International Journal of Computer Applications Technology and Research
Volume 5–Issue 10, 657-663, 2016, ISSN:-2319–8656

Content Distribution for Peer-To-Peer Overlays on Mobile Adhoc Networks to Fuzzy Cross-Layer Approach

Maryam Ravaei
Department of Computer
Science and Research Branch
Islamic Azad University, Kish, Iran

Javad Mirabedini
Department of Computer
Islamic Azad University
Central Tehran Branch, Iran

Ali Harounabadi
Department of Computer
Islamic Azad University
Central Tehran Branch, Iran

Abstract: Peer-to-peer (P2P) networks existing on a MANET are a natural evolution since both are decentralized and have dynamic topologies. As MANETs grow in use due to the increasing popularity of wireless mesh and 4G networks, it is expected that P2P applications will remain as a popular means of obtaining files. Network coding has been shown as an efficient means of sharing large Files in a P2P network. With network coding, all file blocks have the same relative importance. This paper presents an efficient content distribution scheme that uses network coding to share large files in a P2P overlay running on a MANET. Peers request file blocks from multiple server nodes and servers multicast blocks to multiple receivers, providing efficient multipoint-to-multipoint communication. Simulation results show that compared to other common download techniques, the proposed scheme performs very well, having lower download time and energy consumption. Also, more peers participate in uploading the file, resulting in greater fairness.

Keywords: Wireless Ad Hoc Networks; peer to peer network; Multicasting

1. INTRODUCTION

Peer-to-peer (P2P) networks are implemented in the form of overlay networks, consisting of upper-layer connections between peers, independent of the underlying or substrate network .Overlay networks thus abstract their connectivity to a higher-level view of the peers that make up the network. The union of the logical connections that peers form with one another shapes the overlay. Peers communicate via the overlay, with queries and responses being routed over multiple hops.
Peers self-organize and handle ‘‘churn’’, the term used for the constant connections and disconnections of peers as they join and leave the network. As a result, P2P networks must be fault tolerant. On the Internet, peers exist on the access links, or ‘‘edge’’ of the network, and many P2P networks in use today scale to millions of simultaneous users.
In a mobile ad hoc network (MANET) devices form a self-organizing network with their peers and communicate with each other in multi-hop ‘‘peer-to-peer’’ mode instead of using a pre-existing infrastructure. All nodes act as both clients and/ or servers, meaning both clients and servers are mobile and resource- constrained.
Infrastructure less networks such as MANETs are essential in some environments, such as in disaster recovery, or military battlefields, where the communications infrastructure has been damaged or is non-existent. In other environments, such as at a company meeting where the participants would like to share documents, such networks may prove more convenient than using a wireless LAN. MANETs are used in combined infrastructure–
Infrastructure less networks such as wireless mesh networks and they are also expected to become an important part of the 4G architecture [1]. Ad hoc nodes will use multi hop communication to provide wider networking coverage and to connect to the fixed-backbone network, thus providing Internet services to areas without preexisting infrastructure. Recently, the Wi-Fi Alliance has announced a new specification called ‘‘Wi-Fi Direct’’ that allows devices to connect to one another directly, without the use of a base station [2].

The code -name for the specification was ‘‘Wi-Fi peer-to-peer’’. These networks are highly flexible and suitable for different programs and situations. Therefore, establishing temporary communications in them does not need a predetermined structure. Because of the limited transmission range of wireless mediums, traffic is relayed between a couple of middle notes for establishing connections between two far away nodes; therefore, these networks are also sometimes called mobile multi hop ad hoc networks.
In the field of MANETs a lot of research has been conducted some of which will be mentioned.
You et al. proposed an exploration routing protocol based on hop counts. In their study, an exploration function has been used to assist routing. Vu et al. [2] suggested an on-demand protocol considering the mobility in nodes and capability of detecting local faults.
Many P2P networks, particularly file-sharing networks, exist to distribute large data files, or content. In a MANET ,due to the non-existence of stationary, wall-powered servers, content that is distributed in P2P fashion is preferred over the traditional client/ server model due to the ease with which a MANET node’s
Band width can be overwhelmed and its limited energy. Therefore, a means by which content can be distributed that reduces bandwidth requirements and energy consumption is desirable. Furthermore, from the user’s perspective, being able to download content more quickly is beneficial.
In this paper we introduce a content distribution scheme that takes advantage of network coding and multi point to-multipoint communication to provide an efficient means of transferring files between peers in the network. It reduces download times, decreases energy consumption, and eliminates the rarest-block problem. Client peers locate server peers and download coded blocks, which enables them to retrieve content

in less time than downloading uncoded blocks. Servers trasnmit blocks via multicast to enable multiple clients to download simultaneously. The remainder of the paper is organized as follows. Section 'Background' provides some background on content distribution.

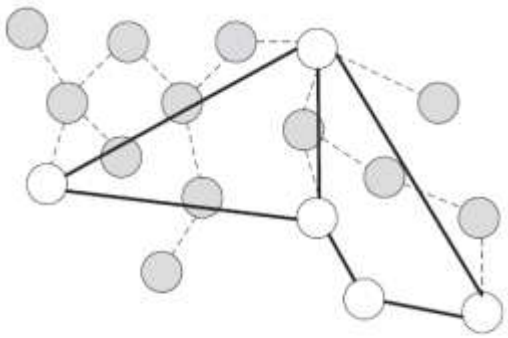


Fig. 1 An example of a peer-to-peer overlay running on a MANET

2. BACKGROUND

Content distribution involves transferring data, or content, from a set of source nodes to a set of destination nodes. Its definition often includes searching for the content as well, but in this paper, we do not include this aspect in our definition of content distribution, and instead focus only on the content transfer component.

In the simplest content distribution system, a client contacts a server and downloads the content from it. However, as the number of clients grows, the server's uplink bandwidth becomes a bottleneck and better techniques become desirable. Taking advantage of the uplink bandwidth of the clients that are downloading is a common way to more quickly distribute content. One approach is to use application-layer multicasting and allow all nodes to contribute to the distribution by uploading the parts of the file they possess [6]. The difficulty with this approach is the construction of the multicast tree, and whether to use source-based trees or a shared-tree. Another approach is to forego multicasting and instead allow nodes to connect directly to one another and send content, as in a P2P overlay. The most popular such approach isBitTorrent [7]. A downside to this approach is that the overlaynetwork does not closely reflect the underlay, resulting in inefficient use of network links. A generalization of the digital fountain idea involves the use of network coding. With this approach, both servers and clients encode data, which mitigates the problem of clients sending packets that may not be of use to one another, as in the digital fountain model. Codes are combined with one another by clients, which reduces the chance of a client receiving redundant data. Network coding was introduced by Ahlswede et al. [11] and linear network coding was introduced by Li et al. [12] as a technique to save bandwidth. There are several tutorial papers covering practical network coding [9–10]. Network coding is a form of information spreading in which nodes, instead of simply forwarding data packets, combine, or encode, several packets together using the XOR operation. The idea is to shift some of the work burden from the network to the nodes' computational abilities. This is a good tradeoff since network bandwidth is generally more expensive, i.e., slower, than computation. One major benefit of network coding is that encoded packets can be further encoded. This allows nodes to encode data without first decoding it, so that they can encode packets for which they do not yet have the completed data. The original packets are associated with a set of coefficients over the field. These coefficients are multiplied by the original packet, s bits at a time, and XORed with the same bit positions of the other packets to be combined. For example, given two original messages, M1 and M2, and two coefficients g1 and g2, the encoded message would be g1M1 XOR g2M2. The linear combination of the data, interpreted as a set of numbers over a finite field, is then transmitted in place of the original packets. The data of size s is considered as a symbol over the field F2s . Therefore, a packet of length L consists of L/s symbols. This encoded data, known as the information vector, is sent along with the set of coefficients, known as the encoding vector. The encoding vector is needed by the receiver to decode the data. The coefficients are selected at random, in a completely independent and decentralized manner. Even with a small field size, such as 28, the probability of selecting linearly dependent combinations is negligible [14]. With network coding, nodes receiving the encoded packets do not need to worry about obtaining specific packets. Instead, they must obtain a sufficient number of linearly independent, or ''innovative'', packets. The encoded packets, along with the given encoding vectors, are considered as a system of equations, where the original messages are the unknowns. For a set of N original messages and m received messages, we therefore have m equations with n unknowns, and so we must have MP n to solve the system. Furthermore, there must be at least n linearly independent equations. Any known technique, such as Gaussian elimination, may be used to solve the system and thus recover the original data.

In the suggested protocol, fuzzy logic will be used for routing multi-media files, which will be briefly explained.

Each type III fuzzy system includes four main parts: fuzzy rule base, fuzzy inference engine, fuzzifier, and defuzzifier (Figure 2.2). Each part will be reviewed below. Fuzzy rule base is a collection of fuzzy if-then rules. It is considered the heart of fuzzy systems, since other components of fuzzy systems are used for establishing these rules in an effective and efficient totality. As mentioned before, the fuzzy rule base includes if-then rules shown below:

If X_1 is A_1 X_2 is A_2, … and X_n is A_n, then y is B.

In this rule, As and B are fuzzy sets; and Xs and y are linguist variables of input and output in a fuzzy system.

A fuzzifier can be considered as a mapping from a real-valued point to another point in a fuzzy set of inference engine.

In the suggested protocol of fuzzy system input, four parameters have been considered: hop count, movement speed of nodes, available bandwidth, and energy at battery level of nodes.

For each input variable, two fuzzy sets having trapezoidal membership functions will be defined (H stands for high level, and l for low level), shown in Figures a-1, b-2. Trapezoidal membership functions are used because they are accurate. For output or stability rate of link, four fuzzy sets with triangular membership functions (H for high level, VH for very high level, M for moderate, L for low level, and VL for very low level) have been used which are shown in Figure 6-5.

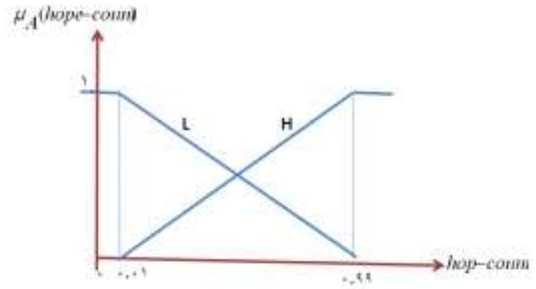


Figure a-1. Membership functions for input variables of hop count

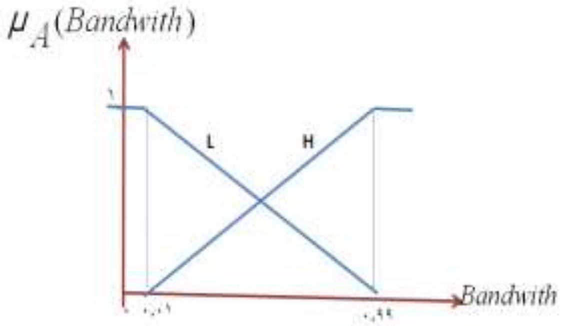


Figure b-2. Membership functions for input variables of bandwidth

3. PREVIOUS RESEARCH

A generalization of the digital fountain idea involves the use of network coding. With this approach, both servers and clients encode data, which mitigates the problem of clients sending packets that may not be of use to one another, as in the digital fountain model. Codes are combined with one another by clients, which reduces the chance of a client receiving redundant data. Network coding was introduced by Ahlswede et al. [6] and linear network coding was introduced by Li et al. [7] as a technique to save bandwidth. There are several tutorial papers covering practical network coding [9–10]. Network coding is a form of information spreading in which nodes, instead of simply forwarding data packets, combine, or encode, several packets together using the XOR operation. The idea is to shift some of the work burden from the network to the nodes' computational Abilities. This is a good tradeoff since network bandwidth is generally more expensive, i.e., slower, than computation. One major benefit of network coding is that encoded packets can be further encoded. This allows nodes to encode data without first decoding it, so that they can encode packets for which they do not yet have the completed data. The original packets are associated with a set of coefficients over the field. These coefficients are multiplied by the original packet, s bits at a time, and XOR e d with the same bit positions of the other packets to be combined. For example, given two original messages, M1 and M2, and two coefficients g1 and g2, the encoded message would be g1M1 XOR g2M2. The linear combination of the data, interpreted as a set of numbers over a finite field, is then transmitted in place of the original packets.

The data of size s is considered as a symbol over the field F2s .Therefore, a packet of length L consists of L/s symbols. This encoded data, known as the information vector, is sent along with the set of coefficients, known as the encoding vector. The encoding vector is needed by the receiver to decode the data. The coefficients are selected at random, in a completely independent and decentralized manner. Even with a small field size, such as 28, the probability of selecting linearly dependent combinations is negligible [14].

With network coding, nodes receiving the encoded packets do not need to worry about obtaining specific packets. Instead, they must obtain a sufficient number of linearly independent, or ''innovative'', packets. The encoded packets, along with the given encoding vectors, are considered as a system of equations, where the original messages are the unknowns. For a set of n original messages and m received messages, we therefore have m equations with n unknowns, and so we must have m P n to solve the system. Furthermore, there must be at least n linearly independent equations. Any known technique, such as Gaussian elimination, may be used to solve the system and thus recover the original data. Avalanche assumes the use of a centralized tracker which coordinates the nodes. In a MANET, such a node is impractical. Furthermore, Avalanche is designed for wired networks and does not consider energy use. Small et al. examine the use of network coding in a P2P network [8]. They conclude that peers are not very likely to receive independent blocks unless the block comes from a peer that has fully decoded the blocks first, meaning it has already received the entire file. Dependent blocks are a waste of bandwidth. The authors also show that the topology of the network has an effect on the usefulness of coding. Dependent blocks are received by nodes due to a common upstream parent and are a function of the ''aggressiveness'' of nodes. That is, the more frequently nodes send blocks to one another, the more likely they are to be dependent. However, decreasing the frequency increases delay.

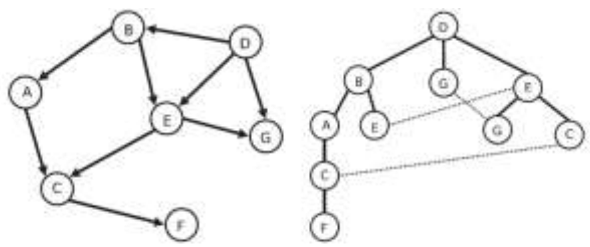
3.1. P2P–MANET content distribution

We propose an efficient content distribution system for peerto-peer computing in mobile ad hoc networks, to our knowledge the first one. The proposed scheme takes advantage of linear network coding to eliminate the rarest-block problem and multicasting to reduce the number of transmissions where possible. This results in reduced energy consumption and decreases download time.

There are currently two primary techniques for distributing data in P2P networks. We refer to these as the whole-file and the multi-block techniques. In the whole-file case, a peer, after locating the content on the network, will attempt to acquire the data from another peer which possesses the desired content. We will refer to this node as a server. The client peer will download the entire content from a single server peer. This is similar to the traditional client/server model, except that there may be several servers within the P2P network, and the client peer, once it has downloaded the entire file, will itself be available as a server peer for other nodes.

The multi-block technique splits up the content into many fixed size blocks of data. A client peer may then download blocks from any server peer such that all blocks that make up the file are eventually downloaded. This allows the client peer to download the content from several server peers simultaneously. An advantage to this technique is that peers which have only partially downloaded the content may still act as server peers for other nodes, able to upload the blocks they possess. Peers with the entire file are referred to as seeds.

One issue with this technique is determining which blocks to download from which server peers. A common problem is the ''rarest block'' problem, in which the file block with the fewest number of replications network-wide may be difficult to acquire. Therefore, many multi-block algorithms try to acquire this block first in an attempt to ''spread out'' the copies of blocks through the network. Network coding has been shown to improve performance in wireless environments [16 -17] as well as in P2P overlays [12– 13]. In wireless networks, network coding allows nodes to combine packets corresponding to different streams, increasing the information content of each transfer and as a result, total throughput. In mobile networks, this also saves device resources, such as energy. In P2P overlays, network coding eliminates the rarest block problem altogether. By encoding across all available blocks, and then transmitting the encoded blocks, all blocks become equivalent, and it is simply a matter for the client peer to acquire enough linearly independent or ''innovative'' blocks. Furthermore, in a MANET the use of multicasting becomes available, and so it is possible for a server peer to multicast its encoded blocks to many client peers. This allows a single transmission on its part to reach many client peers. Since the blocks are all identical, there is no need for client peers to locate and identify specific blocks. The requisite number of innovative blocks suffices. In turn, client peers may download from multiple server peers simultaneously, enabling an efficient multipoint- to-multipoint content distribution mechanism. System model we consider a peer-to-peer mobile ad hoc network, in which the peers are already joined in the overlay and the topology of the network has been determined. Any bootstrap mechanism and topology control algorithm may be used. the client peer determines this information to be part of the file query mechanism portion of the P2P network, not the content distribution part, and thus outside the scope of this paper.



(a)Topology view (b) Tree view

Fig. 2 Multipoint-to-multipoint content distribution.

4. SIMULATION RESULTS

We are attempting to determine the performance of distributing fairly large amounts of content. Therefore, we have selected two file sizes that each client peer is trying to download: 100 MB and 1 GB in size, split into either 1000 or 10,000 blocks each of size 100 KB. In all our simulations, we use 100 MANET nodes, with the number of nodes participating in a P2P overlay varying from 50 to 100 in. increments of 10. The network area is 1500 m · 1500 m, the transmission rate is 54 Mbps. Each experiment runs for up to four simulation hours, sufficient time for all peers to download the file for all systems tested. MAODV [16] is used as the multicast routing protocol and AODV [17] is used as the unicast routing protocol. The two ray ground radio propagation model along with an omnidirectional antenna are used by all nodes. The random waypoint mobility model is used, with all nodes evenly distributed in the simulation area. Nodal velocities are distributed according to a uniform distribution, with a minimum 1 m/s and a maximum 3 m/s, and a uniformly distributed pause time with mean 60 s, thus mimicking a moderate walking pace with infrequent stops.

Table 1 Energy consumption constants used in simulations.

m_{send}	1.89 μW s/byte
b_{send}	246 μW s
m_{recv}	0.494 μW s/byte
b_{recv}	56.1 μW s
$b_{sendctl}$	120 μW s
$b_{recvctl}$	29.0 μW s

The linear model proposed by Feeney [6]. Each MAC layer operation takes a certain amount of power as defined by cost= m · size + b where m is the incremental cost of the operation, b is the fixed cost, and size is the amount of data sent or received. The constants are obtained by physical measurements for a Lucent IEEE 802.11 Wave LAN PC Card from Feeney [6] and are summarized in Table 1. In this section, we present and discuss the simulation results for the numerous cases considered in our experiments.
The proposed algorithm had near-perfect success rates for all overlay sizes and seed counts. The use of network coding and multicasting allows essentially all peers to obtain the file. The scheme with network coding but no multicasting also has a fairly high success rate, though it tends to fall slightly as the overlay size increases due to the large increase in network traffic. Here, the proposed algorithm succeeds with a higher rate because its use of multicasting causes a slower increase in traffic as the overlay size increases. The rarest first block technique also performs reasonably well, with a decrease in success as the overlay size increases, also due The proposed algorithm had near-perfect success rates for all overlay sizes and seed counts. The use of network coding and multicasting allows essentially all peers to obtain the file. The scheme with network coding but no multicasting also has a fairly high success rate, though it tends to fall slightly as the overlay size increases due to the large increase in network traffic. Here, the proposed algorithm succeeds with a higher rate because its use of multicasting causes a slower increase in traffic as the overlay size increases. The rarest first block technique also performs reasonably well, with a decrease in success as the overlay size increases, also due to the increase In traffic. With more peers requesting the content, the network traffic causes congestion and dropped packets. For the whole file scheme, the success rate starts off lower and falls more severely as the overlay size increases. Since all peers must obtain the file only from those that have the entire file, initially all peers must obtain the file from the initial seed nodes. As the overlay size increases, the excessive traffic causes congestion around the initial seed nodes, and the success rate falls.
In this article, for simulating the suggested protocol of fuzzy system and comparing it with AODV protocol, OPNET Modeler 10.5 simulation software has been used. The structure and instructions for installation of the software are mentioned below.

For simulation, after running OPNET 10.5, from the 'Scenario' menu, 'Scenario Components', 'Import', we have entered 'OPNET Scenarios', and selected 'MANET-AODV Stream-mobility-Scenario-with -Demands'.Using 'Duplicate scenario' from 'Scenario' menu, 'AODV Stream' has been copied and the new scenario has been named 'Fuzzy System'.

In both scenarios of 'AODV Stream' and 'Fuzzy System', one variable has been defined for each node which determines for a node to use a standard 'AODV Stream' or 'Fuzzy System' protocol. This variable has been disabled for 'AODV Stream' scenario nodes, and enabled for 'Fuzzy System' scenario nodes. As seen in Figure 4.6, the simulated topology includes 25 nodes which have been distributed on an area of 1127 m by 1127 m. For the model of nodes' movement, the tragedy was first changed to "Vector" model to move in a vector form, and a random cyclical mode from 0 to 359 degrees has been defined for each node. The time of cycle and movement speed of each node have been considered 0-200 s and 1-10 m/s respectively. Figure 4.1 shows a view of node's editor.

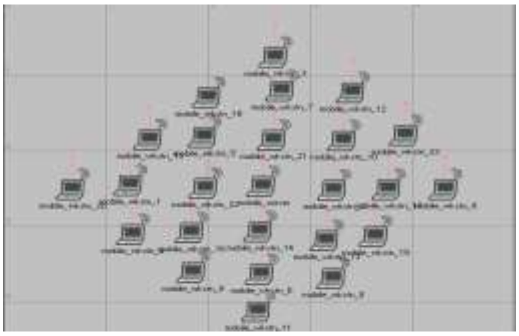


Figure 4.1. A view of network topology for the simulated mode

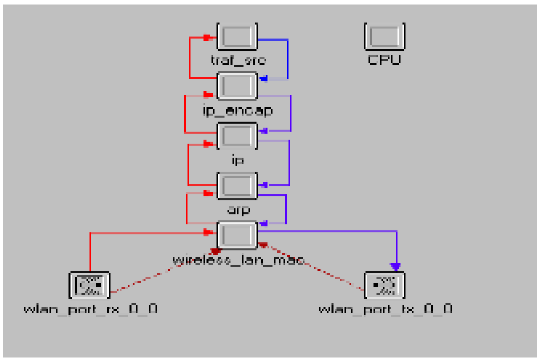


Figure 4.2. A view of node's editor for the simulated model

To make changes in the process model, 'Open' is selected from 'File' menu and then 'Process model' is selected for the type of files. By opening 'AODV Stream – rte' file form 'manet' folder, a diagram of mode with two modes of 'init' and 'wait' are observed which had already been available. A mode named 'TURN' has been added which is responsible for circulation of nodes in an expressed form. This process model is shown in Figure 4.3.

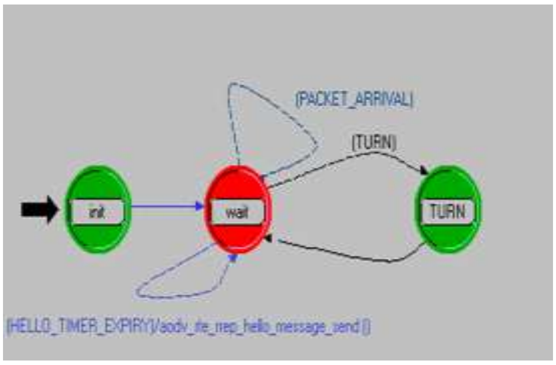


Figure 4.3. A view of processing model of simulated model

To see the code, the 'Edit Function Block' button is used. The main changes in codes for obtaining the suggested protocol includes calculating amounts of available bandwidth, energy level, node speed, hop count, radio frequency, antenna pattern field, signal productivity, and the level of radio sensitivity, and sending them to evaluation functions of reinforcement learning before updating the table in the functions: 'aodv_rte_rreq_pkt_arrival_handle'and'aodv_rte_rrep_pkt_arrival_handle' which are respectively the function of carrying package of route request and the function of carrying the package of route response.

The Fuzzy System software package includes two folders named 'manet' and 'op-model'; To run the simulated model, the contents of 'manet' folder must be copied in 'manet' folder in:

'C:\Program Files\OPNET\A\models\std\manet10.5'

and then the contents of 'op_model' must be copied in:

'C:\Document and Settings\user\op_models' After running 'Opnet', we should select 'Edit' menu, enter 'preferences', and copy the address below to the existing paths:C:\Document and

Settings\user\ op_ models \Rein forcimen t_ AODV Stream' Now, we can run 'Fuzzy System' project from 'Fuzzy System' folder by going to 'File' menu and choosing 'Open'.

After simulation, diagram of results are observable in different forms using the 'View Result' button.

The number of package losses in the network is because they have no access to a route toward destination. Over time, 'Fuzzy' has fewer losses of package compared with 'AODV'. In Figure 4.9 the numbers of package losses are demonstrated.

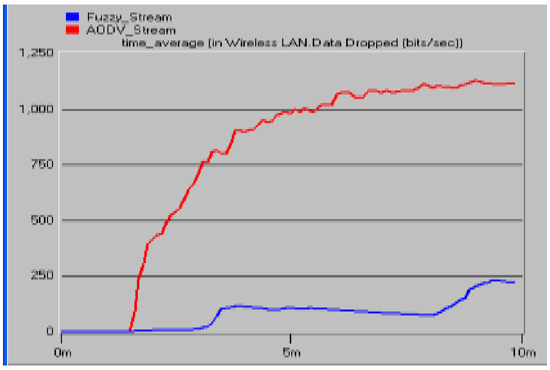


Figure 4.4. A view of the numbers of package losses

Mean time is calculated from the start of transmission of package from the source node until the time the package is delivered to a destination. A remarkable reduction in delay parameter in wireless ad hoc network is received for 'Fuzzy' compared with 'AODV'. By creating more stable routes, in other words, a couple of routes are created which are likely to take packages to destination, and there is no need for rediscovering the route. In Figure 4.10, the total delay in the network is shown.

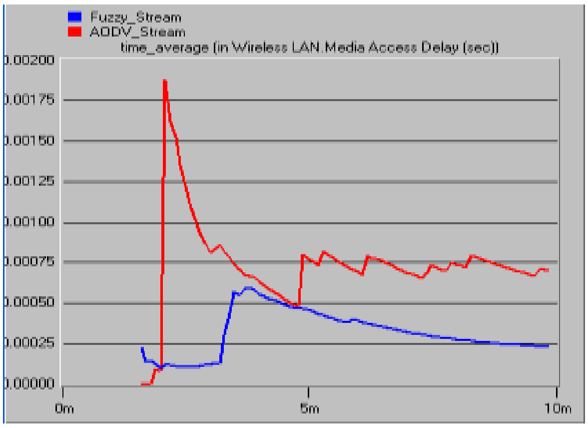


Figure 4.5. A view of total delay in the network

more in 'Fuzzy' than in 'AODV'. In Figure 4.11 transmission or delivery rate of packages are shown

5. DISCUSSION

In the suggested protocol of fuzzy system input, four parameters have been considered: hop count, movement speed of nodes, available bandwidth, and energy at battery level of nodes.

. Each fuzzy rule includes two parts; one is introduction: "if energy at battery level is low, hop count is high, bandwidth is low, and the amount of movement in nodes is low", and the other part is conclusion: "then the link stability is low". In the suggested method, fuzzy inference engine has been considered as minimum Mamdani. Mean time is calculated from the start of transmission of package from the source node until the time the package is delivered to a destination. A remarkable reduction in delay parameter in wireless ad hoc network is received for 'Fuzzy' compared with 'AODV'. This suggested protocol considers two important criteria for accessing more stable routes, and tries to create optimal routes to destination and increase their availability by creating routes with high available bandwidth and higher remained energy levels. In other words, it creates more stable routes for later more effective usages.

6. CONCLUSIONS

This paper proposed an efficient content distribution scheme for P2P–MANETs. It used network coding in addition to multicasting to transfer blocks of data between peers. Network coding created an equivalence of all file blocks, allowing nodes to obtain any blocks they could find from servers, without concern for locating specific blocks. This technique eliminated the rarest block problem. Furthermore, multicasting the blocks allowed servers to efficiently deliver them to many receivers and reduced transmissions at the server node.
The performance of the content distribution scheme was compared to downloading the entire file from a single seed, downloading blocks from multiple servers, and network coding Without multicasting. It was shown that the proposed scheme consumed less energy, provided speedier downloads, had a greater success rate than the competing algorithms, and was fairer in the sense that more peers participated in uploading blocks.

7. REFERENCES

[1] L. You, J. Li, C. Wei, C. Dai, J. Xu, L. Hu, "A hop count based heuristic routing protocol for mobile delay tolerant networks", Hindawi Publishing Corporation, Scientific World Journal, 2014.

[2] Vu TK, Kwon S, "Mobility-Assisted on-Demand Routing Algorithm for MANETs in the Presence of Location Errors", Hindawi Publishing Corporation, Scientific World Journal, 2014.

[3] B. Sivakumar, N. Bhalaji, D. Sivakumar, "A survey on investigating the need for intelligent power-aware load balanced routing protocols for handling critical links in MANETs", Hindawi Publishing Corporation, Scientific World Journal, 2014.

[4] H. Xia, Z. Jia, X. Li, L. Ju, EHM. Sha, "Trust prediction and trustbased source routing in mobile ad hoc networks", Ad Hoc Networks, 2013.

[5] A. Moussaoui, F. Semchedine, A. Boukerram, "A link-state QoS routing protocol based on link stability for Mobile Ad hoc Networks", Journal of Network and Computer Applications, 2014.

[6] K. Kumar, VP. Singh, "Power consumption based simulation model for mobile ad-hoc network", Wireless Personal Communications, 2014.

[7] S. Sridhar, R. Baskaran, P.Chandrasekar "Energy supported AODV (EN-AODV) for QOS routing in MANET", Social and Behavioal Science, 2013.

[8] A.Moussaoui, F.Semchedine, A.Boukerram "A link-state QoS routing protocol based on link stability for Mobile Ad hoc Networks", Journal of Network and Computer Applications, 2015.

[9] V. Kanakaris, D. Ndzi, D. Azzi, "Ad-hoc Networks Energy Consumption: A review of the Ad-Hoc Routing Protocols", Journal of Engineering Science and Technology Review, 2010.

[10] H. Xia, Z. Jia, X. Li, L. Ju, EHM. Sha, "Trust prediction and trust-based source routing in mobile ad hoc networks", Ad Hoc Networks, 2013.

[11] Y. Kumar Jain, R. Kumar Verma, "Energy Level Accuracy and Life Time Increased in Mobile Ad-Hoc Networks Using OLSR", International Journal of Advanced Research in Computer Science and Software Engineering, 2012.

[12] Sh. Tabatabaei, M. Teshnehlab, S.J.Mirabedini, "A New Routing Protocol to Increase Throughput in Mobile Ad Hoc Networks", Wireless Personal Communications, 2015.

[13] D.Palma, M.Curado, "Onto scalable Ad-hoc networks: Deferred Routing", Computer Communications, 2012.

[14] E. Atashpaz-Gargari and C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialist competitive, IEEE Congress on Evolutionary computation, Singapore (2007), 4661–4667.

[15] Braun, T. D., Siegel, H. J , A taxonomy for describing matching and scheduling heuristics for mixed machine heterogeneous computing systems, Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems, pp. 330335, 1998.

[16] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.,"BGSA: binary gravitational search algorithm". Nat. Comput. 9(3), 727–745, 2010.

[17] J.Kolodziej and F.Xhafa, Enhancing the genetic-based scheduling in computational grids by a structured hierarchical population, Future Generation Computer Systems,Elsevier 27(8)(2011),1035-1046

[18] A.Abraham,R.Buyya and B.Nath,Nature's heurustics for scheduling jobs on computational grids, Proceeding of the 8th IEEE International Conference on Advanced Computing and Communications, India(2008), 1-8.

[19] C.Weng and X.Lu, Heuristic scheduling for bag-of-tasks applications in combination with QOS in the computational grid, Future Generation Computer Systems 21(2)(2005),271-280.

[20] Chang, R. S., Chang, J. S., Lin, P. S., "Balanced Job Assignment Based on Ant Algorithm for Computing Grids", The 2nd IEEE Asia-Pacific Service Computing Conference, pp. 291295,2007.