A Warning Based Preemptive Routing Scheme for QoS Maintenance in Wireless Ad Hoc Networks

Sk Mohammad Rokonuzzaman
Clayton School of IT
Monash University
VIC 3800, Australia
rokon@infotech.monash.edu.au

Ronald Pose
Clayton School of IT
Monash University
VIC 3800, Australia
rdp@infotech.monash.edu.au

Iqbal Gondal
Gippsland School of IT
Monash University
VIC 3800, Australia
iqbal@infotech.monash.edu.au

ABSTRACT

Traditional QoS routing protocols find paths that meet the QoS requirements while discovering routes. The chosen route is used while the QoS is met. If the QoS is violated rerouting takes place. Violation of QoS may be unacceptable for delay sensitive traffic. We introduce a preemptive QoS re-routing scheme. The end-to-end delay of packets in each session is monitored and if it seems likely that QoS violation might occur, a preemptive QoS re-routing process is initiated. This helps maintain QoS in dynamic wireless ad hoc networks. Schemes for triggering QoS re-routing are investigated. We have reduced the number of late packets and improved the overall end-to-end delay of the communications.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols – *application, protocol verification, routing protocols.*

General Terms

Algorithms, Management, Performance, Design, Reliability.

Keywords

QoS, preemptive routing, Wireless ad hoc networks.

1. INTRODUCTION

Wireless ad hoc networks are dynamic environments even when nodes are not mobile, making it difficult to support applications with a range of Quality-Of-Service (QoS) requirements. Typical QoS requirements include bandwidth, end-to-end delay, jitter, error rate, etc. As wireless ad hoc networks may have unreliable and shared links and random node failures, it is a challenge to find multi-hop paths that meeting communications sessions' QoS requirements. The dynamic nature of such networks makes maintaining QoS performance particularly difficult. Our network is interesting in that we have a dynamic network topology based on each node having multiple directional links to neighboring nodes, implemented using Smart antenna technology. While we do not currently support mobile nodes we still have quite dynamic routing and topology changes. Here we focus on maintaining communications routes that meet strict QoS delay requirements.

Quality of Service (QoS) routing schemes [1-6] are used to find paths that meet user QoS requirements and make the best use of limited network resources. Typically they search among routes that meet the QoS requirements of the new communication session without breaking QoS of existing communications. In general communications are multi-hop with some nodes involved in multiple communication sessions. This makes queuing delay and other performance dependent factors quite dynamic depending on current traffic and reception conditions. Hence, actual QoS of communications may change over time.

In a QoS guaranteed system the destination node monitors the received packets checking QoS. If the QoS is not met, the source node is advised it to find a better route. The source node then initiates a re-routing process. There may be a hiatus for that communication session. This may be avoided using a preemptive QoS routing technique whereby an improved route is sought before the QoS has deteriorated too much. There has been some work on preemptive routing protocols in wireless ad hoc networks using omni-directional antennas [7, 8]. In these cases QoS was not considered. These methods check the received signal level on each node and if below a threshold a warning about a future link breakage is issued and the re-routing is initiated.

In this study, two QoS metrics have been considered - minimum bandwidth and maximum end-to-end delay. Our preemptive QoS routing process uses end-to-end delay to anticipate a potential QoS delay violation, although other QoS parameters can be used. We monitor end-to-end delay at the destination. If the end-to-end delay crosses a threshold a warning message is sent to the source. Delay at each intermediate node is recorded in the packet. If there is a bottleneck node we try to avoid it when re-routing. We initiate route discovery at an intermediate node before the bottleneck.

Wireless ad hoc networks using omni-directional antennas and a single radio channel minimize channel contention using CSMA style IEEE 802.11 protocols. Even so, network scalability is poor due to inter-flow and intra-flow interference [9, 10]. To reduce interference directional antennas may be used [11, 12, 13, 14]. Each node in our network uses multiple beam directional antennas, each beam using its own transceiver enabling concurrent operation. Moreover we use adaptive beam forming smart antennas [15] that allow the beam direction and beamwidth to be changed dynamically. This allows us to change the topology dynamically. We use a dynamic multi-beam directional topology and a preemptive re-routing approach to maintain the QoS.

Next we give our network model. Section 3 outlines the QoS routing process. Preemptive QoS re-routing is in Section 4. In Section 5 a simulation model is presented followed by results and discussion in Section 6. Conclusions are given in Section 7.

PE-WASUN'09, October 28–29, 2009, Tenerife, Canary Islands, Spain.

2. NETWORK MODEL

Our network nodes are not mobile but can join or leave the network at any time. Each node has one omni-directional beam (*OB*) and *m* directional beams ($DB_1, DB_2, \ldots DB_m$) as in Figure 1. Each beam has its own transceiver. Directional beams may use the same or different radio channels but the omni-directional beam always uses a separate radio channel. The beamwidths are $\theta_1, \theta_2, \ldots \theta_m$ and the beam directions are denoted by $\alpha_1, \alpha_2, \ldots \alpha_m$.

One adjusts θ and α of the *DB*s of the linked nodes to form the network topology. *A priori* knowledge of network traffic patterns is not usually available so the topology control scheme should be able to deal with arbitrary communications. In our network topology is changed dynamically by changing θ and α of *DBs* in an adaptive manner to allocate communications resources where they are needed. Electronically steered and beam-forming multi-beam Smart Antennas facilitate this and maximize the number of concurrent communications [15]. The network may end up partitioned so we need a way to reconfigure the Smart Antennas across the network.

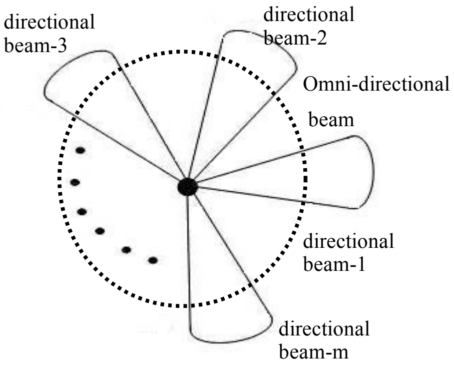

Figure 1. Node with omni-directional & *m* directional beams.

A separate omni-directional (*OB*) control network supports protocols for adaptive topology control and routing. Its performance is not critical since more demanding communications requirements use directional beams.

3. QoS ROUTING PROTOCOL

A modified version of AODV [16] has been used with two QoS metrics - bandwidth, and end-to-end delay. Routes are discovered when required. Each node stores the next hop in the routing table . Each node maintains 1) the Current Routing Table (CRT), and 2) the Alternate Routing Table (ART). The paths discovered in the current directional topology are stored in the CRT and used by the current communications and the QoS re-routing process. These paths are discovered considering the QoS parameters. Paths discovered in the omni-directional control topology irrespective of QoS are stored in the ART. This is used by the topology control process to find paths not in the current directional topology.

During the Route Discovery process each node sends the Route Request (RREQ) message in all the *DBs* to find paths in the Directional Network. The RREQ packets include the QoS parameters - minimum bandwidth required (BW_{min}) and maximum end-to-end delay (D_{max}). For any link, initial Available Bandwidth, $BW_{available}$ = Total Bandwidth of the link; and the bandwidth is updated using the following algorithm.

Algorithm 1: Update-Available-Bandwidth

```
$BW_{available} = BW_{available} - BW_{consumed}$
        Send an update message to the other node(s) of the link
        using the omni-directional Control Channel.
```

Each node records available bandwidths for its beams, and forwards the RREQ message only on beams with adequate bandwidth available. Each node also checks the cumulative delay and more than the maximum end-to-end delay required by the communication, it drops the RREQ packet. A Route Reply (RREP) packet is sent by the destination towards the source using the reverse path. When forwarding RREP packets intermediate nodes again check the available link bandwidth. The QoS Routing Algorithm (Algorithm 2) follows. Here D_{RREQ} is the end-to-end delay of RREQ packet, P_{DATA} and P_{RREQ} are sizes of data and RREQ packets in bytes respectively, δ is the time to transmit RTS and CTS messages and h is the number of hops between source and destination.

Algorithm 2: QoS Routing Process

```
Generate RREQ packets with $BW_{min}$ and $D_{max}$.
Check the available bandwidth ($BW_{available}$) on each link.
If ($BW_{available} > BW_{min}$)
  Call Update-Available-Bandwidth
   Source node sends RREQ packets on each link with adequate
   bandwidth available to support QoS.
   Source node locks minimum required bandwidth on links for
   the duration 2* $D_{max}$.
   Each intermediate node checks RREQ packet delay so far by
   checking the timestamp field and checks available bandwidth.
   if the (delay < D) && ($BW_{available} > BW_{minimum}$)
     send RREQ on directional links except incoming link.
   else
     don't forward the RREQ packets.
   if there is no RREP within 2* $D_{max}$
     release the bandwidth.
   else
     when a RREP is received lock the bandwidth for 2* $D_{max}$.
   if there is no data packet from the source within 2* $D_{max}$
     release the bandwidth.

   RREP packets include end-to-end delay of RREQ packets.
   Check all returned paths for delay constraint using formula,
```

$$D_{DATA} = (\frac{P_{DATA}}{P_{RREQ}} X\, D_{RREQ}) + (\delta\, X\, h)$$

```
        Store all eligible paths in CRT sorted by Shortest Widest Path.
        Send the RREQ packets without specifying QoS parameters in
        the control network using the omni-directional antenna.
    Store all the paths in the ART.
```

The source node reserves bandwidth on the appropriate link for the duration of the communication and sends a Bandwidth Reservation (BR) message to the next node. All intermediate nodes follow the same process. Once the BR packet is sent by the source, it sends test data packets to the destination. If they are received successfully and there is no QoS violation the admission control system admits the communication.

We record how long the last packet took to reach the destination from a node. This helps us look for alternate paths during a delay constrained preemptive QoS re-routing process. Once a packet reaches its destination, the packet arrival time is piggybacked on the ACK message, likewise for all intermediate nodes. Each node also stores the arrival time of each packet until it receives the piggybacked ACK message containing the arrival time of that packet at the destination. From these two values the node calculates how long the last packet took to reach the destination and stores it in the routing table.

4. PREMPTIVE QoS RE-ROUTING

The aim of the preemptive QoS re-routing process is to find a better path than the current one. Our preemptive QoS re-routing process uses the end-to-end delay of received packets as a trigger. If packet end-to-end delay crosses a threshold the destination node generates a warning message. On receipt of a warning message the source node initiates a route re-discovery process to find another path with improved end-to-end delay.

The algorithm has the following steps: 1) Calculate threshold value for generating the warning message, 2) Node selection and route re-discovery initiation. This method is useful for delay sensitive traffic.

4.1 Calculating the Threshold Value

Let the packet size be P bits, and transmission rate of a node be T bits/second.
We can calculate the theoretical minimum delay on each node or hop by considering only the transmission delay and ignoring the queuing delay (considering propagation delay is negligible). Let this delay be expressed as $DHop_{min}$.

$DHop_{min} = (P/T)$ seconds.

If the destination is h hops away, theoretical minimum end-to-end delay is, $D_{min} = DHop_{min} * h = (P/T) * h$ seconds.

Let the maximum allowable end-to-end delay for the negotiated QoS be D_{max}, and the actual end-to-end delay for a packet be D_{act}.

If $D_{act} > (\delta * D_{max})$ [where, $0 < \delta \leq 1$]

then send a warning message of potential QoS violation.

Here $(\delta * D_{max})$ is the threshold value of the preemptive QoS routing algorithm. Choosing δ is important. If δ is too low there will be unnecessary warning messages generated too early. If δ is too high the warning message is generated too late and there may be a late packet before a new path is found. We tried different values for δ, and are considering adjusting δ dynamically.

4.2 Node Selection/Route Discovery Initiation

The source node puts packet creation time in the header. Nodes on the path to the destination give the delay to the MAC layer, and put the transmission delay in the packet header. This delay is the sum of queuing delay plus the $DHop_{min}$ of that node. The source node creates these fields in the packet header and each individual node fills this field or each node can create and fill this field.

When for a packet, $D_{act} > (\delta * D_{max})$, the destination node sends a warning message to the source node or an intermediate node using the omni-directional control channel. The warning includes the end-to-end delay that generated the warning message. Then the source node or the intermediate node initiates a re-routing process for that destination to find a path with delay less than D_{act} of the packet that generated the warning message. If there is such a path the communication session switches to the better path. Otherwise the current path is used as there is still no QoS violation.

If there is a late packet, the destination node generates a QoS violation message and the source node initiates the route discovery process with delay requirement of D_{max}.

The preemptive re-routing process is used to find a better path. With a low threshold there will be many preemptive re-routing requests and often a better path will not be found. If preemptive re-routing fails to find a better path we can discard the warning messages after a while (10 seconds in this study). We do not retry immediately as the network state is almost the same and there is little chance of success. But the network state may change after a while, as existing communications may finish, new communications begin, or the directions of beams may change.

To discover a route in the re-routing process we use the omni-directional control channel so extra control traffic generated by the preemptive routing process does not affect the data traffic in the directional network. As described in Section 4.1 each entry in the Current Routing Table (CRT) contains the time to reach a destination based on how long the last packet took. Each node also keeps track of the available bandwidth for its links. Using this information alternate paths can be found. Alternate paths are returned to the source using the omni-directional control channel along with the estimated end-to-end delay of that path. The source node chooses the path with the smallest end-to-end delay and sends a test data packet to the destination node using the directional network to verify the end-to-end delay of this path. The destination node returns an Acknowledge (ACK) message including the end-to-end delay of the test data packet. If the delay is less than D_{act} of the packet that generated the warning message, that path is chosen., otherwise the current path is not changed.

When $D_{act} > (\delta * D_{max})$ the destination node checks delays at each node. If the delays are similar (considering D_{act}/h) the destination sends the source a warning using the omni-directional control channel and the source node initiates preemptive re-routing.

But if delay at a node ($> 3 * D_{act}/h$) is high compared with other intermediate nodes, this node is considered a bottleneck. The destination node sends a warning to the node before the bottleneck using the omni-directional control channel and that intermediate node initiates preemptive re-routing for that destination while the current path is still used. We want to avoid the bottleneck node. Updating a route from an intermediate node in this way is possible as we use hop-by-hop routing. The destination node informs the source, or an intermediate node, of a potential QoS violation due to exceeding the maximum delay. That node may initiate the route re-discovery process.

Initiating route re-discovery from the node before the bottleneck node is more efficient than from the source if a route is found. However we have a directional topology which is less connected than an omni-directional counterpart. It may not be possible to find another path to the destination from the node before the bottleneck. But a node which is two or three hops away from the bottleneck node towards the source may find an alternate path avoiding the bottleneck node.

Our approach, the 'Backtrack-Preemptive-Routing (BPR)' tries to initiate the route re-discovery process from an intermediate node as close as possible to the bottleneck. If there is a bottleneck node detected by the destination, a warning message is sent to the node just before the intermediate node. This node tries to find a path avoiding the bottleneck node. If such a path exists the communication switches to this new path, otherwise that intermediate node sends the warning message to the node before it towards the source, and that node tries to find a path avoiding the next node. If necessary this process is followed until the source node. In that case the source node initiates route discovery.

BPR may have overhead in terms of time and control packets if the re-routing process ultimately reaches the source node. But the current path is still functional and usually we find a path from some intermediate node thus saving time.

5. SIMULATION MODEL

The performances of our approaches have been evaluated using GloMoSim [17], which is designed using PARSEC [18]. A wireless network is used with 200 nodes placed randomly on a 4 sq. km area. Data packet size is 1024 bytes. For each session the QoS metrics are - minimum bandwidth 160 kbps and maximum possible end-to-end delay 0.05 seconds. We start with 10 communications and add one communication every 10 seconds starting at the 10th second, up to a limit of 20 concurrent communications. Each communication has Constant Bit Rate (CBR) UDP traffic between randomly selected source-destination pairs. Average packet inter-arrival time is 50ms. Nodes have 3 directional beams.

6. DISCUSSION OF RESULTS

In the figures and tables we use a basic preemptive routing approach, PR. Routing re-discovery is initiated by the source node after receiving a warning message. δ represents the threshold as a proportion of maximum QoS delay.

Table 1. Statistics for Different Values of δ for All Sessions

δ	D_{act} that triggered warning	No. of warning or late pkt msgs	No. of warned sessions	Delay reduction after re-routing	Late pkts	Sessions with late pkts
1.00	1.05	7	5	24%	7	5
0.95	0.97	12	6	21%	5	4
0.90	0.93	21	8	19%	4	2
0.85	0.90	39	11	17%	1	1
0.80	0.85	71	13	14%	0	0
0.75	0.81	113	17	11%	0	0
0.70	0.77	147	20	8%	0	0

Table 2. Statistics for Different Values of δ for Session-1

δ	D_{act} that triggered warning and re-routing	Number of warning or late pkt msgs	Delay reductn. after re-routing	Late pkts
1.00	1.08	2	22%	2
0.95	0.97	4	18%	1
0.90	0.94	5	16%	1
0.85	0.90	8	13%	1
0.80	0.86	10	11%	0
0.75	0.82	15	8%	0
0.70	0.78	18	4%	0

Table 3. Statistics for Different Values of δ for Session-2

δ	D_{act} that triggered warning and re-routing	Number of warning or late pkt msgs	Delay reductn. after re-routing	Late pkts
1.00	1.05	1	24%	1
0.95	0.97	2	20%	1
0.90	0.94	4	19%	1
0.85	0.89	5	17%	0
0.80	0.85	6	12%	0
0.75	0.82	8	9%	0
0.70	0.77	10	7%	0

Table 4. Statistics for Different Values of δ for Session-3

δ	D_{act} that triggered warning & re-routing	Number of warning or late pkt msgs	Delay reductn. after re-routing	Late pkts
1.00	1.02	1	27%	1
0.95	0.96	1	22%	0
0.90	0.93	2	21%	0
0.85	0.91	3	19%	0
0.80	0.84	4	17%	0
0.75	0.81	5	13%	0
0.70	0.79	7	10%	0

Table 1 shows statistics for different values of δ for the basic preemptive re-routing approach. It shows the average end-to-end delay constraint that is used in the routing discovery process as a percentage of d_{max}. This value is higher than δ. The preemptive re-routing process tries to find a better path than the current path and if found it uses that path. The next column shows end-to-end delay improvement after re-routing. Table 1 also shows the number of warning messages generated, the number of sessions involved in generating these warning messages, the number of late packets and the number of sessions that generated the late packets.

We see that as δ is decreased the number of late packets drops, and from δ=0.8 it becomes 0. This is due to high δ causing warning messages to be generated too late, and allowing late packets before finding an alternate path. If δ is too low the warning message is generated prematurely.

Tables 2, 3 & 4 show statistics for three different individual sessions. Session-1 generates the largest number of warning messages among all communications. Session-3 generates the fewest messages of those that generate any warning messages.

Session-2 lies in between. The delay reduction after re-routing process is different for these sessions.

As seen in Table 4 there are no late packets when preemptive re-routing is used. There are few warning messages for this session and delay reduction is higher than both Session-1 and Session-2. This indicates that for Session-3 in most cases we were able to find a better path than the current one when the warning messages were generated. Tables 1, 2, 3, and 4 also indicate that the preemptive re-routing process improves the overall routing performance of the network, as it does not allow any packets to be late if an appropriate threshold is chosen, and a new path is used if it provides an improved end-to-end delay.

Tables 2, 3, and 4 show the performance and delay improvement of a session that initiated the preemptive re-routing process. In wireless ad hoc networks paths may be shared by many sessions. When preemptive re-routing changes a path there are impacts on delay for other sessions sharing part of the old path with the session, and for other sessions that share part of the new path. Table 5 shows this impact for the re-routing processes of Session-1 when δ=0.8. Each row indicates a preemptive re-routing process and shows the impact on delay for Session-1 and other related sessions. In Figure 3 we show when these eight re-routing processes take place. Most re-routing occurs when new communications are added. This is not the case for other sessions. Other sessions generate fewer preemptive re-routing processes. The '-' sign indicates a reduction or improvement in delay, whereas the '+' sign indicates a delay increase. As seen in Table 6, sessions that shared part of the old path get improved delay, as a session has been removed from this path. However there is a slight increase in the delay of sessions that share the new path of Session-1. '0%' in the table indicates the new path is not shared with any other sessions. Thus preemptive re-routing also helps balance network load by using underused network resources.

Table 5. Delay Performance of other related Sessions after the Re-routing Process of Session-1 when δ = 0.8

Serial# of re-routing process	Delay reductn after re-routing for Session-1	Impact on delay for other sessions that shared part of old path of Session-1	Impact on delay for sessions that share part of new path of Session-1
1	11%	-7%	0%
2	13%	-6%	0%
3	10%	-7%	+5%
4	14%	-8%	+4%
5	10%	-9%	+7%
6	11%	-7%	+5%
7	10%	-6%	+5%
8	10%	-6%	+4%

Figures 2, 3, and 4 show when re-routing takes place for Session 1 of Table 3. A session generating many warning messages has been chosen and is also used for these three plots. Vertical lines in the plots indicate unsuccessful re-routing processes where better paths could not be found. Dots in the plots indicate a successful re-routing process. The graphs show δ as 0.7, 0.8, and 0.9. Figure 5 shows maximum end-to-end delay of all communications in a 10 second interval for different values of δ. When δ is 0.95 or 0.9 the maximum end-to-end delays are close to when δ is 1. In this case the warning message is generated too late and some packets get delayed before finding another path. But when δ is high the re-routing process is always successful. On the other hand when δ is 0.7 there are too many preemptive route discovery requests. As seen for PR-0.75 and PR-0.7, in most of the interval the delay crosses the threshold initiating route re-discovery. Sometimes the constraint for end-to-end delay is quite low and the chance of finding an improved path is small, as seen in Figure 2. When δ is 0.8 or 0.85 we get reasonable maximum end-to-end delay. In this case the warning message is neither too early nor too late and usually a suitable path is found, as can be seen in Figure 3.

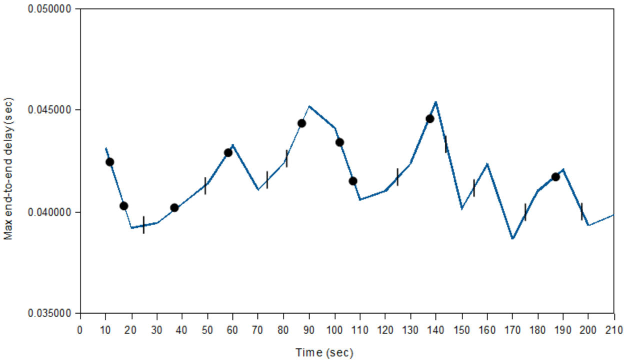

Figure 2. Re-routing processes vs Time when δ = 0.7

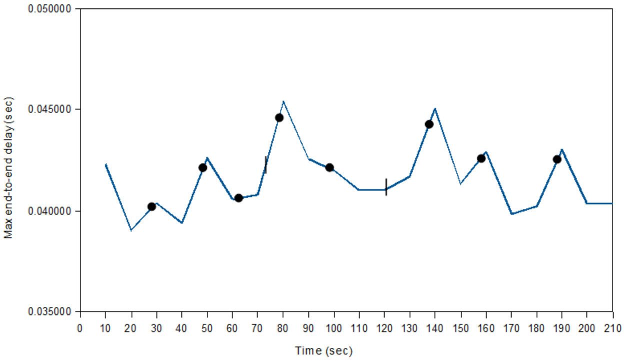

Figure 3. Re-routing processes vs Time when δ = 0.8

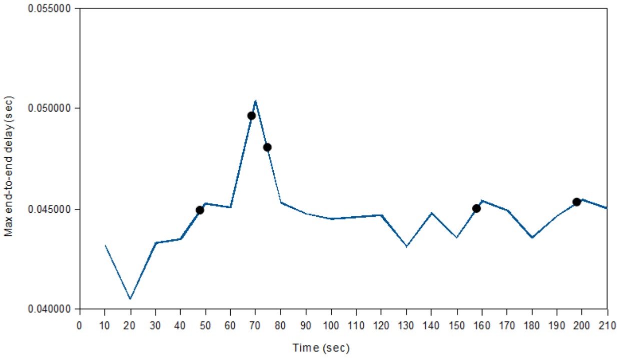

Figure 4. Re-routing processes vs Time when δ = 0.9

Figure 6 shows end-to-end delay of the preemptive bottleneck-node routing approach (indicated by 'PBR'). After finding the bottleneck node if present, route re-discovery is initiated 1 hop, 2 hops or 3 hops away from the bottleneck node. We also simulated our Backtrack-Preemptive-Routing (BPR) approach and the source initiated routing approach (PBR-Source). As seen in Figure

6, BPR performs better compared with the 1 hop, 2 hops, 3 hops or source initiated route re-discovery approaches. This is because finding a path from 1 hop away from the bottleneck is efficient but the chance of finding such a path is low in a directional topology. Since the current path is still functional, BPR keeps trying to find a path by backtracking hop-by-hop. The bottleneck is avoided in route re-discovery and usually a path avoiding the bottleneck is found. Thus BPR improves end-to-end delay.

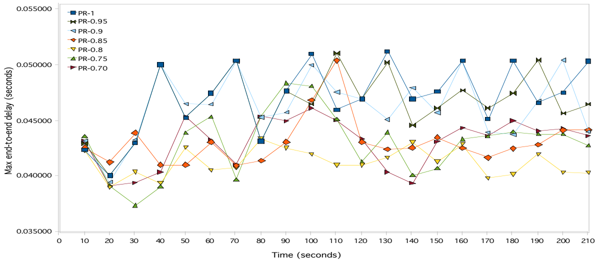

Figure 5. Max end-to-end delay vs Time

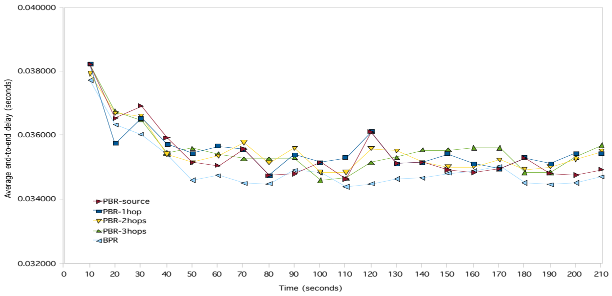

Figure 6. Average end-to-end delay vs Time for δ = 0.8

7. CONCLUSION

We presented a preemptive re-routing approach based on end-to-end delay of packets. The destination node generates a warning message when the end-to-end delay of a packet crosses some threshold. We looked for bottlenecks avoided a bottleneck node during re-routing. Simulation results show our approaches reduce late packets by allowing a communication to change the path beforehand. Overall end-to-end delay is reduced, which allows admitting more traffic to the network, and also helps in network load balancing. Future work will focus on setting the value of δ dynamically depending on current network conditions.

The QoS preemptive re-routing scheme can also use other QoS parameters so is quite general. Computational overhead is not a constraint in our SAHN network model that does not rely on mobile, power-constrained nodes. We plan to do further experimentation with a range of QoS constraints.

8. REFERENCES

[1] Q. Xue and A. Gunz, “Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks,” *Journal of Parallel and Distributed Computing*. Vol. 63, pp. 154-165, 2003.

[2] R. Gupta, Z. Jia, T. Tung and J. Walrand, “Interference-aware QoS Routing (IQRouting) for Ad-Hoc Networks,” in *Proc. IEEE GLOBECOM,* 2005.

[3] J. Tang, G. Xue and W. Zhang, “Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks,” in *Proc. ACM MobiHoc*, 2005.

[4] G. Chauhan and S. Nandi, “QoS Aware Stable path Routing Protocol for MANETs,” in *Proc. IEEE ICETeT*, 2008.

[5] M. M. Islam, R. Pose and C. Kopp. “Challenges and a Solution to Support QoS for Real-Time Traffic in Multi-Hop Ad-Hoc Networks”, in *Proc. IEEE WoCN*, March 2005.

[6] Z. Wang, J. Crowcroft, “Quality of Service Routing for Supporting Multimedia Applications”, IEEEJSAC, vol. 14. No 7, September 1996.

[7] T. Goff, et al. “Preemptive Routing in Ad Hoc Networks”, in *Proc. ACM SIGMOBILE,* Rome, Italy, 2001.

[8] W. Zhu, X. Zhang, Y. Liu and N. Li. “Improve Preemptive Routing Performance in Mobile Ad Hoc Networks with Cache-enabled method”, in *Proc. of Third International Conference on Communications and Networking in China (ChinaCom)*, 2008.

[9] S. M. Rokonuzzaman, R. Pose, and I. Gondal. “A Cross-layer Approach for Using Multiple Radio Channels with Directional Beams in a Suburban Ad Hoc Network,” in *Proceedings of ATNAC-08,* December 7-10, 2008, Adelaide, Australia, pp 192-197.

[10] M. Gong, S. Midkiff, and S. Mao. “A Cross-layer Approach to Channel Assignment in Wireless Ad Hoc Networks,” *Mobile Networks and Applications*, Volume 12, Issue 1, January 2007.

[11] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, “Ad Hoc Networking with Directional Antennas: A Complete System Solution,” IEEE J. Selected Areas in Comm., vol. 23, 2005.

[12] S. Roy, D. Saha, S. Bandyopadhyay, T. Ueda and S. Tanaka, “A Network-Aware MAC and Routing Protocol for Effective Load Balancing in Ad Hoc Wireless Networks with Directional Antenna,” Proc. ACM MobiHoc, 2003.

[13] R. Choudhury and N. Vaidya, “Deafness: A MAC Problem in AdHoc Networks When Using Directional Antennas,” Proc. 12th IEEE Int'l Conf. Network Protocols (ICNP '04), 2004.

[14] G. Jakllari, W. Luo, and S.V. Krishnamurthy, “An Integrated Neighbor Discovery and MAC Protocol for Ad Hoc Networks Using Directional Antennas,” IEEE Trans. Wireless Comm., vol. 6, no. 3, 2007.

[15] Alexiou, Angeliki, and Haardt, Martin, “Smart Antenna Technologies for Future Wireless Systems: Trends and Challenges”, *IEEE Communication Magazine*, September 2004.

[16] C. Perkins, E. M. Royer and S. Das. “Ad hoc On-Demand Distance Vector (AODV) Routing”. IETF RFC 3561, 2003.

[17] GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/.

[18] PARSEC, http://pcl.cs.ucla.edu/projects/parsec.