

Finding Multiple Optimal Paths in Ad Hoc Networks based on QOS metrics: Maximum Bandwidth and Minimum Delay

S.Nageswara Rao
Asst professor, CSE dept.
MITS college, Madanapally.

Dr.C.Shoba bindhu
Assoc professor, CSE dept.
JNTUA, Anantapur,

ABSTRACT

Ad hoc networks have characteristics such as flexibility, fast and easy deployment, and robustness which make them an interesting technology for various applications. Adhoc networks are considered as the most promising terminal networks in future mobile communications. Providing sufficient bandwidth and time delay for multimedia applications in ad hoc networks is an urgent task because of the rising popularity of multimedia applications and potential commercial usage of adhoc networks. Providing QOS(Quality Of Service) guarantee is a critical and challenging problem in ad hoc networks because of multihop, mutual radio interference and node mobility. This paper focuses on QOS metrics: maximum bandwidth and minimum delay. While general approaches are available in literature that can't work better for MANET where node life time is much smaller. This paper proposes a protocol QMPRP(QOS-aware Multiple Path Routing Protocol) to find out QOS oriented multiple paths to avoid reestablishment of paths in case node failures, The results are graphically shown and compared with existing models, the proposed model out performs the existing models in a significant way for frequent node failures.

General Terms

This paper exposes a protocol QMPRP, for finding QOS oriented optimal multiple paths for a pair of source and destination nodes.

Keywords:

Mobile adhoc networks, Quality of Service, QMPRP, Delay and bandwidth.

1. INTRODUCTION

Ad-Hoc network is a dynamic multi hop wireless network that is established by a set of mobile nodes on a shared wireless channel. Each mobile host performs local broadcasts in order to identify its existence to the surrounding hosts. Surrounding hosts are nodes that are in close proximity to the transmitting host. In that way each mobile hosts becomes potentially a router and it is possible to dynamically establish routes between itself and nodes to which a route exists.

Ad-Hoc Networks were initially proposed for military applications such as battlefield communications and disaster recovery, but the evolution of the Multimedia Technology and the commercial interest of Companies to reach widely civilian

applications made QOS in MANETs an area of great interest. Although much progress has been done in QOS for wire-based networks, there are still many problems in wireless networks. Moreover the problems that exist for QOS in wireless-based networks are: a)the Bandwidth Constrains, since a MANET has usually poor bandwidth resources, b) the Dynamic Topology of the MANET, since nodes are continually changing location, connecting and disconnecting from the network making connections many times unreliable, and c)the Limited processing and Storing capabilities of mobile nodes, in contrast with routers on the Internet.

This work is mainly focusing on QOS metrics Band width and Time delay in finding optimal multiple paths from source to destination and proposed a protocol(QMPRP) by using graph traversal methods and modified DSR reactive routing protocol, which can significantly reduce the end to end delay by avoiding reestablishment of connection from frequent node failures.

2. RELATED WORK

We first introduce DSR briefly because QMPR is based on the DSR.

2.1 DSR Protocol (Dynamic Source Routing)

Dynamic Routing Protocol (DSR)[4] is an on demand protocol designed to restrict the bandwidth consumed by control packets in ad-hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach[6].

If a source node that does not have route to the destination and if it has data packets to be send to that destination, it initiates a Route Request (RREQ) packet. This Route Request packet flooded throughout the network. Each node upon receiving a Route Request packet rebroadcast the packet to its neighbors if it has not forwarded already or if the node is not the destination node, provided the packets time to live (TTL) counter has not exceeded. Each Route Request carries a sequence number generated by the source node and the path it has traversed

A node upon receiving a Route Request packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate Route Request. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same Route Request by an intermediate node that receives it through multiple paths. Thus, all nodes except the destination forwarded the Route Request packet during the route construction phase.

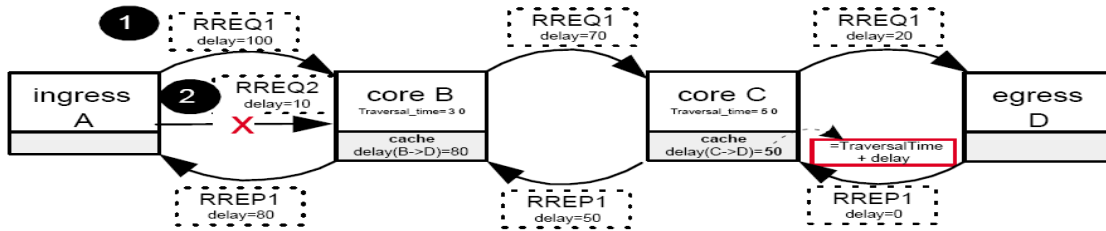


Fig1. RREQ forward or drop is based on the delay demanded

A destination node, after receiving the first Route Request packet, replies (RREP) to the source node through the reverse path the Route Request packet had traversed. This protocol uses a route cache that stores all possible information extracted from the source route contained in a data packet. Node also can learn about the neighboring routes traversed by data packets if operated in the promiscuous mode. This route cache is also used during the route construction phase. If an intermediate node receiving a Route Request has a route to the destination in its route cache, then it replies to the source node by sending a Route Reply with the entire route information from the source node to the destination node.

If any link on a source route is broken, the source node is notified using a route error (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source if this route is still needed. DSR makes very aggressive use of source routing and route caching. No special mechanism to detect routing loops is needed. Also, any forwarding node caches the source route in a packet it forwards for possible future use.

2.2 QOS Routing in MANETs

QOS Routing in MANETs is an essential component to realize a complete QOS MANET Architecture. The QOS Routing procedure can inform a source node of the bandwidth and QOS availability to destination node in the network. This knowledge enables the establishment of QOS connections within the network and the efficient support of real-time multimedia traffic. There are generally many proposed [1][2] [5] solutions for QOS routing in MANET. In this section we are going to explore the QoS version of reactive protocols [3].

2.2.1 Minimum Delay

Indicates the Minimum number of seconds allowed for a transmission from a source (or from an intermediate node forwarding the RREQ) to the destination. Every time a node receives a RREQ it subtracts (from the Delay indicated in the RREQ) the NODE_TRAVERSAL_TIME, which is the time required by this node to process the RREQ. In fig1 The NODE_TRAVERSAL TIME is by default set to 40 milliseconds, but could have a different value; in the case this node has more or less processing power. If the NODE_TRAVERSAL_TIME is bigger than the delay time indicated in the RREQ the node will simply discard the RREQ and not process any further. The graphical representation in figure shows how RREQ1 would be forwarded by the intermediate (core) routers during the path discovery process. At every step the delay field in the RREQ message is reduced by the Traversal Time of the router. At the end egress D will reply a RREP message which will have a starting delay value of 0. This delay value will be added to the

Traversal time of each node and registered (cached) in the Routing Table for future RREQs. The caching of the delay value will make the future discovery of that route a trivial task. So for example a future RREQ2 done by another node A will be directly dropped by core node B since the demanded delay (10ms) can't be met (since it is 80ms).

Of course in the future a node, such as core C, may have increased load which would change its NODE TRAVERSAL TIME from 50ms to 100ms. This change would affect all depending nodes such as B and A. For this reason node C will forward an ICMP_QOS_LOST message to all potentially nodes affected by the QOS parameter. This is also the reason why each node had initially stored a list of depending nodes, the "List of Sources Requesting Delay Guarantees". The ICMP_QOS_LOST message is quite short and it is sent recursively to all nodes affected.

2.2.2 Maximum Available Bandwidth

Available bandwidth is a field which indicates the requested amount of bandwidth for a specific link (route). Every time a node receives a RREQ it must compare its available link capacity with the capacity of bandwidth requested in the RREQ. If the requested bandwidth is not available then the node will again, as the delay example, discard the RREQ and not process any further. If the bandwidth is available then the request will process until the egress router D is reached as shown in Fig 2.

At that point the egress router D will respond with a RREP message which will be initialized with a bandwidth value equal to infinite (a very large number). Each node forwarding the RREP compares the bandwidth field in the RREP and its own link capacity and maintains the minimum of the two in the Bandwidth field of the RREP before forwarding the RREP. This bandwidth value will be registered (cached) in the Routing Table bandwidth value for future RREQs. The caching of the bandwidth value will again make the future discovery of that route a trivial task. We can see again that RREQ2 (which is route discovery message from A to D) request won't be satisfied because its request is 80Kbps exceeds the available 50Kbps which is now Cached in nodes' B cache and hence be directly dropped by core node B.

As in the Delay case, a node in the future may have a drop in link capacity which will lead in the generation of an ICMP_QOS_LOST message to all potentially nodes affected by the QOS parameter. The list of nodes that are affected by this property is now stored in the "List of Sources Requesting Bandwidth Guarantees".

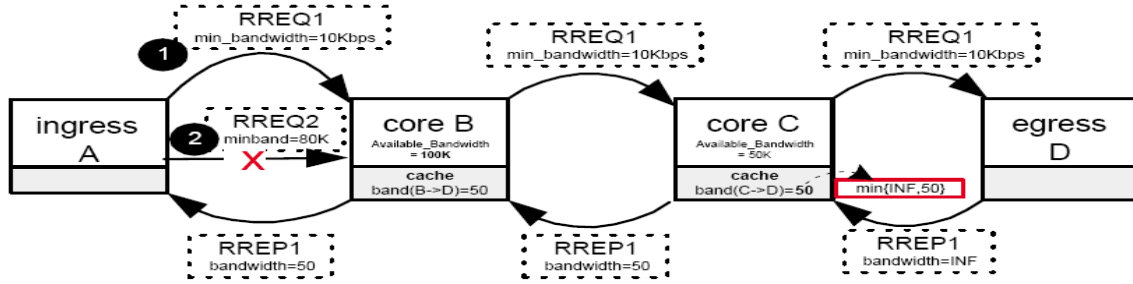


Fig2. RREQ forward or drop is based on the bandwidth demanded

3. PROPOSED WORK

Similar to all the existing reactive models, this work (QMPM) also uses Route Request (RREQ) and Route Reply (RREP) to track node status in the MANET, and used some recursive formulas of graph theory to find out the multiple paths for a pair of source and destination.

3.1 QMPM for QoS metric: ‘Minimum delay’

Recursive Relation:

$$T(x, dest) = set \{x_d + T(x_{adj}, dest)\} \quad (1)$$

In above equation (1), T() is a function, which list out all paths according to their delay for a pair of nodes, x_d is a node processing delay, x_{adj} is set of all adjacent nodes of x.

Algorithm

- 1) Every node in MANET maintains a set of entries with the fields routing delay(RD) and Routing vector set(RV).
- 2) Source node updates entry fields as RD=0, RV = \emptyset and creates RREQ with the fields Source ID(S.id), Destination Id(D.id) and Required routing time delay(Rrt) to all its adjacent nodes.
- 3) Once an intermediate node receives a RREQ, the node compares its processing delay (x_d) with Rrt available in RREQ if $x_d < Rrt$, then it sets up all its entry fields like RD = x_d and RV = \emptyset , and forwards the RREQ by updating S.id with its own Id to All its adjacent nodes.
- 4) Destination node after receiving RREQ updates its entry fields, RD= 0, RV= its Id, and using the same information it creates RREP(RD,RV) and sends back to all adjacent nodes from those it got RREQ.
- 5) Once an intermediate node receives RREP updates its next available entry fields like node

$$RD = RD + RREP's RD, \\ RV = RREP's RV$$

And compares the node RD with the Rrt,

if $RD < Rrt$ then it creates the RREP like

$$RREP's RD = Node's RD, \\ RV = \{Node's Id\} \cup \{Node's RV\}$$

And sent it to all node from those it had received

RREQs

- 6) The same process will be repeated even for the source node, finally it collects all set of possible routing paths and their corresponding routing delay from source to destination.

Fig 3 . Algorithm to find out QOS(node delay)multiple paths

In the above figure, algorithm will list out all paths and their Delays, source node selects the minimum delay path, if any node failes along the path, then instead of reestablish of connection source node immediately will pick up the next optimal path, which doesn't involve the failure node.

3.2 QMPM for QoS metric:Maximum Bandwidth

Recursive formula:

$$B(x, dest) = set \{ \min(x_{bw}, B(x_{adj}, dest)) \} \quad (2)$$

In above equation (2), B() is a function, which list out all paths according to their bandwidth for a pair of nodes, x_{bw} is x node available link band width, x_{adj} is set of all adjacent nodes of x.

Algorithm

- 1) Every node in MANET maintains a set of entries with the fields Route Bandwidth (RB) and Routing vector set (RV).
- 2) Source node creates RREQ with the fields Source ID(S.id), Destination Id(D.id) and Required routing Bandwidth (Rrb), next it compares its available link bandwidth (x_{bw}) with Rrb in RREQ, if $x_{bw} > Rrb$, then it sets up all its entry fields like RB = x_{bw} and

$RV = \emptyset$, and forwards the RREQ to all its adjacent nodes.

- 3) Once an intermediate node receives a RREQ, same procedure will be repeated as in source node but before forwarding RREQ to adjacent nodes, it updates S.id with its own Id.
- 4) Destination node after receiving RREQ updates its entry fields $RB = \infty$, $RV =$ its Id, and using the same information it creates RREP(RB,RV) and sends back to all adjacent nodes from those it got RREQ.
- 5) Once an intermediate node receives RREP updates its next available entry fields like node

$$RB = \min \{ \text{Node's RB, RREP's RB} \}$$

$$RV = \text{RREP's RV}$$

And compares the node RB with the Rrb, if $RB > Rrt$ then it creates the RREP like ,

$$\text{RREP's RB} = \text{Node's RB,}$$

$$RV = \{ \text{Node's Id} \} \cup \{ \text{Node's RV} \}$$

And sent it to all nodes from those it had received RREQs

- 6) The same process will be repeated even for the source node, finally it collects all set of possible routing paths and their corresponding routing Bandwidth from source to destination.

Fig4. Algorithm to find out QoS (band width) multiple paths.

In the above fig4, algorithm will list out all paths and their bandwidths; source node selects the maximum band width path, if any node fails along the path, then instead of reestablish of connection source node immediately will pick up the next optimal path, which doesn't involve the failure node.

4. THE TRAFFIC AND MOBILITY MODELS

Continuous bit rate (CBR) traffic sources are used. The source-destination pairs are spread randomly over the network. The mobility model uses the random way point model in a rectangular field with 1500mX1500m field whereas network size is 80 nodes. Simulations are run for 100 simulated seconds. Maximum speed is varied at 0-10m/s.

5. SIMULATION SETUP

Simulation is done using QualNet5.0.2 simulator. The performance metrics that we evaluated are average End-to-End Delay for finding multiple paths for a pair of nodes while maintaining QoS metrics bandwidth and delay.

6. RESULTS AND ANALYSIS

In order to show the improvement of end to end delay in QoS paths(Band width and time delay) DSR model was taken as the base for comparisons[8]. This is due to the fact of the superiority of DSR model is already well established in the literature. The proposed model hence forth referred as QMPRP model.

6.1 Average End-to-End Delay in Maximum Bandwidth path

In the below figure, the proposed protocol(QMPRP) shown minimum Avg end-to-end delay over existing work DSR for QoS metric band width, when the number of transmission are high, this difference is significantly increased.

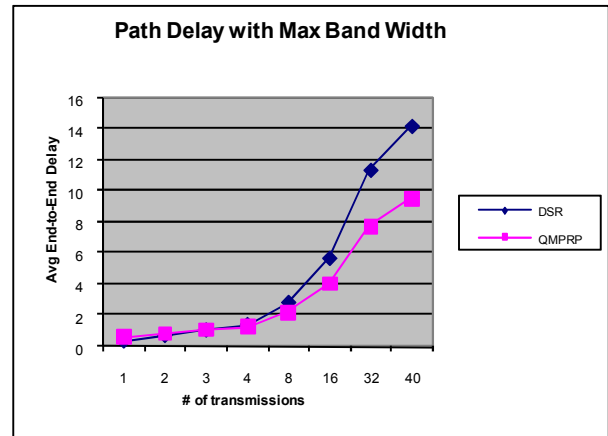


Fig.5 Avg End-to-End delay for QoS Metric bandwidth

6.2 Average End-to-End Delay in Minimum time delay path

In the below figure 6, the proposed protocol(QMPRP) shown minimum Avg end-to-end delay over existing work DSR for QoS metric node delay, when the number of transmission are high, this difference is significantly increased.

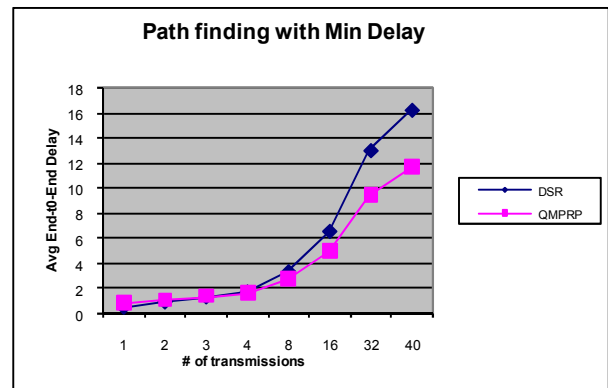


Fig.6 Avg End-to-End delay for QoS metric node delay

7. CONCLUSION AND FUTURE WORK

The changing expectations of wireless users towards high quality, group-oriented, mobile multimedia communication forces today's networks to support adhoc QoS multicasting. Even though quite a few models are available in the literature to support QoS routing in MANETs, they do not work efficiently for frequent failures of nodes in Adhoc networks. This paper proposes a protocol QMPRP(QoS-aware Multiple Path Routing Protocol) to find out QoS oriented multiple paths to avoid reestablishment of paths in case of failures of node, The results obtained are graphically presented and show a significant improvement over the previous models.

8. REFERENCES

- [1] Keming DU and Yahui YANG “ A QoS routing for Maximum Bandwidth in Ad Hoc Networks”. Proceedings of the Second International Conference on Future Networks2010.
- [2] Lei Chen and Wendi and B.Heinzelman A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks. Proceedings of IEEE Network November /December 2007.
- [3] Zeinalipour-Yazti Demetrios, A Glance at Quality of Services in Mobile Ad-Hoc Networks. Int’l Conf.on Computer & Communication Technology2001.
- [4] David B. Johnson David A. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. Monarch Project at Carnegie Mellon Univ.
- [5] Dr.A.Venugopal Reddy and B. Sateesh Kumar . QOS Multicast Routing Based on Bandwidth and Resource availability in Mobile Adhoc Networks. Proceedings of the Seventh Interational Confrence on Networking(2008)
- [6] Yadu Kishore K and G Kakde. Optimization based Topology Control for Wireless Adhoc Networks to meet QoS requirements.Proceeding of 29th IEEE International Symposium on Reliable Distributed Systems(2010).
- [7] Kannan .T and Dr.A.M.Natarajan Fortification of QoS Routing in MANETs using Proactive Protocol. Proceedings of ICWCSC 2010X.
- [8] Prof. S. P. Setty and Syed Basha Shaik. Performance Comparison of AODV, DSR and ANODR for Grid Placement Model.Proceedings of International Journal of Computer Applications (0975 – 8887) Volume 11–No12, December 2010.