Simulation of Carry Protocol (c-protocol) for MANET Network

Ahmed Alghamdi, John DeDourek, Przemyslaw Pochec
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
E-mail: a.alghamdi, dedourek, pochec@unb.ca

***Abstract*— The current protocols used in Mobile Ad Hoc Networks require an end-to-end connected path between source and destination to transmit data. We present a group of protocols (c-protocols) that address the issue of dropping packets each time the end-to-end path breaks. In the c-protocol, rather than being dropped, the packets are allowed to be carried at any node for a time until the carrying node gets connected to another node and retransmits the packets. The simulation shows that the c-protocol allows data to be transmitted and delivered even if the end-to-end connection never occurs. The GPS enabled versions of the c-protocol produced the highest throughput in the simulated network.**

Keywords- store-forward protocls, disturbance–tolerant network, MANET, network simulation.

I. INTRODUCTION

A network is a number of computers or devices that are connected to each other through physical or wireless links. The main advantage of having networks is to be able to share resources and services among the connected devices. In order to have sufficient and beneficial communications, the data transfer between the connected devices is governed by a protocol (a communication rule). There are a large number of protocols that are used in networking. Each protocol has its own services and operates over specific types of network. Wired networks have specific protocols or specific versions of a protocol that would not necessarily work efficiently over wireless networks. These protocols have to be modified or changed in order to serve different types of networks.

With the presence of the Internet, having a home or an office network connected to the world is relatively a simple and easy task, especially these days with the huge improvement in the Internet services. However, this is not the case in every network. There are some networks that have to operate in extreme environments. These types of networks have to be treated in different ways to achieve the most efficient type of communication. Examples of such networks are Terrestrial Mobile Networks, Exotic Media Networks, Military Ad-hoc Networks and Sensor/Actuator Networks [1] [2]. In Terrestrial Mobile Networks, the mobility of the nodes and the change in the strength of the signal may cause the network to be partitioned. Due to the large distance in the case of the Exotic Media Networks, there is a high latency in delivering a message. The Military Ad-Hoc Networks usually operate in hostile areas where mobility, danger and other environmental factors would cause the disconnections to repeatedly occur in the network. Generally, there are different types of problems that occur in such networks. For example, high latency and low data rate are common problems that need to be dealt with in order to have a satisfactory connection. Another problem is the repeated disconnections that prevent the network from having normal end-to-end connections which lower the data transmission rate to unacceptable levels. These problems could be solved by some special architectures like MANETs (Mobile Ad Hoc Networks) or DTNs (Disturbance-Tolerant Networks).

In order to exchange messages or packets between any two nodes, the existing MANET networks require an end-to-end direct path; otherwise messages are dropped when the connection breaks. In order to overcome this problem, messages may need to be carried by the intermediate nodes for a period of time until the nodes get reconnected. Then, the messages are retransmitted again. Our objective is to test such a scenario by simulating different versions of data carrying protocols.

We start by presenting a literature review in Section II. We review Ad Hoc, MANET and DTN networks. In Sections III and IV, we review in details the protocols used in MANETs and DTNs respectively. In Section V, we describe our proposed solutions. In Section VI, we describe the simulation environment. In Section VII, we present the results. Finally we conclude the paper and give some ideas for future work.

II. STATE OF THE ART

We start with a review of different types of mobile networks.

A. Ad Hoc

An Ad Hoc network is a type of local area network (LAN). Each individual device in this network can communicate directly with any other device in a peer-to-peer style. This style eliminates the involvement of a central device that acts as a base station or a router. Ad Hoc networks operate with the absence of a fixed infrastructure. The nodes in Ad Hoc networks can be hosts as well as

 ISBN: 978-1-61208-217-2

routers which allows a message to be transmitted form node to node through the network until it reaches its final destination. The setup of an ad-hoc network is easy and simple and almost all operating systems support this type of network.

B. MANET

The MANET (Mobile Ad Hoc Network) is a type of Ad Hoc network with mobile nodes moving around. The mobile nodes in a MANET change locations and configure themselves on the move and use wireless channels to be able to communicate with each other [1]. The wireless medium could be a Wi-Fi medium, a cellular medium or a satellite medium. Each node could play the role of a source, a router and/or a destination [3].

A simple example of MANET topology could, for example, contain two nodes far away from each other that behave as a source node that sends messages and a destination node that receives the messages. Between these two nodes, there are a number of mobile intermediate nodes that act as hosts as well as routers [3]. Each intermediate node has a limited range in which it can make a connection to a neighbour node (i.e., two nodes that have a range of one meter have to be at most one meter away from each other in order to establish a connection and exchange messages). The message to be sent is then transmitted between those nodes until it reaches the destination. Due to the movement of the nodes and whether or not they are connected to each other, a direct path between the source and the destination is not always guaranteed.

C. DTN

A Disturbance-Tolerant Network (DTN) (also called Delay-Tolerant Network) is a network architecture that could be applied to operate over long distance communication. In this sort of communication, there are challenges that occur during a communication session. Some of these challenges include node mobility, lack of end-to-end paths, the limited transmission range associated with each node and other challenges. Due to these challenges, the existing TCP/IP routing protocol will not work efficiently and has to be modified or changed to work better with the characteristics of DTNs. Zhang (2006) categorized the routing protocols associated with DTNs into different categories: deterministic, stochastic, model-based, control movement and coding-based approaches [4]. Routing in DTNs has a direct relation with the work and development of the c-protocol discussed in this paper.

III. PROTOCOLS USED IN A MANET

Many protocols have been proposed to work in MANETs. Each one of these protocols has specific properties and structure to deliver a solution to a problem facing MANETs. A wide range of these protocols have been categorized into three major categories. These categories are Proactive Routing Protocols, Reactive Routing Protocols and Hybrid Routing Protocols [3] [5].

In proactive routing protocols, each node in the MANET network stores a table holding information about the other nodes in the network. Depending on the implementation of the protocol, the table might hold information about every other node in the network or some selective nodes [5]. These tables are updated periodically or whenever the topology of the networks changes [3] [5] [6]. The protocols in this category differ in the way they update the table/s and the information kept in them. Examples of protocols that fall under this category are DSDV (Destination Sequenced Distance Vector), WRP (Wireless Routing Protocol), GSR (Global State Routing), FSR (Fisheye State Routing), STAR (Source Tree Adaptive Routing) and DREAM (Distance Routing Effect Algorithm for Mobility).

Instead of updating the tables of all the nodes in the network, in Reactive Routing Protocols, updates are only performed on the nodes that need to send data at a specific time. This is called On-demand routing [3] [5]. This means the route from the source to the destination is determined upon sending. Usually the source floods packets into the network to determine the best route to the destination. The packets flooded are small packets known as route request packets (RREQ) [3]. Based on the acknowledgment/response/reply resulting from sending (RREQ), the best route is chosen to deliver the data. Reactive/On-demand routing is further categorized into two categories known as hop-by-hop routing and source routing [3] [5]. The difference between these two categories occurs in the header of the sent packets. In source routing, the full information of the address is stored in the packet header. In hop-by-hop routing, only the addresses carried by a packet are the final destination address and the next hop address. The source routing is reported to be inefficient due to the overhead resulting from carrying too much information in the packets headers [8]. Examples of Reactive/On-demand protocols are AODV (Ad Hoc On-demand distance vector), DSR (Dynamic Source Routing), ROAM (Routing On-demand acyclic multi-path), LMR (Light-weight Mobile Routing), TORA (Temporally Ordered Routing Algorithm) and ABR (Associativity-Based Routing).

Hybrid Routing protocols adopt a mix of the first and second categories' properties [5]. Examples of this category are ZRP (Zero Routing Protocol), ZHLS (Zero Based Hierarchical Link State), DST (Distributed Spanning Trees based routing protocol) and DDR (Distributed Dynamic Routing).

IV. PROTOCOLS USED IN DTN

Due to the repeated end-to-end connection loss, routing in DTN is challenging. A store-and-forward approach is used often in such networks. In Store-and-forward, a message is stored in an intermediate node until the node

 ISBN: 978-1-61208-217-2

sees an opportunity to retransmit the message. This gives the DTNs the advantage of delivering the message without the need of an end-to-end connection [7]. The routing protocols in a DTN are designed to overcome the problem of repeated disconnections. There are two categories of DTN protocols: Deterministic Routing Protocols and Dynamic or Stochastic routing protocols [7]. In Deterministic DTNs, the future topology of the network is known or could be predicted simplifying finding a route.

In Dynamic DTNs, the topology is not known. Dynamic Routing Protocols differ in the way they make decisions to which node a message is forwarded. A simple routing algorithm is called Direct Delivery where a node retransmits a message only if it gets in range with the destination [7]. Another routing algorithm is the First Contact algorithm [7]. In the First Contact algorithm, each node retransmits a message to a randomly-selected, in range, node. The decision made to choose a random node is not efficient since this randomly-chosen node might not be moving towards the destination. Epidemic routing is another approach where each node sends the message to be delivered to each other node in range (flooding). A node accepts a message only if it does not already have another copy of the same message in its buffer.

In Dynamic Routing, because of repeated transmissions, a lot of storage space is wasted. This raises the need of having a recovery scheme to deal with the copies of the data left in the network after a message is delivered. One solution is to introduce a life time parameter where a message is discarded if it has been carried for a period exceeding its life time. This life time scheme is optimal since the message would not reach the destination if the life time is too short. If it is too long, the storage capacity would be wasted. Another recovery scheme introduces acknowledgments that are flooded into the network once a message is received at the final destination. Each node in the network receives such acknowledgments. Then, it deletes the corresponding message stored in its buffer. These acknowledgments could be used as a way to guarantee successful delivery.

V. PROPOSED SOLUTIONS

Different versions of the c-protocol (also known as a store-and-forward-protocol) that do not relay on maintaining the end-to-end connections for a successful data transfer are proposed and simulated in this paper. The difference between these versions is in the algorithms that are used to forward the packets. The common property, which all versions share is the ability to carry/store a message for a while until a reconnection occurs.

These versions are; (1) First hop in the list routing (FLR), (2) closest hop routing (CHP) and (3) farthest hop routing (FHR). By introducing a GPS location (Global Positioning System), so that the distance to each node in the topology is known, (4) the closest to the destination routing (CGPS) and (5) forwarding to the hop that has the best next location to the destination (NGPS) are proposed. One last version of the c-protocol is simple flooding. In order to understand the underlying implementation of each version, a brief discussion is mandatory.

In First in the list routing, the node listed first in the next hop table, by the node currently carrying a packet (the carrying hop), is the one the packet is forwarded to. In closest hop routing, the distance between each connect node is calculated and the packet is forwarded to the one that is closest to the carrying node (in other words, the one having the strongest transmission signal). In farthest hop routing, the packet is forwarded to the farthest node from the carrying node (in other words, the one having the weakest transmission signal).

In GPS enabled routing, the current position of the destination is known to all the nodes in the network. In closest to destination routing, the distance between every connected node and the destination is calculated and sent to the carrying hop. The packet then is forwarded to the closest node to the destination. Since there is movement involved in the network, it cannot be guaranteed that the closest node to the destination is not moving away from the destination. To overcome this issue, forwarding to the closest *next location* to the destination is proposed. (i.e., next location is the location that a node is moving towards. Once the next destination is reached, the node changes direction) Rather than sending the packets to the closest current location of the node to the destination, they are forwarded to the node that has the closest next location to the destination.

In the simulation, the connectivity between the nodes is stored and maintained as a matrix of 0s and 1s, which means not connected and connected respectively. The movement of the nodes is considered random in this work. The way the nodes move is by generating a random X and Y coordinate (treated as the next location and bounded by the network area) and then move at a constant speed towards this next location. After the next location is reached, it is set to be the current location and a new next location is generated. The movement pattern used in the simulation is the same pattern used by the SetDest utility supplied with Network Simulator 2 (NS2) [8].

VI. SIMULATION ENVIRONMENT

When designing any protocol, a set of requirements has to be specified such as: guaranteed delivery, in-order delivery, packet duplication, etc. In this particular type of network topology, the movement pattern and density of the intermediate nodes plays a big role in designing the c-protocol. For testing, the implementation of the actual mobile nodes could be expensive and time consuming. We considered using an existing simulator (like NS2) but require a significant effort to add a new protocol [8]. Instead, a customized JAVA simulator was used to simulate the network and to develop the c-protocol. The JAVA simulator provides a controlled environment for testing the c-protocol and provides a full control over the parameters and the algorithms used by the c-protocol.

A. *Assumption*

To simplify the simulation, some assumptions have been made. To be able to accurately monitor the data flow, it was decided that there was only one source sending and one destination giving only one data flow. To better visualize the network as well as to better understand how the intermediate nodes move, the source and the destination were assumed to be stationary with all other nodes mobile. Another assumption is related to the forwarding mechanism. It is assumed that packet transmission and delivery takes exactly 200 ms for every node (200 ms forwarding cycle).In 500 byte packets, this corresponds to the packet transmission rate of 20 kbps (assuming zero propagation time). In each forwarding cycle, only one packet is sent from each node if the node has a packet to transmit. Another assumption is regarding the type of the data flow and the data generator. The data generator is assumed to generate CBR traffic (Constant Bit Rate traffic), with no acknowledgements required, at the source with fixed interval (one packet every 800 ms) between each packet generated. This corresponds to a generation rate of 5kbps. We implemented a limit on time for carrying a packet in a node: A packet is carried in the node no longer than 10 seconds. Then it is dropped. Buffers were introduced in each node buffering no more than 50 packets at a time. If the buffer is full, it is assumed that the node is not going to receive any more packets until the buffered packets get resent or dropped due to the effect of TTL. In case of the buffers in the sending and receiving ends, a drop tail queue was introduce at the source which allows dropping the new generated packets if the buffer is full. At the receiving end, the buffer size is assumed to be infinity due to the fact that the receiving end is the final destination of the packets.

Table 1 contains the parameters used in the simulation along with their values.

Table 1 The values of the parameters used in the simulation

Parameter	**Value**
Network area	400m x 400m
Number of nodes	N = 7(Low density)
Transmission range	50m
Velocity	1 m/sec
Packet generation interval	800 millisecond
Forwarding cycle	200 millisecond
Location update	10 millisecond
End-to-end connection	Updated every 1 millisecond
Throughput	Calculated every 5 sec
Buffer Size	50 packets
Packet size	500 bytes
Carry time	10 sec
Total simulation time	180 seconds

VII. RESULTS

Two simulation scenarios were used in testing. One with N=7 mobile nodes and the other with 50.

Figure 1 shows the end-to-end connectivity between the source and destination during the simulation captured from the first simulation scenario (N=7).

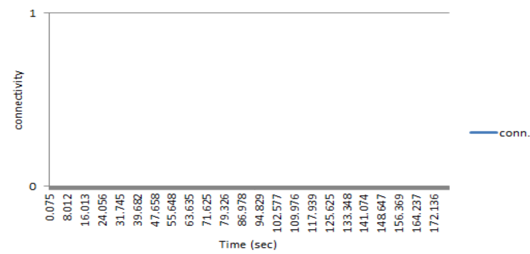

Figure 1: End-to-end connectivity N=7.

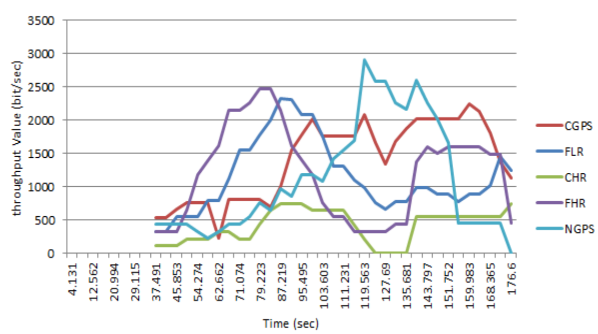

Figure 2: Throughput N=7.

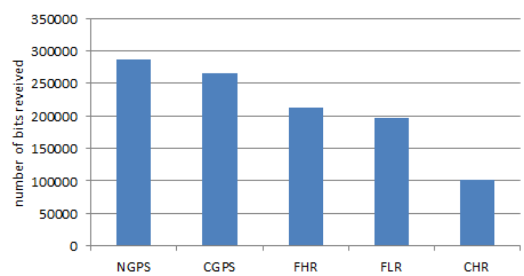

Figure 3 Total number of bits received N=7.

The graph shows that, in this experiment, the source and the destination are never connected by a closed path. Lack of end-to-end path in conventional MANETs prevents the data from being delivered. This issue has been addressed by introducing carrying feature implemented in c-protocols.

Figure 2 shows the throughput resulting from implementing different versions of the c-protocol. Although, in this experment there was no end-to-end path between the

ISBN: 978-1-61208-217-2

source and the destination, the c-protocol allowed data to go through and be delivered to the destination. GPS enabled versions, CGPS and NGPS, allow for a larger amount of data to be delivered than other versions.

Figure 3 shows the total number of bits successfully received at the destination as a result of using different versions of c-protocol. Again, the graph shows that the GPS enabled versions have the highest delivery.

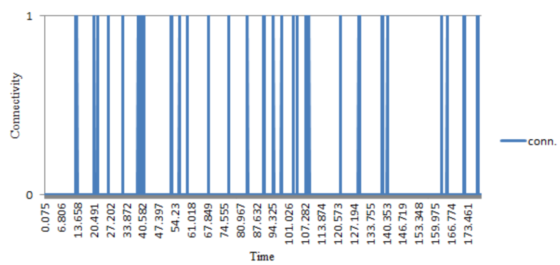

Figure 4 connectivity when N = 50

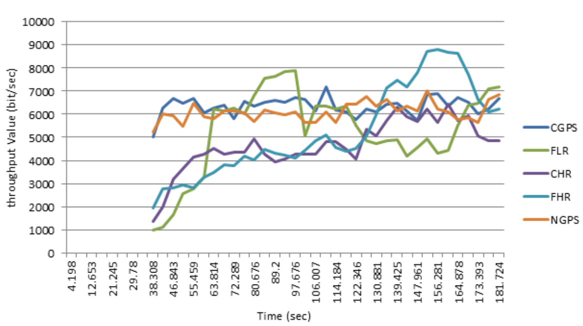

Figure 5 Throughput when N = 50

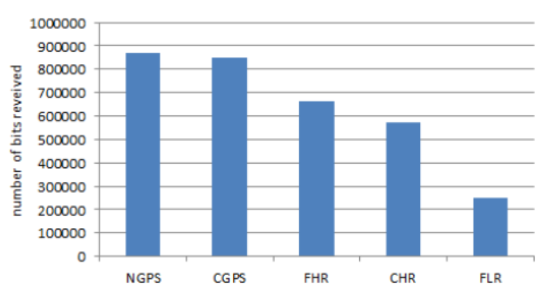

Figure 6 Total number of bits received N=50.

Figure 4 shows the end-to-end connectivity between the source and the destination in the second experiment when the number of nodes is 50 (N=50). The end-to-end connectivity occurs only periodically during this experiment.

The total time of end-to-end connectivity recorded in this experiment was 3600 ms. If we use 5kbps generation rate we can estimate the maximum number of bits (that can be potentially) delivered over the 5kbps connection is 18,000 bits. This would be the maximum throughput achievable using a conventional MANET.

The throughput for the c-protocol is shown in Figure 5. The total number of bits delivered is in Figure 6. For NGPS protocol, in our simulation we recorded almost 900,000 bits received in the course of the experiment.

The results show that introducing the c-protocol in MANET offers a significant advantage over conventional end-to-end protocols used currently in MANETs.

VIII. CONCLUSION

Currently in MANETs, it is essential to have an end-to-end connection in order to deliver packets; without end-to-end connectivity no packets would be delivered. C-protocol addresses this problem by introducing the carry (store/forward) mechanism that allows packets to be delivered even with the absence of an end-to-end connection.

There are two different types of protocols introduced in our work. They include the c-protocols that work without the need of using GPS location of the final destination and the c-protocols that make use of such a GPS location. Both types of c-protocols are reported to be able to deliver data to the destination even without end-to-end connection. The c-protocol versions that use GPS result in largest throughput in the simulated network.

IX. FUTURE WORK

The future work will include investigating the role of mobile node characteristics in a MANET, like the buffer size and the data retention time of the node, on the throughput of a MANET. We also plan to investigate other performance characteristics of the c-protocol like the propagation delay.

X. ACKNOWLEDGMENTS

This work is sponsored and funded by the Ministry of Higher Education of Saudi Arabia through the Saudi Arabian Cultural Bureau in Canada.

REFERENCES

[1] Fall, K. (2003) A Delay-Tolerant Network Architecture for Challenged Internets. Karlsruhe, Germany.*SIGCOMM'03*

[2] Yu, W., Wu, C., and Hu, X. (2010). Spray and Routing for Message Delivery in Challenged Networks, *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on*, pp.472-475

[3] Kulkarni, N.S., Gupta, I., and Raman, B. (2009). On Demand Routing Protocols for Mobile Ad Hoc Networks: A Review, *Advance Computing Conference, 2009. IACC 2009. IEEE International* , pp.586-591.

[4] Zhang, Z. (2006). Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges, *Communications Surveys & Tutorials, IEEE* , vol.8, no.1, pp.24-37.

[5] Abolhasan M., Wysocki, T., and Dutkiewicz, E. (2003). A Review of Routing Protocols for Mobile Ad Hoc Networks. Elsevier B.V.

[6] Latiff, L. A., and Fisal, N. (2003). Routing Protocols in Wireless Mobile Ad Hoc Network - a review, *Communications, 2003. APCC2003. The 9th Asia-Pacific Conference on*, pp. 600- 604

 ISBN: 978-1-61208-217-2

[7] Socievole, A.; De Rango, F.; and Coscarella, C.(2011). Routing approaches and performance evaluation in delay tolerant networks, *Wireless Telecommunications Symposium (WTS),* pp.1-6.

[8] Ekram, H., and Issariyakul, T.(2009). Introduction to Network Simulator NS2. Springer.

[9] Agrawal, D.P., and Zeng, Q. (2011). Introduction to Wireless and Mobile Systems.[Third Edition]. Brooks/Cole.

[10] Balakrishnan, M., Birman, K., Pleisch, S., and Renesse, R. (2006). MISTRAL: Efficient Flooding in Mobile Ad-hoc Networks. [Electronic Version]. MobHoc, Florence, Italy.

[11] Basagn, S., Conti, M., Gioradano, S., and Stojeneoric, I. (2004). Mobile Ad Hoc Networking. Willy-IEEE Press.

[12] Dai,F., Yang, S., and Wu, J. (2007). Logarithmic Store-Carry-Forward Routing in Mobile Ad Hoc Networks, Parallel and Distributed Systems, IEEE Transactions on , vol.18, no.6, pp.735-748.

[13] Seneviratn, A., and Sarikaya, B. (1998). Cellular networks and mobile internet, *Computer Communications,* 21(14), 1244-1255.

[14] Zhang, J., Li, W., Cui, D., Zhao, X., and Yin, Z. (2009). The NS2-Based Simulation and Research on Wireless Sensor Network Route Protocol, *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on* , pp.1-4, 24-26.