Australian Journal of Basic and Applied Sciences, 8(21) Special 2014, Pages: 42-47

AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Auto Mobile Ad-Hoc Mechanism in Delay Tolerant Network

[1]M.A. Azman, [1]S.H.S. Ariffin, [1]N. Fisal, [2]M. Abbas, [1]M.H.M. Fauzi, [1]S.K. Syed-Yusof

[1]*UTM-MIMOS Center of Excellence, Universiti Teknologi Malaysia, UTM Skudai 81310, Johor Bahru Johor*
[2]*MIMOS (M) Bhd, Technology Park 57000, Kuala Lumpur Malaysia*

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form
8 May 2014
Accepted 20 May 2014
Available online 17 June 2014

Keywords:

ABSTRACT

Delay Tolerant Network (DTN) is known as the solution to an Internet network where connectivity is an issue. There are existing project which uses smartphone as a physical transport of data between these intermittent networks. However, each smart phone needs to connect to each other via an infrastructure which will result in lower successful transfer rate. An automated mechanism is proposed for Ad-Hoc connection between the smart-phones is to ensure connectivity which leads to higher successful transfer rate. This report presents the automated mechanism called Auto Mobile Ad-Hoc Network in Delay Tolerant Network that is able to provide better reliability for data transmitted through DTN. This mechanism also allows application of the current Delay Tolerant Network to be connected to other networks and other nodes automatically using Ad-Hoc mode.

To Cite This Article: M.A. Azman, S.H.S. Ariffin, N. Fisal, M. Abbas, M.H.M. Fauzi, S.K. Syed-Yusof., Auto Mobile Ad-Hoc Mechanism in Delay Tolerant Network. ***Aust. J. Basic & Appl. Sci.,*** *8(21): 42-47, 2014*

INTRODUCTION

Internet plays an important role to connect people regardless of distance. However, not everyone has the opportunity to use the Internet, because of the challenging issues due to delay and disruptions. One of the most challenging issues for a certain geographical area to have communication infrastructure is the lack of infrastructural and facilities. Delay Tolerant Network (DTN) architecture for challenged network are categorized in (Fall, K., 2003) where DTN can be implemented in four different challenged networks shown in Fig 1. There are terrestrial mobile networks where users are mobile and commuting from one network to another, there is also exotic media networks where near-earth satellite communication that is prone to latencies. There are two other challenged networks in DTN which are the military Ad-Hoc network where intentional jamming might be the cause of the disconnection and network need to compete for bandwidth, and sensor/actuators network where this networks are consist of millions of low powered nodes and communication within this network often scheduled to conserve power.

There are projects which uses the DTN approaches such as in (Fall, K., S. Farrell, 2008; MacMahon, S. Farrell, 2009; Cerf, V., 2007) to apprehend these challenges. One of the projects is Bytewalla: DTN on Android (Ntareme, H.,) that uses mobile devices to transfer data between nodes. This project aims at connecting African rural villages using Android phones with delay-tolerant networking. The idea is that people carrying their mobile phones and travel from villages to cities will carry data with them. Once they reach the city, they will connect to any WiFi access point and upload the data. For the past years, Bytewalla had evolved from version 1 to 5, each with new features such as security and routing protocols. However, there are several issues in Bytewalla that can be improved. In the current version of Bytewalla both nodes (i.e transmitter and receiver) need to be connected manually to an infrastructure or the same network to allow data to be transferred. If both nodes are not connected in a same network, the data will not be forwarded, and will be stored in the transmitter node until the data expires, which affects the reliability of data transfer.

In the present report, we will discuss the implementation of an automatic mechanism called Auto Mobile Ad-Hoc function using Bytewalla. This mechanism is to improve the probability of successful bundle (a protocol data unit of the DTN bundle protocol) send and receive by enabling all nodes to connect to each other automatically rather than to wait for an available Access Point (AP). Each node will act dynamically according to the certain situation. This implementation is still using the conventional way of transferring data by adding another layer in the Internet protocol suite which is the Bundle Protocol (Scott, K. and S. Burleigh, 2007) and will be implemented on a smart-phone. This method is to enhance the existing Bytewalla software and to

Corresponding Author: M.A. Azman, UTM-MIMOS Center of Excellence, Universiti Teknologi Malaysia, UTM Skudai 81310, Johor Bahru Johor
E-mail: {sharifah, sheila, kamilah}@fke.utm.my, maffandyaz@gmail.com

Australian Journal of Basic and Applied Sciences, 8(21) Special 2014, Pages: 42-47

increase reliability data transfer as well as reduce transmission delay. Figure 1 shows the challenged network in DTN.

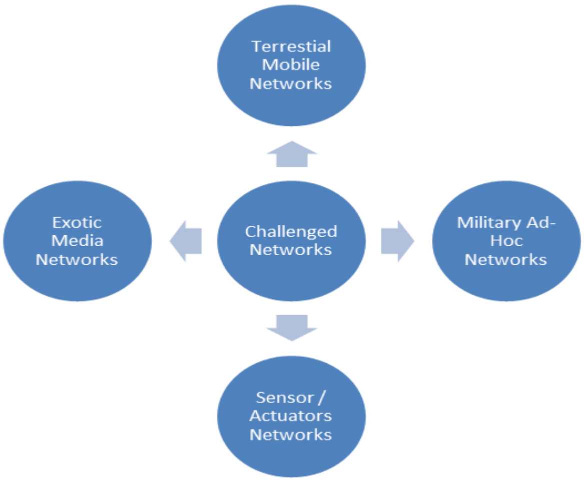

Fig. 1: Challenged network in Delay tolerant network.

The report is organized as follows: Section 2 describes the protocols used in DTN and the transmission routings. Section 3 presents the proposed concept of Auto Mobile Ad-Hoc Mechanism. In section 4 development and implementation in experimental DTN test bed is presented in detail. In section 5, results and discussion are elaborated. Last but not least, section 6 concludes the report.

1. Data Protocols and Routings in DTN:

The DTN is introduced to solve technical issues in networks that may lack of continuous network connectivity. The DTN is designed to operate effectively over extreme distances. An example of networks that are facing issue is the space-network, such as Inter Planetary (IPN) Internet Project (Akyildiz, I.F., 2003; Burleigh, S., 2003). The DTN is used to tackle problems such as the intermittent connectivity, a long and variable delay, asymmetric date rates, and high error rates by using the store and forward switching (Chuah, M., 2006; Macmahon, A., S. Farell, 2009). In order to implements this method, a device with a persistent storage is needed to hold the message indefinitely. Example of the device is a smart-phone. Figure 2 shows the store and forward method used.

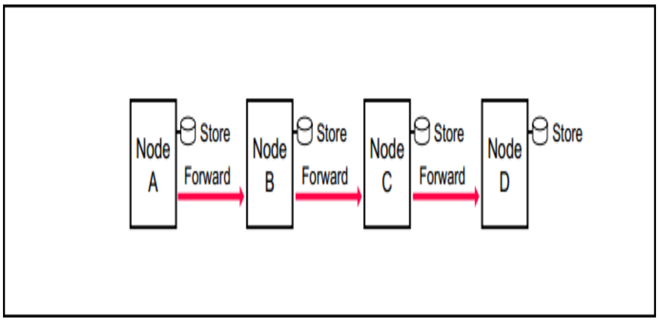

Fig. 2: Store and Forward Method.

It is possible for DTN to implement the store and forward method by overlaying a new protocol layer which is the Bundle Protocol (BP), RFC5050 on top of the Transport Layer. Basically, a bundle is a packet which contains a source-application user's data, control information and a bundle header. Each node is identified by Endpoint Identifiers (EID), which can be treated as address. There are several practical applications in DTN that have been done by other researchers. One of them is DakNet, which was developed by the MIT media Lab researchers (Pentland, A., 2004). Their goal was to provide remote villagers with low-cost digital communication by equipping busses with a mobile access point that travels between village kiosk and city collecting data.

Routing is the main aspect of DTN. It determines the delivery success rate and the delay of the bundles. Currently there are two routing strategies in DTN. The first strategy is flooding, where the message is replicated to several nodes in order for destination EID to be received. Each node acts as a relay to store the bundle until they are able to contact with another node. One of the routing protocols that use flooding concept is Epidemic routing (Jain, S., 2004), where nodes replicate and transmit messages to newly discovered contacts

Australian Journal of Basic and Applied Sciences, 8(21) Special 2014, Pages: 42-47

continuously. The problem with Epidemic routing is that it may congest the network in clustered areas while wasting network resource (bandwidth, storage and energy).

The second strategy is forwarding strategy which uses the best path to the destination by making use of network topology and local or global knowledge to find the best route path to deliver the message to the destination without replication which can lead to less bandwidth and consumption, and faster. One of the routing protocols that use forwarding strategy is the Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) (Delay Tolerant Network, 2012). This routing protocol uses algorithm which maintain the set of probabilities for successful delivery for known the destination. The downside for this protocol is that the actual value for the probability is crucial for successful bundle transfer. The routing protocol used in this project is the Epidemic routing. This is to ensure the successful transfer of bundle because of the high transfer probability rate in Epidemic routing.

2. Auto Mobile Ad-Hoc Network:

With the current DTN mechanism, all transmission from one mobile device to another need to be done manually even though there are potential node that can take the data to be forwarded later. A mechanism is added to solve this connectivity issue where the application will not connected to any network / node, if it is not initiated manually. This proposed mechanism will scan and connect automatically to another network / node.

The architecture in the experiment consists of three nodes (i.e. Android phones) which had been rooted (i.e. ability to access to system files and modify). Figure 3 illustrates the overall experiment scenario architecture. Each node will store and forward received bundle. The bundle will keep on forwarded until it has reached the destination EID. Each of the nodes is connected with each other through Ad-Hoc network.

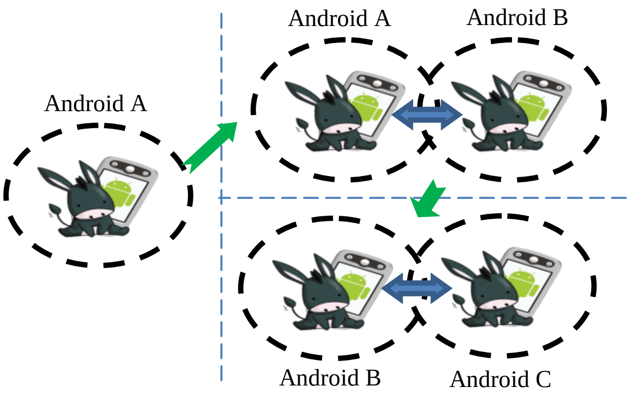

Fig. 3: Experimental Scenario of the Auto Mobile Ad-Hoc Network.

Figure 4 shows the state diagram for Auto Mobile Ad-Hoc mechanism in the DTN, which shows the flow for each node's network connection. The node will continuously check for its own network status, if the node is connected to any network, the node will stay connected and resume the checking phase. If the node is not connected to any network, the node will then scan and attempt to connect to any nearby Ad-Hoc node. If there are no Ad-Hoc nodes nearby, the node will then create its own Ad-Hoc network.

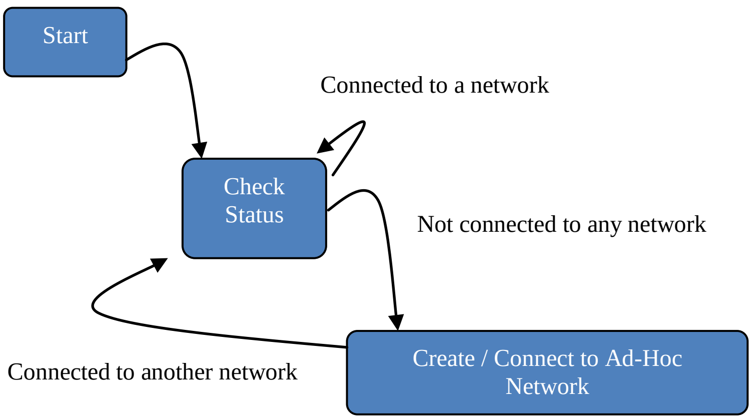

Fig. 4: State diagram of the auto Mobile Ad-Hoc Network.

3. Experimental DTN Test bed:

The experimental DTN test bed consist of DTN Servers and mobile node that have been configured and embedded with auto mobile Ad-Hoc function.

3.1 DTN Server:

The DTN 2 application uses DHCP server, DTN2 software on Ubuntu Platform. In order to send a bundle, an EID and a payload (message) is needed. Each node haves their own unique EID. For the smartphone, the EID is determined by the MAC address. An EID can be treated as an address for each node. Referring to Fig. 3, The bundle will be send to the mobile node that is in the same network with the server (i.e. network A), which is Android A.

3.2 Mobile Node:

For the mobile side, an application is called Bytewalla (Bytewalla is a DTN application for Android) is used. Each mobile node is installed with a modified version of Bytewalla which has the Auto Mobile Ad-Hoc functionality. Each mobile node is also rooted, to enable the use of Ad-Hoc in mobile node. Each node has its own unique EID. For the Android phones, the EID is determined by the MAC address. An EID can be treated as an address for each node. Referring to Fig. 3, a bundle will be sent to Android C which is the EID, originated from Android A. All of the nodes are not originally connected to each other. When Android A is in range with a nearby node (in this case Android B), Android A and Android B will initiate a connection automatically and the bundle will be send to the Android B from Android A. The bundle will then store and forwarded to the next node until it has reached the EID.

Fig. 5: Test bed set up.

RESULTS AND DISCUSSIONS

We investigate the effects of message size and number of nodes to the overall delay. The overall delay is calculated by calculating the time taken for a bundle to arrive to an EID from source. The measurement of the overall delay for each message size and number of nodes is carried out three times.

In the first experiment, we determined the delay based on the message size and different approaches, Android A and B were used as the nodes. Android A sent the bundle to Android B and both nodes are originally not in the same network. In this experiment we compared the delay of data transmission between infrastructure network and Auto Mobile Ad-Hoc network.

Figure 6 shows the results of the first experimental setup for the delay of transmitted data with variable data size. As the data size increased, the delay also increased. Using the infrastructure setup, the minimum delay for 200kB of message size is about 14 seconds. The delay increased gradually based on the data size to about 23 seconds for 1000kB of message size. For the Ad-Hoc setup, the minimum delay is about 34 seconds for 200kB of message size and increased gradually to about 42 seconds for 1000kB of message size. The difference in delay between Ad-Hoc and Infrastructure is because of the time needed for an Android phone to create/establish an Ad-Hoc network which takes around 30-32 seconds.

In the second experiment, we determined the delay based on the number of nodes and here, Android A, Android B and Android C were used. In this setup, all of the nodes are originally not in the same network. Android A will be the source sending the bundle to Android B or Android C which is the EID. In the second experimental setup, we compared between 2 nodes and 3 nodes of data transmission from source to destination.

In the 2-node setup, Android A and B are used. In the 3-node setup, Android A, Android B, and Android C are used.

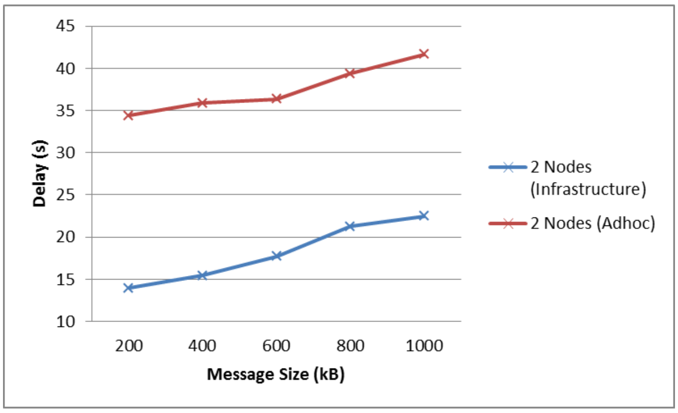

Fig. 6: Delay vs message size for infrastructure and Ad-Hoc mode DTN.

Figure 7 shows the results from the second experiment of the delay with variable number of nodes. For 3 nodes, the minimum delay is about 65 seconds for 200kB of message size which increases to about 83 seconds of delay for 1000kb of message size. As we increased the number of nodes, the overall delay also increased almost double. The delay increases from 2 nodes to 3 nodes because all of the nodes are not originally in the same network, thus each nodes creates their own Ad-Hoc which consume time.

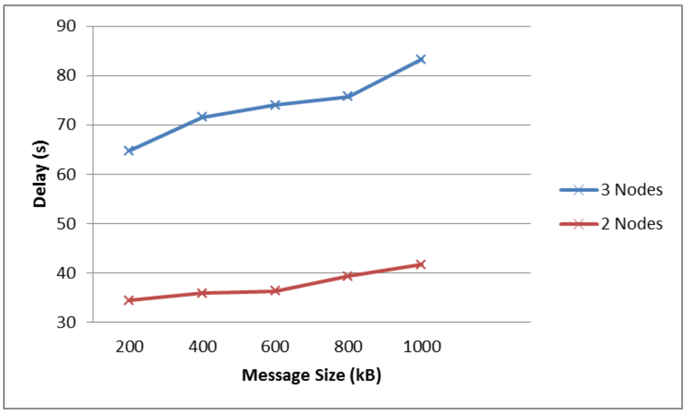

Fig. 7: Delay vs the number of mobile nodes the message are transmitted in DTN.

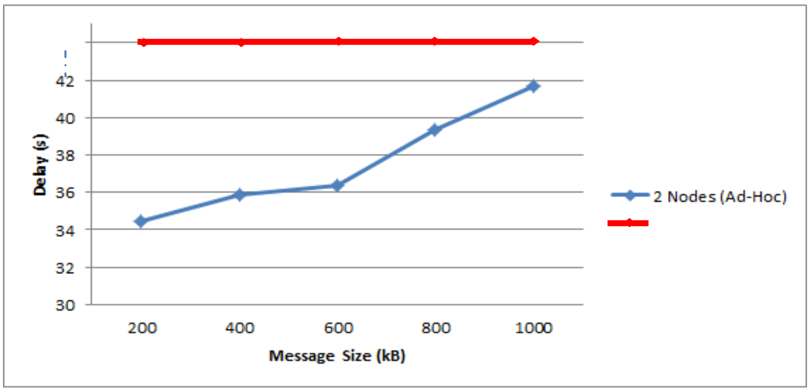

Fig. 8: Reliability of the Ad-Hoc function in DTN compared with the original version of DTN.

Figure 8 shows the delay compared between the original Bytewalla and the Auto Mobile Ad-Hoc Network. As shown in Fig. 8, the Auto Mobile Ad-Hoc, the delays increased as we increase the message size to a maximum of about 42 seconds. However, for the original Bytewalla, because of the nodes originally not in a same network, the nodes will not be connected to any other node and the bundle will not be send to the next node, hence, the delay is infinity.

4. Conclusion:

This report has presented the implementation of Auto Mobile Ad-Hoc function in the existing DTN application using Bytewalla software embedded in mobile nodes. The Auto mobile Ad-Hoc function is to improve the probability of successful bundle transfer to the destination by adding the ability to connect to each node automatically. This network concept can be very fruitful for the current situation since the bundle will keep on transferring to the next node instead of relayed only on one node. From the results shown, the Auto Mobile Ad-Hoc Network function is able to connect automatically to other nodes without user prompt which improved the probability of successful bundle transfer. However, there are delays involves, which is the time needed for a node to create its own Ad-Hoc network and connects to other Ad-Hoc network. The Auto Mobile Ad-Hoc Network certainly has its drawback by having about 32 seconds of delay for the Ad-Hoc network creation. However, it will ensure higher reliability of data transfer through the automated connection of nodes through Ad-Hoc network.

ACKNOWLEDGMENTS

The authors would like to thank all those who contributed toward making this research successful. The authors wish to express their gratitude to Ministry of Higher Education (MOHE), Malaysia Research Management Center (RMC) for the sponsorships, Telematic Research Group (TRG), Universiti Teknologi Malaysia and MIMOS (M) Bhd for the financial support and advice of this project. (Vot number Q.J130000.7823.4L550).

REFERENCES

Akyildiz, I.F., O.B. Akan, C. Chen, J. Fang, W. Su, 2003. “InterPlaNetary Internet: state-of-the-art and research challenges”, Computer Networks Journal (Elsevier Science), 43(2): 75-112.

Burleigh, S., A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst and K. Scott, 2003. “Delay-tolerant networking: an approach to Interplanetary Internet”, IEEE Communications Magazine, 41(6): 128-136.

Cerf, V., A. Hooke, L. Torgerson, R. Dust, K. Scott, K. Fall, H. Weiss, 2007. “Delay-Tolerant Networking Architecture”. IETF RFC 4838.

Chuah, M., P. Yang, B.D. Davidson, 2006. “Store-and Forward performance in DTN”, IEEE VTC.

Delay Tolerant Network, 2012. Protocols and Applications, A.V. Vasilakos, Y. Zhang, T. Spyropoulus, CRC Press, ISBN 1439811083.

Fall, K., 2003. “A Delay-Tolerant Network Architecture for Challenged Internets”, ACM SIGCOMM 2003, Germany.

Fall, K., S. Farrell, 2008. “DTN: An architectural retrospective”, IEEE Journal on selected Areas in Common, 26(5): 828-826.

Jain, S., K. Fall and R. Patra, 2004. “Routing in A Delay Tolerant Network”, SIGCOMM '04 Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, 34(4): 145-158.

Macmahon, A., S. Farell, 2009. “Delay and Disruption-Tolerant Network”, Internet Computing IEEE.

MacMahon, S. Farrell, 2009. “Delay-And Disruption-Tolerant Networking,” IEEE Internet Computing, 13(6): 82-87.

Ntareme, H., M. Zennaro, B. Pehrson, “Delay Tolerant Network on Smartphones: Applications for Communication Challenged Areas”, ExtremeComm11’, ACM Int. Conferences Proceedings Series, doi 10.1145/2414393.2414407.

Pentland, A., R. Fletcher, A. Hasson, 2004. “Daknet: Rethinking Connectivity in developing Nations”, Computer, 37(1): 2004, doi: 10.1109/MC.2004.1260729.

Scott, K. and S. Burleigh, 2007. “Bundle Protocol Specification”, IETF RFC 5050.