Dynamic Multipath Allocation in Ad Hoc Networks

Yosi Ben-Asher , Sharoni Feldman
Computer Science Department, Haifa University, Israel.
{Yosi, Sharoni}@cs.haifa.ac.il}

Moran Feldman
Computer Science Department, Technion, Israel.
moranfe@cs.technion.ac.il

Abstract
Ad Hoc networks are characterized by fast dynamic changes in the topology of the network. A known technique to improve QoS is to use Multipath routing where packets (voice/video/…) from a source to a destination travel in two or more **maximal** disjoint paths. Observe that the need to find a set of maximal disjoint paths can be relaxed by finding a set of paths S wherein only bottlenecked links are bypassed. In the proposed model we assume that there is only one edge along a path in S as a bottleneck and show that by selecting random paths in S the probability that bottlenecked edges get bypassed is high.
We have extended the MRA protocol to use multipath routing by maintaining a set of random routing trees from which random paths can be easily selected. Random paths are allocated/released by threshold rules monitoring the session quality. The simulations show that: 1) session QoS is significantly improve, 2) the fact that many sessions use multiple paths in parallel does not depredate overall performances, 3) the overhead in maintaining multipath in the MRA algorithm is negligible.
Keywords: Ad-Hoc, wireless, streams, multi-path.

1. Introduction

Ad hoc networks [8] are targeted to create a communication infrastructure in a dynamic environment characterized by high mobility of nodes with a limited transmission range. Routing problems caused by dynamic changes of links/edges in the network topology are handled by common ad hoc routing algorithms such as Ad-Hoc Distance Vector [9] (AODV). Another problem that is not handled by these algorithms is congestion where some edges become a bottleneck slowing down the rate in which packets are received (QoS). Overcoming congested edges in ad hoc networks is harder than in static networks due to the rapid changes in the network's topology. Multipath routing is a basic technique used to bypass congested edges in networks and improve QoS. There are two ways to bypass a congested edge: Abandon the current path and use an alternative path (hopefully without congestion) or use an additional path transferring packets in parallel through both paths. Using parallel paths in ad hoc networks has been proposed by Lee and Gerla[5]. We observe that obtaining disjoint paths maybe too costly in ad hoc networks and propose instead to bypass a few most congested edges along the current path.

Consider for example the following fragment of an ad-hoc network (Figure 1) where packets of a given session S are transferred. Each edge is labeled by a number representing the "potential capacity" of this edge in relation to the given session. The potential capacity indicates the bandwidth available for S' packets on that edge. Ideally all edges could allocate the maximal potential capacity (12 in this example) however some edges are congested and can allocate less than 12. The minimal potential capacity along a given path determines the overall throughput of that path. In here, a congested edge with potential capacity 2 dominates all other "high" potential capacities (10, 7, and 12) of the selected path. Using an additional path 10-5-3-8-12 increase the potential capacity of the session to 5. Clearly selecting maximal disjoint paths will increase the potential capacity however this may be a waste of bandwidth since we only have to bypass the most congested edge (assuming there is only one such edge) to improve the overall potential Capacity.

Let u,v be two nodes in a random radius graph and let $P_{u,v}$ be the set of all shortest paths connecting u and v. For a given edge e on a path from u to v let x be the number of edges in a cut through e then the bypassing property holds if the number paths in $P_{u,v}$ passing through e is 1/x. Thus the probability that a random path from $P_{u,v}$ will bypass the "most congested" edge e is 1/x. This property of the set of all shortest paths is backed up by simulation results and some additional formal claims. Note that intuitively random selection of the additional path implies that:

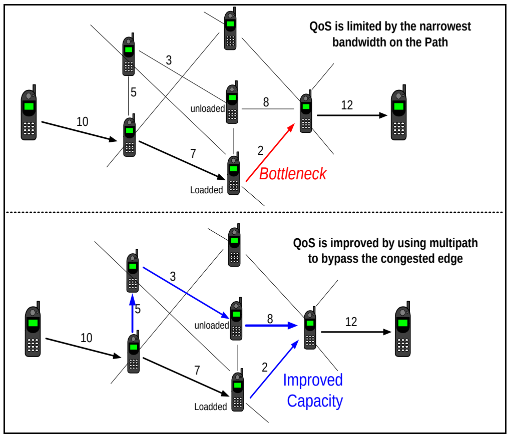

Figure 1: Multipath Example

- If congestion changes from e to e' then the probability that the additional path bypasses e' remains high.
- Every time we select a path at random, the probability it bypasses a congested edge remains high even if the underlying topology is utterly changed.

The random selection is therefore suitable for ad-hoc networks where the topology and the set of congested edges can rapidly change.

Focusing on shortest paths is natural as ad hoc routing algorithms attempt to rout packets through shortest paths. In addition we will show that extending the family of shortest paths destroys the bypass property and hence in some weak sense selecting from the set of shortest paths is essential.

A straightforward use of multipath abilities is to let every session use k > 1 random paths, however this may increase the overall congestion as each session attempt to increase its potential capacity. The solution is to allocate bypassing paths only to session whose QoS is dropped bellow a certain threshold and de-allocate bypassing paths when the QoS of session improves. For sending a stream of packets (video/voice/pictures) the QoS of a given session is measured by counting the current size of "holes" (missing packets) in a stream buffer at the destination. The full streaming protocol we have used combines three types of requests from the destination to the source:

1) Re-transmission of lost/delayed packets in "small holes". 2) Rewind the transmission from a given point. 3) Allocating/de-allocating a new bypassing path.

Our experimental results prove that using multipath over the MRA algorithm can improve session quality by 40% without increasing the overall congestion (i.e., the use of mulipath transmissions at some sessions did not degrade the QoS of other sessions). In particular the simulations prove that the combination of multipath transmissions and streaming control is essential for maximizing the QoS.

The main contribution of this work is the use of a random selection of the shortest path out of a current group of “shortest paths” in order to bypass a congested edge, as opposed to the allocation of maximal disjoint path. The use of random paths significantly reduces the overhead involved with multipath transmissions compare to maintaining disjoint paths. A second contribution is the combination of allocating bypassing paths in a streaming protocol.

2. Related Work

Multipath routing protocols have been deeply explored in wired networks [10][11]. This exploration broadened in Ad Hoc networks due to the special properties of these networks. The well known AODV[4] routing protocol is the origin for a group of multipath routing protocols. The AODV-BR[6] as a reactive routing protocol builds routs on demand via a query and reply procedure. The primary route and alternate paths are established during the route reply phase. The main idea in AOMDV[1] is that multiple link-disjoint paths are computed from the source top destination through a modified route discovery process.

An attempt to solve the aging problem is presented by the MP-AOMDV routing algorithm [2]. It attempts to solve the problem of unrefreshed paths by periodically revalidating each of the alternate paths, while introducing a minimum of control overhead. AODVM[7] protocol has the ability to find node-disjoint paths using a subset of reliable (R-nodes).

Common to all protocols described above is the effort to allocate a full path, mostly a disjoint one. The MRMA protocol presented here takes a different direction. It is targeted to allocate efficiently additional paths to a support a session. The exclusiveness of the

MRMA is its ability to select on demand new one or more paths that bypasses the congested edges.

3. The Metrical Routing Multipath Algorithm

The MRA[3] algorithm is the foundation for the MRMA multipath algorithm. In this section we will describe the extension to the MRA algorithm upgrading it into a multilevel algorithm supporting retransmissions in the first level and multipath in the second level. The MRA algorithm organizes the nodes in the field in rooted trees with shortcuts (STSC).

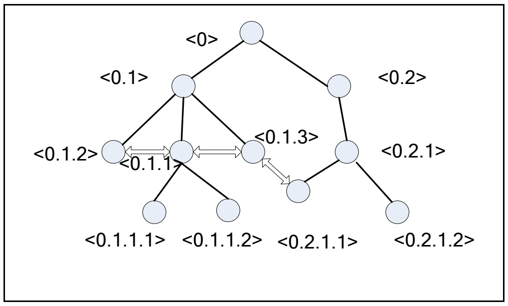

Figure 2: Sample tree

Figure **2** presents a STSC (the root gets the address <0>, the root children gets the address <0.X> etc.). The arrows between nodes <0.1.2>, <0.1.1> <0.1.3> and <0.2.1.1> presents shortcuts. A session is not compelled to use the structure induced by the tree structure. It uses shortcuts to minimize the path length or avoid potential bottlenecks. Only nodes that belong to one tree can create sessions among themselves. To ensure the maximal connectivity, all nodes will organize themselves in a single tree. In the rest of this paper the terms STSC and "tree" are interchangeable.

3.2 Random creation of STSCs in the MRMA algorithm

The MRMA algorithm utilizes the trees creation process of the MRA algorithm with minor changes. While the MRA algorithm uses a single tree to create a session path between any two nodes. The MRMA algorithm is able to create a set of independent trees on the same nodes collection. The creation rules select for example the node with the highest node-id to be the root in the primary tree and the node with the lowest node-id to be the root in the first secondary tree. Note that the node-ids are unique. Similar rules apply to the creation of the other secondary trees. The control information maintained by the primary tree is used also by the secondary trees and as a result there is a significant time saving in the time required to allocate a secondary path compared to the time required to allocate a primary path. The foundation of the multi-path algorithm is its ability to establish dynamically and instantly distinctive routes between the session end points (*S* and *T*). Our solution minimizes the overhead needed to create and maintain multiple session paths and is based on replication of the STSCs mechanism of the MRA.

The first STSC is the "primary" STSC and the other STSCs are "secondary" STSCs. Targeted to keep the management of the secondary STSCs light, we used the messages and control structures of the primary STSC for the control of the secondary STSCs. The number of secondary STSCs is not limited.

3.3 Sessions stream protocol

A session between two nodes is initiated by the originating node which searches and allocates a session path. We will define this path as the "primary" path. The MRMA allocates a bi-directional session path induced from the primary tree structure. The direction from $S \rightarrow T$ is used to transfer data packets while the direction $T \rightarrow S$ on the primary path is used to transfer the protocol control actions described later. As soon as the originating node gets the indication that the session creation process succeeded, it starts to transmit data packets towards the target node. Every data packet is numbered consecutively. The target node monitors constantly the inbound stream of packets. An audit process in the target node (*T*) evaluates continuously the inbound packets and stores them according to the packet sequence number in a test buffer (TB). Based of the analysis on the density of the missing packets done by the audit process, the target node can take the following actions:

1. Proceed with the transfer process. This will not trigger any corrective actions as no packets were lost.
2. Request the originating node to retransmit specific packets.
3. Request a range retransmission (for example retransmit packets 3 to 7).
4. Activate a secondary session path. This will trigger the allocation process of a secondary path. As soon as the path is established, the traffic will be distributed between the primary and secondary path. Every one of the existing paths can be supported by another secondary path that will carry "half" of the traffic of the

original path. Figure 3 presents a full process of allocating secondary paths.

5. Terminate a secondary session path. This action is activated when it is possible to merge back two paths into a single path. The path merge process runs in reversed order to the activation process of new paths.

During the allocation process of a new path, the participating nodes reserve resources for the path, when the main resource is bandwidth. Every node on the path links the incoming and outgoing sides of the path. As soon as the path is established, the source starts to send packets as a stream without waiting for an explicit acknowledge for every package. Every packet sent from the source node (*S)* gets a sequential number, which is used as an index to a TB in the target node. A packet received in the target node is stored in a slot, according to its number, inside the TB.

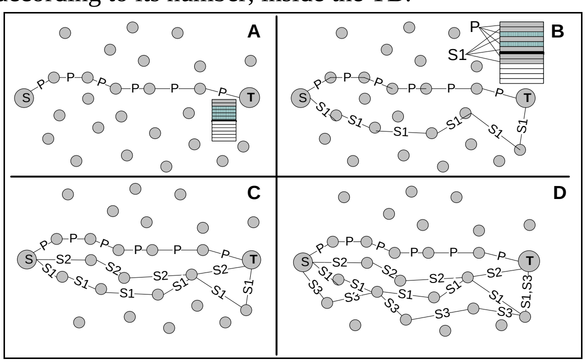

Figure 3: Session path allocation (P=Primary path, S=Secondary path)

The TB in the target node uses a double window mechanism. While storing the received packets in one (active) window, it analyzes the content on the other (inactive) window and if needed initiate corrective actions. The source node transmits packets in a constant rate of 3 windows per second and this is the rate that the target node expects to receive the packets. The first level of corrective actions will resume the transmission of missing packets that will fill the empty slots in the inactive window. A trigger to switch the windows will be an event where the last empty slot of the active buffer gets a packet or the last packet index is higher than the indexes in the active window. In this case, the content of the inactive window will be stored on disk, and the buffers will switch. A packet will be declared as an unrecoverable packet when the non active window must change its state to an active window.

In case that the Quality of Service interpreted to number of missing packets and their density of a path crosses a predefined threshold, the target (*T)* node will initiate the allocation of an additional path. In case that the QoS betters and crosses an appropriate threshold a merge process will merge secondary paths.

A secondary path will merge back with its original path when the QoS betters and crosses a threshold.

<u>Events Impact on Delays</u>

Table 1 depicts the four events related to the management of packet loss and creation of a secondary path. From the analysis of the events we learn that the saving of time while using the multipath, results from the decrease of traffic on the overloaded path and its diversion to secondary paths implies less packets loss. This traffic decrease on the primary path increases indirectly the QoS of other sessions that use nodes on the primary path. The allocation of a secondary path in not engaged with delays as this process is performed in parallel to the existing flow of messages. Only after the establishment of the secondary path, 50% of the traffic will be diverted to the secondary path.

Table 1: Events Impact on Delays

Event	**Recovery Method**	
	Retransmissions	**Multipath**
Allocating secondary path	***Not Applicable***	***No delay***
Packet transmission On secondary path	***Reduces the number of retransmissions on primary path***	***Reduced load on all paths. less retransmissions***
Packet retransmission	***Delays transmission***	***Not Applicable***
Request retransmission	***No delay***	***Not Applicable***

Similarly, a request for retransmission of a lost packet does not "cost" time. However, the resending of a lost packet results in a delay that will be aggregated to the total session time.

4. Justification of the bypassing property

Let $P_{s,t}$ be the set of all shortest paths between two nodes *s* and *t* and let *u* be a congested node in a pre-designated path in $P_{s,t}$. For a given node *u* in $P_{s,t}$ the bypassing property for the triplet <s,u,t> indicates that if we select a random path *p* in $P_{s,t}$ then the probability that *p* bypasses the congested node *u* is greater equal ½ (assuming that there are at least two nodes in the level of *u* in the induced graph of $P_{s,t}$). An empirical justification for this property can be easily obtained. Figure 4 presents the results of a set of simulations performed on a theater hosting from 50 to 300 nodes. The “good triplet” presents the number of triplets <s,u,t> satisfying the bypassing property and “bad triplets” presents the number of triplets that do not satisfy this property. Note that the “good triplets” curve grows approximately to the square number of nodes, while the number of “bad triplets” grows linearly. Thus empirically this property holds.

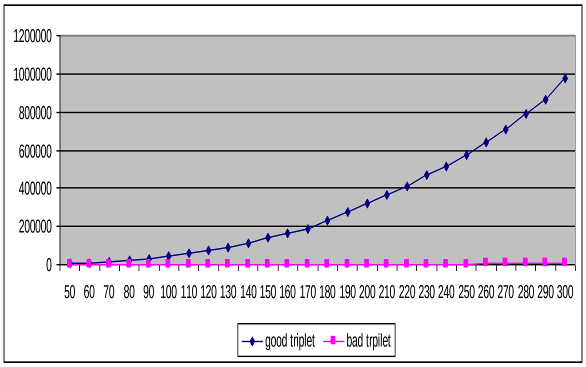

Figure 4: Number of good triplets Vs. bad triplets

A more formal basis for the claim that in ad hoc networks most triplets <s,u,t> satisfy the bypassing property can be obtained by considering level graphs. An acyclic graph *G* is a level graph (LG) if we can associate a level l with each node in *G* such that each node in level l is connected only to nodes in levels l+1, l-1 (except end point nodes l=0 & l=maxlevel connected l=1/l=maxlevel-1). For every LG we consider two special nodes *s,t* where *s* is connected to all nodes of the first level and *t* to all nodes of the last level. It follows that $P_{s,t}$ is always a LG as it contains only shortest paths between *s* and *t*. Figure 5 depicts such a case where a LG is formed in a given snapshot of an ad hoc network. It is thus sufficient to show that the bypassing property holds for LGs. We first prove this for "fixed degree" LGs (where the in-degrees and out-degrees are the same at every level) and extend it for LGs where the in-degrees/out-degrees are selected at random from a bound range.

Due to space limitations the proofs of the following claims, corollaries and theorems was omitted.

Claim 1: Let *G* be a *LG* such that the in-degrees/out-degrees at each level are the same. Let *s* and *t* be the two end nodes then the number of paths from *s* to *t* is equally partitioned between the nodes of every level.

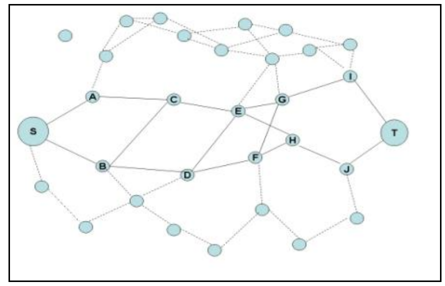

Figure 5: A leveld graph in an ad hoc network

Corollary 1: Let *G* be a LG where nodes at each level have the same in-degree/out-degree respectively then if the number of nodes at each level (m) is greater equal 2 then *G* satisfies the bypassing property.

A similar claim can be obtained for edges in LGs with the same in-degree/out-degree at each level.

Corollary 2: Let *G* be a LG where nodes at each level have the same in-degree/out-degree respectively then the number of paths through every incoming/out-going edge of the same level is the same.

It can be shown that even a small deviation from LGs with fixed in/out-degrees will lead to a violation of the bypassing property. In spite of this the bypassing property holds for random LGs. Consider the case of a LG with two nodes at each level.

A graph G is a random binary level graph (RBLG) if G is a LG with two nodes at every level such that the connections between each level are uniformly selected from the following set of five possible "connectors":

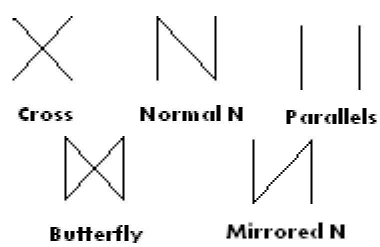

Figure 6: Possible Connectors

There are two special nodes s/t connected to the first and the last level respectively. The path relation at level i is the ratio $s\left(G_{i,1}\right)\cdot t\left(G_{i,1}\right)/s\left(G_{i,2}\right)\cdot t\left(G_{i,2}\right)$ where $s(G_{i,j})$/ $t(G_{i,j})$ is the number of paths between *s/t* and node *j* (*j=1,2*) of level *i*.

Theorem 1: For a given RBLG G with *n* levels the probability that the path-relation of more than 10% of the levels is larger than 121 is exponentially small in *n*.

.10. Claim: Let *G* be a RBLG and let *i* be a level of *G*. Let *S* be a segment of level *i* of length *n* in *G* and let *h*

denote the height of level i in S. The following inequalities hold: If S was chosen as the Segment-Normal-N then $\mathbf{s(G_{i,1}) \cdot t(G_{i,1}) / s(G_{i,2}) \cdot t(G_{i,2}) \leq (h+1) \cdot (n\text{-}h+1)}$. If S was chosen as the Segment-Mirror-N then $\mathbf{s(G_{i,1}) \cdot t(G_{i,1}) / s(G_{i,2}) \cdot t(G_{i,2}) \geq (h+1)^{-1} \cdot (n\text{-}h+1)^{-1}}$. Returning to the main proof.

For practical purposes we can look at the expected segment length. A segment can be terminated by two of the five connectors at each level. This means that the length of the segment has a geometrical distribution, with 2/5 as the parameter. The expected value of such a distribution is 5/2, implying that the expected length of the segments is 2.5 and the expected maximal $\mathbf{s(G_{i,1}) \cdot t(G_{i,1}) / s(G_{i,2}) \cdot t(G_{i,2}) \geq}$ ratio in a segment is $(2.5 / 2 + 1)^2 = 5.0625$.

Though this proof is to the special case of RBLGs it can be extended to any random LG with a constant number of nodes at every level. The formal justification for the use of the bypassing property is thus due to the assumption that at any given time $P_{s,t}$ in a given ad hoc network is a random LG with small constant number of nodes at each level. This is a reasonable assumption as in ad hoc networks cases usually the number of connections for most of the nodes is bounded in a small range. Note that the negation of this assumption implies dense pockets where large number of nodes is concentrated in a small area.

Figure 7: Improved quality of an LD session and an LD session 0.73

5. Simulations

The simulations were performed on an existing ad-hoc simulator expanded to support not only voice sessions but also large data (LD) sessions. This includes the addition of the MRMA algorithm for multi-path transfers and the session stream protocol. The simulator runs in parallel voice and LD sessions. This combination allows us to create a high load on the network resources. The traffic in the network is constructed of two types of LD sessions: "short" and "long" unidirectional LD sessions with the duration 10 seconds and 3-5 minutes respectively. In addition to the LD sessions there is a third type of voice sessions.

Every LD session gets a final score that presents the success of this session. A LD session that succeeds to transfer 100% of the packets gets the score 1. A session that succeeds pass 95% of its packets gets the score 0.95. An example of a LD that got the grade 1 is presented in the left side of Figure 7, and the same picture that got the grade 0.73 is presented in the right side. Recall that packets can be declared as unrecoverable due to the double window mechanism. Many tests were performed to validate the ability of the proposed multi-path technique to improve the quality of LD sessions, however due to space limitations they can not be included here. Instead, we illustrate the usefulness of the proposed technique by showing how the quality of an LD session for transferring a picture improves (Figure 7) due to the use of multi-path.

7. References

[1] M. K. Marina, S. R. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks," in Proceedings of the IEEE International Conference on Network Protocols (ICNP),. 2001, pp. 14-34.

[2] P. Sambasivam, A. Murthy, E. M. Belding-Royer, "Dynamically Adaptive Multipath Routing based on AODV, "*Proceedings of the 3rd annual Mediterranean Ad hoc Networking Workshop* 2004.

[3] Y. Ben-Asher, M. Feldman and S. Feldman, "Ad-Hoc Routing Using Virtual Coordinates Based on Rooted Trees," SUTC 2006, Taichung – Taiwan.

[4] C. E. Perkins, E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing and Applications,* 1999, pp. 90-100.

[5] S. J. Lee, M. Gerla, " Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," In *Proceedings of IEEE ICC*, pp. 3201-3205, 2001.

[6] S. Lee, M. Gerla, "AODV-BR Backup Routing in Ad Hoc Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference on Computer Communications* 1997, pp. 1405-1413.

[7] Z. Ye, S. V. Krishnamurthy, S. K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," INFOCOM 2003

[8] S. Sesay, Z. Yang, J. He, "A Survey on Mobile Ad Hoc Wireless Network," Information Technology Journal 3 (2): 168-175: 2004

[9] C. E. Perkins, E. M. Royer. "Ad hoc On-Demand Distance Vector Routing." Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90-100.

[10] S. Vutukury, J. J. Garcia-Luna-Aceves, "Mdva: A distance-vector multipath routing protocol", proceedings of the IEEE INFOCOM, 2001.

[11] W. Zaumen, , J. J. Garcia-Luna-Aceves, "Shortest multipath routing using generalized diffusing computations", IEEE INFOCOM. pp. 1408-1417, 1998

in Ad Hoc Networks", Computer Networks 39 (6) (2002) 809-8