Multipath extensions to the DYMO routing protocol*

Marga Nácher, Carlos T. Calafate, Pietro Manzoni
Departament of Computer Engineering
Polytechnic University of Valencia
Camino de Vera s/n, 46022 Valencia, Spain
email: marnacgo@doctor.upv.es , {calafate, pmanzoni}@disca.upv.es

Abstract

Multipath routing is a technique that can improve performance, specially in mobile ad hoc networks. Due to traffic dispersion it can perform load-balancing, minimize the energy consumed by nodes or prevent traffic analysis. In this work we focus on enhancing the DYMO protocol to support multipath routing. We study the impact of traffic dispersion on both UDP and TCP traffic when varying a set of parameters.

1. Introduction

A Mobile Ad Hoc Network (MANET) is a wireless network composed by a group of computing devices, usually called nodes, that communicate without requiring any sort of infrastructure or centralised management.

A MANET node is characterised by moderate/high degrees of mobility, limited resources such as energy, and a maximum transmission range. Multiple hops are typically required to exchange information between two MANET nodes. Due to the aforementioned properties of these nodes, routes are typically unreliable and short-lived.

Traffic dispersion is a technique that can help to prevent the threat of eavesdropping, to do load balancing or to minimize the energy consumed by nodes. Dispersion means that, for a same source-destination pair, communication simultaneously uses different paths (i.e., multiple paths) instead of a single one. Despite of its benefits, this technique has a strong impact on throughput, delay, number of hops traversed, and the routing overhead introduced.

In the literature we can find multipath routing solutions such as [1, 2, 3] which cope with problems such as the instability of routes, and also to increase performance and perform load-balancing. Other works focus on data encryption to enhance the data confidentiality [4, 5], in most of cases [6, 7, 8] defining a new routing protocol usually based on DSR [9] or AODV [10].

In this work we focus on the DYMO routing protocol [11], and we enhance it to support multiple path data delivery. We will study the interaction between the routing layer and UDP and TCP traffic in both single and multipath routing environments.

The rest of this paper is structured as follows: in section 2 we describe the different enhancements made to DYMO so as to support multipath traffic dispersion. Simulation results are presented in section 3. We begin with a comparison between single path and multipath routing with UDP traffic in section 3.1 and we conclude with a similar analysis focusing on TCP in section 3.2. Finally, in section 4, we present our conclusions along with references to future work.

2. ENHANCING DYMO TO SUPPORT MULTIPATH TRAFFIC DISPERSION

The Dynamic MANET On-demand (DYMO) protocol is a reactive routing protocol being developed within IETF's MANET working group.

Typically, all reactive routing protocols rely on the quick propagation of route request packets throughout the MANET to find routes between source and destination. While this process typi-

*This work was partially supported by the *Ministerio de Educación y Ciencia*, Spain, under Grant TIN2005-07705-C02-01

cally relies on broadcasting, route reply messages that are returned to the source rely on unicasting.

DYMO is basically an enhacement of the AODV protocol [10]; as for AODV every node records the next hop to send a packet to a specific destination. This technique prevents path selection to be done by the originator (as for DSR). The source node only records one possible next hop to every destination,and so we have to modify the route discovery process. In the multipath route discovery process, if several route replies arrive to the source through different neighbour nodes and different path identifiers, the DYMO agent keeps these nodes as next hops in the destination entry of its route table, which enables extending the path selection algorithm to make traffic dispersion possible.

The usual solution is to split traffic through several node disjoint routes. It is nevertheless important to differentiate among routes so as to maximise their degree of disjointness. Since we have to indicate the kind of disjointness before starting the discovery process and, in general, only few node disjoint routes are found, we decide to look for link disjoint routes.

As mentioned earlier, DYMO is an evolution of the AODV protocol. So, we based our changes on the route discovery process proposed for the AOMDV protocol [12]. In that paper the authors introduce the advertised hop count to prevent loops and a header extension (the last hop field) to identify the path. Last hop is the destination neighbour. The path is unequivocally identified with the pair next-hop/last-hop. This method presents the route cut-off problem resolved in [13] for the AODV protocol.

In the current work we solved this problem for the DYMO protocol in a simpler way. During the request phase, every intermediate node has to save the path to the request packet's originator in order to send the corresponding reply message to it. That's why every intermediate node registers all the paths with different last hops though they may arrive through the same neighbour (next hop in the path register). Figure 1 shows the request phase, the letter after the hyphen is the identifier (last hop) of that path. Nodes E and F will save two paths with Destination O and the same *Next hop*, the figure 2 shows this situation.

We now detail the reply process. When Destination node (D) receives a route request, it sends the reply back through the neighbour node from which it received the packet; the *last hop* value is the same one contained in the request packet. The first path used by each intermediate node with this last hop is the valid path and determines its next hop; the node removes the other paths with the same next hop although with a different last hop. We explain this with an example: supose that, first of all, node D receives the route request from E with last hop A. D sends a route reply with last hop A to E (RREP-A). Although D receives another route request from E, with a different last hop (in this case B), D discards the packet and it does not record this path. Similarly, if D hears a request from another node with last hop A, it obviously discards the packet too. Only if D receives a request from another node (F) with different last hop (B), does it save this path and send a new reply (RREP-B).

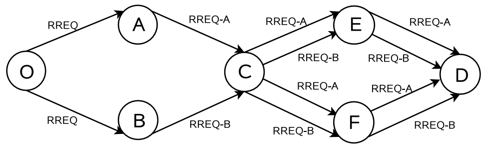

Figure 1. Request Phase

Destination O	...	LastHop A	NextHop C	...
		LastHop B	NextHop C	...

Figure 2. Nodes E, F route table, result of request

When node E receives RREP-A it searches the path to node O with last hop A and removes other paths with the same next hop as the selected path (e.g., it would remove the second row of the table in figure 2). Node F removes the first row when it processes the RREP-B. This way we solve the route cutoff problem.

After the route discovery process, every node will have one or more routes for every possible destination. Therefore they must decide how select them.

Our enhancements to DYMO's route selection mechanism also include a configurable parameter: the maximum number of different routes to be used (N).

This parameter (N) affects the discovery

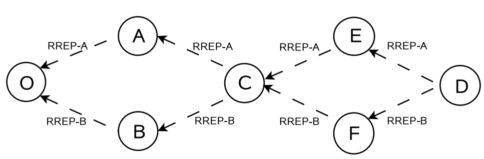

Figure 3. Reply Phase

phase. Nodes are not allowed to save more than N routes for every destination.

Our route selection works in this way: for each data packet the node always chooses the route with the lowest timeout value. It then updates the timeout of this route so that it becomes the route with the largest timeout value. Afterwards, the route with the lowest timeout will be a different one, and so routes are selected cyclically, in a round-robin fashion.

In this paper we are interested in characterising the impact of multipath routing with DYMO on both UDP and TCP traffic. Therefore, in our study, we also assess the impact of the aforementioned parameter on overall performance. This will be the topic of the next two sections.

3. PERFORMANCE ANALYSIS THROUGH SIMULATION

In this section we present the results of our experiments.

For our study we relied on the well-known ns-2 simulator [14] and we based on the University of Murcia's DYMO implementation [15]. We simulate a mobile ad hoc network composed of 60 nodes that move within a 1000m x 1000m area according to the random waypoint mobility model. Nodes speed value depends on the specific simulation, however the pause time is set to a null value in all the cases. The radio transmission range of nodes is 250m, and the MAC/PHY parameters are configured to simulate IEEE 802.11g technology at maximum speed (54 Mbit/s).

With regard to data sources we have six distinct source-destination pairs that send packets with a size of 512 bytes at a constant bit rate of 10 packets/second. Simulation time is 600s, though traffic is only started after the first 100 seconds to allow the mobility model to converge.

We proceed by assessing the impact of multipath DYMO on TCP performance. In this second set of experiments maintain node speed at a constant value of 10 m/s.

All the results depicted in the next section are an average over 10 simulation runs, obtained with 10 different scenarios.

In section 3.1 we determine the impact of multipath traffic dispersion against the default single path solution and with different nodes speed and CBR traffic. We show the results when generated traffic follows the TCP protocol in section 3.2.

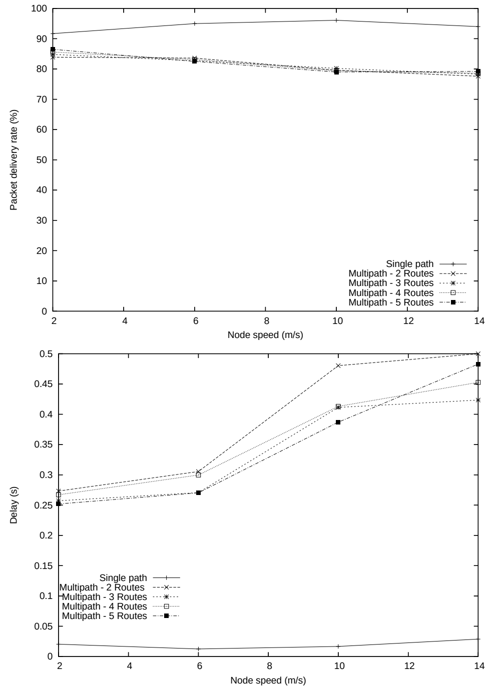

Figure 4. Impact of multipath traffic dispersion in terms of throughput (top) and end-to-end delay (bottom) with UDP

3.1. Impact of multipath data dispersion on UDP

We assess the effectiveness of the proposed multipath routing protocol in terms of packet delivery rate, end-to-end delay, number of hops and routing overhead at different levels of mobility. For this analysis we rely on CBR/UDP traffic to maintain the level of congestion in the MANET under control.

The purpose of this section is to assess the trade-off obtained between path dispersion and performance. In our experiments we vary the number of routes composing the auxiliary route cache (N) used by each node and the nodes speed.

Figures 4 and 5 show the simulation results related to packet delivery ratio, delay, number of hops and overhead when performing per-packet traffic distribution of UDP data but different node speeds. We observe a slight decrease in terms of delivery ratio as we increase the average node speed, which is due to a greater probability of link breakage. This also caused delay values to increase, but the average number of hops is maintained.

Focusing on the comparison between single

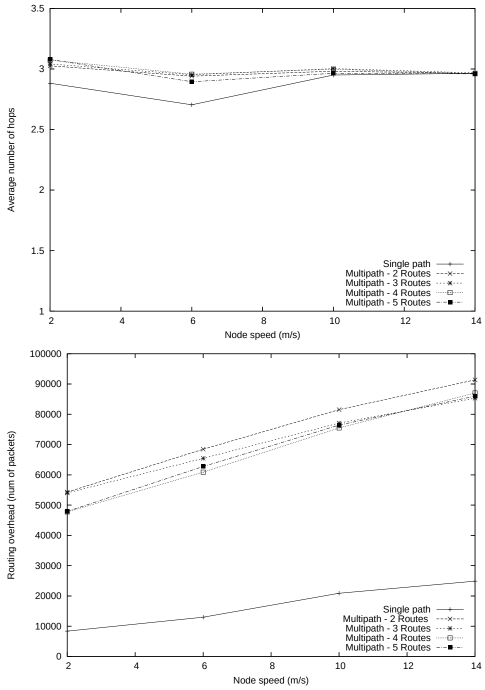

Figure 5. Impact of multipath traffic dispersion in terms of average number of hops (top) and routing overhead (bottom) with UDP

vs multi path DYMO, we observe that in each multipath simulation, the use of paths that are worse than the single-path one (i.e., the shortest route) provokes a decrease in terms of throughput independently of the nodes speed and the number of routes used in multipath data delivery. Moreover, path dispersion increases end-to-end delay values. This is due to the modifications introduced in the route discovery process, causing every intermediate node to add processing time. In terms of the average number of hops, an expected increase occurs since the pool of routes used includes routes that are of equal or greater length than the one used by default.

An important conclusion from these experiments is that the set of simultaneous routes used (N) does not have a significant impact on either throughput, delay or number of hops. This occurs because the number of disjoint routes available is typically very low.

In terms of routing, the proposed algorithm requires simultaneous routes to be found and maintained. So, we observe a steady increase of routing overhead with speed.

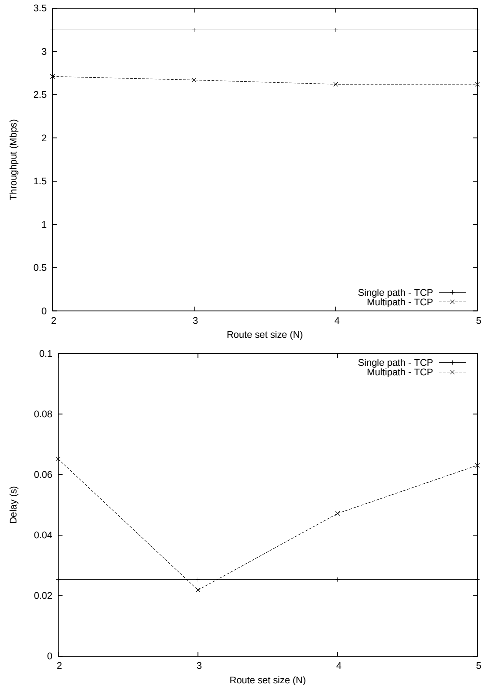

Figure 6. Impact of multipath traffic dispersion on TCP throughput (top) and delay (bottom)

3.2. Impact of multipath data dispersion on TCP

In this section we fix node speed at 10 m/s and we repeat the simulations made for UDP, this time with TCP traffic. Concerning traffic, the six distinct source-destination pairs transfer FTP/TCP traffic during the entire period of activity, from second 100 on. Figure 6 shows the simulation results related to throughput and delay and figure 7 shows the number of hops and routing overhead simulation results.

We observe that, despite multipath data delivery may cause packet reordering (which TCP interprets as congestion), TCP throughput only decays slightly.

Just like in the previous section, we observe a decrease in terms of throughput, an increase in terms of delay and the expected increase in terms of average number of hops. Causes are explained in section 3.1.

4. Conclusions

In this paper we propose enhancements to DYMO's route discovery and packet forwarding

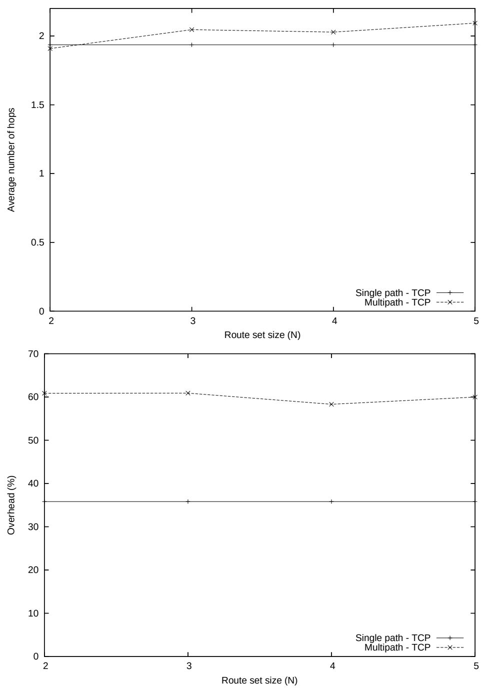

Figure 7. Impact of the route set size in terms of average number of hops (top) and normalized routing overhead (bottom) with TCP traffic

processes in order to support multipath routing and traffic dispersion policies, which are tunable through a set of parameters.

We use a simpler algorithm to solve the cut-off problem than previous solutions used for the AOMDV which is the basis for our proposal.

Based on a set of experiments conducted in a MANET environment, we conclude that, by introducing multipath routing, the throughput of both UDP and TCP connections is reduced; however, results show that the expected throughput decay is under 20%. Besides, we find that limiting the number of simultaneous paths used has little influence on throughput.

We find that all delay, average number of hops and routing overhead increase compared to the single path solution. This is expected since we now find alternate paths, usually with more hops than the original single route.

As future work we plan to compare multipath DYMO with multipath DSR in the scope of secure and anonymous routing.

References

[1] A. Tsirigos and Z.J. Haas. Multipath routing in the presence of frequent topological changes. *IEEE Communications Magazine*, 2001.

[2] H. Lim, K. Xu, and M. Gerla. Tcp performance over multipath routing in mobile ad hoc networks. In *IEEE International Conference on Communications (ICC'03)*, 2003.

[3] C. T. Calafate, M. P. Malumbres, and P. Manzoni. Mitigating the impact of mobility on h.264 real-time video streams using multiple paths. *Journal of Communications and Networks*, 2004.

[4] W. Lou, W. Liu, and Y. Fang. Spread: Enhancing data confidentiality in mobile ad hoc networks. In *Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 2404–2413, 2004.

[5] S. Bouam and J. Ben-Othman. Data security in ad hoc networks using multipath routing. In *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications. PIMRC 2003*, pages 1331–1335 vol.2, Beijing, China, Sep 2003.

[6] N. Wisitpongphan and O.K. Tonguz. Disjoint multi-path source routing in ad hoc networks: transport capacity. In *IEEE Semiannual Vehicular Technology Conference*, 2003.

[7] P. Papadimitratos, Z.J. Haas, and E.G. Sirer. Path set selection in mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002.

[8] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *IEEE International Conference on Communications, 2001 (ICC)*, 2001.

[9] D.B. Johnson, D.A. Maltz, and Y. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt, July 2004.

[10] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. http://www.ietf.org/rfc/rfc3561.txt, July 2003.

[11] I. Chakeres and C. Perkins. Dynamic manet on-demand (dymo) routing. http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-05.txt, June 2006.

[12] M. K. Marina and S. R. Das. Ad hoc on-demand multipath distance vector routing, July 2002. Technical Report.

[13] Y. Yuan, H. Chen, and M. Jia. An optimized ad-hoc on-demand multipath distance vector (aomdv) routing protocol. In *Conference on Communications*, 2005.

[14] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[15] Implementation of the dymo routing protocol-dymoum-0.3. http://masimum.inf.um.es/?Software:DYMOUM.