to general ones. We adopt content popularity as the representative criterion in this paper.
- The proposed method reduces system overhead, and specifically the lookup count of BC table is reduced to 37% of the naive BC system.
- We introduce newly formulated popularity model for generating requests.
- We quantitatively clarify the superiority of the proposed method in comparison to the most popular and widespread Content Delivery Network (CDN) approach.

The rest of the paper is organized as follows. Section II describes qualitative comparison of between our proposed method and other CON researches, and shows our earlier study. In section III, we propose BC-Scoping on Popularity. Section IV describes the performance evaluation of our proposed method, and the section consists of the followings: newly formulated popularity model (section IV-A), the simulation scenario and its results for comparison of our proposed method with other Breadcrumbs systems (seciton IV-B and), and the simulation scenario and its results for comparison of our proposed method and CDN. Finally, section V shows our conclusion.

II. Related works and Breadcrumbs

A. Related Works

One of the main focuses in this paper is routing. Among routing methods in CON [3] - [8], our proposed routing method is based on Breadcrumbs taking a passive and simple approach, just logging a direction in which a content is forwarded, at each router. This simplicity reduces computational cost of processing or maintaining routing information compared with other schemes. Breadcrumbs's passive and simple approach allows us to make scalable and feasible CON autonomously in cooperation with cached contents in network. We use Breadcrumbs as the base of a solution against a scalability problem on searching for wide variety of contents.

CCN [8] adopts one of active approaches: a flooding-based approach. This approach is the most primitive idea of routing and will produce huge amount of traffic for retrieving a content. In contrast, users just issue a request for a content in Breadcrumbs and downloading the content makes a BC trail, which means a series of guidance information to the located cache.

DONA [6] adopts another one of active approaches: advertising routing information and forming routing trees. This approach achieves high efficiency and performance but needs to exploit information of network topology. Maintaining routing trees will require periodic exchange of network topology information and some computation. By contrast, Breadcrumbs does not use any detailed topology information with the exception of basic information about which domain each node belongs to, like subnet mask, especially for BC-Scoping on Domain. And thus, smaller overhead is required over the entire network.

PSIRP [4] adopts a routing scheme based on distributed-hash table (DHT). This approach achieves flat and scalable routing burden among routers but needs that participating nodes have to cooperate closely. On the other hand, in Breadcrumbs architecture, what each node does is just to log a direction and to forward packets. The nodes do not exchange extra information other than about IP routing. This simplicity provides easier management of nodes in network.

In the terms of efficient content distribution, CDN [11] [12] is the most practical and well-known techniques in the current Internet. CDN enables content publishers to make their contents widely available and to distribute the contents efficiently. In order to accomplish the efficient content distribution, Akamai [12], which is a first class example of commercialized CDN operators, monitors the state of service, network and servers through several ways including servers' periodic reporting and end-to-end agents' measuring. CDN architecture takes a centralized approach, which is appropriate for the operators to make decisions for controlling their system. On the other hand, Breadcrumbs architecture takes a decentralized approach, each node in the network works autonomously. In other words, CDN and Breadcrumbs have a fundamental difference in design concept, and they have a complementary relationship. We can then use the both of them simultaneously.

Another one of the important points in our research is that Breadcrumbs is a bridging architecture between the current network and the future network. There are some researches [4] – [6] [8] which adopt a clean-slate approach in order to establish content-oriented network, but it is hard to put the approach into practice in a large scale and in a short period by replacing current network. On the other hand, Breadcrumbs easily forms content-oriented network in cooperation with a network with any routing protocol (e.g., IP network). This characteristic ensures high feasibility of our Breadcrumbs-based approaches. We can use current network equipments supporting only IP if we install middleware for supporting

 ISBN: 978-1-61208-377-3

TABLE I. BC ENTRY

Attribute	Description
ContentID	Global file ID
UpHop	ID of node from which the file was forwarded
DownHop	ID of node to which the file was forwarded
DownloadTime	Time when the file passed through the node lastly
RequestTime	Time when the file was requested at the node lastly

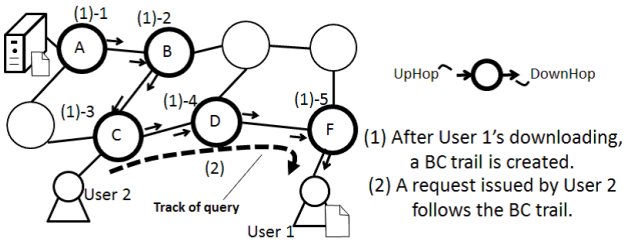


Fig. 1. An example of BC trail

Breadcrumbs on them, without fundamental and drastic replacement or re-installation of the current system [13][14]. In addition, even when Breadcrumbs architecture is partially deployed in the current IP network, the architecture works well by means of forming overlay network with simple IP tunneling technique [15]. Furthermore, we can operate content-oriented system with the support of the current IP system. In other words, we can create, improve and prepare the environment for future content-oriented system with operating the current Internet system.

B. Breadcrumbs

In this section, we give an outline of Breadcrumbs [7]. Breadcrumbs architecture is stupidly simple from local view but efficient from global view to let application and network cooperate. Through primitive and simple processing such as logging a download path and caching a content, we can place, locate and obtain the content efficiently. We assume that the cache replacement policy follows Least Recently Used (LRU).

When a content is downloaded, each router on the download path makes a *breadcrumb (BC)* entry, which is a minimal information to route requests, and a cache of the content. Note that, because we assume more feasible network environment, core routers do not have any cache space, and only the *user end*: edge router, ONU, STB or user's PC, who downloaded the content, caches a copy of it. Each BC entry is composed of a 5-tuple data shown in Table I. Each of ContentID, UpHop, DownHop, DownloadTime and RequestTime has one value. When a request for a content encounters a BC entry for the content at a node on the way to an original content server by conventional IP forwarding, the request is routed to DownHop in the BC entry until it reaches a node with an intended cache (see Figure1). Thus, through tracing a series of BC entries, a request can follow the content downloaded previously. This series of BC entries is defined as *trail*. If DownHop or UpHop in a BC entry at a node is *null*, the node is at the end of the BC trail.

III. PROPOSAL

In this paper, we focus on popularity of contents as a criterion for BC-Scoping. We propose BC-Scoping on Popularity, which distributes both of the guidance information and caches adaptively based on content popularity.

A. Motivation

Name-based forwarding is one of the most important factors of content-oriented network. It provides us some advantages over the conventional location-based forwarding. One is smaller overhead to get the nearest copy of contents, and another one is robustness over dynamic change of content location by cache replacement. However, we should note that we can take these advantages only when we request popular content. "The nearest copy" implicitly means that there are many cached copies of a target content, but there are few ones of unpopular contents. "Dynamic change of locations" also implicitly means that the target content is requested many times and then many copies are frequently created and replaced in a variety of places, but unpopular ones are not so much.

Now, we can get a hint from [16]; taking different approaches based on target contents may provide better performance.

B. BC-Scoping Framework

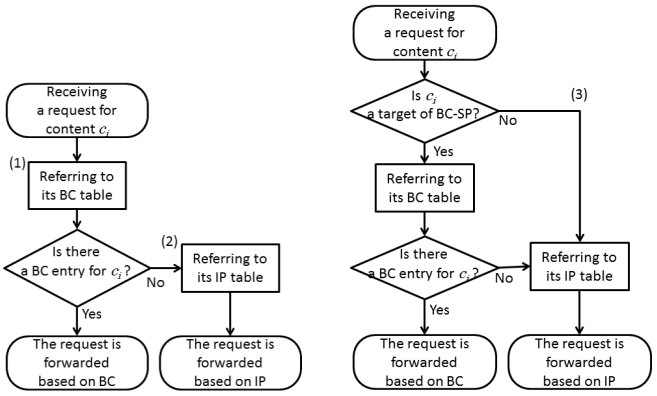
We have proposed BC-Scoping framework, which manages BC's distribution scope according to some criteria. We can reflect constraints on arbitrary criteria, including network domain, access count, content popularity and something. We would rather use simple metrics as criteria of BC-Scoping than complicated ones even if the metrics are too simple to bring us strictly optimal solution. This is because we believe that network system as infrastructure should be as simple as possible.

We already focused on hierarchical network topology and developed BC-Scoping on Domain [10], which limits the distribution scope of guidance information according to network domains. In other words, BC-Scoping on Domain enables or disables intermediate routers to create guidance information based on whether or not the destination node of a downloaded content belongs to the same domain as the router does, respectively. BC-Scoping on Domain promotes intra-domain communication so that it achieves small-hop content retrieval and reduction in traffic volume.

C. BC-Scoping on Popularity

In this paper, we focus on popularity of contents as a criterion for BC-Scoping. We propose BC-Scoping on Popularity, which distributes both of the guidance information and caches for only popular contents. In the proposed system, when a router forwards a content, the router at first checks the popularity of the content. If the popularity rank is higher than the threshold rank, the router creates a BC entry or updates the

 ISBN: 978-1-61208-377-3



(a) Without BC-Scoping on Popularity (b) With BC-Scoping on Popularity

Fig. 2. Router's request handling flow in BC system

corresponding entry after forwarding the content. Otherwise, the router does nothing after forwarding. In addition, when *user end* receives a content, the user end also checks the popularity of the content. If the popularity rank is higher than the threshold rank, the user end caches a copy of the content. Otherwise, the user end does nothing. The most simplest way to check the above condition is to utilize a packet header. Specifically, in advance, each origin server adds popularity information to contents. Then, according to the popularity information attached to target content, active or inactive flag is changed in the header of transferred packets. It requires the minimal cost.

D. Benefits of BC-Scoping on Popularity

BC-Scoping on Popularity provides the following benefits.

First, explicit separation of forwarding policy reduces overhead of looking up forwarding table at each router. In the naive BC system, when a router receives a request, the router always refers to its BC forwarding table at first for name-based forwarding regardless of content popularity ((1) in Figure 2(a)). In most cases, there is no cached copy of unpopular contents in network, and there is no corresponding BC entry, either. The router eventually refer to IP table ((2) in Figure 2(a)). In short, routers have to refer to two different forwarding tables for unpopular contents in most cases. With BC-Scoping on Popularity, when a router receives a request for unpopular contents, the router skips reference to BC table, and then refers to IP tables at first ((3) in Figure 2(b)). Though the router may refer to two different forwarding tables for popular contents, the size of BC table is smaller and we have a lower possibility of no cached copy in network compared with the case without BC-Scoping on Popularity. From these assumptions, BC-Scoping on Popularity can reduce system overhead.

Second, it prevents frequent replacement of a cache of popular content by the one of unpopular content. As a result, more caches of popular contents are distributed in network. Since cached copies of contents with high popularity are clearly more valuable in terms of traffic reduction and small-hop content retrieval, we will get better performance on those metrics.

IV. Evaluation

We conducted computer simulations on 2 scenarios with newly developed popularity distribution of request for contents. The distribution used in the simulations is developed, by combining 2 characteristics of content popularity [18][20]. 2 scenarios are the following:

1) the proposed method vs BC+
2) the proposed method vs CDN.

In the former evaluation, we compare the proposed method with the conventional Breadcrumbs methods. In the latter evaluation, we also compare the proposed method with the simplified model of CDN, which play the important roles in the current content distribution.

How to get the popularity information of each content is out of scope in this paper, and we assume the information is attached in each content beforehand. From the aspect of implementation, one of promising ways is that, according to request frequency, content servers attach the flag indicating whether in-network guidance information should be logged via the traversing route to each content or not.

A. State-of-the-art popularity distribution of request for contents

There are some observations on users' request pattern [18] –[20]. In [18], YouTube's web pages were observed, and it was showed that video popularity almost follows a Zipf distribution with an exponential cutoff and also has fetch-at-most-once like behavior. Fetch-at-most-once [19] behavior makes a gap between actually traced data and a logical Zipf distribution in high popularity rank. [20] reported this gap can be expressed using Zipf-Mandelbrot's law instead of Zipf's law. Under this distribution, the probability of an occurrence of a request for the content with the k-th popularity is defined by the following equation:

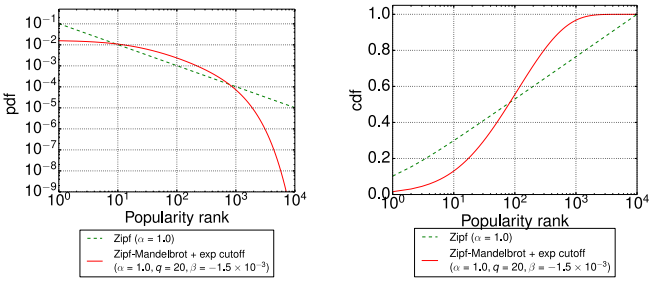
$$f(k; N, q, \alpha) = \frac{1}{(k+q)^{\alpha}} \times \frac{1}{\Sigma_{i=1}^{N} \frac{1}{(i+q)^{\alpha}}}. \quad (1)$$

N is the total number of ranks, q, α are the parameters which determine the shape of distribution. The above two were independently discussed so far. Therefore, we developed a new distribution model by combining two factors, Zipf-Mandelbrot with exponential cutoff distribution. Under this distribution, the probability of an occurrence of a request for the content with the k-th popularity is defined by the following equation:

$$g(k; N, q, \alpha, \beta) = f(k; N, q, \alpha) \times e^{\beta k} \times a. \quad (2)$$

β is the parameter to determine the shape of distribution. a is the parameter for normalization.

 ISBN: 978-1-61208-377-3



(a) pdf (b) cdf

Fig. 3. Dist. of requests for contents

We show the probability density function (pdf) and the cumulative distribution function (cdf) of this distribution in Figures 3(a) and 3(b), respectively.

B. Simulation scenario (vs BC+)

We set each parameter as shown in Table II and other settings are as follows. Note that we assume that contents are cached on not the core routers but the edge of the network, *user end* which includes edge router, users' storage, ONU, or STB, unlike original Breadcrumbs [7], for higher feasibility.

- Network topology

We generate a flat router-network based on the Waxman model [17]. The parameters for generating each network are as follows: A edge length of a square is 1000, $\alpha_{waxman} = 0.3$ and $\beta_{waxman} = 0.05$. Every router is connected with five users and the location of each server is chosen in a uniformly random manner. In the generated topology, a packet can be transferred between any two routers in 18 hops at most.

- Requests for contents

Each user determines which content to request at random following the distribution described in Section IV-A, where $N = 10000, \alpha = 1.0, q = 20, \beta = -15/N$. Each user requests a content at the independent, identical, and exponentially-distributed random interval.

- Packet size and delay

According to the paper of the original Breadcrumbs [7], we assume that the network is not congested, and each node has an enough routing capability to traffic load so that some parameters in the system are constant for simplicity. Size of packets is consistently 1,500 Byte. A request or a control packet consists of 1 packet. On the other hand, a content consists of 768,000 packets. This content size is nearly equal to the size of a content with a bit rate of 5 Mbps and length of 30 min. A delay for a packet to travel to the adjacent node is always 2.3 ms, which includes delays for processing, queuing, propagation and transferring. This delay means that a throughput of any connection is consistently 5 Mbps.

- Compared methods

TABLE II. PARAMETERS

parameter	value
# Routers	1000
# Users	5000
# Servers	50
# Contents	10000
Cache capacity per user	2
Interval of request generation per user	2000
T_f for purge of BC	90000

We compared the following 3 methods.

1) BC+ : Naive BC+ system without BC-Scoping
2) BC+ (rate) : BC trails and caches are created at a constant rate.
3) BC-S (pop) : Proposed method.

In this simulation, "popular" contents mean "top 1%" popular contents among all, and we call those contents $target$ contents. BC trails and caches for only the target contents are created in network.

We prepare a method $BC+(rate)$ as an object for comparison in terms of reduction in overhead. In BC+ (rate) system, BC trails are created at a constant rate, where each server and cache node decide whether or not to create BC trails based on access count. This method is the simplest way to reduce overhead. We can consider this frequency as a criterion for BC-Scoping. We set the rate to 55.5% in this simulation because requests for top 1% popular contents occupy 55.5% of all generated requests. This means that we have the same number of opportunities for BC trail creation in BC+ (rate) and BC-S (pop) (see Figure 3(b)).

We should note that we use Breadcrumbs+ [9] instead of Breadcrumbs for the following evaluation, because the original BC [7] has a routing loop problem due to a broken BC trail, where requests are transferred forever within a series of routers with forming a loop of route and cannot reach the intended contents. The original Breadcrumbs is too simple to solve the problem, and hence we revised attributes in a BC entry and improved the invalidation operation of BC. We discussed the detail of this problem in [9].

C. Simulation results

We show the results in Table III, and in Figures 4 - 10. We used the following evaluation metrics: BC operation count, lookup count of routing table, request hop count, content hop count and server access ratio.

With these results, we can obtain the following considerations. The first two considerations are on overhead, the latter two are on performance trade-off.

1) BC operation count: BC operation count is the total number of the events operating BC entries at each router. Smaller BC operation count means that each router spends smaller resources to manage Breadcrumbs system.

Figure 4 shows that BC+ (rate) reduces total BC operation count to approximately 56% compared with BC+. This reduction in operation count is proportional to that in creation

 ISBN: 978-1-61208-377-3

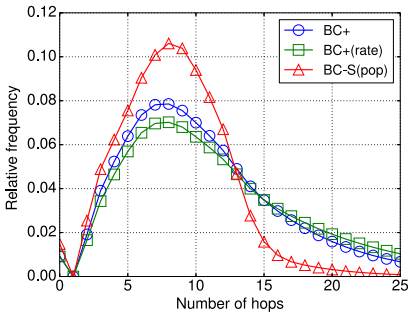


Fig. 6. Dist. of request hop count

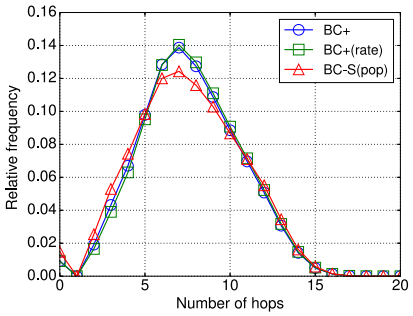


Fig. 8. Dist. of content hop count

TABLE III. MEAN VALUES

Method	Request hop count	Content hop count	ServerAccess Ratio
BC+	10.93	7.62	0.132
BC+(rate)	12.48	7.71	0.109
BC-S (pop)	8.47	7.54	0.453

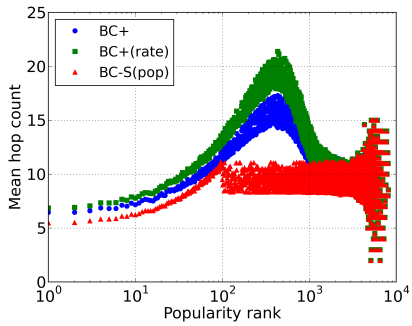


Fig. 7. Dist. of mean request hop count by popularity rank

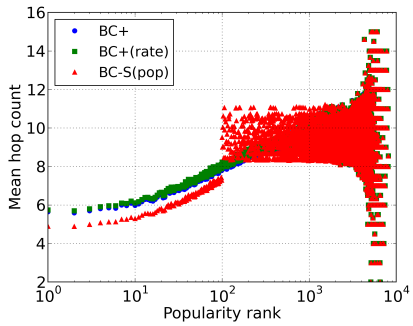


Fig. 9. Dist. of mean content hop count by popularity rank

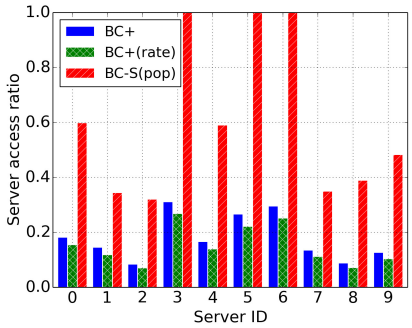


Fig. 10. Server access ratio

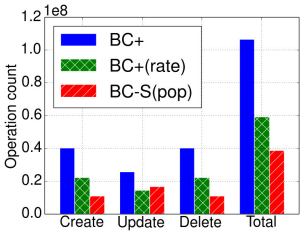


Fig. 4. BC operation

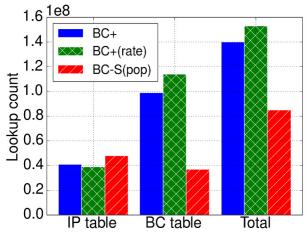


Fig. 5. Lookup count of routing table

opportunities of BC trail and cache. On the other hand, our proposed method of BC-S (pop) reduces total BC operation count to approximately 37% compared with that of BC+ even though requests for popular contents ranked in top 1% account for 55.5% of all generated requests. The reason is the following. In all the three methods, cache replacement causes to delete a part of BC entries for the evicted content, and then to create a new BC entries for the newly cached content. In BC-S (pop), however, an opportunity of cache replacement is reduced, and then just updating BC entries, which means refreshing expiration times of the BC entries, is enough to maintain the system.

2) lookup count of routing tables: Lookup count is the total number of events referring to routing table at each router.

Figure 5 shows that BC-S (pop) has the lowest total lookup cost of routing table which is approximately 61% compared with BC+. Focusing on lookup count in IP table, BC-S (pop) increases lookup cost to approximately 117% of BC+ whereas BC+ (rate) decreases to approximately 95%. Focusing on that in BC table, BC-S (pop) decreases lookup cost to approximately 37% of BC+ whereas BC+ (rate) increases to approximately 115%. We should note that BC-S (pop) increases the lookup count in IP table but decreases the total lookup count, whereas BC+ (rate) increases the total lookup cost. These results mean that we cannot reduce lookup count through just reducing an opportunity of BC entry creation without ingenuity.

3) Hop count: Request hop count is the number of links that a request traverses before it reaches the target content from the requesting user. Content hop count is the number of links that a content traverses before it reaches a requesting user from a node providing the target content.

Table III shows that BC-S (pop) reduces hop count of both request and content. Figure 6 shows that BC-S (pop) increases the amount of request with smaller hop count (13 or less) and decreases with larger hop count (14 or more), and Figure 8 shows that BC-S (pop) also increases the amount of content with extremely smaller hop counts (5 or less). These results are due to the two reasons shown in Figures 7 and 9. One is that popular contents ranked in top 10^2th are retrieved more frequently from near caches with smaller hop count in BC-S (pop) compared with the other methods, because BC-S (pop) increases the amount of cached copies of popular contents. The other is that unpopular contents ranked under 10^2th are retrieved from servers mainly with between 8 and 11 hop counts in BC-S (pop). These hop counts are smaller than ones

 ISBN: 978-1-61208-377-3

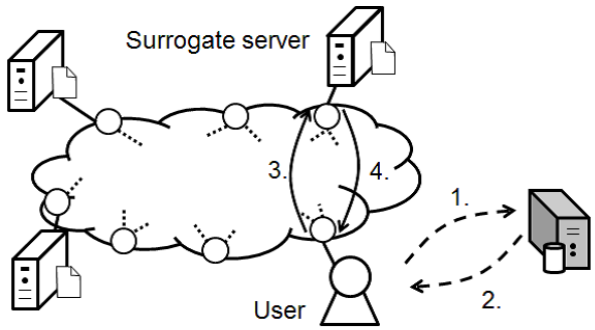


Fig. 11. CDN system applying DNS redirection

from servers and caches in the other methods, because BC-S (pop) does not create caches and BC trails of unpopular contents, which guides requests to a far cache.

4) Server access ratio: Server access ratio is defined as

$$\frac{\text{The number of requests which reached a server}}{\text{The number of all the generated requests}}.$$

As server access ratio becomes lower, more requests reach cache-nodes, and this results in higher cache utilization.

Figure 10 shows that BC-S (pop) increases server access because requests for unpopular contents are always forwarded to server in IP routing. On the other hand, Table III says that almost all requests for popular contents reached caches of the corresponding contents since requests of popular contents occupied 55.5% of all generated requests in this simulation setting shown in Figure 3(b). We can improve this metric through just loosing the limitation on scoping by expanding the target range, but it may cause to worsen other metrics including hop count described above. In short, there is performance trade-off relationship between server access ratio and other metrics.

D. Simulation scenario (vs CDN)

In this section, we illustrate the CDN model, which we developed and used in the simulation to evaluate the proposed method by comparing with the current major content distribution technique.

The core idea of CDN [11] is to replicate contents into some *surrogate servers*, which provide the replicated contents instead of the *origin server*, and to redirect web requests for the contents to one of the surrogate servers. One of the typical implementations is shown in Figure 11.

In order to make a CDN model, we should consider the following topics: surrogate placement, surrogate selection on request redirection, object selection on replication, surrogate selection on replication, and decision on amount of replication.

- surrogate placement

We applied *greedy algorithm* [21] to determine where to place surrogate servers. The idea of this algorithm is as follows. At first, among all N routers, the first surrogate server is placed on a router to which the summation of the numbers of hops from all users is minimum. Then, the second surrogate is placed on another router so that the summation of the numbers of hops from all users to the nearest surrogate of all becomes minimum. We iterate this process until M surrogate servers are placed. Note that in this simulation scenario N is 1000 as described in Table II and M is set to 1 and 5, as described below.

In the real world, the number of potential sites depends on where we can place data center facilities and what ISP provides the Internet connection for the data center.

- surrogate selection on request redirection

We simplified *DNS redirection* technique in our simulation. The core idea of this technique is illustrated in Figure 11. When the DNS receives a name resolution request for a content, it resolves the request so that a web request is forwarded to the “best” surrogate server. In order to realize this best selection, an actual CDN operator monitors the state of service, network and servers [12]. In our simulation, we did not consider the monitoring overhead in order to simplify the scenario; we just redirect web requests to the nearest surrogate server according to pre-computed topology information. In other words, we assume that all surrogate servers and network links have the unlimited capacity.

Three topics: object selection on replication, surrogate selection on replication and decision on amount of replication, mean what content should be replicated, where a replica of the content should be stored, and how many replicas should be made, respectively. We simplified these 3 topics in our simulation; all surrogate servers statically store each replica of all kinds of contents under the CDN’s control. Although there are some strategies [11] about these topics, our assumption enables us to reduce simulation parameters.

In the earlier part of this sub-section, we described the simplified CDN model, which we developed in this paper. We then show the other parameters in the rest part.

- Compared methods

We compared the following 3 methods.

1) IP CDN (1) : the simplified CDN model which has 1 surrogate in network.
2) IP CDN (5) : the simplified CDN model which has 5 surrogates in network.
3) BC-S (pop) : Proposed method.

In this simulation, target contents are also “top 1%” popular contents among all like in Section IV-B. We replicated only the target contents in the surrogates in *IP CDN(1)* and *IP CDN(5)*.

We set the other parameters same as setting in Section IV-B.

 ISBN: 978-1-61208-377-3

TABLE IV. MEAN VALUES

Method	Request	Data
IP CDN(1)	8.45	8.45
IP CDN(5)	7.30	7.30
BC-S(pop)	8.47	7.54

TABLE V. FAIRNESS INDEX OF CONNECTION COUNT ON LINKS

Method	Router NW
IP CDN(1)	0.1209
IP CDN(5)	0.3341
BC-S(pop)	0.4969

E. Simulation results (vs CDN)

We show the results in Tables IV and V, and in Figures 12 - 17. We used the following evaluation metrics: request hop count, content hop count, connection count on link in the router network and cumulative distribution function (CDF) of upload connection count on each surrogate.

1) Hop count: With respect to request, BC-S(pop) requires more hop count for requests than IP CDN (n). This is because BC trail sometimes causes a long travel of request described as a long tail in Figure 12. In IP CDN (n), location of the content is fixed at each surrogate, and the request redirection provides the smaller hop content travel for the target contents (see Figures 13 and 15). On the other hand, BC-S (pop) and IP CDN (5) have similar content hop count. In BC-S (pop), distributed large amount of user-cache with higher popularity occurs small hop travel of contents.

2) Connection count on links: Table V shows Fairness Index of connection count on each link which interconnects routers. In Table V, BC-S(pop) achieves higher fairness of connection count on links than each IP CDN (n) does. This is because Breadcrumbs architecture makes a distributed system whereas CDN makes a centralized system, even though the both system are on a client-server model. Figure 16 shows complementary cumulative distribution function (CCDF) of connection count on links. In Figure 16, each IP CDN (n) forms a straighter shape than BC-S(pop) like the Pareto distribution. This means that a few links near surrogate servers are extremely crowded but most of links are not in IP CDN (n). In BC-S(pop), caches of target contents are distributed everywhere and this promotes better load-balancing compared with IP CDN (n). This difference on fairness arises from the difference of fundamental policy on architecture design; CDN takes a centralized approach whereas Breadcrumbs does a decentralized approach.

3) Upload connection count on each surrogate server: Figure 17 shows that the distribution of connection count on each surrogate server follows normal distribution and the connection count is stable. In our simulation, each connection consumes 5 Mbps on the link bandwidth and each access link for surrogate servers may be able to accept these connections at the same time, but some surrogate may not be because it requires expensive equipment in terms of workloads. Comparing IP CDN (1) with IP CDN (5), multiple surrogates placed at distributed locations achieve load-balancing but increasing a new surrogate at another place is not easy. On the other hand, our proposed method based on Breadcrumbs uses distributed and totally large user-cache space, and upload connection count on each user in BC-S (pop) is approximately 1.4. This is extremely smaller than that on each surrogate, and this difference on workloads also arises from the difference of the fundamental policy on architecture design.

4) IP CDN (5) vs IP CDN (1): IP CDN (5) outperforms IP CDN (1) in the all metrics because of more surrogates without any cost. More surrogates require more management cost but we ignore the cost in this simulation for simplicity. In addition, we should note that more surrogates at different locations in this simulation means more data center facilities in the real world. It is therefore easier and more feasible to enhance the existing surrogates than to increase new surrogates.

V. CONCLUSION

Our research goal is to establish a scalable and feasible architecture, which integrates the conventional IP network and a new content-oriented network. In terms of scalable routing on CON itself, not an active approach but a passive approach should be adopted for simplicity. Therefore, we established the routing method based on Breadcrumbs among researches on CON routing. Breadcrumbs is an architecture with guidance information (routing information) to a node holding a cached content. Breadcrumbs's passive and simple approach allows us to make scalable and feasible CON autonomously in cooperation with cached contents in network. In this paper, we focused on popularity as a criterion of BC-Scoping framework, and we proposed BC-Scoping on Popularity, which distributes the guidance information of the only popular contents. As for unpopular contents, IP routing is definitely applied, which is desirable from the viewpoint of routing overhead. Such contents are requested less frequently, hence most of the guidance information for the contents are not used effectively. The proposed method clearly separates the forwarding policy based on popularity of content so that it promotes cache utilization of popular contents and server utilization of unpopular contents. This method brings the following benefits: reduction in system overhead including BC operation cost and lookup cost of BC table, small-hop content retrieval. We should note that the latter benefit is in return for increase of server access because there is trade-off relationship between hop count and server access. Finally, we conducted simulations and demonstrated the effectiveness of our proposal. We can highlight that we introduced the state-of-the-art requesting model based on content popularity for evaluation, and that we quantitatively clarified the superiority of the proposed method in comparison to most popular and widespread CDN approach. We have the following two tasks as our future works. First, we will consider a way to estimate popularity of contents at each servers or cache node. One idea is that each server and cache node can count access frequency at itself autonomously, and another is that we prepare dedicated servers for managing content popularity. Second, we will implement Breadcrumbs architecture [13] and evaluate the effectiveness of our proposal in the real world.

 ISBN: 978-1-61208-377-3

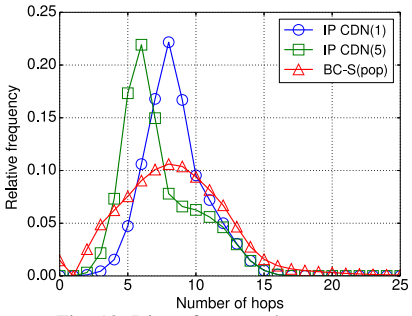


Fig. 12. Dist. of request hop count

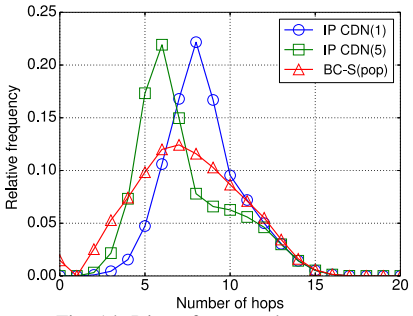


Fig. 14. Dist. of content hop count

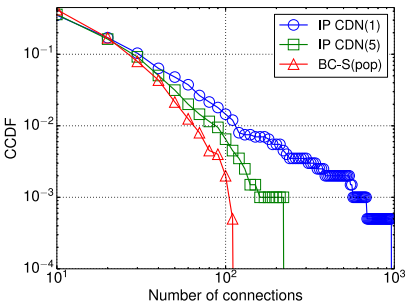


Fig. 16. CCDF of connection count on links

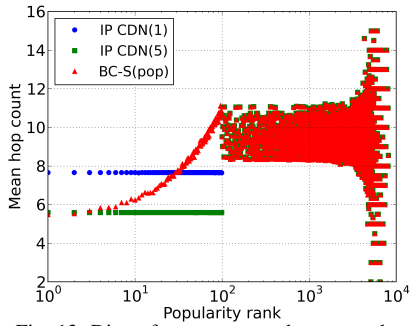


Fig. 13. Dist. of mean request hop count by popularity rank

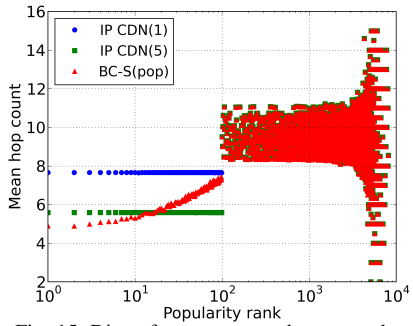


Fig. 15. Dist. of mean content hop count by popularity rank

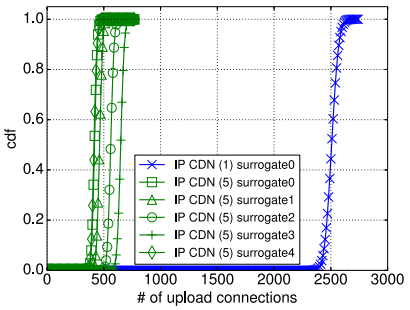


Fig. 17. CDF of upload connection count on surrogates

ACKNOWLEDGEMENT

This research was supported in part by National Institute of Information and Communication Technology (NICT), Japan. We also appreciate insightful comments of Prof. J. Kurose and Dr. E. J. Rosensweig, University of Massachusetts, and the other project members.

REFERENCES

[1] G. Barish and K. Obraczka, "World wide web caching: Trends and techniques," IEEE Communications Magazine, vol. 38, no. 5, May 2000, pp. 178–184.

[2] J. Wang, "A Survey of Web Caching Schemes for the Internet," ACM SIGCOMM Computer Communication Review, Vol. 29, Issue 5, Oct. 1999, pp. 36–46.

[3] J. Choi, J. Han, E. Cho, K. Kwon, and Y. Choi, "A Survey on Content-Oriented Networking for Efficient Content Delivery," IEEE Communications Magazine, vol. 49, no. 3, Mar. 2011, pp. 121–127.

[4] D. Lagutin, K. Visala, and S. Tarkoma, "Publish/Subscribe for Internet: PSIRP Perspective," Towards the Future Internet - A European Research Perspective, 2010, pp. 75-85.

[5] I. Stoica, D. Adkins, S. Zhuang, and S. Shenker, "Internet Indirection Infrastructure," IEEE/ACM Transactions on Networking, vol. 12, no. 2, Apr. 2004, pp. 205–218.

[6] T. Koponen et al. "A Data-Oriented (and Beyond) Network Architecture," in Proc. ACM SIGCOMM 2007, Oct. 2007, pp. 181–192.

[7] E. J. Rosensweig and J. Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," in Proc. IEEE INFOCOM 2009, Apr. 2009, pp. 2631–2635.

[8] V. Jacobson et al, "Networking named content," in Proc. ACM CoNEXT 2009, Dec. 2009, pp. 1–12.

[9] M. Kakida, Y. Tanigawa, and H. Tode, "Breadcrumbs+: Some Extensions of Breadcrumbs for In-network Guidance in Content Delivery Networks" in Proc. SAINT 2011, July 2011, pp. 376–381.

[10] M. Kakida, Y. Tanigawa, and H. Tode, "Distribution Method of In-network Guidance Information for Inter-AS Content-Oriented Network Topology," Proc. WTC 2012 Poster Session, Mar. 2012.

[11] A. Passarella, "Review: A survey on content-centric technologies for the current Internet: CDN and P2P solutions," Computer Communications, vol. 35, no. 1, Jan. 2012, pp. 1–32.

[12] J. Dilley et al, "Globally Distributed Content Delivery," IEEE Internet Computing, vol.6, no.5, 2002, pp. 50–58.

[13] T. Yagyu, M. Kakida, Y. Tanigawa, and H. Tode, "Prototype Development of Active Distribution of In-network Guide Information for Contents Delivery," in IEICE Technical Report, vol. 111, no. 468, NS2011-211, Mar. 2012 (in Japanese), pp. 179–184.

[14] T. Yagyu, "Synergic Effects among Plural Extensions of Breadcrumbs for Contents Oriented Networks," Proc. AFIN 2013, Aug. 2013, pp. 35–42.

[15] T. Tsutsui, H. Urabayashi, M. Yamamoto, E. Rosensweig, and J. Kurose, "Performance Evaluation of Partial Deployment of In-Network Query Direction Method in Content Oriented Networks," in Proc. Future Net V, IEEE ICC 2012, Canada, June 2012, pp. 7386–7390.

[16] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino "Exploit the Known or Explore the Unknown? Hamlet-Like Doubts in ICN," Proc. ICN'12, ACM SIGCOMM 2012, Aug. 2012, pp. 7–12.

[17] B. M. Waxman, "Routing of multipoint connections," IEEE J. Selected Areas in Communications (Special Issue on Broadband Packet Communication), vol. 6, no. 9, Dec. 1998, pp. 1617–1622.

[18] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," IEEE/ACM Transactions on Networking, vol.17, no.5, Oct. 2009, pp. 1357–1370.

[19] K. P. Gummadi et al, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," Proc. ACM SOSP 2003, Dec. 2003, pp. 314–329.

[20] O. Saleh and M. Hefeeda, "Modeling and Caching of Peer-to-Peer Traffic," Proc. ICNP 2006, 2006, pp. 249–258.

[21] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," Proc. IEEE INFOCOM 2001, 2001, pp. 1587–1596.

 ISBN: 978-1-61208-377-3

Mobile Edge Computing: A Taxonomy

Michael Till Beck, Martin Werner, Sebastian Feld
Ludwig Maximilian University of Munich
{michael.beck,martin.werner,sebastian.feld}@ifi.lmu.de

Thomas Schimper
Nokia Networks
thomas.schimper@nsn.com

***Abstract*—Mobile Edge Computing proposes co-locating computing and storage resources at base stations of cellular networks. It is seen as a promising technique to alleviate utilization of the mobile core and to reduce latency for mobile end users. Due to the fact that Mobile Edge Computing is a novel approach not yet deployed in real-life networks, recent work discusses merely general and non-technical ideas and concepts. This paper introduces a taxonomy for Mobile Edge Computing applications and analyzes chances and limitations from a technical point of view. Application types which profit from edge deployment are identified and discussed. Furthermore, these applications are systematically classified based on technical metrics.**

***Index Terms*—edge deployment, cellular networks, classification**

I. Introduction

Increasing utilization of network resources is one of the most apparent challenges for mobile network operators. Data traffic contributes heavily to today's overall mobile network traffic. Many mobile applications rely on data and services hosted in remote data centers. This induces high network load, since data have to be up- and downloaded to and from mobile devices and data centers connected to the Internet.

Moreover, new mobile applications accessing Internet services are expected to further contribute to this trend: In fact, bandwidth demands are expected to continue doubling each year [1]. And this trend does not yet incorporate for effects due to wearable devices and the Internet of Things, which add new devices such as Google Glass to the mobile ecosystem. With the increased computational power of these devices, novel application scenarios become realistic including augmented reality leading to an even higher bandwidth demand.

To keep up with these increasing demands, network operators are obliged to enhance and upgrade capacities of existing network resources continuously. Furthermore, they are impelled to integrate novel technologies into their infrastructure in order to provide sufficient quality of experience for mobile end users. New technologies like LTE Advanced introduce higher bandwidth capacities and lower latency. Higher edge capacities, however, also directly affect utilization within the network core and entail further investments. Both, enhancing existing resources and integrating new technologies, comes with significant operational cost.

Mobile Edge Computing (MEC) has recently been proposed as a promising technology to overcome this dilemma in certain scenarios. MEC aims at reducing network stress by shifting computational efforts from the Internet to the mobile edge. Traditionally, devices deployed at the mobile edge solely act as mobile access points: Base stations forward traffic, but do neither actively analyze nor respond to user requests. Thus, they do not provide computing resources for hosting edge services beyond network connectivity. MEC introduces new network elements at the edge, providing computing and storage capabilities at the edge. Therefore, new devices are deployed and co-hosted at base station towers. In the following, these devices are referred to as MEC servers.

Fig. 1 depicts the MEC ecosystem and the integration of MEC servers into the mobile network topology. There are four stakeholders involved in this scenario: 1) Mobile end users using User Equipment (UE), 2) network operators owning, managing, and operating base stations, MEC servers, and the mobile core network, 3) Internet infrastructure providers (InPs) maintaining Internet routers, and 4) application servive providers (ASPs) hosting applications within data centers and content delivery networks (CDN). Mobile devices (UE) connect to the eNodeBs which translate Radio signals so they can be routed through the wired access and core networks. MEC Servers are deployed in close proximity of the eNodeB, typically by physically attaching it there and looping the traffic through the MEC server for further processing the data. The MEC server is capable of participating both in user traffic and control traffic (S1-U and S1-C interfaces). MInP and ASPs deploy rulesets, filters, and MEC services at the MEC servers, defining how to handle specific traffic. In this way, MEC services are capable of managing specific user requests directly at the network edge, instead of forwarding all traffic to remote Internet services. MEC servers either process a request and respond directly to the UE or the request is forwarded to remote data centers and content distribution networks (CDNs).

Being directly handled by services hosted on MEC servers, these requests do not need to be forwarded through the core infrastructure. Traditionally, all data traffic is routed through the core network to a base station which delivers the content to mobile devices. In the MEC scenario, MEC servers take over

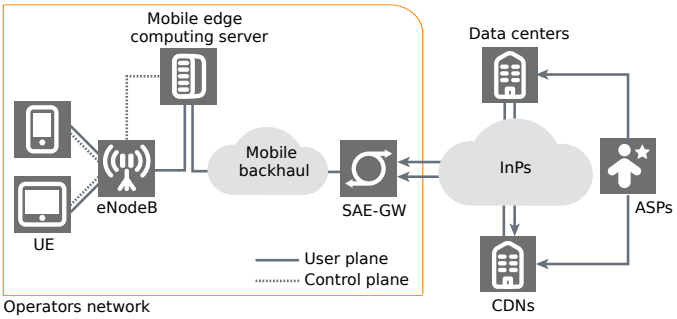


Fig. 1: Mobile Edge Computing Topology

 ISBN: 978-1-61208-377-3

some or even all of the tasks originally performed by Internet services. Being co-located next to base stations, computing and storage resources of MEC servers are also available in close proximity to mobile users, eliminating the need of routing these data through the core network. Therefore, MEC is seen as a future, promising approach to increase quality of experience in cellular networks [2]–[4]. Furthermore, it enables the deployment of novel application types at the mobile edge.

This paper provides an analysis of technical chances and limitations of MEC by identifying, discussing, and classifying various applications and application types for the deployment at the mobile edge.

II. Related Work

Until today, MEC servers have not been deployed in cellular networks; thus, the MEC concept has been discussed only from a theoretical perspective so far. However, there are some related approaches that are similar to this concept. For example, mobile cloud computing is highly related to mobile edge computing. A survey on mobile cloud computing is given by Dinh et al. [5], describing cloud-affine mobile application types. As an example for mobile cloud computing, the cloudlet concept is discussed by Satyanarayanan et al. [4]: Cloudlets are trusted, resource-rich, mostly stationary computers with fast and stable Internet access, offering computing, bandwidth, and storage resources to nearby mobile users. While MEC servers are operated by mobile infrastructure provider, cloudlets are owned and managed by mobile end users. Mobile users access cloudlet via a local area network such as Wi-Fi in order to instantiate services. Not being connected to the mobile network, cloudlets do not share network operator related knowledge. Thus, cloudlets are suitable for offloading resource-intense tasks from the mobile end user device in order to increase execution speed or battery lifetime.

Fesehaye et al. [6] focus especially on interactivity when stating that cloudlets are also capable of caching and transferring content. A content-centric local networking approach is introduced, using interconnected cloudlets. Their contribution is threefold: First, a mobile infrastructure as a service cloud is defined, using both cloud technology and cloudlets. Second, in order to realize content-centric features, a wireless routing protocol is proposed in order to enable communication between two cloudlets as well as between two mobile users via a cloudlet. Third, the impact of cloudlets on interactive mobile cloud applications like file editing, video streaming, and messaging is analyzed. However, offloading and content caching are just two of the use cases for MEC. In contrast to cloudlets, MEC servers are widely deployed and available to all mobile users, not just to some specific ones. Being co-located with base stations, MEC servers provide additional features such as being able to access position and mobility information. This paper discusses these features, listing and classifying application types for the deployment at the mobile edge.

Until now, the MEC concept itself has mainly been discussed from a non-technical perspective. E.g., IBM discusses economical benefits for businesses and M2M applications [3]. A first real-world MEC platform was introduced and motivated by Nokia Networks [2] in 2014. In this concept, MEC servers are standard IT equipment with processing and storage capacity directly placed at mobile network's base stations. Being placed at the mobile edge, MEC servers are capable of collecting real-time network data like cell congestion, subscriber locations, and movement directions. Furthermore, some individual application types (but neither analyzed nor classified from a technical perspective) are motivated for running at the mobile edge. This concerete mobile edge computing platform will be described in the following section.

III. A first Mobile Edge Computing Platform

As discussed in the previous section, some initial ideas on Mobile Edge Computing have been discussed in literature before. However, these discussions are limited to a more or less theoretical perspective, since no real implementation of MEC servers was introduced so far. Just in 2014, Nokia Networks has introduced a very first real-world MEC platform [2]: Radio Applications Cloud Servers (RACS) represent concrete incarnations of MEC servers.

This section shortly discusses NSN's approach as an example for a realistic MEC deployment. The section is structured as follows: First, hardware configuration is described; then, the software architecture is explained; and third, traffic forwarding and filtering rules are depicted.

In line with Figure 1, NSN's MEC servers are deployed next to base stations: they are co-hosted with base stations and are directly linked to them. MEC servers are equiped with commodity hardware, i.e. usual server CPUs, memory, and communication interfaces. Application deployment is based on cloud technology and virtualization. Therefore, RACS provide a VM hypervisor (see Fig. 2) for the deployment of VM images running MEC applications.

VMs have to fulfil certain requirements, like providing a self-monitoring service that keeps sending heartbeat messages to the RACS system. The hypervisor will reboot the VM if heartbeat messages are not sent by the VM, ensuring that the VM is automatically reinitialized after some applications crashed. Furthermore, for security considerations, VMs have to be signed before deployment. This enables the operator to verify that the VM state has not been altered by malicious offenders. VMs are able to communicate with the RACS platform via a message bus, as most applications running on the mobile edge are expected to be event driven. Via the message bus, VMs subscribe to message streams, i.e., topics. This way, VMs are able to retrieve UE data streams and cell-related notifications. As an example, some subscription topics refer to specific traffic classes sent or received by mobile devices. VMs can subscribe to all traffic with a specific destination address or port number.

 ISBN: 978-1-61208-377-3

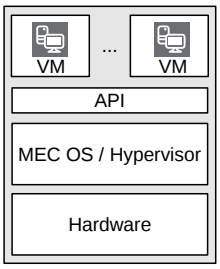


Fig. 2: RACS Architecture

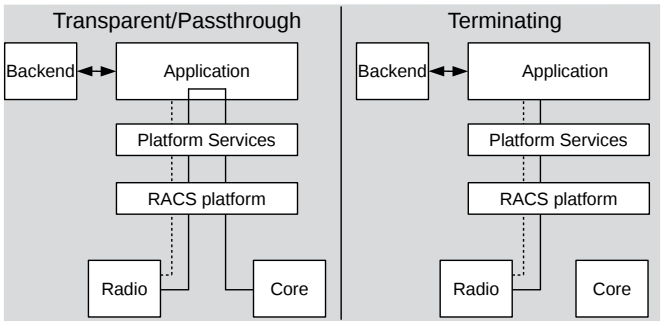


Fig. 3: Application Types

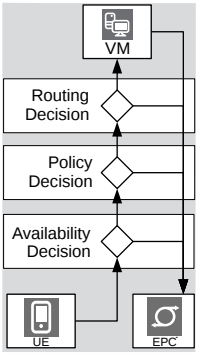


Fig. 4: Forwarding and Filtering Rules

Two main categories of applications can be deployed at RACS servers, depending on the traffic flow: transparent/pass-through and terminating applications (cf. Fig. 3). The dotted line in Fig. 3 represents the control plane, which is available in both applications types. The line having arrowheads represents out-of-band communication that both application types are optionally able to perform. The solid line is the user plane.

Transparent/pass-through applications are capable of monitoring, rerouting, and augmenting UE traffic. In this situation, additional header information can be introduced to HTTP requests including network-specific information, which is not available to ASP services in traditional cellular networks.

For terminating applications, UE traffic is encapsulated into IP packages with a virtual IP address. This packet flow is then routed into a VM, where it terminates (note the truncated solid line in Fig. 3). If the VM is not running or no MEC server is co-hosted with the current base station, the encapsulated packages are routed to a server in the mobile network core handling the requests. Packages are rerouted transparently, which means that developers of mobile applications can refer to the same URL in order to access the service which is provided by the MEC server's infrastructure were applicable, or by a backend service, where needed.

Mobile network operators define forwarding and filtering rulesets for traffic routed through MEC servers. Based on both privacy considerations and application providers' demands, these rulesets specify which data are sent to which type of application. In accordance with the subscriptions to the topics mentioned above, mobile traffic is routed through the VMs. Fig. 4 provides an overview of this static decision tree: UE traffic is sent to the base station and its co-located MEC server. For each rule, it is validated whether the application corresponding to this rule is up and running, i.e., whether the VM hosting the application is active. If this is the case, the filtering ruleset is applied to identify information that is permitted to be accessed by the application. The last step is the actual routing decision. After a positive result, information is visible to the application.

In the following, application types and use cases will be discussed which are promising candidates for being hosted by MEC/RACS servers.

IV. Applications and Use Cases

Introducing a Mobile Edge Computing platform into a cellular network allows for applications to be executed directly at the serving base station. While the concept of Mobile Edge Computing has been introduced in literature before, it still remains an open question, which applications profit from being deployed at the edge. This section categorizes and discusses several application types which are promising candidates for the deployment at the mobile edg: Subsection A introduces a classification scheme for MEC applications; subsection B discusses several applications and subsection C highlights the main benefits of Mobile Edge Computing.

A. Classification

This section introduces a classification for several approaches. It is based on three levels of abstraction (cf. Figure 5). As a first distinction, the application classes "Offloading", "Edge Content Delivery", "Aggregation", "Local Connectivity", "Content Scaling" as well as "Augmentation" were identified. While there might be additional classes of applications which could benefit from edge computing, we believe that these application classes will have the strongest impact. In order to further organize application examples and their demands, subgroups of applications showing a similar footprint with respect to resource demands were introduced. Then, concrete examples of applications were given to show the variability of applications inside classes exploiting mobile edge computing resources in a similar way. Another perspective on the classification of applications is given by starting with advantages of edge computing: The most obvious advantage of edge computing inside cellular networks is given by a reduction of end-to-end delay. When packets do not have to travel through the evolved packet core to the application server on the Internet, an application can provide real-time services with strong, constant and known bounds on the delay. A reduced delay motivates the deployment of applications from all given classes: In every case, a solution without edge computing would involve a transmission through the core network as well as through Internet links towards

 ISBN: 978-1-61208-377-3

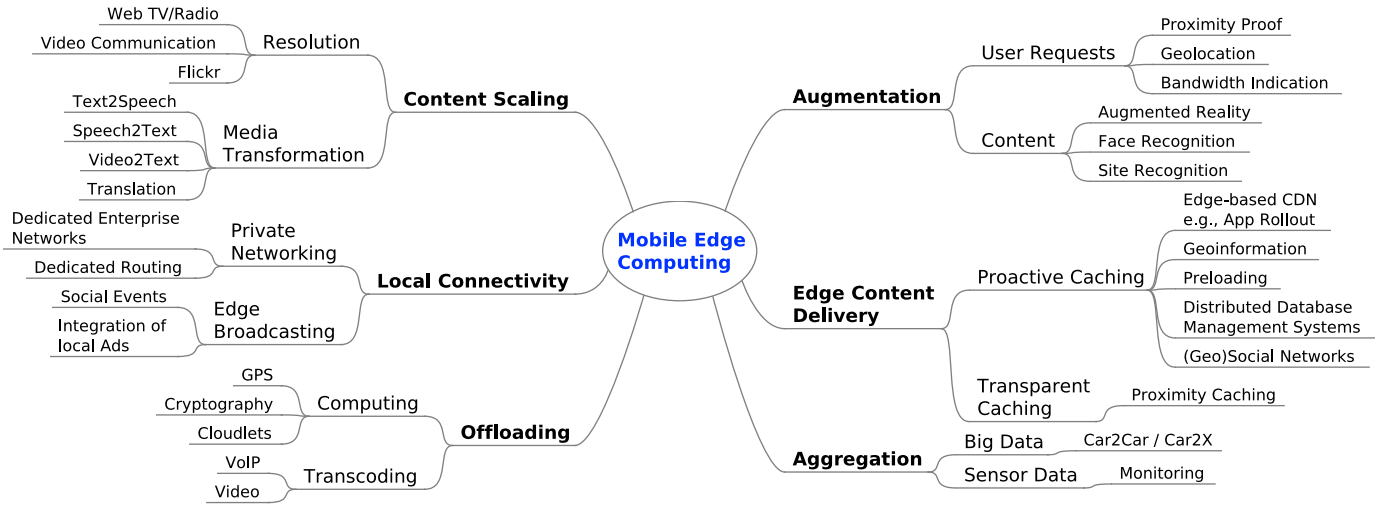


Fig. 5: Mobile Edge Computing: Applications and Use Cases

the application host and back. Additionally, the use of edge computing makes offloading feasible in more cases, as today's radio bandwidth is much higher compared to usable Internet bandwidth and tasks that would typically be performed on the mobile device due to the size of their input can be performed on the edge. Finally, a third motivation for edge computing is given by the local nature of the edge computing servers: By storing only relevant information for the coverage of a single cell, many computational tasks can be performed on small datasets reducing overhead and possibly increasing privacy: A location-based information service, for example, can provide its service inside each cell and in the case of edge-terminating traffic, the location of mobile users keeps inside the mobile network, where it was previously known.

In the following, advantages and disadvantages of each application class are discussed using the examples given in Figure 5. To this end, several key metrics are discussed and a rating for each of these metrics is provided with respect to the three main entities: Mobile user (UE), cellular network provider (MInP), and application service provider (ASP). For the purpose of this evaluation, the following metrics are considered in order to evaluate the feasibility of edge computing for a given class of applications:

Power Consumption: The effect on power consumption of each power-consuming device including the mobile device and also the base stations. Furthermore, power consumption is relevant to network operators, though with a lower impact.

Delay: The effect on delay introduced due to modified communication, computation, and system's complexity.

Bandwidth utilization: The effect on bandwidth demand and cost for each entity.

Scalability: The effect on scalability of algorithms exploiting the available location information at the edge.

B. Applications

Figure 6 gives a qualitative assessment of the impact of the identified application classes on the three main stakeholders. The following sections explain examples for each application class and motivate the decisions made for the assessment of the table.

Offloading: Even today, many mobile applications delegate resource- or power-intense tasks to remote services due to limited hardware capabilities. MEC servers offer additional capacities for hosting such services at the mobile edge. The concept is expected to increase limited computing, storage, bandwidth, or battery capacities of mobile devices by referring to external, resource-rich systems. Compute-intense tasks are offloaded either because they can not be executed in-time by the UEs due to limited hardware capabilities or they are offloaded in order to reduce power consumption of mobile devices in cases where the power consumption needed for computation exceeds the power consumption needed for wireless transmission. If no MEC server is available, mobile devices can degrade gracefully to a more distant MEC server, Internet cloud servers, or fallback to their own hardware resources [4], [6]. For example, calculating GPS positions is a power-intense task which can gainfully be offloaded to remote servers. Also, asymmetric encryption requires much more battery power than symmetric approaches. Therefore, asymmetric encryption is offloaded, and more battery-friendly encryption methods are chosen in order to encrypt communication between UEs and the base station. Offloading of power-intense tasks like transcoding of multimedia traffic also falls into this category. VoIP applications transcode traffic depending on the current load of the base stations, enabling real-time bitrate adaptations and better QoE.

With respect to our metric system, offloading is motivated by reduced delay due to the fact that traffic between MEC servers and mobile devices has not to be routed through the core network. Another objective of offloading is the reduction of power consumption of the mobile device. Of course, the sending of the task request and response does not have to consume more power than the local execution. Core and ASP

 ISBN: 978-1-61208-377-3

Class	Entity	Metric			
		Power Consumption	Delay	Bandwidth Usage	Scalability
Offloading	UE	++	++	o	o
	MInP	+	+	++	o
	ASP	o	++	+	+
Edge Content Delivery	UE	o	++	o	o
	MInP	o	++	++	+
	ASP	+	++	+	++
Aggregation	UE	o	-	o	o
	MInP	+	+	++	+
	ASP	++	-	++	+
Local Connectivity	UE	o	++	o	o
	MInP	+	+	++	++
	ASP	o	++	++	++
Content Scaling	UE	o	++	o	o
	MInP	-	+	+	++
	ASP	+	++	++	++
Augmentation	UE	o	+	o	o
	MInP	o	o	o	o
	ASP	o	+	o	o

Fig. 6: Classification of Mobile Edge Computing Applications

can benefit indirectly from offloading, namely when more calculations are performed at the edge as compared to the situation without offloading. Offloading introduces cost due to higher system complexity: Even simple systems become complex distributed systems and have to deal with communication, marshalling, availability, and errors.

Edge Content Delivery: MEC servers offer resources for the deployment of additional content delivery services at the network edge. Content traditionally hosted by Internet services/CDNs is now shifted more to the network edge. MEC servers operate as local content delivery nodes and serve cached content. Caching techniques, not only in the context of Mobile Edge Computing, can be classified as being either reactive/transparent or proactive. *Transparent Caching*: Caching is transparent if neither the UE nor the ASP are aware of the caching MEC server. As shown by Ericcson, 10% of mobile data traffic is expected to be generated by web browsing, and more than 50% by video data [7] in 2019. Therefore, caching content at the edge is a promising approach to reduce communication quantity and latency for core network providers. *Proactive Caching*: Content is non-transparently cached before it was requested, since it is expected to generate high network utilization in the future. One example here is the roll-out of software updates before they are actually requested by mobile devices. Another example is caching proximity-related data: Geo-Social Networks (GSNs) like Google Latitude and Yelp store region-related content. Mobile users often use these services to request information about geographically nearby locations and places (restaurants, etc.).

Proactive caching is highly related to content distribution networks and is expected to lead to further improvements in terms of bandwidth reduction for the core net and the ASP, and in terms of shorter transmission delays for the mobile devices. ASPs play an important role in this scenario, since they provide relevant information on which content should be distributed throughout the network. Another example is the pre-loading of user content. In order to reduce transmission delays at the UE site, ASPs can preload content that is expected to be requested by the UE user. In contrast to proactive caching, decisions whether (and which) additional content should be send to MEC servers depend on actions performed by each specific UE user. Pre-loading is well known and actively used by companies like Amazon, for example: Amazon silently pre-loads content on the client side that might possibly be requested by the user in the near future while the user is browsing the Amazon website [8]. This leads to decreasing transmission delay and an improved user experience. In the context of Mobile Edge Computing, pre-loading is shifted from UEs to MEC servers in order to decrease power consumption caused by the transmission of data to the ME.

Both approaches can be used either isolated or shared. In the isolated scenario, each cache works independently of other caches: Content already cached by other MEC servers is not shared. In the shared scenario, MEC servers cooperate and obtain content from other MEC servers.

Technically, edge content delivery reduces network utilization and network delay. Similar to distributed database management systems (DDBMS), edge content delivery aims at storing data in close proximity to where they are usually requested. This kind of data localization leads to a reduction of computational complexity, compared to centralized database systems. But it also decreases access delays with respect to latency, since communication paths are kept short [9]. Also, overall bandwidth usage decreases, since less network resources are needed to transfer data: On the one hand, MEC servers have to synchronize with each other to ensure that data are stored consistently, which comes with additional communication overhead. But, on the other hand, UEs frequently requesting data can fetch them directly from a DDBMS instance nearby instead of having to establish remote

 ISBN: 978-1-61208-377-3

connections to a centralized service. Therefore, this leads to an overall reduction of communication overhead in the core network and also in the ASP network. Of course, the applicability of edge content delivery depends on the locality of the data. Utilization of the core network resources decreases if UEs frequently request data stored by the local DDBMS instances.

Aggregation: Instead of routing all UE data to core routers separately, MEC servers are capable of aggregating similar or related traffic and, thus, reduce network traffic. As an example, many Big Data applications like Car2Car solutions generate a lot of similar and region-related event notifications which can be aggregated. This also applies in the context of monitoring applications where many devices measure similar data that can be aggregated at the edge.

Due to the fact that the quantity of data received by ASPs would decreases, aggregation has a positive effect in terms of ASPs' bandwidth utilization, power consumption, and scalability. However, delay increases since data need to be processed by MEC servers. Since core network traffic decreases, the same applies for bandwidth utilization, power consumption, and scalability of the MInP. Operating MEC servers comes with additional power consumption cost, however, total power consumption is expected to decrease as a result of lower core utilization.

Local Connectivity: With traffic being routed through MEC servers, servers are capable of separating traffic flows and redirecting traffic to other destinations. An application of this class is connecting enterprise users directly via base stations deployed on enterprises' rooftops to the enterprise network. As an example, this applies on sports/music events where cameras catching additional viewpoints broadcast their content among users in the cell. Furthermore, Local Breakin allows for local redistribution of data fed into the cell, for example, advertisements and information related to the geographical location of the base station. Thus, MEC servers broadcast locally generated and locally relevant content within the cell.

Traffic is routed by circumventing Internet routers, leading to lower communication delay for UEs and ASPs. Furthermore, MInP's bandwidth utilization and power consumption is reduced, since traffic is not routed through the core network. Reduced network utilization has a positive impact with respect to MInP's communication delay.

Content Scaling: MEC enables downscaling of user-generated traffic before it is routed through the mobile core network. Content scaling can also be applied to traffic sent by Internet servers. Scaling UE-generated content before it is delivered to ASPs' data centers decreases bandwidth demands of ASPs. As an example, image sharing sites like flickr and facebook downscale user generated content in order to reduce storage demands. Downscaling UE content directly at the edge also reduces MInP's core network utilization. Additionally, MEC also enables real-time scaling of Internet content – if traffic congestion occurs at base station site, MEC servers are able to downscale traffic in order to both reduce stress of MInPs' base stations and increase network speed.

Augmentation: Since additional information is available at the base station site, these data can be shared with ASPs in order to enhance quality of experience. To this end, mobile network operators enhance requests sent by the UEs by also including statistics on the number of connected UEs, bandwidth utilization, and so on. As an example, current and expected cell congestion are two factors enabling real-time adaption of ASP's service parameters like content resolution as well as communication and notification behavior.

MEC enables mobile network operators to also provide user-related information, since these data are available in the cellular network and get lost as soon as packets are processed by Internet routers. Thus, in order to provide enhanced services tailored to the needs of the UE user, mobile network operators can inject additional data (e.g., age, sex, postal address, cell movement patterns, etc.) into the original requests. Obviously, privacy aspects have to be taken into account when applying these feasibilities in the real world. In addition to this non-technical enhancement, MEC-based augmentation comes with reduced network delay due to the fact that ASPs are able to adapt service parameters in real-time, rather than reactively: MEC enables ASPs to tailor content in real-time to the needs of the UEs.

C. Advantages of Mobile Edge Computing

The following considerations can be concluded from the previous subsections: From a technical perspective, end users benefit mostly from reduced communication delay. Here, one interesting application class is offloading: Due to its close proximity to the end user, MEC servers enable new kind of applications to be considered as offloading candidates. From the MInPs point of view, the most interesting aspect of MEC is bandwidth reduction and scalability. Here, interesting applications are edge content delivery, aggregation, and local connectivity. ASPs profit with respect to scalability and faster services. MEC enables them to host services at the edge, which results in lower bandwidth demands within data centers. Furthermore, augmentation enables novel possibilities for ASPs, since cellular network specific information can be integrated into the traffic flow that are, due to technical limitations, not available in conventional networks.

V. Conclusion and Future Work

This paper discussed several applications for the deployment at the mobile edge and classified them based on six categories. These categories were evaluated based on the technical parameters power consumption, delay, bandwidth usage, and scalability. Benefits for stakeholders, namely mobile end user, network operator, and ASPs were analyzed. As discussed before, in most deployment scenarios, mobile end users and MInPs profit from reduced network delay, and, thus, faster services. Furthermore, from the ASPs' point of view, MEC enables the integration of additional, congestion- or user-related information into the traffic flow.

 ISBN: 978-1-61208-377-3

Several questions remain open for future work: Whilst being quite promising in this context, offloading has not been analyzed so far with respect to MEC. In contrast to offloading approaches that apply in cloud networks, several constraints have to be taken into account in the MEC scenario: Mobile applications have to be aware of the fact that MEC servers are deployed in a decentralized way and, since the mobile user might move from its current geographical position, connectivity between MEC servers and end user device is constrained. Thus, applications that rely on MEC services have to be mobility-aware and need to fallback gracefully to other MEC servers, distant cloud servers, or even the UE itself. Beyond, efficient and power-saving offloading approaches for VoIP systems have not been discussed in this context. We already initiated some measurements and experiments in that direction, which look quite promising. Offloading decision factors need to be evaluated, deciding when to offload data to MEC servers (e.g., depending on link quality, interferences, and congestion). With respect to Edge Content Delivery, proximity-aware caching algorithms are needed, deciding when and how MEC servers request remote data for storing at the edge, avoiding congestion and enhancing Quality of Experience.

ACKNOWLEDGMENT

We would like to thank Uwe Puetzschler (Nokia Networks) for kindly supporting our work.

REFERENCES

[1] Ericcson, "Ericcson Mobility Report – June 2013."
[2] Intel and Nokia Siemens Networks, "Increasing mobile operators' value proposition with edge computing."
[3] IBM Corporation, "Smarter wireless networks; add intelligence to the mobile network edge."
[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.
[5] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches: A survey of mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 13, pp. 1587–1611, Dec. 2013.
[6] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, "Impact of cloudlets on interactive mobile cloud applications," *16th International Enterprise Distributed Object Computing Conference*, pp. 123–132, Sept. 2012.
[7] "Ericsson Mobility Report – November 2013."
[8] P. Jones, C. Newcombe, R. Ellis, D. Birum, and M. Thompson, "Method and system for preloading resources," Feb. 22 2011. US Patent 7,895,261.
[9] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems, Third Edition*. Springer, 2011.

 ISBN: 978-1-61208-377-3

On Delay-Aware Embedding of Virtual Networks

Michael Till Beck, Claudia Linnhoff-Popien
Ludwig Maximilian University of Munich
{michael.beck,linnhoff}@ifi.lmu.de

***Abstract*—Network Virtualization is seen as a key technology for the Future Internet. In fact, today, Network Virtualization is actively used by telecommunication providers. One of the key challenges in this context is the embedding of virtual networks into the substrate network topology: Virtual Network Embedding algorithms aim to assign substrate nodes and substrate paths to virtual nodes and links in an optimal way. Several embedding algorithms have been proposed in literature, pursuing various optimization goals. Most of them strive to increase cost-efficiency of the embedding. This paper discusses communication delay in the context of the Virtual Network Embedding problem. While embedding cost has already been extensively discussed, queueing delay has only sparsely been analyzed in this context. This paper introduces a delay model that is based on queueing theory and considers demands of virtual network requests. Based on this model, optimization objectives for delay-aware embedding algorithms are presented. Furthermore, delay related evaluation metrics are introduced for analyzing the effectiveness of delay-aware virtual network embedding approaches.**

***Index Terms*—Virtual Network Embedding, Delay**

I. Introduction

Network Virtualization is a promising approach to overcome the ossification of the current Internet infrastructure. Core protocols of the Internet infrastructure are perceived as being difficult to change. This is widely known as the ”IP-waist”. Network Virtualization enables infrastructure providers to separate network capacities into several isolated networks, each capable of running its own communication protocols. First, Network Virtualization has been actively used in the context of Internet testbeds like G-Lab [1] and 4WARD [2]. Today, Network Virtualization is seen as one of the most promising technologies to overcome the resistance of the Internet infrastructure towards novel core protocols. In fact, Network Virtualization is actively used by today's telecommunication providers as a tool to enhance the flexibility of their network infrastructures.

In Network Virtualiation, several virtual networks are deployed on top of a substrate network topology [3]. From an abstract point of view, substrate networks are a collection of network nodes (representing, e.g., physical servers), connected by network links (representing physical communication links, e.g., Ethernet cables). Similarly, virtual networks consist of virtual nodes and virtual links, both demanding network resources (like CPU and bandwidth capacities) provided by the substrate network.

This leads to the Virtual Network Embedding problem: The objective of an embedding algorithm is to embed virtual networks on top of a shared substrate network in an optimal (or near-optimal) way.

Several Virtual Network Embedding approaches have been discussed in literature so far. Many aim to reduce embedding cost, i.e., the amount of substrate network resources that are needed in order to embed the virtual networks. By keeping embedding cost low, the infrastructure provider is able to allocate additional virtual network requests. Besides embedding cost, another key objective in the context of telecommunication networks is to keep network delay low. Despite of the fact that several embedding approaches have been presented in literature so far, delay is only sparsely discussed in this context.

Therefore, this paper presents a delay model for future virtual network embedding approaches that considers the dynamic components of communication delay. The model is based on queueing theory and takes into account both transmission delay and queuing delay. Furthermore, delay-aware optimization objectives are discussed in this context. As discussed in this paper, embedding virtual networks in a delay-sensitive way comes with additional embedding cost. On the one hand, infrastructure providers usually aim to assign network resources in a cost-efficient way in order to increase the amount of resources that are available for future virtual network requests. On the other hand, a delay aware embedding tends to consume additional network resources. Thus, there is a tradeoff between delay-awareness and cost-efficiency. This paper motivates why both aspects should be taken into account when embedding virtual networks. Finally, evaluation metrics are discussed for measuring the effectiveness of embedding results.

II. Background and Related Work

This section shortly motivates the virtual network embedding problem and presents the network model used in this work. Furthermore, related work is discussed.

A. The Virtual Network Embedding Problem

Figure 1 depicts the virtual network embedding problem. Several virtual network requests (VNR) have to be assigned to a shared substrate network. Substrate resources are limited: E.g., substrate nodes provide CPU resources and substrate links offer bandwidth resources that can be assigned to VNRs. Virtual networks demand those resources. Virtual nodes demand CPU resources and virtual links demand bandwidth

 ISBN: 978-1-61208-377-3

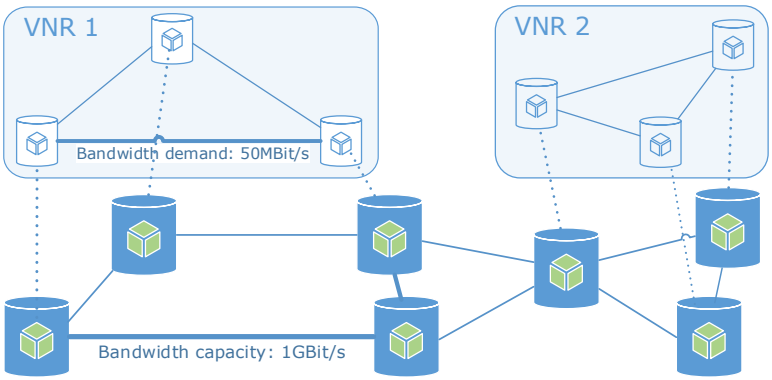


Fig. 1. The Virtual Network Embedding Problem

resources. For the embedding, virtual nodes and links have to be assigned to substrate resources offering sufficient resources. While a virtual node is assigned to just one single substrate node, a virtual link can be embedded to multiple substrate links, i.e., to a substrate path. The embedding algorithm has to ensure that each segment of that substrate path provides sufficient bandwidth resources.

The problem of optimally embedding virtual networks to the substrate network is NP-hard. Therefore, to reduce computational complexity, several heuristical embedding approaches have been introduced, aiming to solve the embedding in a nearly-optimal way [3].

B. Network Model

This subsection shortly describes the network model used in this work. It is based on the one presented in [3].

Both the substrate network and virtual networks are modeled as network graphs: A substrate network SN is represented by a set of substrate nodes N connected by substrate links L. A substrate node $n \in N$ provides CPU resources $\mathrm{res}_{\mathrm{cpu}}(n)$; a substrate link $l \in L$ offers bandwidth resources $\mathrm{res}_{\mathrm{bw}}(l)$. Similarly, the i-th virtual network request VNR^i is a set of virtual nodes N^i and links L^i. CPU demand of a virtual node $n^i \in N^i$ is modeled as $\mathrm{dem}_{\mathrm{cpu}}(n^i)$, and bandwidth demand of a virtual link $l^i \in L^i$ is modeled as $\mathrm{dem}_{\mathrm{bw}}(l^i)$. The virtual network embedding problem is composed of the node mapping problem and the link mapping problem. The node mapping step is described as a function $m_{\mathrm{node}} : N^i \rightarrow N$, and the link mapping step as a function $m_{\mathrm{link}} : L^i \rightarrow L' \subseteq L$. For a more readable representation, we will refer to $R_{\mathrm{total}}(l)$ as being the total bandwidth of a substrate link l; $R_{\mathrm{occupied}}(l)$ represents the amount of bandwidth resources that are allocated to virtual links assigned to a substrate link l, and $R_{\mathrm{available}}(l)$ refers to available bandwidth resources of that link that were not allocated so far.

C. Related Work

Many embedding approaches have been discussed in literature so far [4]. Most of them aim to reduce embedding cost in order to increase the number of networks that can be embedded onto the substrate topology. Besides cost-efficient algorithms, several other embedding approaches aiming for other optimization objectives have been proposed, focussing on energy efficiency [5], workload distribution [4], [6], resilience, etc. An extensive survey on virtual network embedding parameters and optimization objectives is given in [3].

However, only few related work on delay-aware virtual network embedding is available so far. To the best of our knowledge, there are only two publications in that direction:

Karthikeswar et al. introduce an embedding algorithm that considers the tradeoff between end-to-end delay and substrate utilization [7]. The problem is formulated as a mixed integer programming formulation and is based on a static delay model. It is assumed that communication delay is a static property of the communication links and does not depend on the utilization of the link.

Liao et al present a multi-agent approach to solve the virtual network embedding problem by considering link delay [8]. This approach aims to minimize bandwidth cost while keeping delay constraints of communication paths within defined limits. The approach considers a linear delay model.

In contrast to related work, this paper introduces, based on queueing theory, a non-linear delay model. Based on this model, delay-aware optimization objectives are formulated and several evaluation metrics are discussed.

III. Delay-Aware Virtual Network Embedding

In this section, delay-awareness is discussed in the context of the virtual network embedding problem. To this end, a delay model for substrate links is presented. Delay is modeled as a function that heavily depends on the utilization of network links. Based on this model, optimization objectives are introduced that aim to reduce network delay by avoiding highly utilized communication paths.

First, the delay model is introduced. Then, optimization objectives for future virtual network embedding algorithms are formulated. Finally, evaluation metrics are presented for measuring the effectiveness of delay-aware embedding approaches.

 ISBN: 978-1-61208-377-3

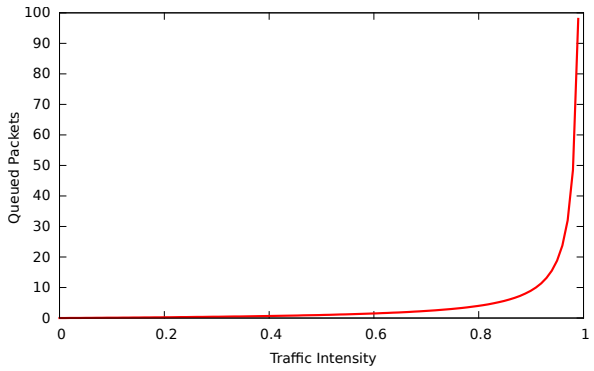


Fig. 2. The average number of queued packets rises if traffic intensity increases

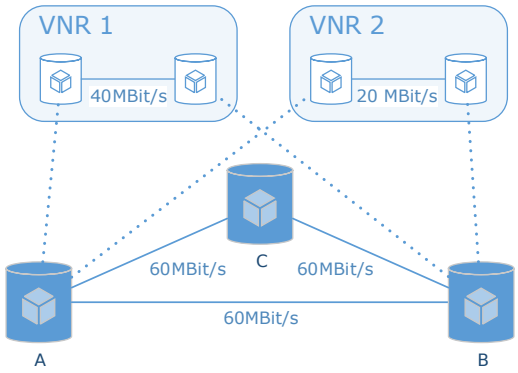


Fig. 3. Delay-Aware Virtual Network Embedding

A. Utilization-Aware Delay Model

In the following, various types of communication delay are being discussed; then, a model is derived that is applicable in the context of the virtual network embedding problem.

In telecommunication networks, there are four types of delay influencing network speed [9]:

- Processing Delay d_{proc}
  Refers to the time needed for processing a packet, e.g., extracting header information, examining how a packet has to be routed and checking whether bit-level errors occurred during transmission. Processing delay is usually in the bounds of nanoseconds and does not depend on the utilization of the routers (unlike queueing delay).
- Transmission Delay d_{trans}
  Time needed in order to transmit a packet of length L from router A to router B. Depends on the transmission rate R of the link. Transmission delay refers to the time that is needed to transmit all bits of a packet from A to B, thus, transmission delay is L/R.
- Propagation Delay d_{prop}
  Time needed to propagate a bit through the communication link. Propagation speed depends on the physical medium and is, in general, nearly equal to the speed of light.
- Queuing Delay d_{queue}
  The time a packet remains in the router's queue before it can be processed. Queuing delay of a packet depends on the number of other packets that arrived before. Queuing delay can vary significantly: If the queue is empty, the router will handle the arriving packet immediately, i.e., queuing delay is zero; if, however, many packets are waiting for transmission, queuing delay contributes heavily to the total end-to-end delay. While previous work focuses on static delay models, this paper also considers the utilization of routers in the embedding process. To this end, transmission delay is considered for the as a non-linear function.

In the following, a delay model that is applicable in the virtual network embedding domain is discussed. As mentioned before, transmission delay depends on bandwidth resources of the substrate network. Furthermore, queuing delay depends heavily on the utilization of the network link. The delay model presented here is based on a well-known utilization-aware queuing delay model: The model considers that queuing delay increases significantly when traffic intensity i of a communication link l increases. As shown in Figure 2, the number of packets that have to be queued increases non-linearly.

Figure 3 motivates delay-awareness for the virtual network embedding problem: In this scenario, two virtual networks are being embedded. It is assumed that, in this scenario, only substrate nodes A and B offer sufficient resources for hosting the virtual nodes. If the virtual link of VNR 1 is assigned to the substrate link between A and B, utilization increases and, as such, also does queuing delay. Still, this link offers sufficient bandwidth resources to the virtual link of VNR2. In fact, current virtual network embedding approaches that do not consider delay as part of their optimization strategy, tend to embed also the second virtual link to this substrate link. However, as utilization is, in this case, 100%, this would significantly increase delay if traffic intensity in virtual networks increases.

Traffic intensity is usually defined as follows:

$$i = \frac{La}{R}$$

with packet length L, arrival rate a and bandwidth R. Consistent with several other work presented in this area of research, it is assumed that a fixed amount of bandwidth resources is assigned to virtual links. Thus, traffic intensity is calculated as

$$i(l) = \frac{R_{\text{occupied}}(l)}{R_{\text{total}}(l)}$$

with R_{total} denoting bandwidth resources of a link l and R_{occupied} bandwidth resources that are allocated to virtual links.

The average number of packets waiting for transmission is calculated based on queuing theory; a substrate link is modeled

 ISBN: 978-1-61208-377-3

as a M/M/1 queuing system; then, the average number of packets waiting in the link's queue is calculated as follows (graph shown in Figure 2) [10]:

$$p(l) = \frac{i(l)}{1 - i(l)}$$

Considering that queue size qs in real systems is limited, queuing delay is then defined as

$$d_{\text{queue}}(l) = \frac{\min\big(p(l), qs\big) \cdot L}{R_{\text{total}}(l)}$$

Thus, total delay is defined as

$$d_{\text{total}}(l) = d_{\text{proc}} + d_{\text{prop}} + \frac{L}{R_{\text{total}}(l)} + d_{\text{queue}}(l)$$

Taking utilization into account, this model describes the delay-behavior of real-world substrate networks more accurately than linear models.

B. Optimization Objectives

Based on the model presented in the previous section, this section discusses optimization objectives for future delay-aware virtual network embedding approaches. Several optimization objectives are applicable in this context:

Minimizing delay is the most obvious objective. This can be done in various ways, depending on the optimization objective of the infrastructure provider: First, one option is to aim for minimum average delay. More precisely, the embedding algorithm aims to minimize average delay of all substrate paths assigned to virtual links. In this case, the infrastructure provider guarantees that on average, delay is below a certain limit. However, in worst case, some communication links do have worse delay properties. Thus, another option is to minimize maximum delay (instead of the average delay). Now, the infrastructure provider is able to guarantee that none of the embedded virtual links suffer from worse communication delay. In general, traffic intensity should not exceed 80-90%, as depicted in Fig. 2.

As a more straight-forward alternative, instead of *optimizing* the embedding towards delay-effectiveness, the embedding can be performed by just *considering* delay constraints: I.e., instead of minimizing delay, the embedding algorithm just assures that communication delay never exceeds a pre-defined limit. For example, virtual links are never assigned to substrate paths if traffic intensity on these links would get too high. This simplified concept can be extended with respect to delay constraints of individual virtual links: In this case, virtual network operators are able to specify delay constraints for individual links (instead of for the whole virtual network). The infrastructure provider assigns these networks in a way that none of these constraints are violated. The traditional embedding problem can be easily extended with respect to both alternatives; in both cases, embedding algorithms just have to validate that none of the constraints has been violated before assigning substrate nodes and links. This works equivalently

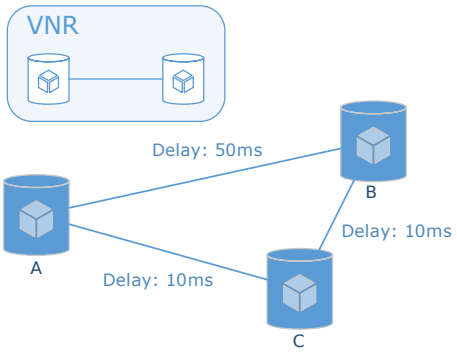


Fig. 4. Delay-Awareness vs. Cost-Efficiency

to the validation of whether substrate resources fulfill CPU and bandwidth demands of the virtual nodes and links. Here, the actual optimization objective of the embedding approaches does not have to be altered.

As mentioned before, the optimization objective of most embedding algorithms is cost-efficiency. Many embedding approaches aim to keep embedding cost low, in order to increase the amount of available network resources, leaving space for future virtual network requests. A delay-aware embedding, however, often comes with higher embedding cost. The tradeoff between delay-awareness and cost-efficiency is depicted in Figure 3. A cost-optimal embedding algorithm assigns the virtual link to the substrate link directly connecting A and B. In this case, the virtual network operator suffers from high delay. A delay-optimal algorithm, however, choses the indirect path A↔B↔C. In this case, the infrastructure provider suffers from high embedding cost. A suitable solution to this dilemma would be to combine delay-awareness and embedding cost by balancing both objectives accordingly.

C. Evaluation Metrics

Several evaluation metrics have been discussed in the context of the virtual network embedding problem so far. A survey on embedding approaches has been presented by Fischer et al. [3], also including an extensive discussion on evaluation metrics. Furthermore, an open source simulation framework implementing many of these metrics has been published [11]. This section presents several new metrics with reference to delay-awareness. Of course, novel embedding algorithms should always be thoroughly analyzed also with regard to other metrics that are not directly related to communication delay, most notably one, the embedding cost metric. Since cost is one of the most notable and well-known metrics in the context of virtual network embedding, it is shortly discussed here.

Embedding Cost: Embedding cost is one of the most common evaluation metrics in the context of virtual network embedding. Cost is defined as follows:

$$\text{Cost}(\text{VNR}^i) = \sum_{n^i \in N^i} \text{res}_{\text{cpu}}(n^i) + \sum_{l^i \in L^i} \sum_{l \in L} \text{res}_{\text{bw}}(l^i, l)$$

 ISBN: 978-1-61208-377-3

Performance of embedding algorithms is usually evaluated through extensive simulations in randomly generated network topologies. Embedding cost significantly depends on the kind of virtual networks that are being embedded: Randomly generated network requests demanding few network resources tend to be much easier to embed than those demanding many. Therefore, the revenue metric is used in order to quantify virtual network requests. Similar to cost, revenue is computed as follows for a virtual network request VNR^i:

$$\mathrm{Revenue}(\mathrm{VNR}^i) = \sum_{n^i \in N^i} \mathrm{dem}_{\mathrm{cpu}}(n^i) + \sum_{l^i \in l^i} \mathrm{dem}_{\mathrm{bw}}(l^i)$$

Thus, to put cost in relation to network requests, the revenue/-cost metric is introduced:

$$\text{Revenue-Cost}(\mathrm{VNR}^i) = \frac{\mathrm{Revenue}(\mathrm{VNR}^i)}{\mathrm{Cost}(\mathrm{VNR}^i)}$$

Path Length: The path length metric reflects how many substrate links were (on average/maximum) assigned to the virtual links. Despite of the fact that communication delay of a substrate path is the sum of the delay on each substrate link segment of that path, long path lengths do not necessarily reflect large delays. As an example, in the scenario depicted in Figure 4, a delay-aware embedding algorithm would chose a longer, but less delay-intense path.

Link Delay: Link delay is defined as the average/maximum delay of all substrate links. This metric is of interest to the infrastructure provider, as it reflect the average/maximum utilization of the substrate network. High link delay indicates that either the infrastructure is not optimally used (e.g., as a result of several non-optimal embeddings of virtual networks) or additional hardware components should be integrated into the substrate network in order to improve network performance and to keep up with increasing virtual network demands. Link delay is therefore also a good indicator for estimating whether sufficient substrate resources are available in order to embed further virtual network requests.

Traffic Intensity: Traffic intensity reflects how many packets (on average/maximum) are being queued by the substrate nodes. Similarly to the link delay metric, traffic intensity indicates which parts of the substrate network suffer from high load and, thus, need to be reconfigured.

Path Delay: Path delay is defined as the sum of the delay of all substrate links that are part of a communication path. I.e., average/maximum path delay is the average/maximum delay of all paths that were assigned to virtual links. This metric is a key indicator to the virtual network operator, as it reflects how well the virtual network has been embedded into the substrate network and how the network performs with respect to communication delay.

Concluding, embedding cost, path length, link delay, and traffic intensity are metrics that are related to the performance of the substrate network. Path delay is a key metric indicating how well a virtual network performs after it has been embedded into the substrate infrastructure.

IV. Conclusion and Future Work

As discussed in this paper, link utilization significantly influences queueing delay of routers; Delay-aware embedding approaches should consider that delay increases non-linearly; the embedding has to be performed in a way such that communication delay is kept within reasonable bounds. In general, infrastructure providers should avoid embedding virtual links to paths such that traffic intensity exceeds 80%. For delay-sensitive applications, new virtual network embedding approaches are needed with delay-aware optimization objectives. As discussed in this paper, there is a tradeoff between delay-awareness and cost-efficiency. Therefore, these algorithms should also consider embedding cost as part of their optimization strategy.

We are currently in the process of implementing a delay-aware embedding algorithm based on the model presented here. Furthermore, evaluation metrics discussed in this paper are in the process of being integrated into the Alevin simulation framework [11], [12] and will, as such, be published as open source.

References

[1] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, "German-lab experimental facility," *Future Internet Symposium (FIS) 2010*, 9 2010.

[2] J. Carapinha and J. Jiménez, "Network virtualization: a view from the bottom," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, VISA '09, (New York, NY, USA), pp. 73–80, ACM, 2009.

[3] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 1888–1906, 2013.

[4] M. T. Beck, J. F. Botero, A. Fischer, H. De Meer, and X. Hesselbach, "A distributed, parallel, and generic virtual network embedding framework," in *IEEE Int'l Conf. on Communications (ICC 2013)*, IEEE, 2013.

[5] A. Fischer, M. T. Beck, and H. De Meer, "An approach to energy-efficient virtual network embeddings," in *Proc. of the 5th Int'l Workshop on Management of the Future Internet (ManFI 2013)*, IFIP, IEEE, 2013.

[6] M. T. Beck, A. Fischer, and H. De Meer, "Distributed virtual network embedding," in *Proc. of the 7th GI/ITG KuVS Workshop on Future Internet*, University of Kaiserslautern, 2012.

[7] K. Ivaturi and T. Wolf, "Mapping of delay-sensitive virtual networks," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pp. 341–347, IEEE, 2014.

[8] L. Shengquan, W. Chunming, Z. Min, and J. Ming, "An efficient virtual network embedding algorithm with delay constraints," in *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*, pp. 1–6, June 2013.

[9] J. Kurose and K. Ross, "Computer networks: A top down approach featuring the internet," *Peorsoim Addison Wesley*, 2006.

[10] T. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Telecommunication networks and computer systems, Springer New York, 2000.

[11] M. T. Beck, A. Fischer, F. Kokot, C. Linnhoff-Popien, and H. De Meer, "A simulation framework for virtual network embedding algorithms," in *Proc. of the 16th International Telecommunications Network Strategy and Planning Symposium*, Networks, 2014.

[12] VNREAL, "ALEVIN2 – ALgorithms for Embedding VIrtual Networks." `http://alevin.sf.net`, May 2014.

 ISBN: 978-1-61208-377-3

Cross-Layer Solutions for Enhancing Multimedia Communication QoS over Vehicular Ad hoc Networks

Ali Hassoune Mustafa, Mekkakia Zoulikha and Bendella Fatima
Department of Computer Engineering,
University of Sciences and Technology of Oran
Oran, Algeria
{mustafa.alihassoune; zoulikha.mekkakia; fatima.bendella}@univ-usto.dz

***Abstract*—Vehicular Ad hoc NETworks (VANET) are a new emergent technology based on wireless ad hoc networks. They are characterized by their high speed nodes, which affect on network topology. This can affect network services, especially those having big packet size and need high bandwidth, such as Multimedia services. Many solutions have been proposed in the literature in order to serve a better multimedia communication over VANET. In this paper, we will focus on cross-layer solutions because of their high performance in term of Quality of Service (QoS). We present a survey of existing cross-layer solutions, which try to enhance multimedia services over VANETs. Works will be classified depending on the nature of information exchanged and their belonging to OSI Layers.**

Keywords- Cross-Layer; QoS; Multimedia; VANET.

I. INTRODUCTION

Vehicular Ad hoc NETworks (VANET) are the main interesting instantiation of mobile Ad hoc networks, which could have an important role in future services. They are characterized by their high mobility due to high speed nodes, which affect on network topology. They are composed of vehicles equipped with On-Board Units (OBUs), and Gateways installed in the streets, which are called Road Side Units (RSUs). VANETs can be deployed in different kind of roads, such as "Highways" where nodes move with high speed (80-120 km/h) and have generally low density. Also, VANETs can be deployed in "Urban roads" where nodes move with less speed (20-60 km/h), which affects network density. The urban roads are also characterized by the existence of buildings, which act as obstacles for VANET communications.

VANETs use a special MAC protocol called 802.11p, which is an enhancement of 802.11, using two types of channels: Control Channel (CCH) and Service Channels (SCH). The CCH is used for the periodical dissemination of control information and for the dissemination of traffic safety messages. The SCH is used to disseminate non-critical information for infotainment applications, such as the Video Streaming. In addition, VANET applications can be classified into Safety and Non-Safety applications (like downloading files or accessing the internet). Safety applications can help to prevent accidents and road congestions, while the non Safety applications may be used for user's convenience like video streaming or Video on Demand (VoD) applications, such as TV Broadcasting.

Sending these kinds of data in such networks is very challenging because of large data size, link breaks, and sensibility to losses. These features are rare in such networks where nodes move with high speed and where topology changes rapidly. Therefore, robust multimedia applications encounter many challenges that oblige applications to be able to tolerate link failures and packet losses in order to have a high quality video at the receiver side. Hence, many solutions have been proposed in the literature in order to enhance Multimedia QoS to serve better video services in VANET. We shall focus in this paper on cross-layer solutions where information is exchanged between different layers in order to have a better multimedia service. These techniques will be classified depending on data exchanging nature.

This paper is organized into three sections. In Section I, we survey existing cross-layer solutions and techniques, which enhance multimedia Qos. Section II is reserved for discussing the solutions and presenting open issues. In Section III, we conclude our work and present future works.

II. CROSS-LAYER SOLUTIONS FOR ENHANCING MULTIMEDIA QOS OVER VANET

Cross-layer approach is an 'escape' from the Open Systems Interconnection (OSI) model, which applies virtually strict boundaries between the layers, data are kept within a given layer. Protocol architectures follow strict layering principles, which ensure interoperability, fast deployment, and efficient implementations. However, lack of coordination between layers limits the performance of such architectures due to the specific challenges posed by wireless nature of the transmission links. Cross-layer solutions remove such strict boundaries to allow communication between layers. Its core idea is to maintain the functionalities associated to the original layers but to allow coordination, interaction and joint optimization of protocols crossing different layers.

In this paper, it is notable that all studied mechanisms are cross-layer solutions, which serve a high quality video on the receiver's end. We can distinguish between the aforementioned techniques based on exchange nature, as shown in Table 1.

TABLE 1. STUDIED CROSS-LAYER TECHNIQUES

Works	APP	NET	MAC	PHY
[1;2;3;4;6;7;8]	√			√
[9;10;11;12]	√		√	
[13;14;15;16;17;18;19; 20;21]	√	√		
[22;23;24;25;26;27;28]			√	√
[29;30;31]		√	√	
[32;33;34]		√	√	√

 ISBN: 978-1-61208-377-3

We classified these works into six categories shown above. The next paragraphs, we present the different works found in the literature.

A. Physical-Application exchange solutions

Physical-Application (PHY-APP) exchange solutions adapt physical layer parameters like vertical handoff periods in function of application layer information. They are presented in this section.

The vertical handoff was proposed by Yan et al. [1]. They presented an algorithm based on the prediction of the traveling distance of a vehicle within a wireless cell. These try to minimize the probability of unnecessary handovers. This probability was constructed by using a speed ratio, which is the ratio between instantaneous speed of a vehicle and maximum speed, function of the technologies radius cell coverage, and of the average handover latency.

Another algorithm, based on physical layer parameters as handover latency and the received signal strength, presented by Kwak, et al. [2], had an objective of reducing the loss of throughput. Their valid ideas are limited to horizontal handovers in Wireless Local Area Networks (WLAN) though with homogeneous topologies.

Chen et al. [3] presented an approach in which a novel network mobility protocol for VANETs was presented. It was a solution that limits vertical handover drawbacks and reduces both packet loss rate and handoff latency.

Another critical issue in any system is the handoff decision policy. Zhu et al. [4] proposed a work for train-ground communication, which can be applied in VANETs. They suggested that video distortion is the most direct QoS performance metric from the perspective of end users, which can be estimated by packet loss rate and encoding parameters. It is formulated as a Semi-Markov Decision Process (SMDP), which is a generalization of a Markov Decision Process (MDP) [5]. The optimal handoff decision and application layer parameters adaption policies can be obtained from the value iteration algorithm of SMDP.

In addition, a new strategy was presented which analyses users' interactive viewing behavior by estimating video segment playback order. That employed pre-fetching of the expected segments, by smoothening the video playback. A cellular network had relatively high stable connectivity merits, but it was more expensive. So, by using VANET, vehicles can forward data to other nodes or RSUs at low costs. However, VANETs easily become disconnected in situations with low vehicle density and high mobility, which needs to switch to another technology. Four novel mechanisms were introduced: distributed grouping-based video segments storage scheme, video segment seeking scheme, multipath data delivery mechanism, and Speculation-based pre-fetching strategy.

Alternatively, Changqiao et al. [6] proposed a Quality of Experience (QoE)-driven solution for VoD services in urban vehicular network environments. Vehicles create a low VANET layer with Wireless Access In Vehicular Environments (WAVE) interfaces and create an upper layer Peer-To-Peer (P2P) Chord overlay on top of a cellular network via 4G interfaces. A novel storage strategy was proposed that distributes the video segments along the Chord overlay, reducing segment seeking traffic and achieving a high success rate and very good video data delivery efficiency.

Sadiq et al. [7] proposed an Intelligent Network Selection (INS) scheme, which ranks available wireless network candidates using three input parameters: Faded Signal-to-Noise Ratio, Residual Channel Capacity, and Connection Life Time. In this proposed scheme, when a vehicle is in busy-mode, its wireless channel can execute various real time and non-real time applications. In order to identify and select the most qualified network candidate as the next wireless access point, each vehicle executes the INS algorithm in the network access area. This avoids any unnecessary handover decisions during real-time sessions. Results showed that INS is more efficient at decreasing handover delays, end-to-end delays for VoIP and video applications, and packet loss ratios.

An adaptive QoS handoff priority scheme was proposed by Zhuang et al. [8] for wireless networks, which reduces the probability of call handoff failures in a mobile multimedia network with cellular architecture. This approach used the ability of most multimedia traffic types to adapt some QoS at the packet level to achieve a smaller probability of dropping at handoff, which has an impact on the multimedia received QoS. So, calls with adaptive traffic can opt to use lower amounts of bandwidth and handoff successfully. Also, it proposed the Adaptive Quality of Service (AQoS) scheme proved more flexibly and efficiently in guaranteeing QoS and proposing a modification, called the Modified Adaptive QoS (MAQoS), which can admit new calls even when the system is in the congestion state, e.g., emergency calls. In addition to the flexibility inherited from the AQoS scheme, MAQoS is even more flexible in decoupling the different components (dropping, and blocking probabilities) of the grade of service metric. The adaptive QoS handoff priority scheme and its modification are studied analytically and compared to those of the non priority handoff and the guard-channel handoff schemes, and have shown better results.

B. MAC-Application Exchange solutions

MAC-Application (MAC-APP) exchange solutions adjust MAC parameters like retransmissions or frame rate in terms of Multimedia QoS like latency or loss rate received from the application layer.

First, an adaptive MAC retransmission limits adaptation scheme proposed by Asefi et al. [9], in which the adaptation was based on an optimization of video streaming quality. It adjusts MAC retransmission limit using channel information and periodically feeds to RSU. It is used to calculate the probability of media access between the RSU and the vehicle. In the Enhanced Distributed Channel Access (EDCA) of 802.11p, Video packets are associated with lower priority compared to

 ISBN: 978-1-61208-377-3

safety messages. To solve this problem, this scheme applies a multi-objective optimization framework at the RSU, which tunes the MAC retransmission limit with respect to channel statistics (packet error rate and packet transmission rate) in order to minimize the probability of playback freezes and start-up delay of the streaming.

To address the problem of adaptive QoS, Mercado and Liu [10] proposed a solution, which choosing the Signal-to-Interference and Noise Ratio (SINR) of each user's channel as the QoS index depending on the nature of multimedia. They tried to solve two problems. The first by increasing the SINR levels for multimedia users. Different users have distinct desired SINR levels according to their requested service types; their algorithm uses an iterative method to drive the SINR levels as close as possible to those desired levels. The SINR levels are improved without deteriorating quality for other types of users. It showed a significant increase in the average SINR levels for multimedia users. The second problem is how to initiate new users into the network by using lower complexity algorithms. The proposed algorithm also included a fast activation scheme that reduces the computational complexity involved in some parts of call initiation, by using a coarser and faster method for finding feasible SINR.

Another solution, presented by Venkataraman et al. [11], is a hybrid IEEE 802.11p-based multihop network communication solution (QOAS). This delivered quality-oriented real-time multimedia data to high-speed vehicles by using both infrastructure and ad hoc modes. A client-server feedback was used in order to support high multimedia quality. QOAS estimates how a user perceives quality and sends feedback to the server, which adjusts video transmission rate. They are based on the fact that random losses have a greater impact on the perceived quality than a controlled reduction. It was specially used to send video stream with high quality to moving vehicles at high speed, and where there was a quick handover between different relay nodes and the sender.

The last MAC-Application solution studied was the work proposed by Bonuccelli et al. [12], which focused on situations of traffic congestion. The transmitter reduces the transmission rate by 50%, when two consecutive frames are not received. The application layer reduces the rate when congestion is detected in MAC level. The transmitter decides whether a packet will reach the receiver in time. If not, the transmitter either drops it or sends it, which may be useful for decoding the next packet.

C. Network-Application exchange solutions

In this section, we present Network-Application (NET-APP) exchange solutions, which adjust parameters of routing or clustering in function of feedback received from the application layer.

A novel user-oriented cluster-based solution for multimedia delivery over VANETs proposed by Irina et al. [13], which is able to personalize multimedia content and its delivery based on the preferences of the passengers and their profiles. It includes two algorithms. The first one focuses on cluster head selection, which makes sure that cluster head function is efficiently distributed among vehicles. The second one is the cluster formation algorithm, which aims to group vehicles based on vehicle characteristics and user interest in content.

Both algorithms take into account the velocity of the mobiles, direction of travel and position of the vehicles. The proposed solution used a client-server architecture based on a hybrid vehicular communication network model. The vehicles are organized in clusters based on the user interest in multimedia content, location, direction of travel and velocity.

Another new application-centric solution is proposed as a routing protocol for streaming video in multihop for VANET. It is based on exchanging information between the application layer and the network layer aiming to select the path. This minimizes the application layer's frame distortion rate, which is the average distortion of the video frames at the destination vehicle. It was proposed by Asefi et al. [14]. A subset of candidates is selected by the node carrying data and is transmitted with video frames of high quality applying application centric optimization.

Another solution, proposed by Asefi et al. [15], focused on routing, which had shown the enhancement of video quality. This was achieved by minimizing distortion, the startup delay and streaming freezes. A virtual link is created between the destination vehicle and the access router and not RSU. They proposed a new protocol, Quality-Driven Routing Protocol. The proposed data delivery model had two modes of operation. The first one is the "straight way" where the vehicle carrying the packet to be forwarded selects N neighboring vehicles that are in its transmission range and are geographically closer to the destination vehicle. It then selects the next hop that minimizes the frame distortion. The second mode is "Intersection" where the vehicle selects the next straight path to forward the packet.

Sajimon and Sojan [16] applied a spatio-temporal similarity measure using Points of Interest (POI) and Time of Interest (TOI). The similarity formed will be used by the remote database to broadcast trigger-based messages to participating vehicles in a neighborhood for a future route. A large quantity of collected trajectories were published and shared across users on many websites. Additionally, it demonstrated a binary encoding scheme in managing road network data, and proposed a structural and sequence similarity measure between travel locations in finding a spatial similarity.

An application layer forwarding algorithm incorporating VANET routing, called Intelligent Adjustment Forwarding (IAF), proposed by Jung-Shian et al. [17], in which a segment-to-segment transmission paradigm was used to enhance the video data delivery. IAF started by performing an intelligent routing discovery process to establish a transmission path to the destination, where the source obtains the position information of all the nodes along the transmission path and then determines which of these nodes should be nominated as Intermittently Connected Points (ICPs). Then, data is

 ISBN: 978-1-61208-377-3

transmitted to the destination by the ICPs using a store-and-forward paradigm. If the source node is unable to locate the destination, the data is transmitted to the segment endpoint, which performs a store-carry-and-forward function in order to deliver the data to the destination.

Also, Asefi et al. [18] proposed a cross-layer protocol whose routing decision was based on application layer objective function. It discussed the encoded transmissions, decoding reception and error prone channels. The appropriate route is chosen to achieve an optimized Peak Signal-to Noise Ratio (PSNR) in different densities networks. In that proposed Cross-Layer Path Selection Scheme, the video streaming was sent from RSU to a vehicle using multi-hop communication. An encoding rate was allocated to each video session at the RSU side. This scheme selects the path with lowest end-to-end distortion for each video packet and for the entire video stream where the total distortion is the summation of all the packet distortions of a video stream.

IBCAV, abbreviation of Intelligent Based Clustering Algorithm in VANETs, was proposed by Mottahedi et al. [19]. Their proposal seeked to improve routing algorithms in VANETs by employing inter-layered methods, where cluster size, speed and density of nodes are the factors, which have been taken into account. Results show that the IBCAV performs better than other routing protocols in terms of packet delivery ratio, end-to-end delays and throughput.

A cooperative overtaking assistance systems based on VANETs is another network-application solution proposed by Vinel et al. [20]. A video stream, captured by a camera installed at the vehicle, is compressed and broadcast to any vehicles driving behind it. They demonstrate that the performance of their scheme can be significantly improved if codec channel adaptation is undertaken by exploiting information from the beacons about any forthcoming increase in the load of the multiple access channel used. Their proposal results give the guarantee of low latency and acceptable visual quality by making use of the additional information obtained from the beaconing.

Finally, DVAC was proposed by Yung-Cheng and Nen-Fu [21]. Distributed Vehicles Adaptive Clustering is an application layer solution for delivering live video streams by forming clusters. This algorithm was designed to form and maintain clusters. They create clusters with vehicles moving in the same direction. The vehicles decide whether they are cluster head, tail or member. It also suggested Vehicles-Adaptive PEer-to-Peer relay (VAPER) method, which was responsible for avoiding duplicates in streaming in the cluster. Both the head and tail act as redirectors in communications between vehicles in the same cluster, and they act as peers in communication between clusters. It showed that even if the signal is lost, there is a continuous play of live video streaming for a considerable time.

D. Physical-MAC exchange solutions

The cross-layer solutions in this section are mainly based on the information flow between MAC layer and PHY layer (PHY-MAC). For example, transmission rate adaptation is the ability to adjust the modulation rate at which packets are transmitted according to the observed channel qualities, such as SNR and packet loss rate.

Camp and Knightly [22] investigated cross-layer designs for modulation rate adaptation in vehicular networks targeted at urban and downtown environments. Their work involved high-level interaction between the MAC and physical layers. They studied two protocols for rate adaptation, which are Loss-triggered and SNR-triggered. The transmitters determine the packet loss rate by monitoring the frame receptions of the packet transmission in MAC layer. If an Acknowledgement (ACK) is received before the timeout event then the transmission is considered to be successful. This type of information is shared between MAC and PHY layers. The transmitter increases the transmission rate after consecutive successful transmissions and decreases the rate after observing consecutive failures.

SoftRate is a bit rate adaptation protocol proposed by Vutukuru et al. [23]. The receivers used physical layer calculated information that was exported to higher layers via an interface called the SoftPHY interface. SoftPHY estimates the channel Bit-Error Rate (BER) upon receiving packet frame.

Also, Chen [24] developed a novel IEEE 802.21 Media Independent Handover (MIH) mechanism for next generation vehicular multimedia network, and they proposed also an adaptive QoS management mechanism. The proposed MIH framework can determine the best available network by obtaining received signal strength parameters. Results showed that using this mechanism can increase overall throughput, which is satisfactory compensation for increased handover time.

Another cross-layer solution, proposed by Rawat et al. [25], is a joint adaptation between MAC and physical layer that mainly focuses on adaptation of transmission power and QoS message prioritization based on node density and contention window size. Network calculates the node density by gathering the neighbors' information within. By adopting a traffic flow model as proposed by Artimy et al. [26], using length of road segment, estimated vehicle density, and traffic flow constant parameters, it calculates the new transmission range. The transmission power is a function of transmission range. To support QoS applications, authors proposed two distinct functionalities to adjust the priority of the packets – transmission power level in physical layer and MAC channel access parameters, such as minimum contention window (CWmin), maximum contention window (CWmax), and arbitration interframe space (AIFs).

Similarly, Caizzone et al. [27] proposed a mechanism that adjusts the transmission power adaptively based on number of neighbors. Each vehicle starts with initial transmission power and incrementally increases the transmission power as long as the number of neighbors is within a minimum threshold, or it reaches maximum transmission power. The transmission power is decreased if the number of neighbors is greater than maximum

 ISBN: 978-1-61208-377-3

threshold.

Finally, Rawat, et al. [28] presented a new scheme for dynamic adaptation of transmission power and Contention Window (CW) size to enhance performance of information dissemination in VANETs. That was achieved by incorporating the EDCA mechanism of 802.11e and using a joint approach to adapt transmission power at the physical layer and QoS parameters at the MAC layer. The transmission power adapted was based on the estimated local vehicle density to change the transmission range dynamically, while the CW size was adapted according to the instantaneous collision rate to enable service differentiation. In the interest of promoting timely propagation of information, VANET advisories were prioritized according to their urgency and the EDCA mechanism is employed for their dissemination. Results show that this scheme brings better throughput and lower average end-to-end delay compared with other similar schemes.

E. Network-MAC exchange solutions

The cross-layer solutions in this section are mainly based on the information flow between MAC layer and Network layer (NET-MAC).

First, Zhang et al. [29] proposed the "in network aggregation" mechanism by employing a cross-layer design. Because the communication traffic statistical data of MAC layer affects the traffic density and the data arrival, they determine the aggregation period on the basis of traffic statistics of MAC layer. Its objective in-network aggregation was to reduce the amount of data to be transmitted as much as possible. They intended to apply SMDP to optimize the aggregation period and gave an approximate solution by exploiting a real-time Q-learning algorithm.

Another NET-MAC approach was the route selection through link prediction. Menouar et al. [30] proposed Movement Prediction-based Routing (MOPR), which was an approach proposing a movement prediction based routing protocol for Vehicle to Vehicle (V2V) communication in VANETs. It takes vehicle movement information available in MAC layer, such as position, direction, speed, and network topology into consideration, in order to improve the routing process. MOPR predicts the future location of intermediate relay nodes, which help in selecting the most stable routes containing stable nodes that are traveling in the same direction or with the similar speed or on the same road as of the destination/source nodes.

Similar to MOPR, Chen et al. [31] proposed a multi-path routing protocol to reduce the frequency of route rediscovery. They proposed a cRoss-layer Ad hoc On-demand Multipath Distance Vector (R-AOMDV) protocol. This method made use of a routing metric that combines hop count and transmission counts at MAC layer by taking into consideration the quality of intermediate links and delay reduction. It relied on two control packets: Route REQuest (RREQ) and Route REPly (RREP). The intermediate first hop nodes in RREQ and RREP packets were used to distinguish between multiple paths from source to destination. To measure the quality of the entire path, we add two additional fields to RREP packets: the Maximum Retransmission Count (MRC) that is measured in MAC layer and the total hop count that is measured in network layer.

F. Physical-MAC-Network exchange Solutions

Solutions in this section are based on the information flow between PHY, MAC and NET layer (PHY-MAC-NET).

A novel CAC algorithm was proposed by Bejaoui [32]. It provides the desired throughput guarantees on the basis of the vehicle density and the nodes' transmission range in 802.11p VANETs. They considered vehicle-to-roadside communications as essential to manage traffic situations. In order to enhance the performance of vehicular communications, this scheme adapts the transmission power physical layer and optimizes the contention window size (in MAC layer) depending on information coming from NET Layer as the vehicle density estimation. Results have shown that this solution improves the performance of the vehicular communication.

Also, Sofra et al. [33] proposed an approach using a Link Residual Time (LRT), which was calculated by using the received power from the physical layer. This value can be used by other layers to make better decisions for handover, scheduling time, and routing decisions. Each vehicle monitors parameters like the arrival time and the received power level for each packet that was received on the link. The estimation of LRT starts by removing the noise from the data, and checks if the link quality is deteriorating, then, it estimates the model parameters required for calculating LRT. Finally, renewing LRT is estimated.

Finally, Singh et al. [34] presented the use of link connectivity information among neighbors to help in addressing the challenges in designing routing protocols for VANET environments. They proposed a cross-layer protocol called Signal Strength Assessment Based Route Selection for OLSR (SBRS-OLSR). In this framework, the link connectivity was based on SNR measurement, and the routing protocol was based on existing Optimized Link State Routing (OLSR). By capturing SNR information from the physical layer, the network layer can provide a better route that improves throughput and delay performance.

III. DISCUSSION AND OPEN ISSUES

In section II, we surveyed the existing cross-layer techniques and solutions for enhancing multimedia QoS. Cross-layer solutions are efficient in serving a better video service by adjusting layers' parameters in regard of other layers' information. Cross-layer approaches try to overcome the lack of coordination between layers that limits the performance of wireless networks. These solutions allow coordination, interaction and joint optimization of protocols crossing different layers.

For example, PHY-APP exchange solutions, which adjust physical layer parameters in terms of application

 ISBN: 978-1-61208-377-3

layer information, can increase Multimedia QoS using efficient handoff solutions. Additionally, MAC-APP solutions significantly help in offering a reliable video communication in VANET by adjusting contention windows and frame rate depending on QoS information received from the application layer. Furthermore, NET-APP solutions help in finding the best route in terms of QoS to disseminate packets, or choosing the optimal cluster head, which will act as a broadcaster of Multimedia data. This in turn results in order to have a better multimedia service.

According to the literature, PHY-APP and MAC-APP are the most effective solutions, which adapt Physical and MAC parameters in terms of interaction with Application layer. Since Physical and MAC layers are the closest to nodes, their impact can be higher than other solutions.

However, there are several issues that need further attention. Below; we will try to evaluate these studies and describe open research problems, which will need to be studied, in addition.

A. *Evaluation metrics and tools*

The evaluation of existing techniques is one of the most relevant open issues, as it is not easy to measure Multimedia communication QoS and it is more subjective rather than objective. The aforementioned works did not necessarily use the same common metrics. Some of studied works simulations use packet loss rate, and/or end to end delay, and/or throughput which are not necessarily significant for Multimedia QoS. Some papers use other metrics like PSNR, Mean Opinion Score (MOS), Video distortion, or other video metrics [35]. In addition, they do not necessarily use the same simulation tools. There are several network simulators, such as NS2, NS3, OMNET++, OPNET, etc.[36]. This makes it more difficult to compare between the results of different solutions. So thus, finding a common tool and metrics function can significantly help in evaluating and comparing different solutions.

B. *Global solution*

As evident from existing research described in the last section, many protocol designs focus on a specific problem in multimedia communication Qos. Some of them try to solve end to end delay problem; others try to reduce packet loss rate. Other works by trying to stop or minimize freezes and video play drawbacks. However, most of these protocols ignore to design a complete solution for Multimedia over VANET, which can take into consideration most of the problems encountered in Multimedia communication QoS in VANET. This kind of solution, to the best of our knowledge, has not been alluded to or studied in the literature. Such solutions can be presented as a global solution working on different ISO layers at the same time (data are exchanged between multiple layers). Therefore, it is still an open research issue to develop a global solution, which takes into account many other factors at time. Our paper may help in developing such complete solutions.

C. *Mobility models*

Another important issue is “the mobility models” which represent the movement of mobile vehicles in changing their positions, speeds and accelerations all the time. Therefore:

- It is very important to use realistic mobility models that reflect reality. This can be very important in analyzing the performance of the different proposed solutions.
- Also, as everyone knows, there are many mobility scenarios types: highway scenarios, intersection scenarios, rural scenarios, etc. Most of works focus their proposed solution on a specific mobility scenario type (even if it is realistic). So it is very important to design a solution which can be compatible with all existing mobility scenarios types.

D. *Existence of other applications in the network*

Most of the mentioned solutions focus on a specific problematic, which is “Multimedia QoS”. They try to provide a better video service. However, this may negatively affect on other applications behavior, because the channel is shared by many devices, so it is difficult to fulfill QoS guarantees. Also, most of these studies realize their simulations without taking into account existence of other applications in the network which may distort their results, such as file transfer, video applications or any other launched application. Therefore, it is still an open research issue to study and develop solutions for. This might take into account existing of other applications in the network.

E. *QoS support in a multicast streaming*

It is an area which requires attention and studies in cross-layer solutions in VANET.

F. *Cross-Layer design and instability*

Additionally, any Cross-Layer design should take attention as undesirable effect on the system performance can occur due to Cross-Layer exchanges. Frantic and extensive Cross-Layer exchanges may lead to a complex mixture design and may lack standardization and compatibility and portability features. So, it is important to have a deep analysis and design of the Cross-Layer solution because it may lead to a state of instability [37], which is very important, especially in VANET, because of large number of nodes (vehicles and RSUs), and multiple sources and destinations.

IV. CONCLUSION AND FUTURE WORKS

In this paper, we focused on the problem of video communication between vehicles in VANET. Many solutions have been proposed in the literature in order to enhance multimedia QoS and provide better video services between vehicles or between RSUs and vehicles.

Cross-layer adaptations are essential for guaranteeing QoS in Multimedia Communications over VANET. We surveyed existing cross-layer techniques and solutions that enhance multimedia QoS. A classification is provided depending on data exchange type (belonging to ISO layers). We present many types of techniques and systems: PHY-APP, MAC-APP, NET-APP, PHY-MAC,

 ISBN: 978-1-61208-377-3

NET-MAC and NET-MAC-PHY, where each layer changes its parameters in function of other layer information. As a future work, a comparison between different classes is needed, in order to deduct the most effective class and technique of mentioned Cross-Layer solutions.

REFERENCES

[1] Z. Yan, H. Zhou, H. Zhang, and S. Zhang. “Speed-Based Probability-Driven Seamless Handover Scheme between WLAN and UMTS”, 4th International Conference on Mobile Ad-hoc and Sensor Networks, Los Alamitos, CA, USA, 2008, pp. 110–115.

[2] D. Kwak, J. Mo, and M. Kang. “Investigation of handoffs for IEEE 802.11 networks in vehicular environment”, ICUFN 2009, Hong Kong, June 7-9, 2009, pp. 89–94.

[3] Y. S. Chen, C. H. Cheng, C. S. Hsu, and G. M. Chiu. “Network Mobility Protocol for Vehicular Ad Hoc Network”, IEEE WCNC 2009, Budapest, Hungary, April 5-8, 2009, pp. 1–6.

[4] L. Zhu , F.R. Yu, B. Ning, and T. Tang, “Cross-Layer Design for Video Transmissions in Metro Passenger Information Systems”, IEEE Transactions on Vehicular Technology, vol. 60, iss. 3, March 2011, pp. 1171 - 1181.

[5] R. S. Sutton, D. Precup, and S. Singh, ”Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning », Journal of Artificial Intelligence, vol. 112, 1999, pp. 181–211.

[6] X. Changqiao et al. , “QoE-driven User-centric VoD Services in UrbanMulti-homed P2P-based Vehicular Networks”, IEEE Transactions on Vehicular Technology, vol. 62 , iss. 5, June 2013, pp. 2273 – 2289.

[7] A. S. Sadiq et al., “An Intelligent Vertical Handover Scheme for Audio and Video Streaming in Heterogeneous Vehicular Networks”, Mobile Networks and Applications, vol. 18, iss. 6, December 2013, pp. 879-895.

[8] W. Zhuang, B. Bensaou, and K. C. Chua ,”Adaptive Quality of Service Handoff Priority Scheme for Mobile Multimedia Networks”, IEEE Transactions On Vehicular Technology, vol. 49, no. 2, March 2000, pp. 494 – 505.

[9] M. Asefi, W. Jon, and Xuemin Shen, “A Mobility-Aware and Quality-Driven Retransmission Limit Adaptation Scheme for Video Streaming over VANETs”, IEEE Transactions on Wireless Communications, vol. 11, iss. 5, May 2012, pp. 1817 – 1827.

[10] A. Mercado and K. Liu, "Adaptive QoS for Wireless Multimedia Networks Using Power Control and Smart Antennas",IEEE Transactions On Vehicular Technology, vol. 51, no. 5, September 2002, pp. 1223 - 1233.

[11] H. Venkataraman, et al., “Performance Analysis of Real-Time Multimedia Transmission in 802.11p based Multihop Hybrid Vehicular Networks”, 6th International Wireless Communications and Mobile Computing Conference, 2010, pp. 1151-1155.

[12] M.A. Bonuccelli, G. Giunta, F. Lonetti, and F. Martelli, “Real-time video transmission in vehicular networks”, Mobile Networking for Vehicular Environments, May 2007, pp. 115-120.

[13] Irina et al., "User-oriented Cluster-based Solution for Multimedia Content Delivery over VANETs", IEEE Computer Society, 2012, pp. 1-5.

[14] M. Asefi, W. Jon, and Xuemin Shen., “An Application-Centric Inter-Vehicle Routing Protocol for Video Streaming over Multi-Hop Urban VANETs”, IEEE International Conference on Communications, June 2011, pp. 1-5.

[15] M. Asefi, S. Cespedes, Xuemin Shen, and W. Jon,”A Seamless Quality-Driven Multi-Hop Data Delivery Scheme for Video Streaming in Urban VANET Scenarios”, IEEE International Conference on Communications (ICC), June 2011, pp. 1-5.

[16] S. Abraham and P. Sojan Lal, “Spatio-temporal similarity of network-constrained moving object trajectories using sequence alignment of travel locations”, Data Management in Vehicular Networks, vol. 23, August 2012, pp. 109–123.

[17] L. Jung-Shian , L. I-Hsien Liu, K. Chuan-Kai, and T. Chao-Ming, “Intelligent Adjustment Forwarding: A compromise between end-to-end and hop-by-hop transmissions in VANET environments”, Journal of Systems Architecture, vol. 59, iss. 10, November 2013, pp. 1319-1333.

[18] M. Asefi, W. Jon, and X. Shen, “A Cross-Layer Path Selection Scheme for Video Streaming over Vehicular Ad-Hoc Networks”, VTC 2010-Fall, September 2010, pp. 1-5.

[19] Mottahedi, et al., "IBCAV: Intelligent Based Clustering Algorithm in VANET", IJCSI International Journal of Computer Science Issues, vol. 10, iss. 1, no 2, January 2013,pp. 538.

[20] A. Vinel, , E. Belyaev, K. Egiazarian, and Y. Koucheryavy, “An Overtaking Assistance System Based on Joint Beaconing and Real-Time Video Transmission”, IEEE Transactions on Vehicular Technology, vol. 61, iss. 5, June 2012, pp. 2319 – 2329.

[21] C. Yung-Cheng and H. Nen-Fu, “Delivering of Live Video Streaming for Vehicular Communication Using Peer-to-Peer Approach”, Mobile Networking for Vehicular Environments, May 2007, pp. 1-6.

[22] J. Camp and E. Knightly, “Modulation rate adaptation in urban and vehicular environments: cross-layer implementation and experimental evaluation”, 14th ACM International Conference on Mobile Computing and Networking, 2008, pp. 315–326.

[23] M. Vutukuru, H. Balakrishnan, and K. Jamieson, “Cross-layer wireless bit rate adaptation”, ACM SIGCOMM Computer Communication Review, vol. 39, no 4, 2009, pp. 3–14.

[24] C. Jiann-Liang, "An Adaptive QoS mechanism for multimedia applications over next generation vehicular network", 5th International ICST , CHINACOM, August 2010, pp. 1-6.

[25] D.B. Rawat, G. Yan, D. Popescu, M. Weigle, and S. Olariu, “Dynamic Adaptation of Joint Transmission Power and Contention Window in VANET”, 2009, pp. 1–5.

[26] M. Artimy, W. Robertson, and W. Phillips, “Assignment of dynamic transmission range based on estimation of vehicle density”, 2nd ACM International Workshop on Vehicular Ad Hoc Networks, 2005, pp. 48.

[27] G. Caizzone, P. Giacomazzi, L. Musumeci, and G. Verticale, “A power control algorithm with high channel availability for vehicular ad hoc networks”, IEEE International Journal on Communications, 2005, pp. 3171–3176.

[28] D.B. Rawat, et al., "Enhancing VANET Performance by Joint Adaptation of Transmission Power and Contention Window Size", IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 9, September 2011, pp. 1528-1535.

[29] L. Zhang et al., "QoS-Oriented Data Dissemination in VANETs", IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, 2012, pp. 2518 – 2521.

[30] M. Menouar, M. Lenardi, and F. Filali, “A movement prediction based routing protocol for vehicle-to-vehicle communications”, 66th Vehicular Technology Conference, 2007, pp. 2101 - 2105.

[31] Y. Chen, Z. Xiang, W. Jian, and W. Jiang, “A Cross-layer AOMDV Routing Protocol for V2V Communication in Urban VANET”, 2009, pp. 353– 359.

[32] T. Bejaoui, “QoS-Oriented High Dynamic Resource Allocation in Vehicular Communication Networks”,The Scientific World Journal, 2014, pp. 1-9.

[33] N. Sofra, A. Gkelias, and K. Leung, “Link residual-time estimation for VANET cross-layer design”, 2009, pp. 1–5.

[34] J. Singh, N. Bambos, B. Srinivasan, and D. Clawin, “Cross-layer multi-hop wireless routing for inter-vehicle communication”, TRIDENTCOM, 2006, pp. 92–101.

[35] A. Chan, K. Zeng, P. Mohapatra, S. Lee and S. Banerjee, "Metrics for Evaluating Video Streaming Quality in Lossy IEEE 802.11 Wireless Networks", INFOCOM, March 2010, pp. 1-9.

[36] S. Siraj, A. K. Gupta, and R. Badgujar, "Network Simulation Tools Survey", IJARCCE, vol. 1, iss. 4, June 2012, pp. 201-210.

[37] B. Jarupan, E. Ekici, « A survey of cross-layer design for VANETs », Ad Hoc Networks journal, vol. 9, iss. 5, July 2011, pp. 1-18.

 ISBN: 978-1-61208-377-3

Applications and Opportunities for Internet-based Technologies in the Food Industry

Saeed Samadi
Food Machinery dept.
Research Institute of Food Science & Technology (RIFST)
Mashhad, Iran
s.samadi@rifst.ac.ir

***Abstract*— In the modern world, information technology (IT) has been incorporated in most development activities. The food production industry is one of the recent industries to embrace IT in their major daily operations. This study discusses applications and opportunities for Internet-based technologies in the food industry and highlights state-of-the-art technologies and trends in the field. These technologies are mainly classified into Radio Frequency IDentification (RFID) for supply chain management, quality and safety monitoring, e-commerce, robotics, Wireless Sensor Networks (WSN), and Geographic Information Systems (GIS). Since all emerging technologies are coupled with challenges, the study addresses both challenges and benefits of incorporating IT in the food industry. Safety and quality of food products is a vital issue in the context of the food industry. As a result, this paper discusses how IT can be integrated to enhance the safety and quality of food products. The paper concludes by arguing that awareness be raised within the agro-food industry on the importance of the adoption of Internet-based technologies as a critical success factor in the twenty-first century.**

Keywords- food industry; information technology; Internet; RFID; e-commerce.

I. INTRODUCTION

Information technology (IT) is one of the individual forces that has contributed to globalization and advancement of life standards. These advancements have been occurring rapidly due to the rate of innovation from the IT industry. Significant incorporation of IT in most of the developmental activities is proof of the spread and importance of this technology [1]. IT has been globally incorporated in construction industries, production, manufacturing, healthcare, education, information management, security, and food and agricultural production. However, IT has been embraced at different levels by the fields mentioned above. Information management ranks as the most advanced field concerning use of IT [2]. On the other hand, agriculture ranks as the least innovative field as far as incorporation of IT is concerned. Other fields of production besides agriculture and food obtain maximum potential production from their fields due to highly incorporated IT systems [3]. Unfortunately, agriculture and food production does not extract its maximum potential because of its low level of IT incorporation. Most of the yield available in the agricultural sector is retrieved from small and medium sized enterprises (SME). Therefore, high scale production firms from the food and agriculture production industry are not constructive parties in the business [4]. SME are characterized by either low or medium financial capacity. This financial background is not able to fully fund state-of-the-art technologies such as radio frequency identification (RFID), wireless sensor networks (WSN), and integration to e-commerce. These technologies are available for application by the food and agriculture industry, and once incorporated, agricultural and food production would be able to maximize its potential [5]-[7].

The rest of this paper is organized as follows. Section II summarizes a review of literature. Section III discusses the use of important and available Internet-based technologies in the food industry. Section IV discusses in finer detail, the aim of the paper. Section V includes an acknowledgement and conclusions.

II. LITERATURE REVIEW

Currently, food and agricultural production has incorporated IT to a significant degree. Unfortunately, there still exist technical challenges that have resulted in the industry incurring losses and gaining a bad reputation. These technical challenges can be corrected through application of the mentioned technologies [8]. IT is signified by techniques that result in faster, efficient production with minimal human effort. Agricultural production is an economic activity that is more dependent on human input relative to machine input than other activities. This does not mean that technologies to minimize human effort and input in the industry are absent. Technologies that can result in reduction of human input exist in the industry, but the prevailing challenge is the cost of operation. As initially stated, SMEs comprise robust producers in the industry and lack sufficient capital to sustain these technologies [9]. IT applications relevant to the field of agriculture require high initial capital, but are cost effective. Areas within the field of agricultural production that can incorporate IT include; supply chains, harvest, standardization, marketing, soil fertility, and yield prediction [10]. These areas can be improved by the following technologies: RFID, WSN, GIS, robotics control, and e-commerce. These technologies are applied in the agricultural and food production industry to fulfill different objectives. These technologies utilize networks for communication. However, some technologies such as RFID have more than one application in the industry. It can be used in supply chain management and also in traceability for standardization [11].

 ISBN: 978-1-61208-377-3

III. AVAILABLE TECHNOLOGIES

A. RFID Technology

This technology uses radio frequency to identify or retrieve information from production. It operates using the same mechanism as barcodes with magnetic strips [12]. Instead, of a barcode, RFID uses microchips that are embedded on the product of interest. RFID has two main advantages over barcodes. In the case of a barcode, it has to be on the line of sight of the barcode reader for information to be obtained from it. RFID is advantageous because the chip and the reader do not have to be on a line of sight to retrieve information from the chip, because the chip produces specific radio frequencies. The other advantage of RFID is that the chip is more reliable than the barcode [9]. This is because validity of barcodes is ruined once the code is scratched or removed. RFID microchips are not easily removed because they are not attached to the surface of the product.

RFID technology can be used in supply chain management and standardization. Food quality has been the cause of controversy in the food industry. Food that has not been properly stored has higher chances of going bad and once food has attained this status, it can become toxic. Food toxicity is dangerous as it can result in complex health disorders or even death. Therefore, a compromise on the quality of food is likely to ruin the reputations of the supplier and manufacturer, and this translates into losses [4]. RFID enables the user to establish the amount and type of ingredients contained in the food product. In addition, it also provides the time elapsed from the time of manufacture to the time of first use. This information is imperative to both the retailer and the consumer. Cases of food poisoning as a result of consuming expired food or allergic substances would be substantially reduced.

The other core challenge in the food production and agriculture industry is supply chain management. Some food products are essential for humans, but their production is unique in specific regions. Therefore, a comprehensive supply chain should be established so as to benefit both the manufacturer and consumer. The supply chain involves the food transit process from harvest, to processing, to distribution to the retailer [13]. Food undergoes this process before reaching the end user. Despite the extensive route, which is undergone before a product's use, monetary value has to be established. This means that the end user is not overcharged and the manufacturer is not underpaid [14]. RFID technology establishes an infrastructure that tracks a food product's location and ingredients, thus enhancing reliability of the end product. Farmers, specifically involved in food production, have been discouraged from expanding their investment due to limited profit from their enterprise. Previously, middle-men have benefited more than either the farmer or the consumer, minimizing profits to these constituents. Currently, with the employment of RFID, profits and satisfaction have improved because the supply chain of the goods has been bolstered by the technology. RFID technology has to be applied from the point of production (farmer) to the consumer. This reduces the bulk cost that could have been incurred by the distributor or supplier [15]. Wal-Mart is among the supply companies that have encouraged manufacturers to incorporate RFID to increase their profits. They encourage manufacturers through financing part of the RFID implementation. This practice is prevailing in most developing countries, as SMEs are financed to increase agricultural food production performance in the international market.

B. Information technologies applied in food quality and safety monitoring

Poor performance in the food industry is due to loss of trust between food distribution companies and the end user. This is as a result of food contamination and poisoning. In China, profits in the food industry have declined by 50 percent as a result of contaminated food. This shows the sensitive nature of the food industry as a single flaw has the potential to bring down the whole industry. In addition, China's food production is also mainly extracted from small scale investors. This means the sector is not fully exploited. One of the technologies that the country has embraced to enhance the status of the food supply is RFID food packaging. This technology uses RFID to identify the ingredients and the inventory of food products [16]. It utilizes disposable biosensors that produce an antigen-antibody reaction to identify any bacterial cells in the food product. When bacteria thrive in an enclosed food product, the result is a bio-chemical reaction that would either make the food product stale or poisonous. Therefore, this technology helps the food industry upgrade their monitoring systems, and the quality and safety of food products is enhanced. As a result, IT has aided in the restoration of trust between consumers and manufacturers. Furthermore, since the introduction of RFID food packaging, the number of health issues associated with food poisoning or food quality has declined by more than 50 percent [12]. RFID technology also has an additional use as biosensors used in the tags containing inventory information that can be used in supply chain management. Traceability of food products from the farmer to the consumer is the other main concern in the food production industry. Effective supply management is a barrier that prevents SMEs from maximizing their potential. RFID detection technology poses a remedy to this barrier; RFID stores ingredients, destination, and the appropriate geographic location of products [16]. This helps the food industry realize their market extent and as a result increase or reduce their production where necessary, thus minimizing losses. This technology enables rapid detection of poisons or derailed quality of finished food products. It also enables automatic identification of food products along a supply chain.

C. E-commerce

Internet technologies within the context of e-commerce have provided a more interactive market that enhances communication between manufacturers and consumers. This can be accomplished through existing social networking sites such as Facebook and Twitter. Manufacturers append their

 ISBN: 978-1-61208-377-3

social networking websites on containers of food products so that in case of a complaint or compliment, the user can directly communicate with the food company [17]. When there is a reliable communication pathway between service or good providers and the end user, performance of the product is likely to be high. This is relative to a scenario where there is no elaborate communication between the user and manufacturer. IT provides better database management systems that portray the accurate needs of consumers. E-commerce expands the food market as the Internet is able to establish new consumers from regions where a specific food product has not yet been sold. E-commerce serves to benefit SMEs more because of their otherwise insufficient capital to market their food products. E-commerce is cheaper than hiring a marketing firm. This system requires less than five users to conduct online marketing and thus is affordable for SMEs [18]. As a result, SMEs can access a larger market without seeking additional financial assistance to facilitate marketing. Therefore, Internet technologies ensure development of a more reliable supply chain, higher quality food products, and a larger market for food products.

The disadvantage of Internet technology among SMEs is that the business owners and staff have to undergo training so that they can understand computer systems [19]. This is an additional cost that a small scale investor aims to reduce by all means necessary. IT exposes SMEs to Internet hazards such as hacking and fraud, which can cause huge losses to investors.

D. Robotics technology

Immense human input, which is visible in the food production industry, can be replaced by machines as a result of IT. Human input is hindered by fatigue and non-uniform output. Opposed to human input, machine input as a result of IT, is both uniform and reliable. In addition, it is faster and produces more profit than human input. Robotics is applied in land preparation, planting, and weeding [19]. A series of corporative IT devices can sufficiently handle agricultural production leaving human application to solely play an oversight role. A combine harvester is one of the machines that has replaced human involvement in harvesting activity (see Figure 1). In cases where the machine has been used, there has been a greater than 100 percent advantage in yield compared to regions where human effort was used in harvesting. This prevailed in areas with the same size and climatic conditions.

E. WSN technology applications

WSN differ from RFID in that it is able to integrate with other network devices in the field while an RFID tag can only be read with the RFID tag reader. WSNs comprise of Wi-Fi, Bluetooth, and ZigBee. The latter two operate within the Industrial Scientific and Medical (ISM) band of 2.4 GHz, which provides license-free operations, enormous spectrum allocation, and global compatibility. Other devices deployed on a farm to aid agricultural activities [20].WSN technology

Figure 1. Diagram showing robotic harvesters

is used in this industry for monitoring and surveillance of crops within a farm. However, weather variation is the sole challenge that affects performance of WSN in the agriculture industry. The technology utilizes radio frequencies that can be interfered with by weather conditions [21]-[23]. The technology is used in maintenance and monitoring of farmlands. This is achieved through installation of sensors and cameras on the field. These devices are linked to the control station on the farm via the mentioned wireless technology. Monitoring fields enables identification of severe conditions on the soil and weather. With this information, farmers make comprehensive decisions concerning planting activities. Wireless technology also enables pest control and irrigation activities that are essential when pursuing maximum yield. Sensors deployed on the soil are able to determine moisture content of the soil. When soil moisture content is below the minimum, the information is transferred to the control that commands the irrigator to sprinkle the soil. Phytophotra is a disease that affects potatoes and is influenced by temperature and humidity conditions. Between 868MHz and 916MHz, motes can be used in determining moisture content on air and temperature [24]. Extreme temperatures can be reflected and relayed to the control station which initiates spray of pesticides.

F. GIS applications

A GIS uses unique colors and shades of colors to represent different atmospheric and soil conditions. It also uses the same set of unique colors to depict different terrains and ground cover. They utilize satellites to obtain aerial images of the Earth's surface. These satellites exist exclusively for GISs as the colors of objects and surfaces are different from ordinary depiction and representation. For instance, a water body would appear blue from ordinary satellites, whereas a GIS satellite depicts water bodies in dark. Food production and agriculture is governed by atmospheric conditions and soil fertility. Globally, farmers' yields are affected by changes in weather and climate. This is because of poor decisions that are dependent on farm activities [25]. For instance, harvesting time is signified by dry weather and medium to high temperatures. Therefore, when a farmer harvests during other atmospheric conditions, the resultant yield will be low. Through GIS technology farmers have been able to obtain atmospheric conditions in

 ISBN: 978-1-61208-377-3

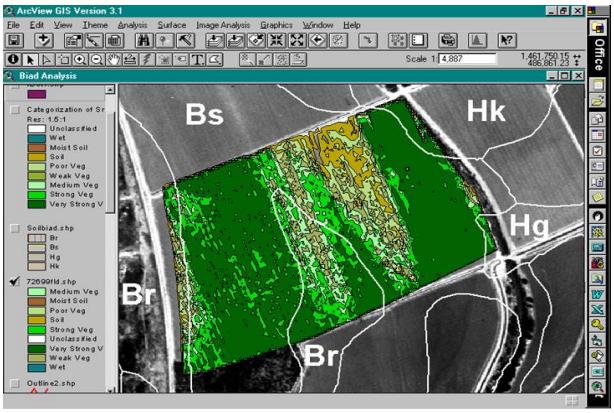


Figure 2. Diagram of remote-sensed image of the soil (GIS image)

real-time that have enabled them to conduct farming activities appropriately. GIS images are specific to natural, physical features. As a result, farmers or investors are able to locate ideal regions that will favor their agricultural investments. Planting on the wrong soil will result in lower yield and losses. Analyzing a soil using the naked eye does not necessarily yield an accurate description of the soil's composition [26]. Therefore, advanced methods induced by IT would result in better land use, thus maximizing yield from the food production and agricultural industry (see Figure 2).

IV. DISCUSSION

The primary goal of IT incorporation in the food industry is to foster food security and extract maximum sustainable yield. Once the primary role has been fulfilled, there are numerous secondary goals that IT ensures are effectively addressed. They include: processing, distribution, marketing, and storage [27]. IT, through the technologies previously discussed, fulfills each of these goals successfully. RFID ensures comprehensive results from supply management, which constitutes a secondary goal of the industry [26]. Regions that have incorporated RFID in their supply chains receive more revenue from the agriculture and food production industry than regions that have not applied RFID technology [29]. Similarly, regions that have incorporated WSN practice sustainable farming on a larger scale than in regions where the technology has not been applied. After production, the other barrier to extracting maximum potential from agriculture is the marketing of harvested goods. Large scale producers in farming have extensive marketing strategies that cover almost ninety percent of their produce. On the other hand, SMEs in agricultural and food production lack elaborate marketing avenues that can ensure intake of their products in the market. The first obstacle is the cost, which is a requirement for establishing an elaborate marketing network. The other obstacle is technology. Technology now offers a solution to its initial problem in that the Internet has contributed positively towards establishing global villages. Farmers are able to establish first person contact between the manufacturer and the user or processing firm. For instance, the Kenyan association of coffee growers has established a direct link to coffee processing firms in England and the United States. This ensures farmers obtain maximum compensation for their products, and therefore represents an appropriate motivation for farmers to expand their farms. As a result, the potential of food and agricultural production is optimized.

Food investors who have embraced robotics and e-commerce receive more income from the food industry than food investors not aware of the technology or those who have shunned it. Consequently, in countries where these practices have been encouraged and are prevailing at significant levels have a better economy than in countries where IT application is limited.

Another added value of internet-based technologies in the agro-food industry is the improvement of efficiency and reactivity from real-time management of supply chains from farm to fork [30]. From a "food miles" point of view, this could result in a reduction in greenhouse gas emissions and in the carbon footprint, e.g., decrease of transport kilometers or empty vehicles, less waste, and better decay management.

The digital divide is a challenge that might hinder the applicability of the technologies discussed in this paper. Digital divide is mainly the gap between those with and those without access to ICT technologies and/or skills necessary to take advantage of ICT services. In addition, there is a widening gap between the urban and rural sectors on utilizing advanced and emerging technologies [31]. To overcome this, measures should be taken to strengthen informatics in the agro-food industry by fostering the development of national information capacity and new databases, linking national and international databases, and adding value to information to facilitate utilizing them at various levels. Also, innovative ways of combining ICT-based information sources (such as agro-food information systems) with traditional ones should be considered.

V. CONCLUSION

This paper gives an overview of major IT-based technologies and their impact on the food industry. It presents how selected fields of application can make a considerable contribution to food industry both in increasing efficiency and making data more available and easily managed. It discusses how these technologies can be integrated to enhance safety and quality of food products and provide advantages such as mobility, transparency and autonomy. The example technologies are mainly built on networked devices or utilize networks for communication. However, much additional work still should be done for a large scale integrated communication and scalable coordination throughout the agro-food networks.

The paper also highlights that there is great opportunity for internet-based applications in developing countries. However, in most developing countries, strategies should be employed to overcome technical and societal barriers that can hinder further development of these technologies in agro-food sector. Therefore, it is a mandate of the ministry of agriculture and/or other governmental authorities to ensure IT techniques are being used in the food and agriculture sector to boost production and create an extensive market for the produced goods.

 ISBN: 978-1-61208-377-3

REFERENCES

[1] Y. Yu, J. Li, and X. Y. Qin, "The Information Key Technologies for Quality & Safety Monitor and Management of Agricultural Products," Advanced Materials Research, vol. 634, 2013, pp. 4004-4010.

[2] N. P. Mahalik and A. N. Nambiar, "Trends in food packaging and manufacturing systems and technology," Trends in food science & technology, vol. 21, 2010, pp. 117-128.

[3] A. Z. Abbasi, N. Islam, and Z. A. Shaikh, "A review of wireless sensors and networks' applications in agriculture," Computer Standards & Interfaces, vol. 36, 2014, pp. 263-270.

[4] D. Prajogo and J. Olhager, "Supply chain integration and performance: The effects of long-term relationships, information technology and sharing, and logistics integration," International Journal of Production Economics, vol. 135, 2012, pp. 514-522.

[5] A. Suprem, N. Mahalik, and K. Kim, "A review on application of technology systems, standards and interfaces for agriculture and food sector," Computer Standards & Interfaces, vol. 35, 2013, pp.355-364.

[6] M. Cinque, D. Cotroneo, C. Di Martino,S. Russo, and A. Testa, "Avr-inject: A tool for injecting faults in wireless sensor nodes," In IEEE International Symposium on Parallel & Distributed Processing, IEEE IPDPS 2009, May 2009, pp. 1-8.

[7] C. Di Martino, G. D'Avino, and A. Testa, "icaas: An interoperable and configurable architecture for accessing sensor networks," International Journal of Adaptive, Resilient and Autonomic Systems, vol. 1 , no.2 ,2010 pp. 30-45.

[8] J. Von Braun, The world food situation: new driving forces and required actions. Intl Food Policy Res Inst, 2007.

[9] M.E. Yüksel and A. S. Yüksel, "RFID technology in business systems and supply chain management," Journal of Economic and Social Studies vol.1, no. 1,2011,pp. 53-71.

[10] J. Wolfert, C. N. Verdouw, C. M. Verloop, and A. J. M. Beulens, "Organizing information integration in agri-food—A method based on a service-oriented architecture and living lab approach," Computers and electronics in agriculture vol. 70, no. 2, 2010, pp. 389-405.

[11] T. Kelepouris, K. Pramatari, and G. Doukidis, "RFID-enabled traceability in the food supply chain," Industrial Management & Data Systems vol.107, no.2, 2007, pp. 183-200.

[12] E. A. Soujeri, R. Rajan, and A. Harikrishnan, "Design of a zigbee-based RFID network for industry applications," Proceedings of the 2nd international conference on Security of information and networks. ACM, 2009, pp. 111-116.

[13] A. Sarac, N. Absi, and S. Dauzère-Pérès, "A literature review on the impact of RFID technologies on supply chain management," International Journal of Production Economics, vol.128, no.1, 2010, pp.77-95.

[14] J. Fontanella, "Finding the ROI in RFID," Supply Chain Management Review, vol. 8, no.1, 2004, pp.13-16.

[15] K. Butner, "The smarter supply chain of the future," Strategy & Leadership, vol. 38, no.1, 2010, pp. 22-31.

[16] X.Zhu, S. K. Mukhopadhyay, and H. Kurata, "A review of RFID technology and its managerial applications in different industries," Journal of Engineering and Technology Management, vol. 29, no.1, 2012, pp.152-167.

[17] N. P. Mahalik, "Processing and packaging automation systems: a review," Sensing and Instrumentation for Food Quality and Safety, vol. 3, no.1, 2009, pp. 12-25.

[18] M. J. Meixell, "Quantifying the value of web services in supplier networks," Industrial Management & Data Systems, vol.06, no.3, 2006, pp. 407-422.

[19] R. A. Tabile, et al., "Design and development of the architecture of an agricultural mobile robot," Engenharia Agricola, vol. 31, no.1, 2011, pp. 130-142.

[20] T. Gulledge, "What is integration? ," Industrial Management & Data Systems 106.1 (2006): 5-20.

[21] L. Ruiz-Garcia, P. Barreiro, and J. I. Robla, "Performance of ZigBee-based wireless sensor nodes for real-time monitoring of fruit logistics," Journal of Food Engineering, vol. 87, no.3, 2008, pp. 405-415.

[22] A. Testa, A. Coronato, M. Cinque, and J. C. Augusto, "Static verification of wireless sensor networks with formal methods," In Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on , IEEE, November 2012, pp. 587-594.

[23] M. C. Cinque, D. Di Martino, and A. Catello Testa, "An effective approach for injecting faults in wireless sensor network operating systems," Computers and Communications (ISCC), 2010 IEEE Symposium on. June 2010.

[24] A. Baggio, "Wireless sensor networks in precision agriculture," ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden. 2005.

[25] P. K. Haneveld, "Evading murphy: A sensor network deployment in precision agriculture," O teu-Delft, Xuño 2007.

[26] T. Kalaivani, A. Allirani, and P. Priya, "A survey on Zigbee based wireless sensor networks in agriculture," Trendz in Information Sciences and Computing (TISC), 2011 3rd International Conference on. IEEE, 2011.

[27] D. Restuccia, et al, "New EU regulation aspects and global market of active and intelligent packaging for food industry applications," Food Control, vol. 21, no.11, 2010, pp. 1425-1435.

[28] M. Bolic, D. Simplot-Ryl, and I. Stojmenovic (Eds), RFID systems: research trends and challenges. John Wiley & Sons, 2010.

[29] M. Canavari, R. Centonze, M. Hingley, and R. Spadoni, "Traceability as part of competitive strategy in the fruit supply chain," British Food Journal, vol.112, no.2, 2010, pp. 171-186.

[30] G. Schiefer, R. Reiche, and J. Deiters, "Transparency in Food Networks-Where to Go," International Journal on Food System Dynamics, vol.4, no.4, 2014, pp. 283-293.

[31] R. Bertolini, "Making information and communication technologies work for food security in Africa," International Food Policy Research Institute (IFPRI), No. 11, 2004.

 ISBN: 978-1-61208-377-3

In-Network Support For Over-The-Top Video Quality of Experience

Francesco Lucrezia, Guido Marchetto and Fulvio Risso

Department of Control and Computer Engineering
Politecnio di Torino, Italy
Email: name.surname@polito.it

***Abstract*—This paper discusses the effects of the network congestion on the goodness of a streaming video session and proposes a solution that the network itself can adopt to recover from the possible Quality of Experience degradation. We consider a Broadband Access Network scenario, where congestion is most likely to occur in current communication networks. In particular, we concentrate on the Asymmetric digital subscriber line (ADSL) technology, which is predominant in the last-mile link toward the user machine. The proposed solution is based on an online heuristic evaluation of the mismatch between the bandwidth requirements of the video and the throughput actually offered in the bottleneck link. When a possible Quality of Experience degradation is observed, our tool reacts by limiting the concurrent traffic. The YouTube application is used as a reference throughout the entire paper due to its wide popularity. Moreover, High Definition videos are mainly considered as they are the most bandwidth demanding and hence the most sensitive to network impairments.**

Keywords–HTTP streaming; Quality of Experience; YouTube; TCP splitting.

I. INTRODUCTION

Since web-content providers started to make available a very huge amount of data, the Internet traffic is inevitably growing, especially the one related to multimedia applications. In particular, HTTP-based streaming traffic is growing steadily due to the increasing popularity of content providers such as Netflix, Hulu, and YouTube. Over-The-Top (OTT) videos embedded in the Internet applications can be watched by each individual equipped with an Internet connection and this also makes customers more and more demanding. For example, the spread of High Definition (HD) video retrieving is continuously increasing. For these reasons, the delivery of OTT content is one of the current challenges that network operators have to deal with.

One of the most important key points for the support of this kind of service is the provision of an adequate Quality of Experience (QoE) to the users. Namely, users have to be satisfied when watching the video, otherwise they are induced to switch to other providers for future content retrieves or, even worse, to suddenly leave before the end of the video. In both cases, this may cause a swift decrease of the provider revenues, e.g., due to advertisement fee losses. User satisfaction can be measured in several ways [1], ranging from the quality of the offered content to the experience offered by the deployed user interface. The former is clearly the most difficult to control as it actually depends on the network condition, which is variable in time and, in general, cannot be deterministically predicted.

Some solutions have been proposed [2][3], which target the mapping between the source of impairments and the final user QoE. However, they are closely related to the specific application under study. Specific solutions are then necessary for the HTTP-streaming case, which is the focus of our work. For such kind of application, the two main QoE metrics are the starting playback delay and the stalling events [4]. Hence, the main source of impairment to consider is the possible congestion of bottleneck links between the server and the clients, especially for the bandwidth demanding HD videos. In the current Internet, this problem might basically reside in the Broadband Access Networks providing the connectivity to final users, where the ADSL is predominant as last-mile technology. In these links, congestion might be due to the connection sharing among different users (e.g., tens in a small office sharing a single ADLS connection, as well as hundreds connected to different DSLAMs, or even both) or also when a single customer produces a lot of additional TCP traffic due to Internet applications that run automatically when host devices are connected to the network (e.g., BitTorrent file transfers).

Some solutions also exist for QoE provisioning in the OTT video streaming scenario, e.g., [5][6]. These all are effective, but require software modifications in the client machine, which is often unfeasible. In this paper we focus on an in-network solution, i.e., a countermeasure that the network itself can adopt to avoid OTT video QoE degradation. In other words, our solution is transparent to the video application, thus avoiding user intervention or client modifications. A trivial approach would clearly be the prioritization of video streaming traffic. However, this would be a static Quality of Service (QoS) solution — rather than one addressing users' QoE — and furthermore would suffer from the well-known starvation problem. At the same time, reservation-based techniques (e.g., [7][8][9] might be helpful in solving the problem, but these are not currently deployed in the Internet. Instead, the idea is to make use of the TCP splitting technique [10] in order to dynamically measure the throughput of the video session and possibly react to avoid stalls when it is lower than expected, thus really addressing a QoE problem. We consider the YouTube application as a reference in the rest of the paper as it is probably the most popular video content provider worldwide. However, it is worth noticing how the solutions described here might be easily extended to other HTTP-streaming services or, even more in general, to other TCP-based streaming applications (e.g., remote visualization tools [11]) as the main operating principles are similar.

The paper is organized as follows. Section II presents the YouTube service and a preliminary characterization work we performed on this application to specifically identifies its

 ISBN: 978-1-61208-377-3

internals, since they are continuously evolving. Section III discusses the proposed solution, while Section IV describes our testbed and reports on some experimental results. Finally, Section V concludes the paper.

II. YouTube

YouTube is a website created in 2005 and that rapidly became one of the most popular video-sharing application. It uses HTTP streaming for video content delivery (in particular, it recently adopted the Dynamic Adaptive Streaming over HTTP (MPEG-DASH) protocol), while it supports both Adobe Flash Player and HTML5 for video visualization at the client side. The transfer of the video content occurs by means of HTTP requests sent by the client for each chunk in which the entire video is divided according to the short-duration media segments (also called fragments) of the MPEG4 video format specifications. Concerning the service architecture, YouTube is based on a front-end Web server providing the home page of the website, while it relies on external resources for content delivery. In particular, when clicking on a link to watch a video, the web server redirect the first HTTP request to a specific video server selected according to the client position, RTT timing and other performance factors [12].

The main factors that characterize the transfer mode from a network perspective are the size of the data chunks, the frequency of the HTTP GET messages and the number of TCP connections used to transfer the chunks during the session. An extensive YouTube traffic characterization is available in literature (e.g., [13], [14]), which can provide an overview of such mechanisms. For example, it is well known that, like in other OTT services, video transfers start with a buffering phase during which a large amount of byte is downloaded by means of short-spaced HTTP GET messages and then proceed with a more smoothed download phase (i.e., the steady state) [15]. However, since YouTube internals are continuously evolving, we performed a characterization work to extract up-to-date information concerning the specific parameters that are of interest in our context. The Google Chrome Desktop browser is considered for this characterization work and in the rest of the paper.

The analysis of some capture files created by means of the tcpdump network analyzer led as to conclude that as soon as a request for a video is sent, the client opens a variable number from two to four different TCP connections with the server and the same applies for the entire duration of the video. This behavior is part of the logic that the application uses to react to a change in the network conditions or to a specific user action (e.g., when he pauses the video). In particular, the number of parallel TCP connections opened by the client plays a fundamental role in the observed performance. Indeed, it is the mean YouTube uses to counter the side effect of the ACK-driven congestion-control of the TCP protocol that causes the throughput lowering of a single TCP connection when dealing with traffic flows experiencing large RTT, packet losses, or congestions. In this way, the video session also becomes more robust and more aggressive with respect to the other competing applications that use a single TCP connection. It is worth noticing that the usage of parallel TCP connections is currently adopted also by several other OTT video applications.

Our analysis also pointed out that the number of chunks transferred by each connection, the size of the chunks, the amount of data downloaded during the buffering phase, as well as the precise pattern following by the variable bitrate in the steady state, strongly depend on the specific video considered and on its resolution. However, all videos analyzed had in common the fact that chunks are of two types: smaller chunks of dimension less than 500 KB and bigger chunks, larger than 1 MB. The same applies for HD videos with the only difference that the larger chunks can be bigger than 7MB for 1080p resolution and bigger than 4MB for 720p. A very common dimension (for any resolution) of the smaller chunks is 479232 bytes; this is an indication of the fact that chunks of smaller dimension actually belong to the audio stream and are transferred by one of the parallel connections opened by the system. Figure 1 details the specific behavior of two videos we used during our analysis.

III. In-network QoE support

Despite the abovementioned robustness of YouTube due to the utilization of parallel TCP connections, the presence of additional TCP traffic that contends for network resources might be a source of impairment. This might cause congestion over bottleneck links, with possible throughput decrease and consequent QoE degradation for video streaming users, especially when HD videos are considered. For this reason, our target is to guarantee an adequate bitrate to YouTube flows when they compete with other applications.

The specific problem statement is the following.

Problem statement. Given the bottleneck link capacity, the number of YouTube TCP flows and the number of competing TCP flows:

- Find the minimum bitrate that must be guaranteed for the streaming session in order to avoid stalling events (or a switching to a lower video resolution);
- Detect the instant in which congestion treats the video session;
- Act accordingly.

As stated in the previous sections, the bottleneck links in the current Internet infrastructure are in the access portion of the network. In particular, in this paper we consider a Broadband Access Network scenario, and the bottleneck links are then represented by last-mile links based on the ADSL technology.

We also point out that in the rest of this study we consider to know in advance which connections carry video streaming traffic and hence must be protected. The identification of these sessions is in fact a completely orthogonal problem and is outside the scope of the paper. Some possible solutions might rely on well-known traffic classification techniques (e.g., [16]) or further extensions of them.

A. Splitting the TCP connection

A TCP splitter [10] is a process able to intercept TCP connections and put itself in the middle of a communication. This is done by redirecting incoming SYN packets to a specific port where the splitter is listening (used to handle the connection with the sender) and then opening another TCP connection with the original recipient of the SYN packet. The splitter is responsible to pass data from one TCP connection to the other and vice versa, transparently to the end users. This

 ISBN: 978-1-61208-377-3

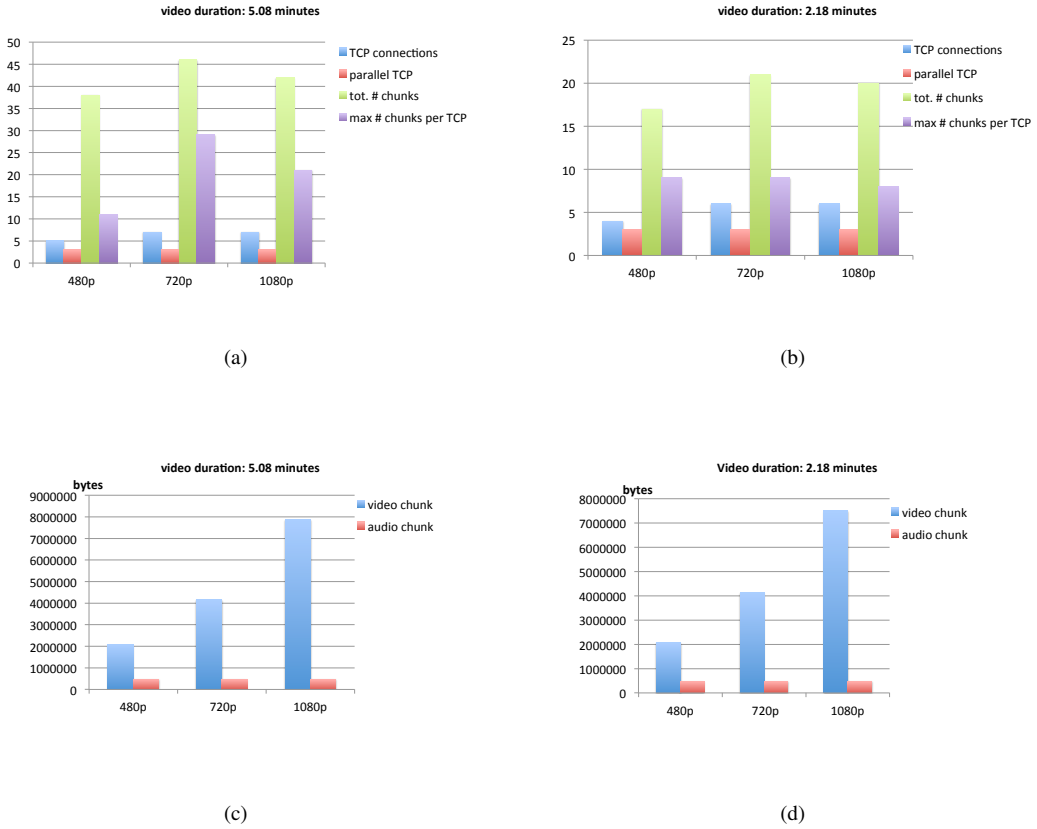


Figure 1. YouTube video characterization in terms of number of chunks, number of TCP connections, chunks per TCP connection and parallel TCP connections.

is usually adopted to increase TCP performance in the case of long end-to-end delays, which might reduce TCP overall throughput. The basic idea is that with two TCP connections instead of one, the weaker link has less influence on the whole path and the end-to-end segment established by the connection is broken into two segments shortening the overall RTT.

In our case, instead, the TCP splitter can be used for a totally different purpose. The idea is based on the fact that if the TCP splitter is located between the server and the bottleneck link — specifically, on the Broadband Remote Access Server (BRAS) of the ADSL system, the data received by the splitter from the server cannot be delivered toward the client at the same rate. In fact, since the receiver socket of the splitter continuously acknowledges packets coming from the server through a high-speed backbone network, the speed of data transfer remains higher than the speed in the bottleneck link. In this way, the splitter becomes the manager of a ”virtual” flow control driven by the bottleneck link and by the related speed with which the TCP socket delivers packets into this link. In a normal file transfer, the splitter receiver is faster than the splitter transmitter for the entire connection duration, because the file is by nature transferred at the maximum speed offered by the network. In a streaming video, instead, after the initial buffering phase (similar to a file transfer from this point of view), the download is throttled by a more sophisticated application logic, mentioned in the previous section. In particular, the server transmits only necessary data, i.e., the video chunks at the defined video bitrate, or a bit faster. Hence, in a given time unit, the amount of data received from the server should be the same of that sent to the client through the bottleneck link, even if at different speed. This is key to assure that the video is actually delivered with the expected bitrate to the client. If this condition is not satisfied, it means that the throughput offered by the bottleneck link is not sufficient to support that video bitrate.

This is the key point of our solution: if a process running on the BRAS is able to compare the throughput of the video session in the server-splitter link with that obtained in the splitter-client link, this process can infer when the video session is suffering and hence react to protect it, thus potentially avoiding stalls (or decreases in encoding quality in the case of DASH streaming). To better clarify these concepts, we report on some results we obtained in our emulated ADSL scenario, depicted in Figure 7 and better described in Section IV-A. Figure 2 and Figure 3 compare the video delivery pattern at an emulated BRAS with and without a TCP splitter running on it, when there is no congestion in the bottleneck link. In particular, Figure 2 shows the progress of received and transmitted bytes of a YouTube HD video at 1080p resolution into the emulated BRAS, without the splitter, while Figure 3

 ISBN: 978-1-61208-377-3

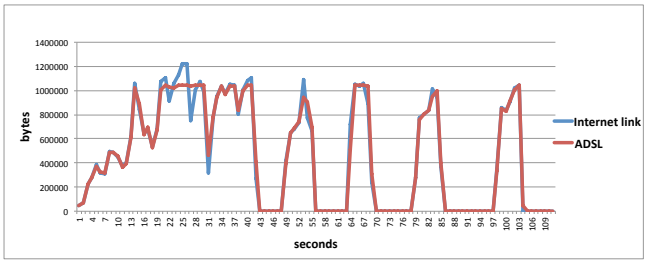


Figure 2. Video delivery pattern at the BRAS without TCP splitting

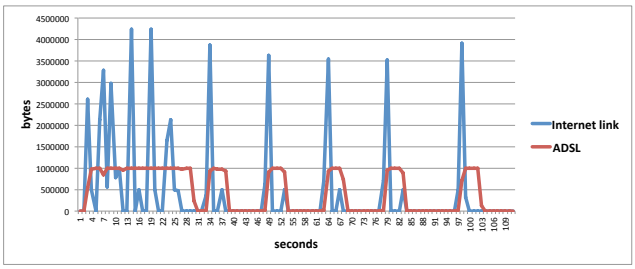


Figure 3. Video delivery pattern at the BRAS with TCP splitting

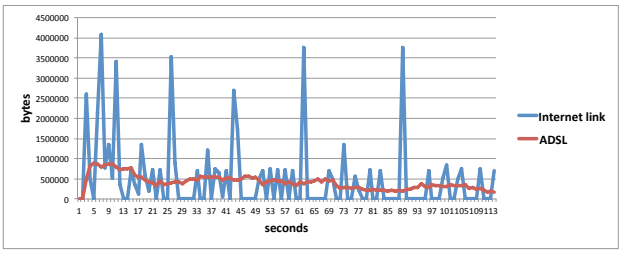


Figure 4. Video delivery pattern at the BRAS with TCP splitting and congested bottleneck link

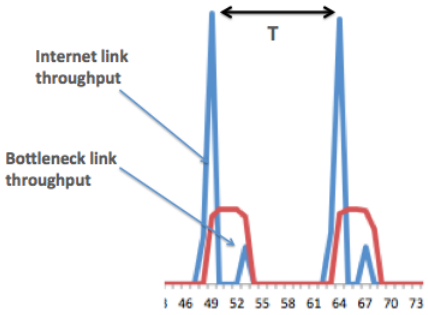


Figure 5. Heuristic

shows the progress of the same HD video with the action of the splitter. The emulated ADSL capacity is set to 7 Mbps in downstream and 400 Kbps in upstream. As said, the network is not congested.

The red line is the quantity of bytes injected at every second into the bottleneck link, while the blue line is the amount of byte received from the YouTube server.

The particular shape of the second pattern stems from the two main factors mentioned above: the peaks represent the download of each chunk characterizing the HTTP streaming, while the trend of the two series is due to the TCP splitting functioning, which permits to exploit the capacity of the fast Internet link by breaking the path of the TCP self-clocking mechanism within the network node. Notice the different maximum amplitude of the peaks between Figure 2 and Figure 3: without the splitter, the maximum amount of received bytes clearly depends on the bottleneck link capacity. In both cases, it is evident how, given a proper time unit (9 seconds in this specific case), the amount of bytes received from the server is equal to the one sent to the client, i.e., the application is not suffering.

Figure 4, instead, shows the video delivery pattern when the bottleneck link is congested. The video was the same of the previous graphs at the same resolution of 1080p. Notice how the traffic pattern significantly changes in the bottleneck link and how the amount of bytes forwarded to the client no longer follows in any way the pattern received from the YouTube server. In fact, we experienced two stalls during this experiment.

B. Video suffering detection

In order to detect when a video session is suffering and needs protection, it is necessary to keep track of the periodic throughput at the splitter. In particular, it is necessary to evaluate the difference between the number of received bytes from the server and the number of transmitted bytes to the client, in the same time interval. Since both the minimum required data rate and the correct time interval to consider cannot be known in advance, we propose a simple heuristic to determine such condition: the area described by the amount of received and transmitted bytes should be the same in the time interval defined by the spaced chunk download in the steady state phase of the video session (Figure 5). The idea is that if the area described by the received bytes is greater than the one described by the transmitted ones, it is likely that the bottleneck link is congested and cannot support the video bitrate of the session. When video suffering is detected, the system reacts by launching a QoS script (described in the next section) that protects the video session. The throughput evaluation has to start after that the buffering phase is finished. In our heuristic, this is detected by observing when the server stops transmitting for some seconds.

The heuristic relies on two vectors `rx[n]` and `tx[n]` where `n` is the discrete time in seconds. Figure 6 details this procedure. The variable `max_attempts` is introduced to be more conservative and avoiding to react in case of temporary congestion.

C. Corrective actions

After having realized that the video is suffering from a situation of congestion, the system should be able to react with some actions. At this point, it is possible to map our QoE problem to a simpler QoS one. In fact, we can control the bandwidth consumption on the link connecting the BRAS to the clients by means of proper scheduling and shaping algorithms applied at the network interface on that link. In essence, we can specify the bitrate to assign to a given streaming flow (which is dynamic and given by the `rx_thrgh` value described in the previous subsection) and

 ISBN: 978-1-61208-377-3

```
1: congestion = 0
2: Each second n:
3: if rx[n] >0 and rx[n − 1] == 0 then
4:     T ← n - n_0
5:     rx_thrgh = ( Σ_{i=n_0}^{n} rx[i])/T
6:     tx_thrgh = ( Σ_{i=n_0}^{n} tx[i])/T
7:
8:     if rx_thrgh >tx_thrgh then
9:         congetion++
10:
11:        if congestion >= max_attempts then
12:            launch QoS script
13:        end if
14:    else
15:        congestion = 0
16:    end if
17:    n_0 ← n
18: end if
```

Figure 6. Heuristic to detect video suffering

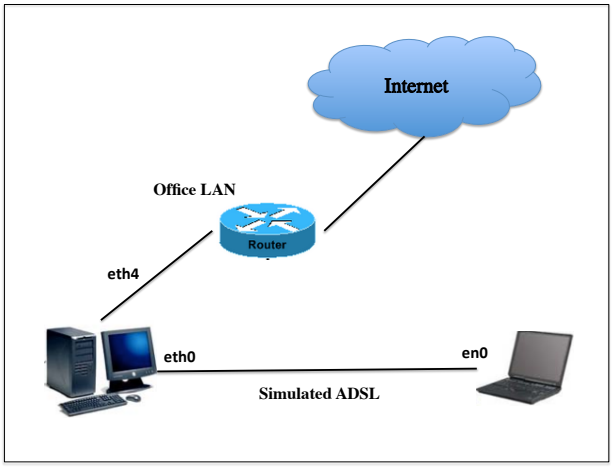


Figure 7. Laboratory environment

guarantee it through shaping and scheduling of the traffic. The algorithms for such a task are various and well-known. All have in common the definition of classes and a proper packet marking scheme, so that incoming packets can be put in the correct queue before being scheduled for transmission.

Although different solutions are possible, we adopted the Hierachical-Token-Bucket algorithm, a classful shaper and scheduler available in the Linux kernel that can efficiently provide bandwidth sharing. The required parameters are the IP addresses of the flows to control and the related rates to be assigned.

IV. EXPERIMENTS

A. Test environment

In order to evaluate our solution, we realized a test environment in our lab, emulating an ADSL scenario. Figure 7 depicts this testbed. The set of elements involved are the following:

- A laptop acting as client PC.
- A desktop PC running the Linux operating system, which emulates a BRAS node.
- Proper tools to emulate the last-mile ADSL link characteristics: *ipfw* and *dummynet* in the client-side machine and the Linux module *NetEm* in the network node.
- A tcp-splitter written in C language, *pepsal* [17], installed in the emulated BRAS.
- The Linux *Traffic Control (tc)* tool for QoS actions

A client browser running on the laptop automatically generates both YouTube traffic and competing file transfer flows while a *dummynet* script is used to limit the upstream bandwidth on the emulated last-mile link. The desktop PC emulating the BRAS is a GNU/Linux machine with two Ethernet cards, one connected to the client PC and the other to the Internet through the high-speed access connection available in our University. At this node, *dummynet* is used for limiting the downstream bandwidth on the emulated last-mile link,

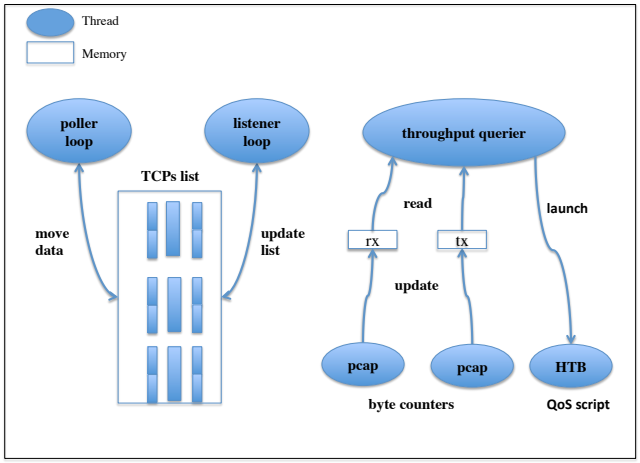


Figure 8. Main components of the tool

while a *tc* script is used for QoS actions.

We linked the TCP splitter, the heuristic video suffering detection, and the *tc* QoS script control in a unique, multithread program depicted in Figure 8. In essence, an ad-hoc thread is used for reading packets belonging to the streaming flows coming from the Internet toward the external interface of the splitter and outgoing to the client toward the internal interface. The cumulative number of bytes read from these two sniffers are saved in two global variable, `rx_bytes` and `tx_bytes`, and used by another thread that wakes up every second and stores in its memory space the instantaneous data rates. After reading the cumulative bytes from the interfaces, this thread resets the global variables to zero so that they become ready for the next sample. The sampling interval of 1 sec is selected as compliant with the YouTube dynamics, in particular with the idle time between chunks download in the steady state and the download time of each chunk, which involve some seconds.

 ISBN: 978-1-61208-377-3

TABLE I. EXPERIMENTAL RESULTS

	With tool	Without tool
No stalls	92%	0
Stalls	8%	100%

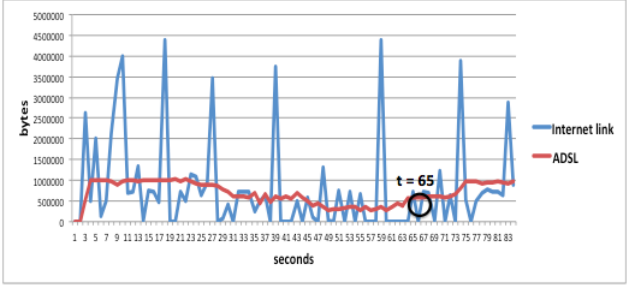


Figure 9. At istant t = 65 the process runs the QoS script

B. Results

In order to evaluate the effectiveness of our solution, we run some experiments considering several HD videos, which are the most bandwidth demanding and hence the most sensitive to the network congestion. In particular, we consider 50 different HD YouTube video sessions, established one at a time in the testbed. In each test, the selected HD video flow competes for the emulated ADSL link with additional file transfers. We analyzed the occurrence of video playback stalls with and without our tool running on the emulated BRAS. Table I reports on the results. First of all, we can see how congestion can significantly affect the perceived QoE of HD videos as at least one stall is observed in all the experiments when our tool is not active on the BRAS. Moreover, we can observe how our heuristic actually reacts to congestion situations in 92% of cases, avoiding stalls and thus increasing perceived QoE. For the sake of completeness, Figure 9 shows the video delivery pattern in a case where the video is suffering from congestion caused by two additional TCP connections; at instant $t = 65$ the reaction mechanism is triggered by our heuristic and it can be noted how the streaming transmission rate increases consequently, thus avoiding stalls.

For what concern the overhead caused by our tool, we have to consider that it is a splitter process working in user-space implementing the computation of the areas described by the download pattern over the time. The evaluation of the area is a simple sum of bytes counted every second. The major constraint is on the number of flows being able to analyze. Since the splitter works in user-space as a single process, the number of open sockets may be constrained by the machine operating system. Moreover, the main memory could be over-loaded before the splitter reaches the maximum number of opened connections. The limitation on the maximum number of opened sockets can be overcome by implementing the tool in the kernel-space. This might be considered for a possible commercial solution. The amount of resources involved depends on the number of flows to be tracked. In our tests we ran one session at a time and the resources consumed was negligible.

V. CONCLUSION AND FUTURE WORK

The paper investigates a possible solution for protecting the overall video quality of an HTTP streaming service under the scenario of congestion caused by competing and concurrent TCP flows. The developed tool exploits the particular traffic pattern of HTTP-based videos, which is maintained at the entrance of the bottleneck link if a TCP splitter is used. By comparing incoming and outgoing patterns, the tool is able to detect when the video is suffering and reacts by protecting video downloads, thus limiting stalling events at the client. Our experimental results showed the effectiveness of the proposed approach, which was able to avoid video stalls in 92% of the considered cases. Possible future work regards the improvement of our heuristic method to detect video suffering, for example by also considering the pattern of HTTP GET messages flowing in the opposite direction, and in particular the time spacing among them during the steady state in both congested and uncongested scenarios.

REFERENCES

[1] K. ur Rehman Laghari and K. Connelly, “Toward total quality of experience: A qoe model in a communication ecosystem.” IEEE Communications Magazine, vol. 50, no. 4, 2012, pp. 58–65.

[2] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service,” Network, IEEE, vol. 24, no. 2, March 2010, pp. 36–41.

[3] S. Jelassi, G. Rubino, H. Melvin, H. Youssef, and G. Pujolle, “Quality of experience of voip service: A survey of assessment approaches and open issues,” Communications Surveys Tutorials, IEEE, vol. 14, no. 2, Second 2012, pp. 491–513.

[4] R. Schatz, T. Hossfeld, and P. Casas, “Passive youtube qoe monitoring for isps,” in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, July 2012, pp. 358–364.

[5] H. Hu, X. Zhu, Y. Wang, R. Pan, J. Zhu, and F. Bonomi, “Qoe-based multi-stream scalable video adaptation over wireless networks with proxy,” in Communications (ICC), 2012 IEEE International Conference on, June 2012, pp. 7088–7092.

[6] M. Jarschel, F. Wamser, T. Hohn, T. Zinner, and P. Tran-Gia, “Sdn-based application-aware networking on the example of youtube video streaming,” in Software Defined Networks (EWSDN), 2013 Second European Workshop on, Oct 2013, pp. 87–92.

[7] M. Baldi and G. Marchetto, “Pipeline forwarding of packets based on a low-accuracy network-distributed common time reference,” Networking, IEEE/ACM Transactions on, vol. 17, no. 6, Dec. 2009, pp. 1936 –1949.

[8] M. Baldi, M. Corra, G. Fontana, G. Marchetto, Y. Ofek, D. Severina, and O. Zadedyurina, “Scalable fractional lambda switching: A testbed,” Optical Communications and Networking, IEEE/OSA Journal of, vol. 3, no. 5, 2011, pp. 447–457.

[9] M. Baldi and G. Marchetto, “Time-driven priority router implementation: Analysis and experiments,” Computers, IEEE Transactions on, vol. 62, no. 5, 2013, pp. 1017–1030.

[10] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, RFC 3135: Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, Internet Engineering Task Force, Jun. 2001.

[11] G. Paravati, V. Gatteschi, and G. Carlevaris, “Improving bandwidth and time consumption in remote visualization scenarios through approximated diff-map calculation,” Computing and Visualization in Science, vol. 15, no. 3, 2013, pp. 135–146.

[12] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. Munafo, and S. Rao, “Dissecting video server selection strategies in the youtube cdn,” in Distributed Computing Systems (ICDCS), 2011 31st International Conference on, June 2011, pp. 248–257.

[13] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “Youtube traffic characterization: A view from the edge,” in Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ser. IMC ’07, 2007, pp. 15–28.

 ISBN: 978-1-61208-377-3

[14] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "Youtube everywhere: Impact of device and infrastructure synergies on user experience," in Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 345–360. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068849

[15] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies, ser. CoNEXT '11, 2011, pp. 25:1–25:12.

[16] K. Takeshita, T. Kurosawa, M. Tsujino, M. Iwashita, M. Ichino, and N. Komatsu, "Evaluation of http video classification method using flow group information," in Telecommunications Network Strategy and Planning Symposium (NETWORKS), 2010 14th International, Sept 2010, pp. 1–6.

[17] C. Caini, R. Firrincieli, and D. Lacamera, "Pepsal: a performance enhancing proxy designed for tcp satellite connections," in Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd, vol. 6, May 2006, pp. 2607–2611.

 ISBN: 978-1-61208-377-3

Multiple Tree-Based Online Traffic Engineering for Energy Efficient Content-Centric Networking

Ling Xu, Tomohiko Yagyu

Cloud System Research Lab
NEC Corporation, Japan
Email: lingxu@gisp.nec.co.jp, yagyu@cp.jp.nec.com

***Abstract*—Content-Centric Networking (CCN) is a new network architecture aiming to solve many fundamental problems of existing IP networks. CCN is unique in that it widely deploys caches on routers to reduce redundant data transmission. Introducing caches into routers, however, causes the networks to consume extra energy. Although great research effort has been put for improving energy efficiency in IP networks, little work has been done for CCN networks yet. We designed an online traffic engineering algorithm called Multiple Tree-based Traffic Engineering (MTTE) to fill this void. Our approach is to make traffic flow on a small portion of edges in the network and shut down underutilized edges. Data are delivered on multiple tree-topology networks that are dynamically created based on the physical network topology and the current network traffic pattern. The trees are carefully constructed so that they contain minimal edges and mitigate congestion. Simulations using network topologies of real-world autonomous systems show that by using MTTE, up to 45% of the edges can be shut down. To the best of our knowledge, MTTE is the first online green mechanism for CCN.**

Keywords–Content-centric networking; energy efficiency; multiple trees; traffic engineering.

I. INTRODUCTION

Recently, *Content-Centric Networking* (CCN) has been drawing considerable attention from the industry and academic community [1]. In conventional IP networks, a great part of data transmission is redundant. CCN reduces redundant data transmission by deploying caches on all routers in the networks. A CCN network consists of hosts and routers. Each host holds contents, and each content has a unique *name*. Each name consists of a *prefix* and a *file name*. For example, the name "/Asia/Tokyo/music1.mp3" contains prefix "/Asia/Tokyo/" and file name "music1.mp3". For host p and its content d, p registers d's name on each router. This process is called *content publishing*, and p is known as the *producer* of d. Suppose that another host, say c, needs d. c sends an *Interest* i to its adjacent router. The Interest is then forwarded toward p according to certain forwarding policies. We refer to c as a *consumer* of d. Having received i, producer p packs d into *content object* packets, denoted as $co(d)$, and sends $co(d)$ back to c. Each router r along $co(d)$'s forwarding path tries to store d in its caches. The next time r receives an Interest for d, if d is still in its cache, r sends $co(d)$ to the consumer directly. In this paper, for ease of exposition, we do not differentiate hosts from routers. For host h and its adjacent router r, when h issues an Interest for content d, we simply regard r as the consumer of d; when h is a producer of d, we regard r as d's producer.

We are concerned about CCN's energy consumption. Networks consume a huge amount of energy, and improving their energy efficiency has been a hot research topic in recent years. CCN networks generally consume more energy than conventional IP networks since in-router caches cost extra energy.

Recent studies have evaluated CCN's energy efficiency [2][3]. However, no effective mechanism has been proposed for reducing CCN's energy consumption.

Although many energy-saving techniques have been proposed in IP networks, they cannot be readily transplanted into CCN networks. One of the general ideas for reducing network energy is to shut down underutilized edges [4]. Routers are the main hardware components and main energy consumers in networks. A router consists of a Central Processing Unit (CPU) and a set of network interfaces. Each interface connects an adjacent router via a piece of cable. Henceforth, we use *edges* to denote interfaces. The general idea for conserving energy is to shut down CPUs and edges, and the critical challenge is to shut down edges without remarkably reducing the system performance. In IP networks, one of the most commonly used techniques for this challenge is to (1) estimate the traffic matrix (i.e., the amount of traffic that will flow between each pair of edge routers) periodically, and (2) based on the estimated traffic matrix, find the edges whose shutdown minimizes performance degradation using linear programming [5]. In CCN networks, however, all routers can cache and provide all kinds of content. The traffic patterns are more complicated than in IP networks. Hence, estimating the traffic matrix is difficult.

Our objective is to design a mechanism that reduces CCN's energy consumption without remarkably reducing the network performance. To this end, we proposed *Multiple Tree-based Traffic Engineering* (MTTE). Previous research has found that edges in modern networks are generally underutilized [4]. Our idea is to shut down as many underutilized edges as possible. We split traffic on multiple tree-topology networks generated based on the physical network. The trees are generated in such a way that the number of edges included in the trees is minimized. Since trees generally contain fewer edges than the original physical network, energy can be conserved.

We face two challenges in our design. First, as fewer edges are used for forwarding traffic, the network is more vulnerable to traffic congestion. We assume that when a network system is heavily congested, most likely the congestion is caused

 ISBN: 978-1-61208-377-3

by a few bottleneck edges. To reduce congestion, we need to reduce the load on the congested edges. In MTTE, the network contains a centralized server called the *controller*. The controller dynamically monitors the congestion status of the system. When the system congestion is severe, the controller generates more trees so that the congestion on the bottleneck edges can be relieved. When the maximal utilization among all edges is low, the controller merges traffic on fewer trees and shuts down more edges.

The second challenge is that creating new trees potentially increases transmission latency. In a native CCN network, contents are delivered on the full physical network; in MTTE, contents are delivered on sub-networks. The (shortest path) distance between each pair of consumer and producer in MTTE is essentially greater than in native CCN network. Hence, the mean length of physical forwarding paths between consumers and produces in MTTE, denoted by the *Mean Forwarding Length* (MFL), is greater. Moreover, under the original CCN protocol, when a content is forwarded, all routers along the forwarding path will try to cache this content. As the MFL increases, the same content is likely to be cached on more routers. Previous studies have shown that repeatedly caching the same content reduces caching efficiency [6]. Hence, adding more trees naively is likely to further increase the system latency. We address this challenge by reducing the diameters of the trees.

We compare the performance of MTTE and native CCN under two metrics, *Live Edge Rate* (LER) and the system latency, via simulations running on real-world autonomous system topologies. Here, *live edges* are the edges contained in trees, and other edges are called *free edges*. LER is the ratio between the average number of live edges used during the simulation and the total number of edges in the physical network. The system latency is the average time between issuing Interests and getting content objects. Simulation results reveal that MTTE can shut down up to 45% of the edges while maintaining a latency comparable to CCN. Furthermore, under heavy congestion, MTTE can shut down 40% of the edges and achieve a latency 90% lower than CCN.

In short, we propose an online energy-saving mechanism for CCN which, to the best of our knowledge, is the first of its kind.

We detail the design of MTTE in Section III and evaluate MTTE's performance via simulation in Section IV. We highlight the relevant prior literature in this field in Section II and present our conclusion in Section V.

II. Related Work

In conventional IP networks, *Traffic Engineering* (TE) is a kind of well-researched mechanism that adjusts routing paths of traffic for relieving congestion, balancing traffic and reducing energy consumption [4]. MTTE is one of the few TE schemes designed for CCN networks. TE mechanisms can be implemented either offline [7] or online [8]. Offline TE mechanisms need to estimate the traffic matrix. Based on the traffic matrix, the TE protocols find the routing paths that minimize the energy consumption of the whole network using linear programming. In CCN, however, due to the wide existence of caches, the traffic between each pair of edge routers are more dynamic, and the traffic matrix would be hard to predict. In contrast, online TE mechanisms monitor the real-time traffic of the network and dynamically change routing paths according to the traffic fluctuation. MTTE is, to the best of our knowledge, the first online TE mechanism for CCN networks that both reduces energy consumption and relieves congestion.

Online energy-aware TE protocols for IP and MPLS networks have been proposed recently [9][10]. Vasić et al. assume that network edges work on multiple fixed energy consumption levels, and edges can switch to lower energy consumption levels when their utilization is low [9]. Each pair of routers maintain multiple routing paths. Each router periodically adjusts the traffic partition among its routing paths so that more edges can shift to lower energy consumption levels. Coiro et al. propose EATE, a distributed energy-aware architecture [10]. EATE is created above the MPLS protocol stack. Routers in the network run a modified OPSF-TE algorithm to periodically compute the shortest paths to each other based on the hop counts and the numbers of sleeping edges along the paths. When congestion occurs, nodes create new routing paths to bypass the congested area. However, these TE schemes are designed for connection-oriented architectures. Whether they can be readily used for CCN networks is unknown.

Chanda et al. implemented an online TE scheme for *Information-Centric Networking* (ICN) for balancing traffic [2] (CCN is a specific architecture implementation of the information-centric networking philosophy). The research objective of [2] is to demonstrate that TE mechanisms can be implemented more efficiently on CCN networks than on IP networks. Xie et al. [3] implemented a TE protocol in CCN with the goal of improving the caching efficiency, but how [3] would affect CCN's energy efficiency is unclear.

Research effort has been made in assessing CCN's energy efficiency using simulation [11][12]. Many of these researches compare the energy efficiency of CCN with existing IP-based content delivery techniques such as content delivery networking and peer-to-peer networking. Song et al. [13] noticed that in modern carrier networks, a great amount of traffic is generated from the edge. [13] uses GreenTE - an existing energy-aware TE mechanism designed for IP networks [4] - for reducing energy consumption of the core network, and uses CCN for eliminating redundant traffic generated by the edge network. However, Song. et al.'s approach does not reduce the energy consumption of CCN itself.

III. Multiple Tree-based Traffic Engineering

To reduce energy consumption, our idea is to shut down as many underutilized edges as possible. From an energy-saving viewpoint, delivering all contents on a single spanning tree would be the most favorable option. This extreme case, however, is impractical since a single tree is vulnerable to congestion and results in the high latency. Hence, we try to deliver the traffic on multiple trees to minimize edges contained in the trees and relieve congestion at the same time.

We use G to denote the whole network. In MTTE, G contains a central server called the controller. We use STs to denote the set of trees created in MTTE. Initially, the controller creates one spanning tree based on the physical network topology and adds this tree to STs.

A. Congestion Detection

The controller periodically measures system congestion status and based on this, decides whether to create new trees or not. Let us briefly explain why congestion and latency occurs.

 ISBN: 978-1-61208-377-3

In a standard network, for each router and each edge of this router, the edge maintains a packet *queue* of a fixed length and has a fixed *capacity*. The edge's utilization is the ratio between the speed at which packets come to this edge to the edge's capacity. When utilization > 1.0, new incoming packets are added to the queue's tail and await forwarding. We call the edges with utilization higher than ϕ_{CE} the *congested edges*, where we empirically set ϕ_{CE} to be 0.9. When packets traverse the congested edges, the system latency is likely to grow. Each router periodically reports the utilization of its adjacent edges to the controller. The controller calculates the *Congestion Rate* (CR) as the maximal utilization of edges in the networks. When $CR > \phi_{tcc}$, where ϕ_{tcc} is a system parameter which we empirically set to be 0.9, the controller creates one more tree and asks routers to deliver the traffic on each tree evenly. We call this mechanism the *Tree-based Congestion Control* (TCC).

B. Creating a New Tree

When the controller creates a new tree, we keep three objectives in our minds: (1) the new tree should introduce free edges into $E(STs)$ so that less traffic traverses the congested edges; (2) the new tree should not dramatically increase the live edge rate; (3) the new tree should not remarkably increase latency. Here, $E(STs)$ represents the set of live edges. Basically, the new tree is created using Kruskal's minimal spanning tree algorithm [14], while the three goals are realized by deciding which edges should be added into the new tree.

To realize goals (1) and (2), the controller assigns weights to edges so that the *uncongested edges* are chosen first, free edges are chosen later, and congested edges are chosen last. Here, uncongested edges are the live edges that have utilization lower than ϕ_{CE}.

We realize goal (3) by reducing the diameters of the trees. By doing so, we can reduce MFL and consequently, the system latency. We need to create a spanning tree st with a small diameter from the underlay physical network. Although theoretical research has been performed on creating minimal diameter trees [15], we use our own heuristic algorithm for simplicity. For each edge e, we compute its "edge betweenness" ($e.be$) - a widely used metric in graph theory [16]. Informally, $e.be$ represents the number of shortest paths in the entire network that traverse e. Imagine that routers c and p are a pair of consumer and producer, and sp is the shortest path between c and p on G. Intuitively, if more edges with high betweennesses are added to st, the probability that the data transmitted on st between c and p are delivered along the shortest path is higher. Accordingly, the diameter of st will be small. Based on this observation, in MTTE, the controller selects edges with higher betweennesses first. When the controller creates the initial tree, for each edge e, it calculates $e.eb$, and sets $e.weight = 1/e.eb$. When subsequent trees are created, the controller makes the weights of uncongested edges directly proportional to the edges' utilization, and makes the weights of free edges inversely proportional to the edges' betweennesses. If the new tree is different from all the existing trees, the controller adds the new tree into STs; otherwise, the network has no more capacity for mitigating congestion, the tree creation fails and the system stays unchanged. Namely, the system will not add trees permanently.

The complete tree-creation algorithm is shown in Figure 1.

1: E_{UE} = Edges in $E(STs)$ with utilization $<= \phi_{CE}$.
2: E_{CE} = Edges in $E(STs)$ with utilization $> \phi_{CE}$.
3: E_{free}: Edges not in $E(STs)$
4: U_{max} = maximum of edge utilization of E_{UE}
5:
6: The controller calculates the betweenness $e.eb$ of each edge e.
7:
8: **if** $|STs|$ = 0 **then**
9: // The controller is creating the initial tree
10: **for all** e in the network **do**
11: e.weight = $1/e.eb$
12: **end for**
13: **else**
14: // The controller creates a new tree for mitigating congestion
15: **for all** $e \in E_{UE}$ **do**
16: e.weight = e.utilization
17: **end for**
18: **for all** $e \in E_{free}$ **do**
19: e.weight = $U_{max} + 1/e.eb$
20: **end for**
21: **for all** $e \in E_{CE}$ **do**
22: e.weight = $U_{max} + 2$
23: **end for**
24: **end if**
25:
26: The controller generates a minimal spanning tree st using Kruskal ' s algorithm on the whole network.
27: **if** st is different from all the existing trees **then**
28: return st
29: **else**
30: return FAILED
31: **end if**

Figure 1. Based on current edge utilization, the controller generates a new spanning tree.

C. Hash-based Traffic Splitting

We briefly discuss CCN's packet (Interests and content objects) forwarding mechanisms. In CCN, each router r contains a *Forwarding Information Base* (FIB). Each FIB contains a set of entries and each entry is a mapping from one prefix to a set of network interfaces. To explain CCN's forwarding process, suppose that r's FIB contains two entries $fe0$="/Asia/":{face 2} and $fe2$="/Asia/Tokyo/":{face2,face5}, and suppose that an incoming packet has a name n="/Asia/Tokyo/music.mp3". r searches in FIB for the entry fe whose prefix matches n's prefix in the longest length. In this example, $fe = fe2$. r has multiple forwarding strategies. According to forwarding strategies, the incoming packets will be forwarded to one or multiple faces in fe.faces. How routers create their FIB entries and set their forwarding strategies is not standardized yet. In this paper, we suppose that routers create FIBs in such a way that packets are forwarded along one of the shortest paths between each pair of routers.

Each time STs is changed, routers update their respective FIBs so that packets can be delivered on the new STs. We split the hash name space of CCN names into $|STs|$ sub-name spaces, denoted by $NS[1], ...NS[|STs|]$. Packets with names whose hash values belong to $NS[i]$ will be forwarded on the i-th tree. Specifically, the controller sends both the topologies of trees and G to routers. For each producer p and each content

 ISBN: 978-1-61208-377-3

d stored on p, p publishes d. We suppose that H is a collision-proof hash function preloaded on each router. We use $N(d)$ to denote the CCN name of d. Producer p broadcasts $N(d)$ along the $(H(N(d)) \ mod \ |STs|)$-th tree. Suppose that two routers $r1$ and $r2$ are adjacent, during the broadcast, $N(d)$ traverses $r1$ first and then $r2$, and f is $r1$'s face that connects $r2$. Upon receiving $N(d)$, router $r1$ adds an entry $N(d).prefix : f$ in its FIB.

D. Tree Removal

When traffic in the network decreases, the controller shrinks STs and makes routers forward packets on fewer trees. That is, when $CR < \phi_{lowUtil}$, the controller removes off the last tree in STs and asks routers to update their FIBs. $\phi_{lowUtil}$ is a preloaded system parameter that we empirically set to be 0.6. Routers shut down their adjacent edges that are not included in $E(STs)$.

IV. Evaluation

This section evaluates MTTE's performance by comparing the system latency and LER between MTTE and native CCN. Our simulation is performed on ndnSIM – a simulation platform developed by UCLA for CCN-related research [17].

A. Performance Metrics

The system latency is calculated in the following manner. Each consumer r issues *Interest Issuing Frequency* (IIF) Interests for random contents per second. CCN's forwarding mechanism ensures that if r issues multiple Interests for the same content d before receiving the corresponding content object, finally r will receive no more than one content object of d. Upon receiving the content object, r calculates a *local latency* as the time interval since r issues the first Interest for d, until the time r receives the first content object of d. At any time point, we calculate the mean value of the local latencies of all consumers since the simulation starts by now as the system latency.

LER is defined as $\gamma/|E|$, where γ is the average number of live edges used in STs during the simulation, and E is the total number of edges in the physical network. We use latency(MTTE) and latency(CCN) to denote the latencies of MTTE and CCN, respectively.

B. Simulator Setting

Our simulations run on the network topology of autonomous system 3257 (AS3257). This topology is provided in Rocketfuel network dataset [18], a dataset that has been used in network research [19][20]. Each node in AS3257 represents a router. We extract the largest connected component of AS3257 and use all the remaining nodes for creating trees. AS3257 contains three types of routers: *cores*, *gateways* and *leaves*. According to the definition of Rocketfuel datasets, leaves are the routers with degrees equal to or less than two, gateways are the routers directly connected to the leaves, and the remaining routers are cores. The numbers of edges and routers in AS3256 are listed in Table I. We have also run simulations on other Rocketfuel autonomous system topologies and obtained consistent performance results.

We assume that in real world CCN networks, consumers are adjacent to leaves, and producers are adjacent to both gateways and leaves. In our simulation, we assign one producer to each leaf and each gateway, and assign one consumer to each leaf. Namely, totally 132 producers and 80 consumers are generated.

TABLE I. NETWORK PARAMETERS

Parameter	Value
Total number of edges	420
Total number of routers	240
Number of gateway routers	52
Number of leaf routers	80

Each producer generates ten random prefixes, and each prefix covers ten unique file names. Therefore, a total number of (producer count) $\times$ 10 $\times$ 10 names are generated.

In each second, each consumer issues IIF Interests with randomly selected names. The requested names are selected according to a Zipf distribution [21][22][6]: the k-th name is generated with a probability proportional to $1/k^{\alpha}$, where α is 0.7 in our simulations. Each simulation lasts 300 seconds. We evaluate the performance when the traffic is light (IIF=5) and heavy (IIF=15). The payload of each content object is 1024 bytes, the capacity of each edge is 10^6 bps, $\phi_{CE} = \phi_{tcc} = 0.9$, and $\phi_{lowUtil} = 0.6$. For each parameter setting, we repeat the simulation ten times and measure the average results. Parameters ϕ_{tcc}, ϕ_C and $\phi_{lowUtil}$ reflect the trade-off between transmission quality and energy efficiency. Generally, under the same traffic, more trees will be created and maintained when ϕ_{tcc} and $\phi_{lowUtil}$ are low. TCC will more aggressively choose free edges when ϕ_{CE} is low. In a real world CCN network, the network administrators can adjust the parameters themselves accordingly to the real needs of the system (high transmission quality or high energy efficiency).

C. Performance under High Traffic

Figure 2 compares the latency between MTTE and CCN when traffic is high (IIF=15). It shows that as times passes by, latency(MTTE) decreases and latency(CCN) increases. At the time point of second 300, MTTE shuts down 40% edges (Figure 3), and latency(MTTE) is 1/9 of latency(CCN).

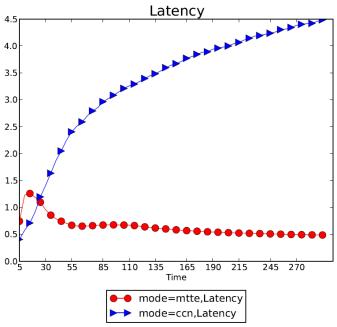


Figure 2. The comparison of the latency between MTTE and CCN when IIF=15. The horizontal and vertical axes represent the time the simulation has elapsed (in seconds) and the average latency (in seconds).

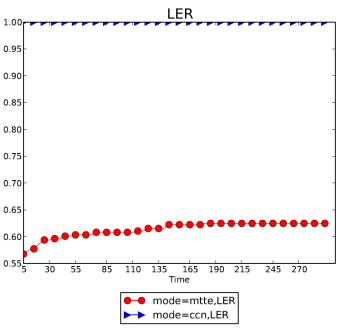


Figure 3. The comparison of LER between MTTE and CCN. The horizontal and vertical axes represent the simulation time and LER, respectively.

As argued in Section III, the system latency is mainly caused by the congestion on a few bottleneck edges (i.e., the congested edges). To validate this, we measure the mean and maximum of edge utilization over all edges (Figure 5 and Figure 6). The utilization of each edge is calculated as the ratio between the Exponentially Weighted Moving

 ISBN: 978-1-61208-377-3

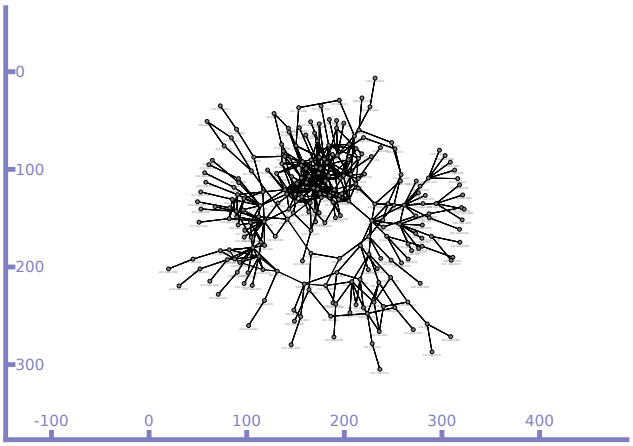


Figure 4. AS3257, the network topology used in user simulation.

Average (EWMA) of the traffic load on this edge to this edge's capacity. Theoretically, edge utilization ranges between 0.0 and 1.0. However, since the utilization is calculated in a EWMA manner, it may slightly exceed 1.0 when the network experiences congestion. In MTTE, the mean utilization is about 0.19, but CR is up to 1.0. We illustrate the topology of AS3257 in Figure 4. This figure also shows that several clusters exist in the network, where clusters are connected by a few edges. These edges correspond to the congested edges.

The reason that latency(CCN) increases with time is that the congested edges are overloaded, packet drop occurs, and consumers cannot receive the required contents. According to CCN's forwarding rules, the consumers will re-send their Interests, which makes the system even more congested and increases the system latency. The reason why latency(MTTE) decreases is that MTTE splits traffic onto multiple trees. As CR exceeds ϕ_{tcc} (Figure 6), new trees are generated (Figure 3). This reduces both the traffic on the congested edges and the Interest retransmission rate, which reduces the system latency accordingly.

In a CCN network, the mean edge utilization can be affected by the maximal routing capacity - the total capacity of the minimal cut of the routing paths [23], and the *Cache Hit Rate* (CHR). Generally, more traffic can be delivered and higher mean utilization can be achieved when the maximal routing capacity is high. Meanwhile, as routers deliver more traffic, the CHR increases (discussed in more detail in Section IV-D) and the mean edge utilization decreases. In MTTE, as more trees are created, the maximal routing capacity and hence the mean edge utilization increase. This trend can be observed at the early stage of the simulation (before second 30, Figure 5). As the maximal routing capacity of the overlay trees approaches the maximal capacity of the physical network, the increase in the mean utilization stops. On the other hand, routing paths in CCN and hence the maximal routing capacity do not change since the beginning of the simulation. The mean edge utilization of CCN generally decreases at the early stage of the simulation (before second 30, Figure 5), which is mainly attributed to the improve of the CHR.

In Figure 2, latency (MTTE) increases before second 10 and henceforth decreases. This is because before second 10, no sufficient trees are created. Packets accumulate on the bottleneck edges, which increases the delay. After that, as more trees are created, the congestion is mitigated and the delay decreases.

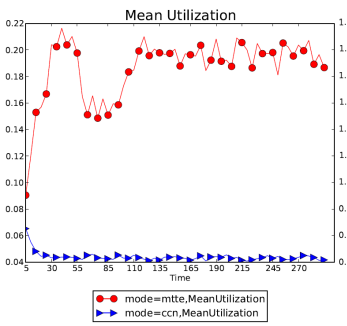


Figure 5. The comparison of the mean edge utilization (vertical axis; ranging between 0.0 and 1.0) between MTTE and CCN when IIF is 15.

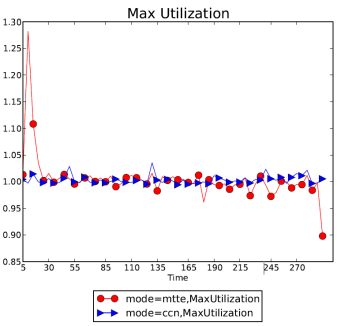


Figure 6. The comparison of the maximum of edge utilization among all edges (vertical axis) between MTTE and CCN when IIF is 15.

D. Performance under Low Traffic

Figure 7 compares the system latency between MTTE and CCN when traffic is low (IIF=5). Specifically, latency(MTTE) is roughly 18% higher than latency(CCN), and MTTE shuts down up to 45% edges (Figure 8).

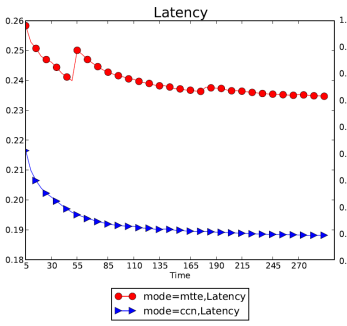


Figure 7. The comparison of the system latency (vertical axis; measured in seconds) between MTTE and CCN when the traffic is low (IIF is 5).

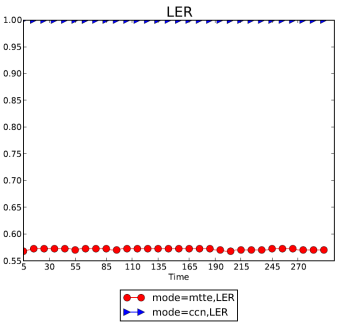


Figure 8. The comparison of LER (vertical axis) between MTTE and CCN when IIF is 5.

As the clock ticks, the system latencies of both MTTE and CCN decrease. To find out why, we measure the CHR. Suppose a router receives an Interest. If the content requested by this Interest is (not) in the router's cache, we say that the cache makes a (miss) hit. Each router records the number of hits (misses) its cache makes during the simulation as the *cache hit count* (*cache miss count*). Then, we calculate the CHR according to (1):

$$CHR = \frac{\text{mean cache hit count}}{\text{mean cache hit count} + \text{mean cache miss count}}. \quad (1)$$

As routers process more Interests, more popular contents are stored in the caches and the CHR increases (Figure 10). Accordingly, that the system latency decreases over time. Note that the CHR is calculated based on the traffic from the past five seconds. The delay is calculated based on the traffic since the simulation starts until the current time point. Hence, the converging speed of the delay is lower than the CHR, where the delay keeps slightly decreasing even when the CHR has largely turned stable at second 55.

The reason that latency(MTTE) $>$ latency(CCN) is that as stated in Section III, MFL(MTTE) is generally larger than MFL(CCN). To see this, we measure the mean hop counts

 ISBN: 978-1-61208-377-3

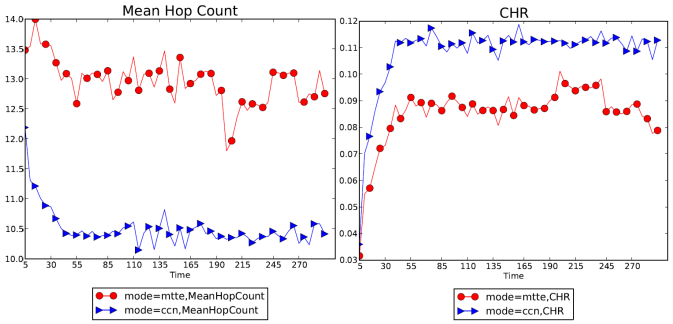


Figure 9. The comparison of the MHC (vertical axis) between MTTE and CCN when IIF is 5.

Figure 10. The comparison of the CHR (vertical axis; ranging between 0.0 and 1.0) between MTTE and CCN when IIF is 5.

(MHCs) of MTTE and CCN as indicators of the MFLs. The MHC is calculated as the average number of hops for content objects to return to consumers. The MHC is not equal to the MFL as the MHC is also affected by the CHR, but it should positively correlate to the MFL. Figure 9 shows that MHC(MTTE) can be 30% greater than MHC(CCN). As the MFL increases, the same content is cached on more routers, making CHR(MTTE) decrease as well. Figure 10 shows that CHR(MTTE) can be 20% lower than CHR(CCN). As the combined result of the high MFL and low CHR, latency(MTTE) is greater than latency(CCN).

E. Performance under Fluctuating Traffic

In order to evaluate the performance when the network experiences fluctuating traffic, we vary the IIF so that the IIF rides a sine wave. The wave shape of the IIF is shown in Figure 11. We expect to see that (1) TCC works correctly, i.e., MTTE adds trees when congestion is heavy and removes trees when network utilization is low, and (2) MTTE keeps both the latency and the LER low.

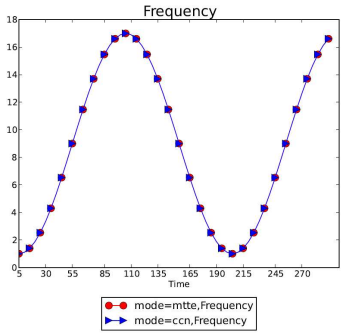


Figure 11. We make the IIF form a sine wave. The horizontal and vertical axes represent the simulation time and the IIF, respectively.

Figure 12 shows that generally, latency(MTTE) is much lower than latency(CCN). As the IIF increases, so does the congestion on bottleneck links. On one hand, MTTE dynamically creates and removes trees (Figure 14), and latency(MTTE) largely remains stable, which proves the effectiveness of TCC. On the other hand, in CCN, packet drop occurs and latency(CCN) increases as the IIF increases. As packet drop occurs, consumers re-send Interests, which makes the congestion deteriorate further. Latency(CCN) remains high even when the IIF peak is over. The peak of the IIF emerges

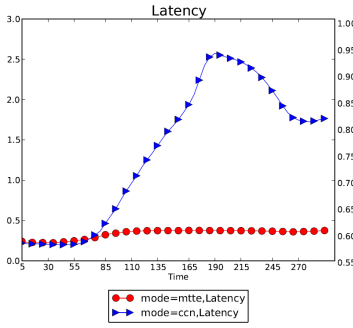


Figure 12. The comparison of the system latency (vertical axis) between MTTE and CCN when the IIF fluctuates.

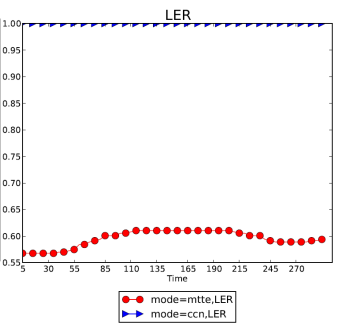


Figure 13. LER changes in MTTE and CCN as the IIF fluctuates.

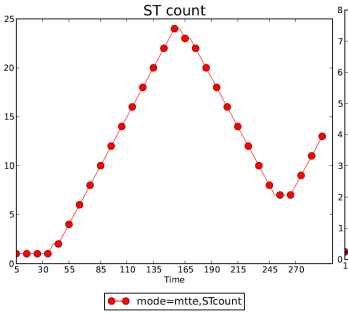


Figure 14. The change in tree count in MTTE. As the IIF fluctuates, so does the tree count.

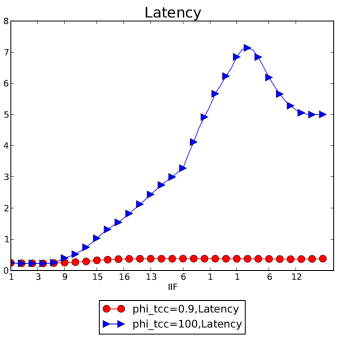


Figure 15. Comparison of latency(MTTE) (in seconds) when TCC is disabled and enabled under fluctuating traffic.

at the 100th second (Figure 11) while latency(CCN) does not start decreasing until the 190th second (Figure 12), meaning that it takes a long time for the network to completely transmit the Interests accumulated when the network was congested. All thought the simulations, MTTE shuts down up to 40% of edges (Figure 13). Since trees created in MTTE are heavily overlapped, the increase in LER(MTTE) is slight even thought the traffic remarkably surges. Figure 15 compares the latency when the TCC mechanism is disabled (by setting $\phi_{tcc} = 100$ so that no tree is created) and enabled ($\phi_{tcc} = 0.9$). We can clearly see that TCC effectively reduces the latency.

V. Conclusion and Future Work

CCN is a promising network architecture that provides many new possibilities. In this work, we concentrated on CCN's energy efficiency, which is a barely-explored but much-needed research topic. With the core idea of shutting down expendable edges, we have proposed a novel multiple tree-based architecture called MTTE, the first online green mechanism for CCN. Through simulation, we have shown that MTTE can shut down up to 45% of redundant edges, and achieve comparable, and in many cases superior, traffic transmission performance, compared to native CCN. As for future work, *we plan to improve MTTE's performance using more sophisticated tree generation algorithms, and evaluate the performance in larger-scale physical networks*. Meanwhile, we will design energy-saving mechanisms for CCN based on more accurate energy models. MTTE uses a centralized controller, which may incur scalability problems. Implementing it in a distributed manner is another future research topic.

 ISBN: 978-1-61208-377-3

ACKNOWLEDGMENT

The work described in this paper was, in part, performed in the context of the FP7/NICT EU-JAPAN GreenICN project.

REFERENCES

[1] L. Zhang et al., "Named data networking (NDN) project," Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 2010.

[2] A. Chanda, C. Westphal, and D. Raychaudhuri, "Content based traffic engineering in software defined information centric networks," Proc. IEEE NOMEN Workshop, 2013, 2013.

[3] H. Xie, G. Shi, and P. Wang, "TECC: Towards collaborative in-network caching guided by traffic engineering," in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 2546–2550.

[4] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering," in Network Protocols (ICNP), 2010 18th IEEE International Conference on. IEEE, 2010, pp. 21–30.

[5] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in IEEE International Conference on Communications, 2009. IEEE, 2009, pp. 1–6.

[6] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache less for more in information-centric networks," in NETWORKING 2012. Springer, 2012, pp. 27–40.

[7] J. C. C. Restrepo, C. G. Gruber, and C. M. Machuca, "Energy profile aware routing," in IEEE International Conference on Communications Workshops, 2009. ICC Workshops 2009. IEEE, 2009, pp. 1–5.

[8] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in ACM SIGCOMM Computer Communication Review, vol. 35, no. 4. ACM, 2005, pp. 253–264.

[9] N. Vasić and D. Kostić, "Energy-aware traffic engineering," in Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking. ACM, 2010, pp. 169–178.

[10] A. Coiro, M. Listanti, A. Valenti, and F. Matera, "Energy-aware traffic engineering: A routing-based distributed solution for connection-oriented ip networks." Computer Networks, vol. 57, no. 9, 2013, pp. 2004–2020.

[11] N. Choi, K. Guan, D. C. Kilper, and G. Atkinson, "In-network caching effect on optimal energy consumption in content-centric networking," in 2012 IEEE International Conference on Communications (ICC). IEEE, 2012, pp. 2889–2894.

[12] U. Lee, I. Rimac, D. Kilper, and V. Hilt, "Toward energy-efficient content dissemination," Network, IEEE, vol. 25, no. 2, 2011, pp. 14–19.

[13] Y. Song, M. Liu, and Y. Wang, "Power-aware traffic engineering with named data networking," in Seventh International Conference on Mobile Ad-hoc and Sensor Networks (MSN), 2011. IEEE, 2011, pp. 289–296.

[14] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," Proceedings of the American Mathematical society, vol. 7, no. 1, 1956, pp. 48–50.

[15] R. Hassin and A. Tamir, "On the minimum diameter spanning tree problem," Information processing letters, vol. 53, no. 2, 1995, pp. 109–111.

[16] M. Girvan and M. E. Newman, "Community structure in social and biological networks," Proceedings of the National Academy of Sciences, vol. 99, no. 12, 2002, pp. 7821–7826.

[17] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim: Ndn simulator for ns-3," Named Data Networking (NDN) Project, Tech. Rep. NDN-0005, Rev, vol. 2, 2012.

[18] "Rocketfuel dataset," https://github.com/cawka/ndnSIM-ddos-interest-flooding/tree/master/topologies/rocketfuel_maps_cch, (retrieved: September, 2014).

[19] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," ACM SIGCOMM Computer Communication Review, vol. 32, no. 4, 2002, pp. 133–145.

[20] C. Yi et al., "A case for stateful forwarding plane," Computer Communications, 2013, pp. 779–791.

[21] A. Ghodsi et al., "Information-centric networking: Seeing the forest for the trees," in Proceedings of the 10th ACM Workshop on Hot Topics in Networks. ACM, 2011, p. 1.

[22] S. K. Fayazbakhsh et al., "Less pain, most of the gain: incrementally deployable ICN," in Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 147–158.

[23] G. Dantzig and D. R. Fulkerson, "On the max flow min cut theorem of networks," Linear inequalities and related systems, vol. 38, 2003, pp. 225–231.

 ISBN: 978-1-61208-377-3

Network Policy-based Virtualization Controller in Software-Defined Networks

Ji-Young Kwak, SaeHoon Kang, YongYoon Shin, Sunhee Yang
Communications Internet Research Laboratory
Electronics and Telecommunications Research Institute
Daejeon, Korea
e-mail: {jiyoung, skang, uni2u, shyang}@etri.re.kr

Abstract— **As the diversity of emerging services has increased, the flow of traffic with various characteristics has occurred over the Internet. These various traffics require different treatments from carrier network in order to meet the Quality of Experience (QoE) requirements for end users. As the emerging technology satisfying these requirements, the Software Defined Networking (SDN) has the potential to enable dynamic configuration and control for the enhanced network management. This paper presents the design of a virtualization controller based on network policy in SDN environments. The proposed controller configures virtual networks in a flexible way for the deployment of various services by using virtual routers as the enabler of QoS policy enforcement. It can also perform dynamically the QoS control of flows, by using a network virtualization approach as a structure for enforcing the QoS and isolation policies of flows. With this approach, it enables various services to be deployed on multiple virtual networks and can achieve a better QoS.**

Keywords - QoS Path Control; Virtual Routing; Network Virtualization; SDN.

I. INTRODUCTION

Network technology has become an integral element in almost all area of human activity. The diversity of network services, as well as the number of available devices will continue to grow according to the increasing demands of customers [1]. However, the architecture of current Internet has reached its limit on carrying out various types of services. It is not suitable to fulfill all the network requirements, such as security, management, mobility and quality of service so that the diversification of coexisting network services can be accepted [2][3]. To overcome the weakness of current Internet, network virtualization technology is essential for the operation and design of future networks. Network virtualization enables the deployment of isolated logical networks over a shared physical infrastructure, so that multiple virtual networks can simultaneously coexist.

Recently, the Software Defined Networking (SDN) paradigm has emerged as an enabler for the network virtualization that automates the deployment and operation of programmable network slices on top of a shared physical infrastructure. The SDN provides flexibility in networking by introducing the concept of network abstractions in which the network control plane is decoupled from the data forwarding plane. This concept in SDN allows a centralized controller to control all switches on a per-flow basis [4], and to install the rules of packet treatment into switches by using a control protocol between controller and switches. The OpenFlow [5] describes actions to install per-flow rules into switches as the standardized protocol for signaling between switches and a controller. The SDN controller enables the deployment of arbitrary services in the constructed programmable virtual networks by slicing available network resources and controlling service flows among the network slices.

To obtain the optimal solution of network virtualization based on SDN, the various challenging issues associated with the composition of multi-tenant environments for security and QoS services should be taken into account. It should be possible to construct virtual networks dynamically according to the request of customers with minimal impact on the underlying physical infrastructure. The mechanism of flow isolation is also required to prevent that a misbehaving virtual network affects the performance of other unrelated virtual networks sharing the same network resources [6]. Furthermore, QoS isolation in virtual networks is a key feature for the deployment of virtual networks satisfying the QoS requirements of diverse services. Without a solution to these challenging issues, network virtualization will have its limitation in the deployment of diverse services.

To overcome this limitation, the present paper presents the design and implementation of a virtualization controller, which configures virtual networks in flexible way for the deployment of various services while hiding the complexity of networks in SDN environments. The proposed virtualization controller provides the scheme of an abstracted logical network to configure virtual networks in a simpler way and uses the concept of virtual router as an enabler for the enforcement and management of flexible network policies. In this scheme, it can offer specific services by simply configuring an abstracted logical network without exposing all details of the underlying physical topology. This simple access to virtual network is allowed by means of the abstraction of complicated tasks in network configuration. Network policies are set to a virtual router in this abstracted logical network to apply network behaviors associated with the connectivity and quality of network services. Thus, this virtualization controller can serve various services in multi-tenant networks by properly configuring such abstracted logical network and describing the requirements of network services to policies. These convenient features for the automatic configuration and management of virtual networks

 ISBN: 978-1-61208-377-3

are eventually provided to operators with this virtualization controller.

This paper is organized as follows. In Section 2, we discuss related works. In Section 3, we present the architecture of virtualization controller managing multi-tenants networks based on network policies. In Section 4, we describe technical details for automated provisioning of virtual networks as well as flexible flow isolation and control based on network policy. In Section 5, we demonstrate the feasibility of flexible flow control through the virtual router-based policy enforcement of implemented prototype. Finally, in Section 6 we conclude the paper with some suggestions for future work and research directions.

II. RELATED WORK

In recent years, SDN has emerged as an active area of research. Most SDN controllers offered a low-level programming interface based on the OpenFlow. Increasingly, recent SDN controllers have focused on supporting advanced features such as isolation, QoS provisioning and virtualization [7]. In this section, we briefly discuss solutions and technology related to the proposed SDN virtualization. Network virtualization refers to the ability to provide the end-to-end networking that is abstracted from the details of the underlying physical network by allocating shared resources efficiently. To support the on-demand provisioning of virtual networks, it is necessary to devise the way of unifying abstraction to enable the configuration of virtual networks in a flexible manner.

As one way to implement network virtualization, the recent solution building on an overlay approach is to use encapsulation and tunneling technologies (e.g., Virtual Extensible LAN (VXLAN) [8], Network Virtualization using Generic Routing Encapsulation (NVGRE) [9], and Stateless Transport Tunneling (STT) [10]). This approach is based on the mesh of IP tunnels connecting virtual switches on servers supporting the same tenant [11]. All tenant traffic is sent through the tunnels with different tunnel IDs in the encapsulation header, so that layer 2 traffic in the tenant can be isolated inside the IP tunnels. In particular, under the current SDN paradigm, an edge-overlay approach has been used on the basis of L2-in-L3 tunneling to achieve the network virtualization using traditional network hardware equipment. Among these solutions, Distributed Overlay Virtual Ethernet (DOVE) [12] is a proposal of network virtualization that supports isolation by creating an overlay network on the basis of VXLAN encapsulation and using the network identifier of DOVE header. However, the tunneling-based approach has some performance and compatibility problems because L2-in-L3 tunneling can cause performance degradation due to the IP fragmentation when performing an encapsulation [13]. Furthermore, this overlay approach has the limitation that can not control flexibly the flow of traffic or change directly the forwarding path of packets between different virtual networks in the data plane.

Another way for the network virtualization is to leverage the OpenFlow protocol so as to construct virtual networks based on the policy by allowing flows to map into the proper virtual network by means of the L1-L4 fields of a header. As the early technology of network virtualization for flow isolation, FlowVisor [14] enforces traffic isolation between slices by managing shared resources allocated among network slices. As a network virtualization layer based on SDN, Flowvisor is deployed logically between control and forwarding paths. It acts as a proxy controller between controllers and OpenFlow devices in the virtualization platform of OpenFlow network. Each controller is allocated to a network slice and controls its own slice. However, FlowVisor provides support for network slicing rather than network virtualization. One of the main limitations of FlowVisor is that virtual topologies are restricted to subsets of the physical topology. Furthermore, the deployment and operation of network slice using FlowVisor brings about configuration and planning overhead for operators.

In contrast, the proposed virtualization controller offers highly customized virtual networks when configuring the virtual network by taking into account the QoS required by the flows of service and the virtualization of infrastructure. It considers the high-level network logic as the set of services and policies through the abstraction architecture where network services and policies are decoupled from the mechanisms for low-level physical connectivity. The proposed controller has focused on the automatic policy-based service management to offer per-flow QoS control in a scalable and flexible manner while supporting critical requirements, such as isolation and ease of operation and configuration in a dynamic multi-tenant environment. It allows the logical network functions that range from the basic connectivity service to the advanced control service, such as QoS and security, by mapping QoS parameters, such as queues and rate limiters on resources available on OpenFlow switches.

III. THE ARCHITECTURE OF VIRTUALIZATION CONTROLLER

In this section, we introduce the architecture of a virtualization controller, which automatically configures and manages multi-tenants networks based on network policies. The abstraction mechanism supported by a virtualization controller allows operators to manage and modify virtual networks in a flexible and dynamic way. The abstraction components displayed to the operator are logical virtual networks and virtual routers while the topology of underlying physical networks is hidden to the operator by these abstraction mechanisms. The operator is able to configure a virtual network automatically according to a defined policy by describing the policy associated with network configuration on the abstraction component representing a logical virtual network. Through the proposed virtualization controller, a virtual router in a logical network manages the policies defining how traffics are handled in terms of the quality of network services and the connectivity between virtual networks. Thus, the topology of logical networks can be changed in a highly flexible and dynamic way according to the policies managed by these virtual routers.

As mentioned earlier, the policy-based approach of configuring virtual network supports the capability of

 ISBN: 978-1-61208-377-3

abstracting the complex management tasks associated with virtual networks and provides flexibility with respect to the management of network resources by using virtual routers. The proposed virtualization controller performs virtual-to-physical mapping and network control functions under the constraints of available resources, so as to isolate multiple logical networks on the shared physical infrastructure. Further, it has the complete view of whole physical topology in the environments of virtual networks formed by a centralized virtualization controller and the collection of distributed switches. Thus, the simple control operation on a virtual network can be translated into multiple actions on the physical control plane while performing the mapping and control functions for the virtual network on the corresponding physical network. By using the OpenFlow, it can dynamically install packet-handling rules to the corresponding switches according to the flexible network policies for the management of virtual networks through the distributed switches.

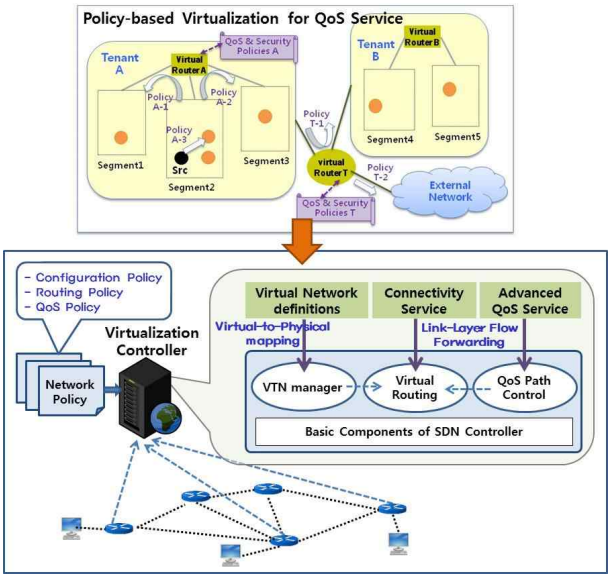


Figure 1. The virtualization controller based on network policy

As illustrated in the Figure 1, the proposed virtualization controller has an architecture where high-level network services and policies are fully separated from the low-level physical connectivity mechanisms. It deploys virtual networks based on configuration policy and sets up the entries of flow tables according to the requested routing and QoS policies, so as to enhance security and QoS in the virtual networks by flexibly controlling the flows of services. A configuration policy is used to create virtual networks, defining the members belonging to a logical group. In the virtual network that hides the details of an underlying physical topology, the routing policy describes the connectivity among logical groups and is used to decide which packets to drop or forward to specific egress ports based on the configuration policy. The QoS policy specifies which paths a flow should follow between the ingress and egress ports in order to minimize congestion or end-to-end latency. This architecture of the proposed controller allows for efficient and scalable performance and policy enforcement through the routing mechanism based on distributed virtual routers. Specifically, it can provide virtual networks in scalable and flexible way under constraint of the control capability by the centralized controller. Furthermore, it supports the simplicity of centralized policy definition and management for segments, tenants, and external networks.

IV. AUTOMATIC CONFIGURATION AND OPERATION OF VIRTUAL NETWORKS

In this section, we describe the abstraction mechanism in a virtualization controller presenting the topology of virtual networks formed by its sets of the partitioned or combined network resources with the separate view of network.

A. Automated provisioning of virtual tenant networks

The proposed virtualization controller creates logically isolated network partitions on shared physical infrastructures by allocating machines in a pool of computing resources to different groups which represent logical virtual networks. It creates tenant networks by grouping machines and configures logical network segments by separating and grouping machines within a tenant network. The virtualization controller provides the logical isolation necessary among the tenant networks or logical network segments created by means of the configuration method in abstracted logical networks specifying what machine is a member in a specific virtual network. Logical network segments are defined in a flexible manner by using different options according to packet information (MAC address, IP address or VLAN) or location information such as the switch and interface attached to a host machine.

The virtualization controller can create multiple virtual networks with virtual topology decoupled from the topology in physical infrastructure by introducing the abstraction mechanism for resource virtualization to aggregate multiple network resources. Thus, it can isolate traffic flows for an additional security or quality of service from multiple tenant networks by creating logical network segments flexibly.

B. Virtual router based policy enforcement for flow isolation

The proposed virtualization controller offers the function of virtual routing to control policy-based connectivity among logical network segments, tenant networks and external network by installing packet-handling rules according to specified network policy on the distributed OpenFlow switches. The function of virtual routing is performed to control connectivity and traffic patterns among logical networks through a set of virtual routers, which conceptually represent abstracted objects in multiple virtual networks co-existed across the infrastructure with OpenFlow switches.

Each tenant network has its own virtual router to manage policies for virtual routing and control the connectivity dynamically among logical network segments within a same tenant network. Being connected with different tenant routers, a system router is used to apply and manage the network policy to define routing rules associated with the QoS and

 ISBN: 978-1-61208-377-3

connectivity among tenant networks or between tenant network and external network. With this concept, the virtualization controller provides a set of distributed virtual routers that can be used to manage defined policies and control the connectivity among logical groups (logical network segments, tenant networks, external network). Accordingly, the function of virtual routing provides flexibility and ease of deployment through the distributed routing mechanism based on a set of virtual routers. Moreover, it performs scalable policy enforcement efficiently while preserving centralized policy definition.

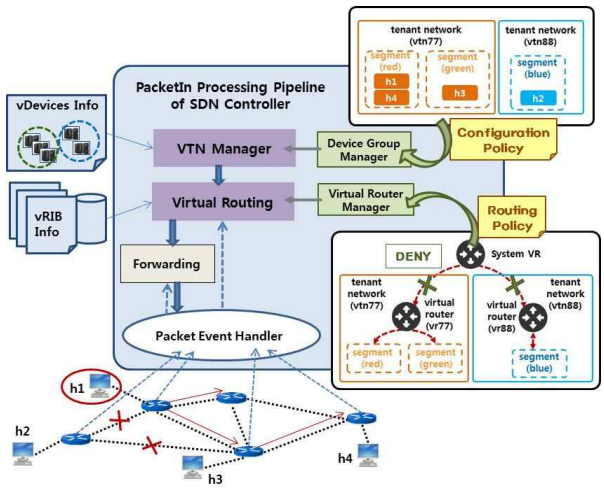


Figure 2. The policy-based flow processing for virtual routing

As shown in Figure 2, it is possible to create tenant routers, and add interfaces and routes with the tenant router. Once virtual routers and router interfaces have been created, router interfaces can connect to a system router or a logical network segment within the same tenant network. If an IP address/subnet mask is assigned to an interface connected with a logical network segment involving a default gateway with the same IP address, hosts within the logical network segment can communicate to other hosts in different subnet, which are connected with that interface.

Once the created virtual routers (vr77 and vr88) and interfaces are connected after the configuration of logical network segments (red, green, and blue segments), the virtualization controller can configure routing tables in virtual routers by specifying policy to describe routing rules (permit, deny). Thus, the tenant routers and the system router can control connectivity of logical groups by specifying routing rules over distributed virtual routers. At the request of incoming flows, it delivers the information of forwarding rules by using the OpenFlow into corresponding switches according to the policy specified in the virtual router, as shown in Figure 2. In accordance with the routing policy (deny) between vr77 and vr88 virtual routers, it installs the forwarding rules blocking traffics destined for a host h2 into corresponding switches, and then the flows over the links in the direction that are passed to a host h2 are dropped.

When a network policy is changed dynamically, the virtualization controller manages the flow of traffic among logical network segments according to the changed policy, by updating flow tables in the corresponding physical switches extracted from the virtual-to-physical mapping method. The function of virtual routing decides on hop-by-hop routing paths based on the specified end-point service policy, and constructs an effective set of forwarding rules that obey the defined policy under constraints of network resources. The constructed forwarding rules are automatically distributed, so as to be updated to forwarding tables in the corresponding switches. The virtualization controller can control the connectivity and traffic patterns among the logical groups in several different forms (logical network segments, tenant networks, external network) by the virtual routing mechanism based on a set of distributed virtual routers.

C. *Flexible resource allocation based on service policy*

An approach to support the diversity of network service is to configure multiple virtual networks on top of a shared physical infrastructure and then customize each virtual network according to specific purpose. Thus, the proposed virtualization controller offers separate virtual networks customized by the traffic type over the physical infrastructure shared among resources for each virtual network. To provide multiple virtual networks available for carrying different kinds of traffic, different QoS mechanisms are specified and then applied on distributed virtual routers. In the structure of these virtual networks, the classified traffic is distributed on multiple virtual networks in different directions depending on the specified policy by applying a proper QoS mechanism for transmitting traffic. Through this virtual router-based customization mechanism, the proposed controller offers flexible forwarding function for the flows of various QoS in order to satisfy the quality requirements corresponding to the certain type of service in each virtual network.

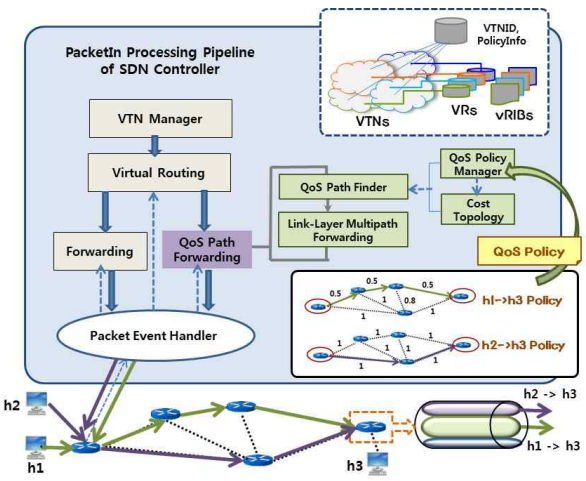


Figure 3. The control of QoS path for services deployment

QoS policy-based routing needs to identify end-to-end paths with enough resources to satisfy performance requirements in terms of metrics, such as loss, delay, the number of hops, and bandwidth optimization. As shown in

 ISBN: 978-1-61208-377-3

Figure 3, the proposed controller performs shortest path routing based on different costs applied on the forwarding paths for various services in virtual networks in terms of QoS requirements. These costs are decided by the proposed controller depending on the application characteristics or the available bandwidth or capacity in each links on the paths for transmitting service traffic. In the case of 'h1->h3 policy' for the service flow destined for a host h3 from a host h1, the costs of this policy are calculated in consideration of the capacity of links on each path due to the characteristics of real-time service, as shown in Figure 3. This mechanism for cost-based path computation is required to allocate optimal resources dynamically and guarantee customized end-to-end services to end users.

V. IMPLEMENTATION OF VIRTUALIZATION CONTROLLER

In this section, we demonstrate the enhanced network functionalities through an implemented prototype based on the design of a policy-driven virtualization controller. To validate the idea of this design, the prototype has been developed by the OpenFlow 1.0 protocol in SDN environments. The main goal of the experimental tests described in this section is to show how the proposed controller controls the flow isolation and connectivity as well as how it performs dynamic path control for the QoS in the multiple virtual networks. All tests were performed on the physical topology composed of OpenFlow switches in the mininet environments. With the screenshot of implemented prototype, Figure 4 depicts that the network policies applied through the REST API are translated to forwarding rules in the OpenFlow switches.

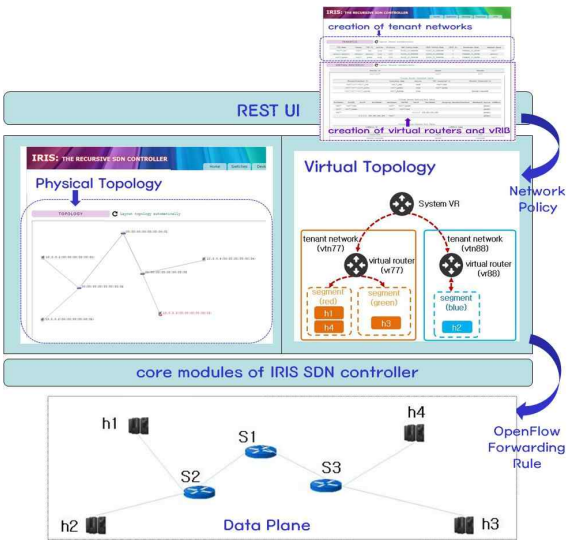


Figure 4. The prototype of virtualization controller

A. Path control of flows (isolation/connectivity control)

By the requests of REST APIs according to the configuration and routing policies, virtual networks are created and then routing rules are set on the virtual routers configured in the virtual networks. These routing rules for connectivity between virtual networks are translated to forwarding rules that are applied to the corresponding physical switches mapped with the virtual network.

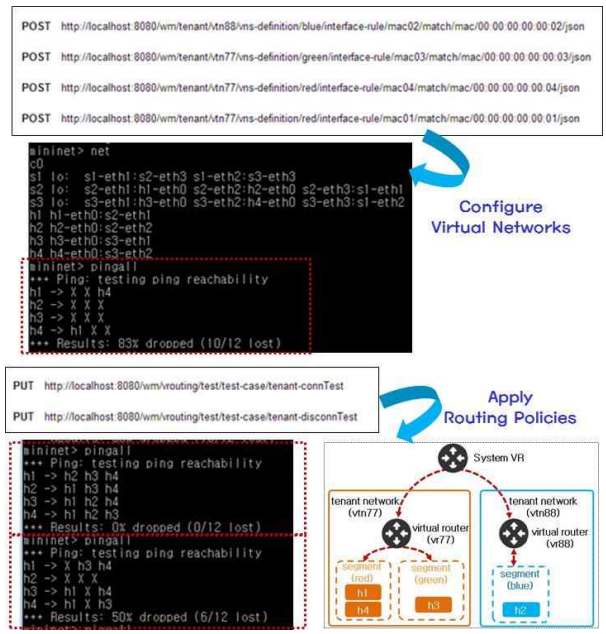


Figure 5. The results of virtual routing according to the network policy

Figure 5 shows the result of virtual routing after changing a routing rule for the connectivity of tenant networks between vtn77 (h1, h3, h4) and vtn 88 (h2).

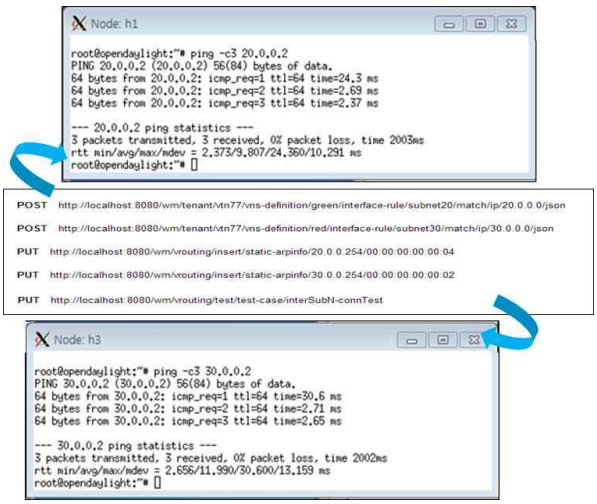


Figure 6. The virtual routing in the environments of different subnets

In Figure 6, we can see that it is also possible to control the connectivity between two hosts, h1 (30.0.0.2) and h3 (20.0.0.2), in the tenant networks with different subnets. Because two hosts in the different subnets are not in the same broadcast domain, they communicate with each other via their subnet gateway.

B. QoS control of flows (bandwidth control)

When there is a matching policy, every packet to the destination from the source is transmitted under the limited

 ISBN: 978-1-61208-377-3

resource depending on the matching QoS policy. In this testing environment, we have created two queues (q1, q2) and set the different bandwidths (20Mbps, 2Mbps) to each queue. Thus, flows will go into the different output queues according to the matching policies and be limited to different bandwidth rate. The controller chooses a higher priority policy when there are more than two policies that match the content of a packet.

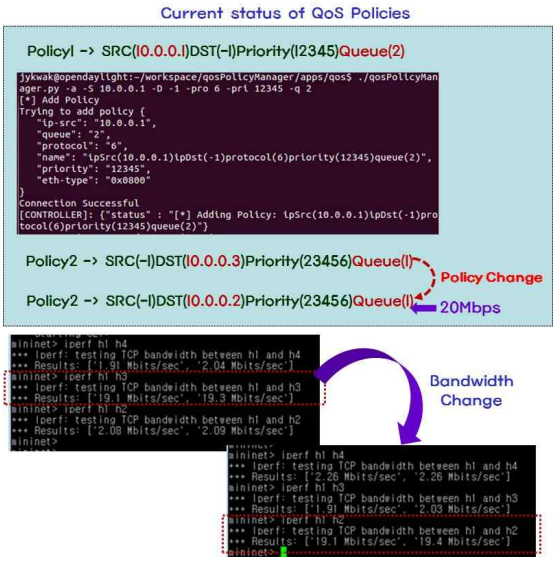


Figure 7. The bandwidth control of flows according to the QoS policy

Figure 7 shows the results of bandwidth test among hosts after changing the Policy2, which indicates that the matched packets will be limited to 20Mbps by the queue 1. As shown in Figure 7, when testing the bandwidth among a host h1 (10.0.0.1) and the other hosts (h2, h3, h4), the bandwidths of h1&h3 and h1&h2 were changed by the modification of Policy2 with higher priority than Policy1. When the Policy2 was modified in this experiment performing the QoS control, the information of destination host in the Policy2 was changed into host h2 (10.0.0.2) from host h3 (10.0.0.3). Thus, the bandwidth in h1&h2 flow increased to 20Mbps from 2Mbps by the change of Policy2. On the other hand, the bandwidth decreased from 20Mbps to 2Mbps in case of the h1&h3 flow because the policy of its flow was changed into Policy1 from Policy2.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed the design of a virtualization controller based on network policy. To deploy various services and achieve a better QoS, the proposed controller configures multiple virtual networks, which are customized with special goals on the same physical infrastructure. By the virtual router in multiple virtual networks, the traffics with different QoS requirements are distributed to the most suitable virtual networks for carrying a particular traffic. As a proof of concept, we have implemented the prototype of policy-driven virtualization controller in the software defined network composed of openflow switches. The implementation of a working prototype has demonstrated feasibility for the autonomic enforcement of QoS policies and the QoS control of flows. In the future work, we aim to improve the scalability of our prototype, so as to control the numerous flows occurring from numerous switches in a large network. Then, we will enhance our implemented prototype by applying optimal path selection mechanisms based on the multipath forwarding of link-layer.

ACKNOWLEDGMENT

This research was funded by the Ministry of Science, ICT & Future Planning (MSIP), Korea in the ICT R&D Program 2014.

REFERENCES

[1] A. Valdivieso, L. Barona, and L. Villalba, “Evolution and challenges of software defined networking,” in Proceedings of the 2013 Workshop on Software Defined Networks for Future Networks and Services, IEEE, November 2013, pp. 61–67.

[2] N. Fernandes et al., “Virtual networks: isolation, performance, and trends,” Annals of Telecommunications, vol. 66, Oct. 2011, pp. 339–355.

[3] P. Szegedi, S. Figuerola, M. Campanella, V. Maglaris, and C. Cervello-Pastor, “With evolution for revolution: Managing federica for future internet research,” IEEE Communications, vol. 47, no. 7, 2009, pp. 34–39.

[4] “Software-Defined Networking: The New Norm for Networks,” white paper, ONF, April 2012.

[5] N. McKeown et al., “OpenFlow: enabling innovation in campus networks,” ACM SIGCOMM Computer Communication Review, vol. 38, April 2008, pp. 69–74.

[6] J. Carapinha and J. Jim´enez, “Network virtualization: a view from the bottom,” in Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures. ACM, 2009, pp. 73–80.

[7] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker, “Composing Software Defined Networks,” In Proc. NSDI, Apr. 2013, pp. 1-14.

[8] M. Mahalingam et al., “VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks,” IETF Draft draft-mahalingam-dutt-dcops-vxlan-04.txt, May 2013, pp. 22.

[9] M. Sridharan et al., “NVGRE: Network Virtualization Using Generic Routing Encapsulation,” IETF Draft draft-sridharan-virtualization-nvgre-03.txt, Aug. 2013, pp. 17.

[10] B. Davie, Ed., and J. Gross, “A Stateless Transport Tunneling Protocol for Network Virtualization (STT),” IETF Draft draft-davie-stt-03.txt, Mar. 2013, pp. 19.

[11] J. Kempf, Y. Zhang, R. Mishra, and N. Beheshti, “Zeppelin - A third generation data center network virtualization technology based on SDN and MPLS,” CloudNet, Nov. 2013, pp.1-9.

[12] K. Barabash, R. Cohen, D. Hadas, V. Jain, R. Recio, and B. Rochwerger, “A case for overlays in dcn virtualization,” in Proceedings of the 3rd Workshop on Data Center-Converged and Virtual Ethernet Switching. ITCP, 2011, pp. 30–37.

[13] R. Kawashima and H. Matsuo, "Performance Evaluation of Non-tunneling Edge-Overlay Model on 40GbE Environment," Proc. NCCA, 2014, pp.68-74.

[14] R. Sherwood et al., "FlowVisor: A Network Virtualization Layer," OpenFlow Switch Consortium, Tech. Rep., October 2009.

 ISBN: 978-1-61208-377-3