A Scalable and Dynamic Distribution of Tenant Networks across Multiple Provider Domains using Cloudcasting

Kiran Makhijani, Renwei Li and Lin Han

Future Networks, America Research Center
Huawei Technologies Inc., Santa Clara, CA 95050
Email: kiran.makhijani, renwei.li, lin.han@ {huawei}.com

***Abstract*—Network overlays play a key role in the adoption of cloud oriented networks, which are required to scale and grow elastically and dynamically up/down and in/out, be provisioned with agility and allow for mobility. Cloud oriented networks span over multiple sites and interconnect using Virtual Private Network (VPN) like services across multiple domains. These connections are extremely slow to provision and difficult to change. Current solutions to support cloud based networks require combination of several protocols in data centers and across provider networks to implement end to end virtual network connections using different overlay technologies. However, they still do not necessarily meet all the above requirements without adding operational complexity or without new modifications to base protocols. This paper discusses a converged network virtualization framework called Cloudcasting, which is a single technology for virtual network interconnections within and across multiple sites. The protocol is based on minimal control plane signaling and offers a flexible data plane encapsulation. The biggest challenge yet for any virtual network solution is to distribute and inter-connect virtual networks at global scale across different geographies and heterogeneous infrastructures. Data center operators are faced with the predicament to re-design networks in order to support a specific virtualization approach. Cloudcasting technology can be easily adopted to interconnect or extend virtual networks with in a massive scale software defined data centers, campus networks, public, private or hybrid clouds and even container environments with no change to physical network environment and without compromising simplicity.**

Keywords–Network Overlay; Network Virtualization; Routing, Multi-Tenancy Virtual Data Center; VXLAN; BGP; EVPN.

I. INTRODUCTION

The Cloud adoption continues to grow; there is an upward trend of applications and services being built in platform independent manner and the scope of connectivity is no longer limited to a single site or a fixed location. As the cloud based applications evolve, the isolated operation and management of tenant networks (sharing common network access) that host these application becomes extremely complex and is different than the underlying physical networks. While infrastructure networks focus on delivering basic functions to ensure that the physical links are reliably available and reachable; the tenants concern with mechanisms to allocate and/or withdraw resources on-demand from different sites and network resource pools. The leading requirement for tenants is to use the networks in the most economical manner and still have sufficient resources available when needed.

The above mentioned motivation was first mentioned in the original Cloudcasting paper [1], which described a network virtualization framework that addresses many shortcomings of existing solutions. The present paper expands on concepts described in the original paper and covers details about prototype experiences, applications and advanced concepts of using Cloudcasting. Since the original work, we have observed that the Cloudcasting architecture applies to almost all virtualization scenarios and can be considered as a generalized framework for infrastructure indepedent virtual networking. The later sections of this paper further validates our observation.

The key characteristics of Cloud-oriented network architectures are resource virtualization, multi-site distribution, scalability, multi-tenancy and workload mobility. These are typically enabled through network virtualization overlay technologies. Initial network virtualization approaches relate to layer-2 multi-path mechanisms such as, Shortest Path Bridging (SPB) [2] [3] and Transparent Interconnection of Lots of Links (TRILL) [4] to address un-utilized links and to limit broadcast domains. Later, much of the focus was put into the data plane aspects of the network virtualization, for example, Virtual eXtensible Local Area Networks (VXLAN) [5], Network Virtualization using Generic Routing Encapsulation (NVGRE) [6], and Generic Network Virtualization Encapsulation (GENEVE) [7]. These tunneling solutions provide the means to carry layer-2 and/or layer-3 packets of tenant networks over a shared IP network infrastructure to create logical networks. Though, due to their lack of corresponding control plane schemes, the overall system orchestration and configuration becomes complex for virtual network setup and maintenance [8].

Even more recently, MultiProtocol-Border Gateway Protocol (MP-BGP) based Ethernet VPN (EVPN) [9] has been proposed as a control plane for virtual network distribution, and has foundations of the VPN style provisioning model. This requires additional changes to an already complex protocol that was originally designed for the inter-domain routing. The deployment of MP-BGP/EVPN in data center networks also brings in corresponding bulky configurations, for example, defining Autonomous System (AS), that are not really relevant to the data center infrastructure network. The solutions like TRILL, SPB and MP-BGP are a class of virtual network architectures that consume data structures of physical (substrate) network protocols, therefore, we refer to them as Embedded Virtual Networks. The term substrate network henceforth will be used to describe a base, underlying, or an infrastructure network upon which tenant networks are built as virtual network overlays. Whereas Cloudcasting protocol is referred to as Extended Virtual Network because it inter-connects different types of virtual networks through its own routing scheme. It can be organized over any substrate network topology and routing arrangements. As a note to the reader, with in the scope of this document, a virtual, customer or tenant network are used interchangeably and mean the same. A cloud is a location and infrastructure network. A virtual network is an entity that shares physical network resources and access with other similar entities; virtual networks are isolated from each other. In the context of this paper agility is understood as being able to responds to the changes in virtual network in real time or as quickly as needed to best serve the customer experience. Whereas elasticity refers to an ability to grow or shrink resource requirements on-demand.

Even though Embedded Virtual Network (the term is inspired from [10]) solutions mentioned above are quite functional, they are faced with several limitations. Of which the most significant and relevant to cloud-scale environments is their dependence on the substrate networks. In addition to being scalable and reliable, a cloud scale network must also be elastic, dynamic, agile, infrastructure-independent, and capable of multi-domain support. There has not been any converged architecture for network virtualization yet. In [1], we proposed Cloudcasting, an Extended Virtual Network framework that operates on top of any substrate network and offers primitives for cloud auto-discovery, dynamic route distribution as needed. As an extension to original paper, several operational concepts have been described. We have provided details of the prototype but most important section deals with the scalable distribution of virtual networks across geographically remote sites.

The rest of the paper is organized as follows. We have kept Section II and III intact from original paper to introduce the reference model and its major functions. Section IV explains different deployment scenarios where cloudcasting applies to. While Section V discusses scalability at global level of the solution, Section VI introduces the Cloudcasting policy framework where in constraints on virtual networks may be specified. Section VII has the qualitative analysis and implementation details and in Section VIII comparison with a few most common already existing solutions are made. Lastly, Section IX briefly lays out an interesting extension of cloudcasting combining services and mobile networks.

II. Cloudcasting Model

A converged virtual routing scheme can be described by two primary factors; an infrastructure-independent virtual network framework, and a unified mechanism to build an overlay of various types of tenant networks with different address schemes. On these basis, a new virtual routing scheme called Cloudcasting, is proposed with the following characteristics

1) ***Auto discovery:*** A signaling scheme that enables us to add, delete, expand and virtualize a tenants network with minimum configuration.
2) ***Auto distribution:*** A signaling scheme that connects multiple virtual networks with each other or asymmetrically as needed.
3) ***Auto Scale:*** The ability to provide and serve high scale of tenants in a location-agnostic manner.

A cloudcasting network is an IP network, which is shared and used by multiple tenant clouds to route traffic within a single virtual network or between different virtual networks. We use the terminology of tenant cloud to emphasize that a tenant or

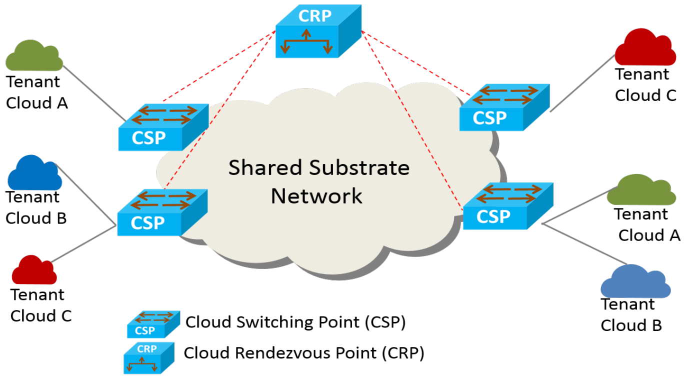

Figure 1: Cloudcasting Reference Model.

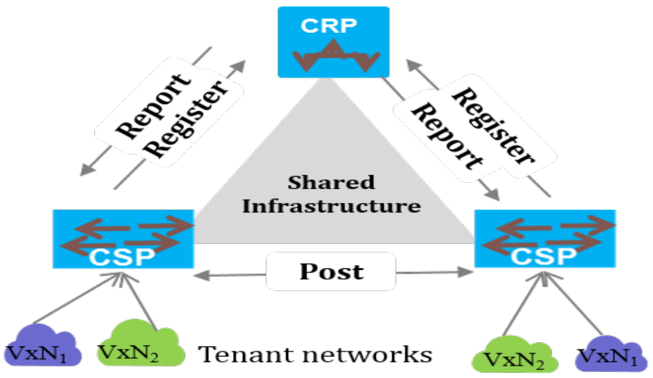

Figure 2: Cloudcasting Framework.

a user network may reside anywhere on the substrate network with a highly dynamic routing table. The IP address space in one tenant cloud may overlap with that in another cloud and these are not exposed to the shared IP infrastructure network. The cloudcasting reference model, is shown in Fig. 1. Each customer has its own network shown as Tenant Cloud A, B and C, a shared substrate IP network that was built independently and can encompass multiple administrative domains. This model describes a centralized conversational scheme, in which tenant clouds or Virtual Extensible Networks (VXNs) announce their presence as well as membership interests to a centralized designated authority, called Cloudcasting Rendezvous Point (CRP), via a cloudcasting network virtualization edge element called Cloudcasting Switching Point (CSP). To communicate among the network elements, a new signaling protocol, called CloudCasting Control (CCC) protocol is defined with three simple primitives facilitating cloud auto-discovery and cloud route distribution. The protocol primitives are defined as below and are further illustrated in Fig. 2.

- *Register message*: A virtual network interest and self-identifying announcement primitive from CSP to CRP.
- *Report message*: A response from CRP to all CSPs with similar virtual network interests.
- *Post message*: A CSP to CSP virtual network route distribution primitive.

The details of aforementioned cloudcasting network elements and their properties in cloudcasting framework are discussed as below.

A. Virtual Extensible Network

A Virtual Extensible Network is a tenant cloud or a user network. It is represented by a unique identifier with a global significance in cloudcasting network. Using this construct, it is possible to discover all its instances on the substrate IP fabric via CRP. VXN identifiers are registered with CRP from CSPs to announce their presence. There are various possible formats to define the VXN, for instance, an alphanumeric value, number or any other string format. In the preliminary work we have defined it as a named string that is mapped to a 28-bit integer identifier, thus enabling support for up to 256 million clouds.

B. Cloud Switch Point

A Cloud Switch Point is a network function that connects virtual networks on one side to the substrate IP network on the other side. It can be understood as an edge of a virtual network that originates and terminates virtual tunnels. A CSP holds mappings of L3 routes or L2 MAC forwarding information of a virtual network. A CSP is cloudcasting equivalent of a Virtual Tunnel End Point (VTEP) [5] in VXLAN networks or an Ingress/Egress Tunnel Router (xTR) in the LISP domain [11] and may similarly be co-located with either on a service providers edge (PE) router, on a top of rack (ToR) switch in a data center, or on both. A CSP participates in both auto-discovery and auto-route distribution. In order to establish a forwarding path between two endpoints of a virtual network or of two different virtual networks, a CSP first registers with the CRP its address and VXN identifiers it intends to connect to. Then the CRP will report to all CSPs that have interest in same VXN. Finally, the CSP will communicate with those other CSPs and exchange their routing information. On the data forwarding plane, a CSP builds a virtual Forwarding Information Base (vFIB) table on per VXN basis and route/switch traffic to the destination virtual networks accordingly.

C. Cloud Rendezvous Point

A Cloud Rendezvous Point is a single logical entity that stores, maintains and manages information about CSPs and their VXN membership. The CRP maintains the latest VXN to CSP membership database and distributes this information to relevant CSPs so that they can form peer connection and exchange virtual network routes automatically. A report message is always generated whenever there is a change in the virtual network membership database. However, CRP is oblivious to any change in vFIB (described above in CSP).

III. Cloudcasting Communication Primitives

Now, we describe cloudcasting communication primitives used among CRP and CSPs. Fig. 3 illustrates the layering of the virtual routing over any substrate layer and overlay control messages between CSP and CRP. The encapsulation message format is shown above in Fig. 4. A predefined TCP destination port identifies the cloudcasting protocol and CCC header contains the specification for the register, report and post messages.

A. Cloudcasting Register Message

An auto-discovery of virtual networks involves two messages. The first message is the Cloudcasting Register that originates from CSPs to announce CSP is interested in a VXN with the CRP. A Register message specification includes the CSP address and list of VXNs it is interested in. An interest is defined as an intent to participate in a specific virtual network. For example, a vxn_{red} on csp_1 expresses interest to join vxn_{red} on csp_2. As an example, consider virtual networks vxn_{red} and vxn_{green} are attached to csp_1. Then, the register message contains a tuple as follows

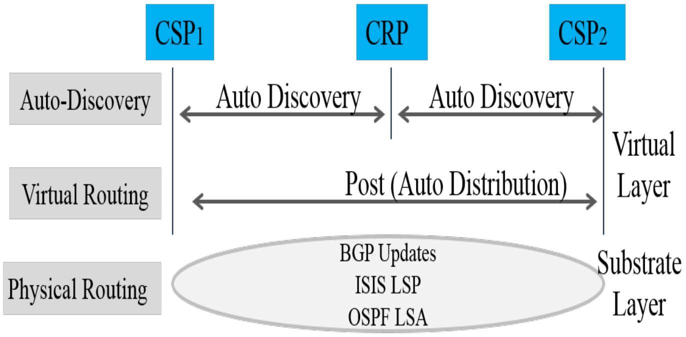

Figure 3: Cloudcasting Protocol Primitives

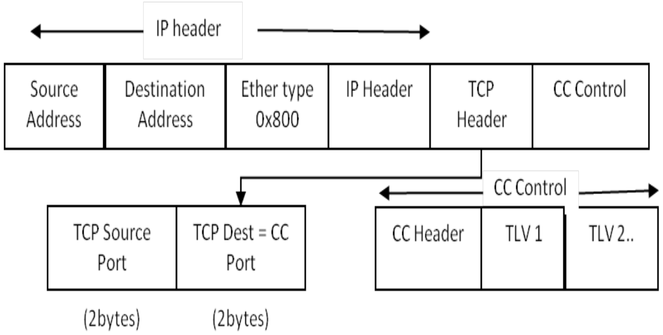

Figure 4: Cloudcasting Control Message Format

$$\textbf{\textit{Register}}\{sender\text{: } csp_1,\ [vxn_{red},\ vxn_{green}]\}$$

After the CRP receives a cloudcasting register message, it scans its CSP membership database to look for the same VXN identifiers. If it finds one (or more), a cloudcasting report message is generated and sent to all the CSPs with same interest, otherwise, it simply logs the VXN in its CSP database.

B. Cloudcasting Report Message

The CRP generates cloudcasting report messages in response to a cloudcasting register message to inform CSPs of other CSPs address and their associated VXN identifiers. If the CRP finds other CSP(s) with the same VXN membership (or interested VXNs), then the Report messages are generated for that CSP as well as the other found CSPs. A Report message is sent to each CSP, that contains other CSP addresses for the shared VXNs. As an example, consider CRP already has csp_2 with interest in vxn_{red}. Upon receiving a cloudcasting register message from csp_1 as described earlier, two report messages are generated as below for csp_2 and csp_1, respectively:

$$\textbf{\textit{Report}}\ (csp_2)\ \{to\text{: } csp_1,\ [interest\text{: } vxn_{red}]\}$$
$$\textbf{\textit{Report}}\ (csp_1)\ \{to\text{: } csp_2,\ [interest\text{: } vxn_{red}]\}$$

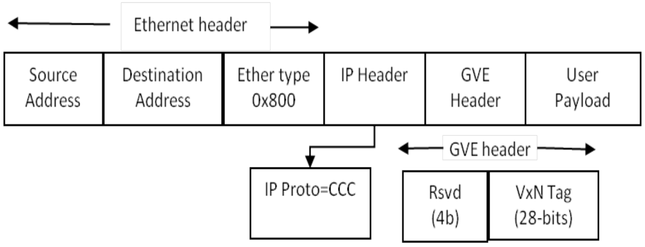

Figure 5: Cloudcasting GVE Protocol Encapsulation

In this manner, auto-discovery of virtual network locations is accomplished that is based on interest and announcement criteria.

C. *Cloudcasting Post Message*

The cloudcasting post messages facilitate route distribution as needed. As a cloudcasting report message is received, the CSP will connect with other CSPs to exchange their routing information that includes VXN identifiers, a Generic VXN encapsulation (GVE) tag and the network reachability information within the VXN along with the address family. The list of network reachability information type includes but not restricted to IP prefixes (such as, IPv4, IPv6), VLANs, MAC addresses or any other user defined address scheme. As an example, when a report as described earlier is received, the following Post will originate from csp1. Post (csp1, csp2) vxnred, gve: i, [AF: IPv4, prefix list] In the example above, it is shown that csp1 sends a post update to csp2 stating that vxnred will use encapsulation tag i; and that it has certain ipv4 prefixes in its IP network. The routing (network reachability) information has the flexibility to support various address families (AF) defined by Internet Assigned Numbers Authority (IANA) as well as certain extensions not covered under the IANA namespace.

D. *Cloudcasting Transport - Generic VXN Encapsulation*

In a cloudcasting network, all network devices will work exactly the same as before on the data plane except the Cloud Switch Points (CSP). A CSP will perform encapsulation and decapsulation by following the VXN vFIB table. A VXN vFIB table includes the routing information for a virtual network on a remote CSP where a packet should be destined to. The route information was learned by exchanging Post messages between CSPs.

The format for VXN encapsulation is shown in Fig. 5 above in which IP protocol is set to GVE and following IPv4 header is the 32-bit GVE-header. If and when Cloudcasting dataplane is adopted by IETF, the protocol number for GVE will be assigned by IANA.

IV. Use Cases

The cloudcasting architecture can be used to deploy tenant networks under many different scenarios. As the cloud based architectures become more prevalent, it will be far more efficient to use a single virtualization technology (at least in control plane) both within a site and for interconnection across multiple sites. The cloudcasting protocol can be deployed for the following use cases

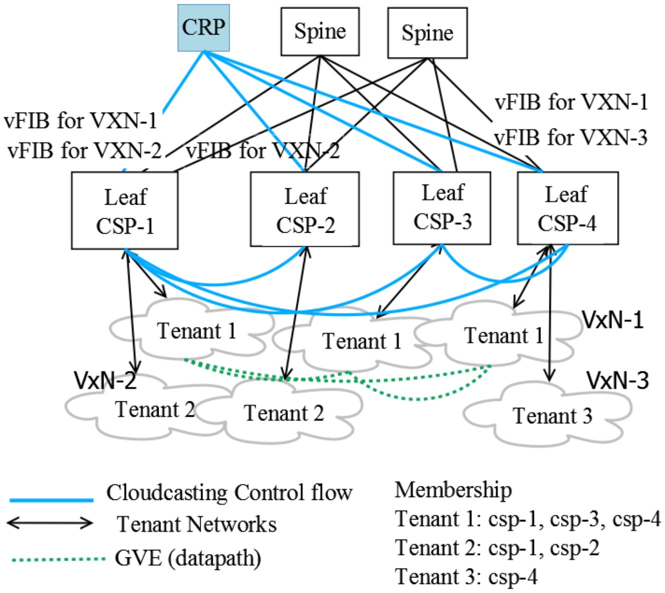

Figure 6: Cloudcasting Enabled Deployment

1) Multi-Tenancy Virtualized Data Centers
2) Multi-Site Interconnection of Data Centers
3) Interconnection of Hybrid Clouds
4) VPN Accesses in service provider environments

In the following sections, these deployments are described in more details, note that the same concept is easily extensible to any environment that requires infrastructure network to provide connectivity for tenant networks.

A. *Cloudcasting in virtualized data center*

Fig. 6 shows a cloudcasting-enabled virtualized data center. As discussed earlier in Section I, the CRP is a logically centralized node that is accessible by all the CSPs.

A leaf-spine switch architecture is used as a reference to explain cloudcasting deployment. A plausible co-location for CRP could be with the spine node, however, it may be anywhere in the substrate network as long as CSPs can reach it with the infrastructure address space. In Fig. 6, several tenant networks are shown as connected to different CSPs and CSP function itself is co-resident with the leaf switches. Each CSP has a virtual FIB table for both encapsulation and decapsulation of traffic along with the tenant network to CSP memberships (dynamically learned through auto-discovery).

The cloudcasting control protocol flow is shown in lighter color lines between CRP and CSPs and among CSPs.

At the bottom of Fig. 6 only the logical GVE data path tunnels with dotted lines for tenant 1 on CSP-1, CSP-3 and CSP-4 are shown.

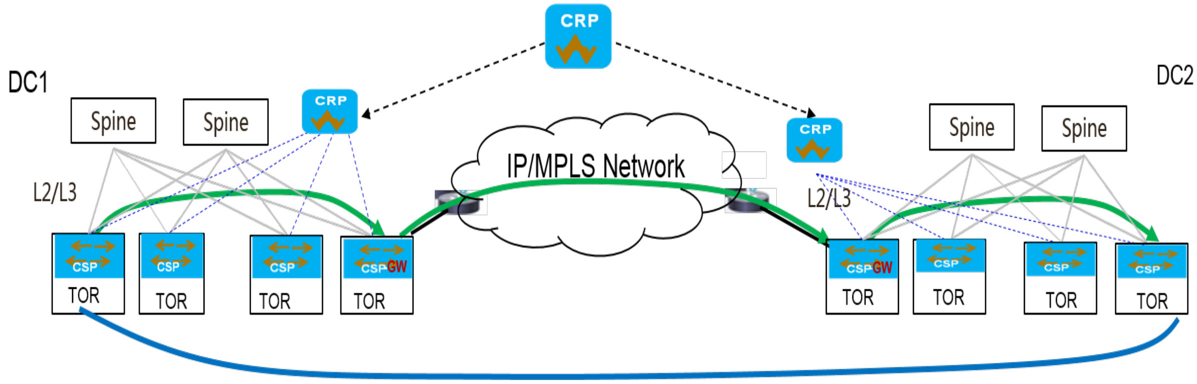

Figure 7: Cloudcasting with Data Center Interconnect

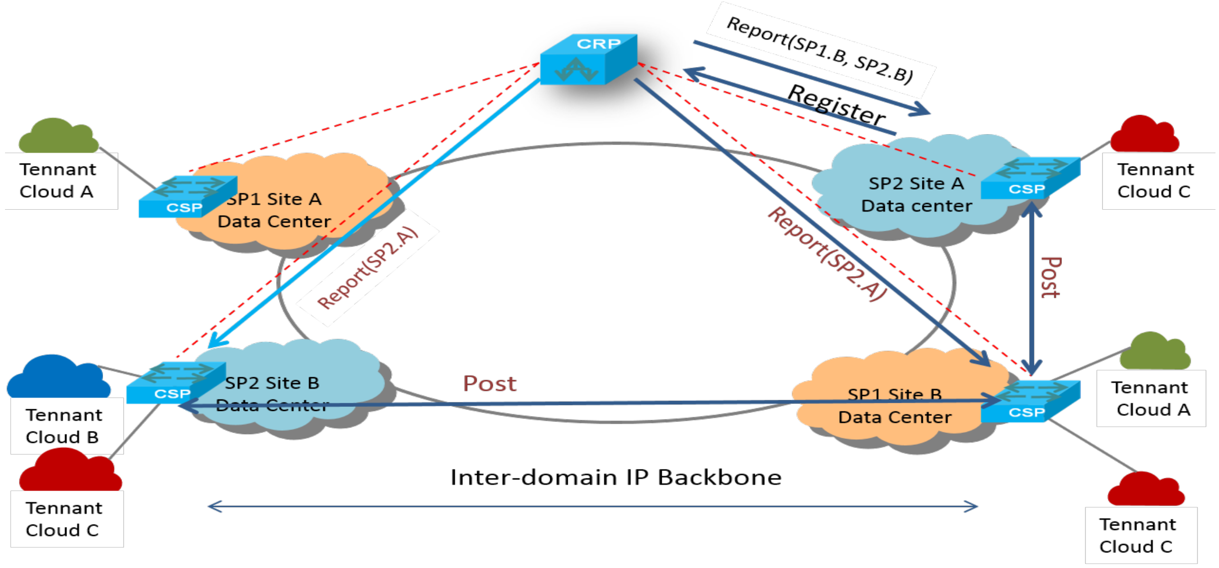

Figure 8: Cloudcasting for multi-site virtual private networks access to Data Centers

B. Cloudcasting As Data Center Interconnect

In a data center inter connect situation, typically data center operator leases MPLS circuits or dedicated link from the service provider. There are several different protocols to provide interconnection between the two data centers such as TRILL, SPB, EVPN, and L3VPN depending upon what is supported by the provider. Instead, cloudcasting can enable all these interconnections very easily without requiring to wait for service provider enabled circuits. In Fig. 7, there are two data centers; both running spine-leaf topologies along with cloud casting enabled network. There are 2 cases shown in this figure. First case is an example that the CSPs in either data centers that need interconnection across data center has infrastructure spaces public IP address. This address is globally routable and therefore, it is possible to directly setup a GVE tunnel in the following manner. Both the CSPs with global space IP address send a CC Register to logical CRP, which facilitates CC Reports. Since CSPs can reach each other, a GVE tunnel can be directly established and CC Post updates may be exchanged as well. This case implements scenario where cloud networks are hosted in two different public clouds, if there are CSPs with global space IP address, the communication between the 2 networks can take place. Often distribution and maintenance of public IP address in not feasible; then a CSP gateway on either data centers can provide a straight forward functionality to translate internal VXNS and bundle multiple GVEs over a single service provider connection.

C. Cloudcasting as VPN in service provider networks

Fig. 8 shows a multi-site VPN connection through cloudcasting. Extending the same concept of CSPs being hosted on each site and they connect to a single logical CRP, cloudcasting enabled VPNs can be formed in the similar manner as described in previous sections. The flexibility of cloudcasting allows to carry layer 2, VLAN, VXLAN, IP or any other network address family through a single virtual routing scheme in a topology independent manner. There is an additional discussion on cloudcasting vs existing technologies in the later section.

V. Scalability And Extensibility In The Cloud

The vision of cloudcasting protocol is many-fold. Firstly, it envisions geographically dispersed Internet-wide multi-tenancy enabled over global infrastructure at a massive scale through a single control signaling mechanism. Secondly, it aims to integrate the data-plane methods in order to normalize the tenant networks forwarding paths.

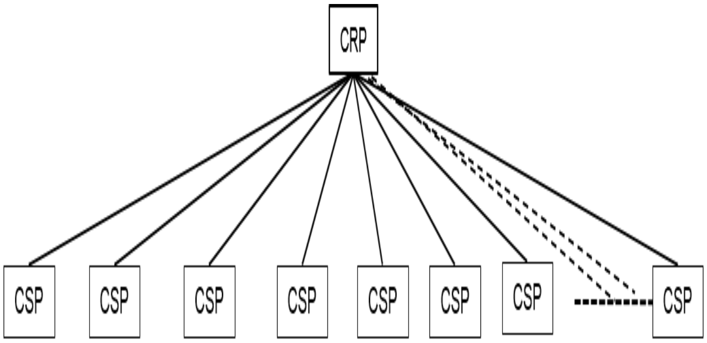

Figure 9: Growth in CRP-CSP connections at scale

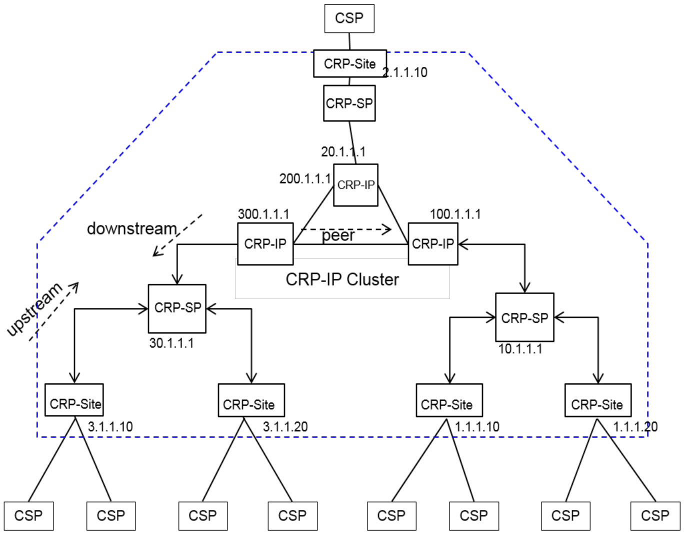

Figure 10: Hierarchical CRP System

A. *Scalability in Controlplane*

Cloudcasting is a virtual routing scheme for cloud based environments that maybe hosted across multiple service provider networks or at multiple sites. In the preceding section, it is assumed that the CRP is a single centralized entity.

1) Hierarchical CRP-system: Cloud networks are expected to be distributed beyond a local site, for example, a tenant network is not scoped with in a single provider domain, but needs to communicate with entities residing behind multiple provider domains. Assuming there are millions of such tenants, then a single CRP managing high number of sessions with as many CSPs in the Internet becomes unmanageable with a flat architecture as shown above in Fig. 9. A centralized CRP node can cause severe performance bottlenecks when servicing large number of CCC messages such as Register and Reports originating from high number of CSPs simultaneously.

The ability to scale is the most important requirement for cloudcasting routing scheme, otherwise, it does not provide a converged deployable solution. With in the cloudcasting protocol a distributed and hierarchical system of CRPs is proposed to exchange CRP control plane signaling. The system builds a connected graph of CRP instances across service provider domains. In order to create a hierarchy, a CRP node is associated with a scope. The scope maybe up to a local site (CRP-site), provider-specific (CRP-SP) or inter-provider (CRP-IP). The definition of cloud networks is also extended now to have scope with in (a) local site, (b) a single provider or (c) multiple providers.

An example of a 3-tier CRP hierarchical system is shown in Fig. 10. In this figure, the CRPs inside the dashed lined box connected together form a CRP system. A CRP-Site (nodes with IP address 3.1.1.10, 3.1.1.20, 1.1.1.10, 1.1.1.20 and 2.1.1.10) is an instance of a local CRP where CSPs from a physical location or a site connect to and is at the lowest level node in CRP system hierarchy. A CRP-SP (nodes with IP address 30.1.1.1, 10.1.1.1, 20.1.1.1) corresponds to a middle-tier CRP in a provider specific space and has a role of interconnecting multiple CRP-Sites in a given region in a single administrative domain. Finally, at the highest-tier of CRP hierarchy is a CRP-IP that supports inter-provider communication. The communication between CRP-Site to CRP-SP and CRP-SP to CRP-IP respectively is required to exchange discovery of cloud networks that are scoped to extend beyond a specific site, administrative domain respectively. In Fig. 10, it is shown multi-provider CRPs, CRP-IPs form a cluster together. Each node (IP address 300.1.1.1, 200.1.1.1, 100.1.1.1) cluster has equal status of cloud network Cloudcasting Information Base (CCIB).

2) CC Protocol Extensions: In order to extend cloudcasting signaling to Hierarchical CRP the following additions to the base protocol are proposed

- *Originating CRP TLV*: It identifies the source of a CC Register in a CRP system hierarchy. It is used maintain mapping of CRP-Site and cloud networks or VXNs in CRP-SP. In addition, CRP Role Attribute (local, provider, global) is also included to determine the scope of CRP.
- *VXN Scope Attribute*: It is used to describe the scope of a cloud network.
- *Cloudcasting Information Base (CCIB*): The CCIB is the control information base maintained at each CRP is aware of the scope of a signaling and originating source of the request. It is a stateful table that is learnt and looked at upon receiving Register and Reports from neighboring CRPs. Additionally, the CCIB state in each CRP may be stored separately for upstream and downstream in CRP-SP. A CC Report is generated and distributed in a similar manner.

3) Single provider scenario: Consider a scenario of single administrative domain on the left side of the Fig. 10 CRP-SP (30.1.1.1) and two CRP-Site with IP address 3.1.1.10 and 3.1.1.20. Further assume that a cloud network Cn is scoped in a single provider SP1 (30.1.1.1). As left CRP 3.1.1.10 receives a Register message from one of connected CSPs, it finds scope to be provider specific and relays message to CRP-SP (30.1.1.1). Before doing so, TLV extensions as per previous sections are added. Receiving CRP-SP maintains (3.1.1.10, Cn) mapping in its information base. At a later time if another CRP say 3.1.1.20 sends a CC Register for cloud network Cn, CRP-SP, 30.1.1.1 generates a CC Report for both CRPs 30.1.1.10 and 30.1.1.20. Finally, CSPs receive Register from their respective CRPs and can continue with route distribution.

4) Multi-provider scenario: A more elaborate collaboration is required to distribute cloud networks across multiple administrative domains, more so when these domains are geograph-

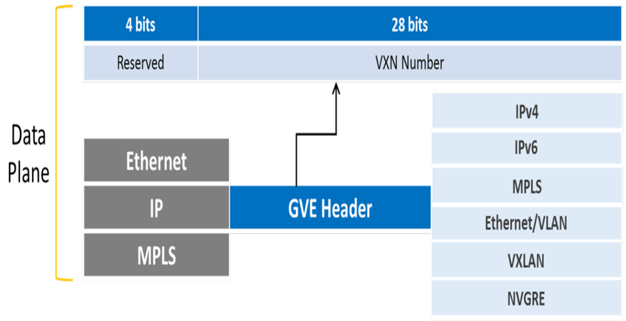

Figure 11: Normalized GVE Data Encapsulation

ically distributed. Therefore, CRP-IP (CRPs for inter provider communications) are clustered to store identical Cloudcasting information base. In this scenario, all CRP-IPS have identical database of cloud networks that are to be distributed beyond administrative domains. A CC Register of scope global is sent from CRP-Site to CRP-SP to adjoining CRP-IP. The receiving CRP-IP distributed this information to all other CRP-IPs in the cluster. Each CRP-IP is then responsible for downstream distribution of CC-Register to attached CRP-SPs, if and only if it has some knowledge in its information base that a CRP-site has interest in same cloud network. For example, lets assume a new global-scoped cloud network C_n is created by CSP attached to CRP-site 3.1.1.20. This CRP sends CC Register with extended TLV to its attached CRP-SP, 30.1.1.1, which in turn relays it to its attached CRP-IP, 300.1.1.1 by replacing originating CRP as itself in the extended TLV. This CC Register is distributed everywhere in CRP-IP cluster. As this is a new cloud and no instances exist, the request stays in the cluster. Similarly, at a later time when a CSP attached to CRP-site, 1.1.1.10 generates a CC Register for C_n, it reaches CRP-IP 100.1.1.1, which determines from its information base that CC Registers need to be sent to 1.1.1.10, does so and also send CC Registers to 3.1.1.20. The CC Reports are generated in exactly the same manner and finally a GVE tunnel is established.

B. *Extensibility through normalized data plane*

The second important aspect of extensibility in cloudcasting protocol is related to the normalization of data plane encapsulation. In preceding section, the GVE encapsulation is defined and Fig. 11 further illustrates it to be highly flexible and scalable. GVE is extensible by virtue of connecting 2 heterogeneous clouds through the multi-protocol information it carries. The figure also illustrates that GVE expects flexibility in terms of its position in outer header. It maybe carried as layer-2, layer-3 or MPLS payload and is capable of translating multiple encapsulations for example IP, MAC, VLAN, VXLAN, NVGRE and so on. The protocol also allows that an instance of a cloud networks at a site may use NVGRE encapsulation and another site of the same cloud may continue to use VXLAN encapsulation. It is of great advantage to migration of connecting 2 islands of a cloud network transparently without changing anything on the local site except for enabling cloudcasting between the edges or gateways.

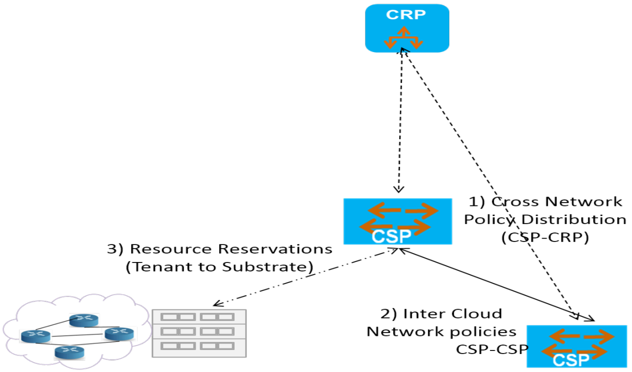
CRP
CSP
1) Cross Network Policy Distribution (CSP-CRP)
3) Resource Reservations (Tenant to Substrate)
2) Inter Cloud Network policies CSP-CSP
CSP

Figure 12: 3-Dimensional Policy Framework for Cloudcasting

VI. Policy Framework for Cloud Networks

A smart policy framework can help build simple orchestration platforms that do not need to excessively interact with the infrastructure and can also adapt independently to policy changes within the virtual networks. Even in the cloudcasting framework, most traffic within and across the cloud networks is still required to be subjected to forwarding or application specific policies. In this section a brief discussion of a new policy model is presented that in general suits better with the cloud networks. The policies in any network of scale such as an enterprise or a campus tend to be fairly diverse, complex and yet quite similar from one tenant to the other. It is difficult to describe network policies because they are designed and created from a business logic perspective. The business logic itself is created centrallly but must be disaggregated and applied in parts across different network segments. In case of cloud-centric environments, it is further obscured because now the environment is virtualized and physical location agnostic. The state of art of policy framework is far too fragmented [12] [13] [14] [15] both in terms of policy description language and common specifications for policy distribution. There are several vendor specific approaches as well open policy frameworks as well. In our view, it is much simpler to break down network policies for cloud networks across three dimensions to address different aspects. In cloudcasting architecture, we separate policy-based interfaces as shown in Fig.12 associating CSP, CRP and the substrate network through 3 different types of policies and their scope. These are explained as below.

1) Cross-Network Policies: In this case, considerations are made to propagate rules that permit or disallow traffic across different cloud networks to the other through policies or Service Level Agreements (SLAs). These type of policies interface at higher level of abstraction. For example, it may also be necessary to specify if dev-test clouds can access the database from production clouds. Within cloudcasting, this is an interface between CRP and CSP. It is extremely simple to associate such policies on CRP, then when the Register request is made, CRP may deny or accept the request to join a certain cloud network. The dotted lines on right-hand side of the Fig.12 shows the scope of such policies.

2) Inter Cloud Network Policies: It is very common to setup policies in a network so that traffic must get steered through specific service chains. For example, traffic from an ingress port is first subjected to firewall then load balancer and finally to an application server. Such policies are not infrastructure related and are within a cloud network that may need distribution within a site or across multiple sites. Once tenant networks are discovered, CSPs have a path to distribute policies through CSP-CSP policy interface. This scope is shown as CSP-CSP through solid line on the bottom right of Fig.12.

3) Tenant to Substrate Network Policies: In section II, it is explained that cloud centric tenant networks borrow and consume resources from substrate networks and tenants do not own any physical resources themselves. Yet, it is necessary to allocate resources to support quality assurance and bandwidth guarantees. Since tenant network operators cannot reserve resources they do not own and any bulk pre-allocation does not align with infrastructure independence, a separate tenant to substrate network policy interface is mandatory. This interface is not related to cloudcasting and therefore, should not be part of cloudcasting. However, there is a need for generalized reservation and administration method, be it a protocol or API based that may be used between tenant and substrate networks. This scope is shown on the left-hand side of the Fig.12.

In previous two sections additional features of cloudcasting such as extensibility, scalability, normalization and policy interfaces were briefly explained to demonstrate that cloudcasting framework is entirely viable solution for interconnectivity of cloud networks. In this paper, emphasis is on core architecture extensions and many details relating to extended TLVs are omitted out. For the same reason, policy interface details are excluded from the paper.

VII. Evaluation And Analysis

The cloudcasting architecture and primitives have been implemented in our research laboratory. We have successfully used the cloudcasting architecture and control protocol to implement the above mentioned use cases. First and foremost, we emphasize that the cloudcasting architecture represents a paradigm shift. It is a truly converged technology for virtual networks, clouds, and VPNs. No matter what the structure of the underlying substrate network is, any/all types of virtual tenant networks can be constructed in the same way by using cloudcasting.

A. Qualitative Analysis

The Cloudcasting suitability and applicability can only be verified vis-a-vis characteristics of the cloud-scale environments. Therefore, we have laid importance on the primary characteristics of cloud centric networks that are elasticity, efficiency, agility, and distribution. The Cloudcasting control plane is elastic, because it can grow and shrink independently of (1) the heterogeneous protocols of the substrate network, (2) number of virtual network attachment points, the CSPs, (3) number of domains (autonomous systems), (4) number of routes within a users virtual network, and (5) mobile nature of the host stations. The Cloudcasting control plane is efficient, because (1) no CSP distributes routes to other CSPs that they are not interested in, (2) thus, no CSP receives and stores routes of virtual networks of non-interest or the ones it is not connected to. In addition, the control plane is fully distributed in such a manner that through a single primitive (post-update); change in the tenant networks can be announced immediately, from the spot of change without configuration changes. The Cloudcasting allows for agile networking. Every time when a new CSP is added, it is only required to configure the newly added CSP by using a few lines of commands. Every time when a CSP is deleted, no additional configuration change or for that matter nothing else needs to be done. This is because cloudcasting has a built-in auto-discovery mechanism that has not been seen in the embedded virtual networks. The Cloudcasting data plane scales as well. Its default GVE encapsulation protocol allows to support 256 million clouds. In other technology such as, VXLAN, it only up to 16 million clouds are supported. Due to the limitation of space, we wont discuss and describe other more desirable characteristics.

B. Prototype Implementation

In our lab, three small-scale data centers were implemented for the demonstration of functionality. Each data center had a CSP network element and also connected to the CRP in cloudcasting enabled network. In addition, each data center also comprised of one or two hosts; and each host had at least 2 VMs spawned with their own private IP addresses. All the traffic from VMs or hosts was default forwarded to the CSP, which performed the data plane encapsulation/decapsulation and forwarding between CSPs. One of the data centers served as media server center and others were clients. The purpose of this setup is to show isolation with in a virtual network domain and VM mobility from one data center to the other. The setup also has a network management system that provisions CSPs about virtual networks and VM hosted with in them. The development environment is entirely based on open source code or is in-house developed. The code is implemented in the following categories -

1) CSP control plane software: CSP software is based on quagga (0.99.24) [16] open source, because it provides an ideal and quick router/switch like development environment to use many features such as command line for configuration, message parsing, daemon and process communication features that are already build in quagga. A csp daemon was created in quagga base and new code was written to provide following functions

- CSP-CRP Connection: CSPs listen to a TCP port and connect to CRP, which is a configurable IP address. On this channel Register and Report messages are exchanged.
- CSP-CSP Communication: CSPs listen to another TCP port to connect to CSP IP addresses received in Report messages. This channel is used for Post updates for virtual route exchanges. Once the routes are learnt from peer CSPs, the datapath process is updated.
- CSP Network Management Interface: CSP also interfaces with a management entity to receive virtual network specifications or changes thereafter. These changes are pushed as an XML file and can easily be changes to REST APIs.

2) CSP datapath: CSP data plane is implemented as another daemon using pcap library [17] to perform tunneling functions for traffic between hosts and CSPs. It maintains two forwarding rules passed from CSP control process, viz. host-CSP and CSP-host. Since the aim was proof of concept and data plane is implemented in software, it is irrelevant to discuss the forwarding path throughput.

3) CRP control plane software: While, CSP control and data plane are developed in C; CRP code is entirely written in Java, consequently both Java and C code base for the CCC protocol exists. CRP uses neo4j [18] based highly scalable graph databases to store and visualize relationships between the virtual networks and CSPs.

4) Network Management Interface: The management system is in-house developed software for a network operator. Written using C# as a web application on IPAD, an operator is able to add or delete new virtual networks to a specific data center as well as add, delete and move VMs from one data center to the other.

The above code may be made available to those interesting in further research in cloudcasting. Due to lack of testbeds and other resources quantitative comparison has not been performed adequately and results are not available yet. It is our intent to demonstrate controlplane efficiency through analysis of bytes and messages transferred under several approaches.

VIII. Related Work

There are several works available that partially solve network virtualization problem; however, they do not provide a complete and consistent solution that sufficiently fulfills all basic requirements discussed earlier in this paper. In what follows, we discuss and compare a few prominent network-overlay approaches.

A. IETF NVO3

The cloudcasting architecture and protocol shares some goals chartered by the IETF working group NVO3 (Network Virtualization Overlays over Layer 3) [19]. The purpose of NVO3 is to develop a set of protocols and/or protocol extensions that enable network virtualization within a data center environment that assumes an IP-based underlay. Cloudcasting varies from NOV3 in that cloudcasting is not just restricted to the data center, and it does not expect a specific structure or protocol conventions in the underlay. The control plane of NVO3 may seem to be a reformulation of the BGP architecture, where NVEs (Network Virtualization Edge) and NVA (Network Virtualization Authority) resemble iBGP speakers and Route Reflectors, respectively, and NVO3-VNTP [20] resembles BGP update messages between an iBGP speaker and its Route Reflector. Therefore, NVA needs to learn and store routes from an NVE and then distribute those routes to other NVEs. In contrast, in Cloudcasting virtual route information is a function between CSPs, the routes are only distributed between the CSPs, the CRP is not involved in routes. CRP is used for cloud membership auto-discovery and thus enables agile provisioning. Auto-discovery functions are also missing from NVO3, where are they are natural to cloudcasting protocol. We should emphasize that CRP has no route database inside that has a significant impact on the size of the database in CSP. This differentiation is common with other related work discussed in the following sections. NVO3 suffers from the existence of multiple encapsulations, the working group has not been able to make progress on a native control plane design and most often resort to EVPN control plane. The group is also divided on the subject of data plane format whether the group shall support a single or multiple encapsulations. In this regard, Cloudcasting GVE supports multiple types of data plane encapsulations inherently as is discussed in earlier extensibility section V (B).

B. VXLAN and EVPN

VXLAN is a data plane format for network overlay encapsulation and decapsulation, and EVPN has been proposed as the control plane for VXLAN [21] [9] [22]. BGP was originally designed for inter-domain routing across service provider networks. Although, EVPN is the only IETF defined distributed control plane protocol, BGP in data center network virtualization leads to may operational overheads as explained in the following ways

1) In order to deploy EVPN, the network operator must configure something like an AS (autonomous system) in substrate networks, which is not really a data center design concept. In addition to this many other BGP-VPN related constructs such as route-targets (RT) are route-distinguisher (RD) must be defined. Configurations can be templatized to reduce complexity, yet to keep the network consistent these parameters must be carefully chosen and during network outages, trouble shooting is extremely difficult because an operator has to be aware of the mappings of RTs and RDs to virtual networks, not to mention that higher number of configuration parameters adds to management traffic. A sample configuration maybe found at [23].
2) Running BGP in a data center requires VTEPs to be iBGP speakers. This can also lead to serious scalability problems of a full-mesh of peering sessions between iBGP speakers (VTEP-BGP). Typically, to address this problem, deployment of Route Reflectors (RR) is recommended. RRs then speaks with every other VTEP-BGP to synchronize their BGP-RIB. As a result, no matter if a VTEP needs a route or not, all the other VTEPs will always send their routes to the VTEP through a Route Reflector, and the VTEP is required to filter out not needed routes through Route Target and other BGP policies. Distribution of not needed virtual routes from RR to VTEP-BGP levies an unnecessary overhead on the substrate network and burn CPU power, processing these BGP messages.
3) BGP in the data centers not only makes operational cost of data centers as high as that of a service providers network it also lacks the agility because BGP heavily relies on configurations (it is well known that configuration errors are a major cause of system failures [8]). For example, when a new BGP-VTEP is added/removed the operator has to configure all the BGP peering relationships by stating which BGP neighbors are peering among each other.

Observe that when BGP was first designed, some distribution and peering principles were built-in; for example, iBGP peers should have received and synchronized the same copies of routes. In the case of clouds, many such principles are not applicable and exceptions need to be added to BGP protocol to address requirements for the cloud networks. Cloudcasting architecture does not suffer from the drawbacks described above. By means of auto-discovery and route distribution, only specific routes of a virtual network are distributed. Moreover, the role of CRP does not require it to be an intermediate hop between two CSPs to distribute the routes. The detailed comparison and evaluation is still in progress and will be published at a later stage.

C. LISP based data center virtualization

Although Locator ID Separation Protocol (LISP) [11] is not an inherent data center virtualization technology, it has a framework to support network overlays. LISP achieves this by distributing encapsulated tenant (customer) routing information and traffic over provider (substrate) network through its control plane based on a mapping system. The LISP architecture includes Ingress/Egress Tunnel Routers (xTRs) and a mapping system (MS/MR) that maintains mappings from LISP Endpoint Identifiers (EIDs) to Routing Locators (RLOCs). LISP requires mapping information to be pulled on-demand and data-driven, xTRs also implement a caching and aging mechanism for local copies of mapping information. Cloudcasting CSPs and LISP xTRs are similar in that they are the virtualization tunnel endpoints performing encapsulation and decapsulation. But the VXN route database and LISP's mapping databases are different as below

1) LISPs mapping system [24] is a separate protocol element and is based on hierarchical design of Domain Name Server (DNS). The xTRs work in collaboration with mapping server (MS) and map resolvers (MR). First and foremost, an xTR must register its EIDs with the mapping system. When a remote xTR is ready to exchange data for an EID, it will query mapping system to find the xTR where EID is located, create the local mapping cache (is referred to as pull method) where entries are aged when not needed. In comparison, CSPs are able to discover each other on the basis of VXN, without registering any EIDs with CRP. Once CSPs and VXN mappings are formed vFIBs are built by post updates. Thus, routes are local and significant only to the CSP.
2) An xTRs local database is built on demand after receiving a data packet without knowing its mapping information, which may expose sender to security risks because the destination is unknown, while CSPs VXN CCIB is signaled through the cloudcasting control protocol over an authorized communication channel. The infrastructure can flexibly make the channel as secure as it prefers using security and encryption protocols.
3) A CSP can auto-discover other CSPs that join the same VXNs, while LISP xTR can only know about another particular xTR after querying the mapping database.

IX. Virtualization in Mobile Networks using Cloudcasting

During our research and study of policy based constraints in Cloudcasting, we came across Fifth Generation (5G) network slices. We concurred with the authors of Next Generation Mobile Networks (NGMN) [25] white paper that 5G networks will be a collection of service aware logical networks. It was obvious that a higher degree of automation is vital in 5G for services to be discovered, provisioned and resources to be apportioned/released. Authors have discussed using Cloudcasting as fundamental block in [26] for auto-discovery of services in mobile network supporting cloud hosted environments. In this work, a network slice corresponds to a VXN in cloudcasting, while service extensions (resource specifications) are newly added and associated with a network slice. The main idea in this paper deviates from symmetric VXN relationship of Cloudcasting. 5G services in [26] have asymmetric producer and consumer association. First, network segments participate in cloudcasting system and network slices are bound to those segments. Then in a producer role, the services announce themselves, their location in the system and their resource requirements. Thus services become available and discoverable with in those slices. Subsequently, in the consumer role, an end user or device attaches itself to the service; network resources are allocated across different network segments. The procedures just described are done dynamically that allows a mobile network system to be easily managed. The idea of auto-discovery is fairly advanced and prototyping of this approach is still being done.

X. Future Work

In this paper, we have presented several extensions to Cloudcasting protocol in terms of policy, services and scalability that makes it more complete. Cloudcasting can be thought of as a generalized virtual routing framework. Its validity, scalability and extensibility as a single mechanism for implementing cloud centric networks. There are several scenarios not explored yet include containers, microservices and SD-WAN.

Since the publishing of original paper, we have designed an integrated policy model as the basis of interface between substrate and virtual network. The model allows distribution of tenant policies using the base protocol. However, resource allocations over substrate network were not found to be as simple and in fact led to new work as an extension to cloudcasting protocol. The corresponding data structures and prototype implementation are an open for further research at the time of writing this paper.

Previous section alluded to Cloudcasting extensions for service distribution in mobile networks. The 5G network slice definition is still evolving, therefore, an opportunity lies in exploring this topic further both from prototyping and validation perspective. Finally, although the prototype for base protocol is available, further comparison study, assessment of control plane signaling overheads, robustness and datapath optimizations related work is not complete yet and we welcome contributions from interested research community.

XI. Conclusion

Cloud-scale networking environments require a technology where virtual networks are first class objects; such that the coarse policies and routing decisions can be defined and applied on the virtual networks. Cloudcasting is a routing system based on converged, unified network virtualization and will evolve better because of lower provisioning costs and enhanced agility through auto discovery. This paper presented several new concepts; it extended original idea from single data center to explain global scale distribution of VXNs across multiple providers, sites and domains. Many use cases are further discussed in great detail. We also shared our perspective on policy model, which plays an important role in interaction between virtual and physical infrastructures to provide operation and management functions. The prototype implementation is discussed at great length and interested readers are encouraged to contact the authors for code. As an interesting application, we have taken the idea of cloudcasting for virtual networks and extended it to the auto-discovery of services in the 5G network slicing context that further bolsters adaptability and flexibility of the framework.

ACKNOWLEDGMENT

The authors are thankful to the members of the Cloudcasting project, particularly those involved in the early design and proof of concept and prototype development for their collaboration and fruitful discussions.

REFERENCES

[1] K. Makhijani, R. Li, and L. Han, "Cloudcasting: A New Architecture for Cloud Centric Networks," in Proc. IARIA Int. Conf. on Comm. Theory, Rel., and Qual. of Serv. (CTRQ), Lisbon, Portugal, February 2016.

[2] D. Fedyk, P. Ashwood-Smith, Allan, A. Bragg and P. Unbehagen, "IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging," Internet Engineering Task Force (IETF), April 2012.

[3] IEEE, "IEEE Standard for Local and metropolitan area networksMedia Access Control (MAC) Bridges and Virtual Bridged Local Area NetworksAmendment 20: Shortest Path Bridging," June 2012.

[4] D. Eastlake, T. Senevirathne, A. Ghanwani, D. Dutt and A. Banerjee, "Transparent Interconnection Of Lots Of Links (Trill) Use Of Is-Is," Internet Engineering Task Force (IETF), April 2014.

[5] M. Sridharan et al., "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," Aug. 2014.

[6] M. Mahalingam et al., "NVGRE: Network Virtualization using Generic Routing Encapsulation, draft-sridharan-virtualization-nvgre-08 (work in progress)," Internet Engineering Task Force (IETF), 2015, pp. 1–10.

[7] J. Gross, T. Sridhar et al., "Geneve: Generic Network Virtualization Encapsulation, draft-ietf-nvo3-geneve-0," Internet Engineering Task Force (IETF), April 2015.

[8] T. Xu, Y. Zhou, "Systems approaches to tackling configuration errors: A survey, article no.: 70, acm computing surveys (csur) volume 47 issue 4," July 2015.

[9] A. Sajassi et al., "A Network Virtualization Overlay Solution using EVPN," Internet Engineering Task Force (IETF), February 2015.

[10] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding Substrate support for path splitting and migration,Computer Communication Review," vol. 38, 2008 2008, pp. 17–29.

[11] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," Internet Engineering Task Force (IETF), January 2013.

[12] VMWare Micro segmentation. web:https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nsx/vmware-microsegmentation-solution-overview.pdf. (2015)

[13] Simplified Use of Policy Abstractions (SUPA). web:https://datatracker.ietf.org/wg/supa/charter/. (2015)

[14] Opendaylight, Group based Policy. web:https://wiki.opendaylight.org/view/Group_Based_Policy_(GBP). (2015)

[15] Cisco Application Centric Infrastructure(ACI). http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/index.html. (2015)

[16] Quagga Routing Software Suite, GPL licensed. http://download.savannah.gnu.org/releases/quagga/quagga-0.99.24.tar.xz.

[17] libpcap, a portable C/C++ library for network traffic capture. http://www.tcpdump.org/.

[18] Neo4j, a highly scalable native graph database. https://neo4j.com/.

[19] M. Lasserre et al., "Framework for Data Center (DC) Network Virtualization," Internet Engineering Task Force (IETF), October 2014.

[20] Z. Gu, "Virtual Network Transport Protocol (VNTP)," Internet Engineering Task Force (IETF), October 2015.

[21] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)," Internet Engineering Task Force (IETF), April 2006.

[22] Sami et al., "VXLAN DCI Using EVPN, draft-boutros-bess-vxlan-evpn-01.txt," Internet Engineering Task Force (IETF), January 2016.

[23] VXLAN Network with MP-BGP EVPN Control Plane Design Guide. http://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/guide-c07-734107.html. [retrieved: April, 2016]

[24] V. Fuller et al. LISP-DDT: LISP Delegated Database Tree, Work in Progress. https://tools.ietf.org/html/draft-ietf-lisp-ddt-08. [retrieved: September, 2015]

[25] N. Alliance, "NGMN 5G White Paper," White paper, February 2015, pp. 1–125.

[26] K. Makhijani, S. Talarico, and P. Esnault. Efficient Service Auto-Discovery for Next Generation Network Slicing Architecture, presented at O4SDI, IEEE NFV-SDN 2016. Unpublished.