A Deadline-Driven Probabilistic Quality of Service Routing for Mobile Ad hoc Networks

Ash Mohammad Abbas, Øivind Kure
Center for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology
O.S. Bragstads Plass, 2E, N-7491 Trondheim, Norway
ash@q2s.ntnu.no, okure@q2s.ntnu.no

Journal of Digital
Information Management

ABSTRACT: *Provision of quality of service (QoS) in an ad hoc network is a challenging task due to their inherent characteristics. In this paper, we present a routing protocol for QoS provisioning in mobile ad hoc networks. Our protocol tries to discover multiple node-disjoint paths between a given source and a destination that are able to satisfy QoS requirements in terms of bandwidth required by a flow of packets. We discuss how one can have probabilistic QoS guarantees using a protocol where nodes are aware of their neighbors that may interfere with them while relaying packets to other nodes along the paths towards the destination. We analyze the probability that the packets arrive at the destination before their respective deadlines.*

Categories and Subject Descriptors
C.1 [Computer Communication Networks] Network Protocols − Routing Protocols; **C.1 [Computer Communication Networks]** Network Architecture and Design − Distributed Networks, Network Topology, Wireless Communication.

General Terms: Quality of service, Routing, Ad hoc networks.

Keywords: Quality of service, Probabilistic QoS guarantees, Node-disjoint mulipath routing, Path independence.

Received: 5 June 2009; Revised 2 September 2009; Accepted 30 September 2009.

1. Introduction

An ad hoc network can be formed instantaneously enabling the participants to communicate without the required intervention of a centralized infrastructure or an access point. An ad hoc network may find applications in battlefield communications, disaster recovery, law enforcement, etc. Those who want to bypass the infrastructure for some reasons and still want to communicate may form an ad hoc network. For example, spying agents would like to bypass the infrastructure to hide their communication while others may wish to bypass for economic reasons. An ad hoc network may provide cheaper and cost-effective ways to share information among many mobile hosts. Some of the characteristics of an ad hoc network differentiate it from other classes of wired and wireless networks. In an ad hoc network, the transmission range of mobile devices is limited, therefore, routes are often multihop. There are no separate routers, therefore, nodes in the network cooperate to forward packets of one another towards their ultimate destinations. The devices used to form such a network are often powered through batteries whose power depletion may cause node failures. Further, nodes may move about randomly and therefore the topology of the network may vary dynamically.

It is desirable to have a provision of quality of service (QoS) in an ad hoc network. However, there are many peculiar characteristics of an ad hoc network that hinder in providing QoS. For example, the absence of any centralized infrastructure and the dynamically varying topology of a mobile ad hoc network make the provision of QoS a challenging task. Further, the topology information of the network is not available apriory at a central node. A node knows only about its neighbors. As a result, a solution or scheme should be able to work with the localized topology information and in a distributed manner. In other words, devising schemes for providing any hard guarantees about the QoS is difficult due to frequent node and link failures. Therefore, one would like to have methodologies that may provide probabilistic QoS.

Multiple paths, specifically, those that satisfy node-disjointness between a given source and a destination may help in QoS provisioning in cases when the bandwidth required by a flow of packets is larger than that which can be provided by a single path. Ideally, these paths should be uncorrelated so as to have the maximum usage in terms of throughput and bandwidth utilization. However, identifying multiple node-disjoint paths that are uncorrelated or independent is a challenging task. In [Wang and Silvester, 1993], the problem of finding the maximum number of node-disjoint paths between a given source and a destination such that there are no cross links between nodes that belong to different paths has already been proved to be NP complete. Finding multiple uncorrelated node-disjoint paths between a given pair of nodes in an ad hoc network is the same as finding a chordless cycle in a graph that contains source and destination nodes [Mohanoor, et al, 2008]. The later problem is an NP-complete problem [Bienstock, 1991]. In [Waharte and Boutaba, 2008], it is pointed out that the problem of finding two node-disjoint paths that do not interfere with one another in a connected network is NP-complete. In other words, one may not be able to identify not only the maximum number of uncorrelated node-disjoint paths but also even two node-disjoint paths that are uncorrelated in a reasonable amount of time.

On the other hand, some specialized methods that use directional antennas to mitigate path correlation among node-disjoint paths are described in [Abbas, et al, 2007]. In [Huang et al, 2007], the maximum flow problem is formulated as an optimization problem using switched beam directional antenna for sensor networks with limited interference. How to select multiple paths based on channel aware multipath metric in mesh networks is described in [Sheriff and Royer, 2006]. An interference-aware metric for path selection for reliable routing in wireless mesh networks is presented in [Tsai and Moors, 2007]. An interference-minimized multipath routing for high rate streaming in wireless sensor networks is presented in [Teo, et al, 2008]. The impact of interference on optimal multipath routing is studied in [Haan, et al, 2007]. Therein, a nonlinear programming problem is formulated to optimize the network performance and an algorithm is proposed that searches an exponential number of linear programmes to approximate the optimal solution.

In this paper, we propose a routing protocol that provides a probabilistic QoS in mobile ad hoc networks. In our protocol, the ensuing data packets may flow through multiple node-disjoint paths, one along one path. However, at each hop the service received by the packet depends upon the following factors: how much bandwidth is available, how urgent the packet is, and if assigned a higher priority whether the packet can meet the deadline till the destination.

The rest of the paper is organized as follows. Section 2 contains problem formulation and major issues. In Section 3, we present the proposed protocol. In Section 4, we analyze the probability of meeting the QoS. Section 5 contains results and discussion. Finally, the last section is for conclusion and future directions.

2. Problem Formulation and Major Issues

In a network with a wireless channel, two or more node-disjoint paths between a given pair of nodes can be correlated. By correlation, we mean that nodes along one path have their neighbors that lie along other node-disjoint paths. In other words, there can be cross links from one path to other paths. Specifically, two paths that have k links from one path to another are said to be k-correlated. Consider a network shown in Figure 1, there are three node-disjoint paths from a given source S to a destination D represented by bold lines. Links between intermediate nodes of more than one node-disjoint paths are shown by dashed lines. Note that two node-disjoint paths $<S,a,b,c,e,D>$ and $<S,f,g,h,D>$ have six links going from nodes of one another, therefore, these two paths are 6-correlated. Similarly, paths $<S,f,g,h,D>$ and $<S,i,j,k,l,D>$ are also 6-correlated.

In a mobile ad hoc network, where generally an omni-directional antenna is used, the transmissions of a node are heard by all nodes that are in the vicinity. Generally, in an ad hoc network, the Medium Access Control (MAC) layer protocol used is IEEE 802.11, which uses carrier sensing mechanism, specifically, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In a carrier sensing multiple access, a node senses the channel. If it finds the channel busy, it chooses a backoff period and waits. When the backoff period is expired, it senses the channel again. Continuing in this manner, either it finds the channel free or it leaves if the number of retries exceeds a threshold limit. One can see the detailed description of 802.11 in the literature. The point to mention here is that nodes contend for the channel with their neighbors. So, if a node belongs to a separate node-disjoint path and some of its neighbors are part of other node-disjoint paths; then the gain that may result using node-disjoint paths in an ad hoc network may not be the same as that expected in other wired/wireless networks. The nodes that belong to separate node-disjoint paths may end up contending simultaneously for the channel and that will bite the gains of having multiple node-disjoint paths. If these paths were independent or uncorrelated, then there might have not been any such contention and packets might have travelled along multiple node-disjoint paths independently. Ideally, one would like to have independent node-disjoint paths from a given source to a destination. However, finding uncorrelated node-disjoint paths between a given pair of nodes is not an easy task as it comes out to be an NP-complete problem. In this paper, we wish to find node-disjoint path such that nodes are aware of the correlation.

On the other hand, packets may be delayed due to several reasons such as unavailability of required bandwidth, the congestion, correlation, node and link failures, node mobility etc. As a result, a packet may or may not be able to meet its deadline till the destination depending upon the service received at upstream nodes. In this paper, we wish to analyze the probability of meeting the deadline by a packet under some reasonable assumptions.

Note that two or more than two node-disjoint paths from a given source to a destination shall never pass through a common intermediate node. However, it may happen that there might be more than one node-disjoint path that belong to different source-destination pairs that pass through a common intermediate node. For example, consider a network shown in Figure 2, there are two communicating pairs of nodes, $S_1 - D_1$ and $S_2 - D_2$. Each of these pair of nodes has three node-disjoint paths between them. However, two node-disjoint paths pass through nodes a,b,c,f,g,i,j, ofcourse, for different source-destination pairs. Even if $S_1 - D_1$ and $S_2 - D_2$ were the only pairs of nodes communicating simultaneously, then loads on these nodes would be larger than that on other nodes in the network. There can be multiple such pairs of nodes communicating simultaneously. As a result, nodes through which a relatively

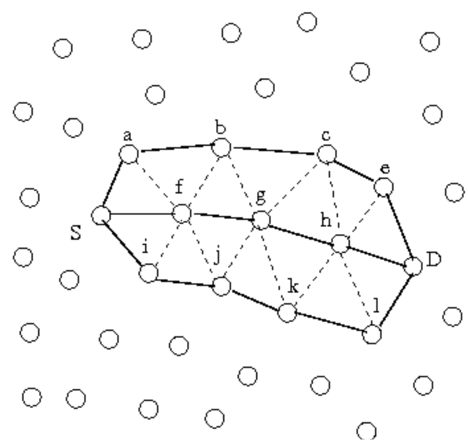

Figure 1. Node-disjoint paths from a given source S to a destination D are correlated

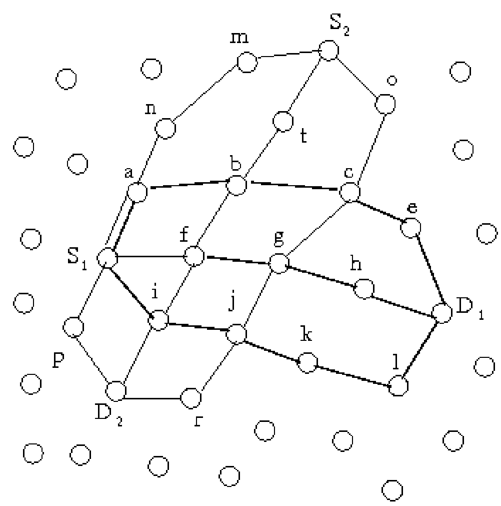

Figure 2. Intermediate nodes that are part of node-disjoint paths may be heavily loaded as they are lying on paths between more than one pair of nodes

large number of routes pass can be considered to be relatively heavily loaded in comparison to those along which none or very few routes pass. Therefore, an issue is how to deal with nodes that are relatively heavily loaded.

3. Proposed Protocol

We call the proposed protocol Probabilistic QoS Routing (PQR). It is an on-demand protocol, and like most of the other on-demand protocols, it is based on the request-reply cycle. The protocol consists of two major phases: *route discovery* and *route maintenance*. The description of each of these phases is as follows.

3.1. Route Discovery

In the *route discovery* phase, the protocol tries to identify multiple node-disjoint paths from a given source to a destination. In other words, whenever a source needs to communicate to a destination node, it looks for a path to the intended destination in its *RouteCache*. If it finds a valid path, it starts sending packets along the path to the destination. If the source does not find a path in its *RouteCache*, it generates a *Route REQuest* (RREQ) packet and sends it to its neighbors. An RREQ contains the following information <*SourceAddress, DestinationAddress, SourceSeqNo, PathTraversed*>. Let us call a node that is neither the source nor the destination of an RREQ as an intermediate node. When an intermediate node receives a copy of an RREQ, it processes the RREQ according to the RREQ forwarding policy. The RREQ forwarding policy that we use at an intermediate node is *All Disjoint Copies* (ADC) [Abbas and Jain, 2006]. In ADC, an intermediate node forwards a copy of the RREQ if *PathTraversed* of the RREQ is disjoint with the copies of the RREQ already forwarded. Otherwise, the copy of the RREQ is discarded.

Eventually, the copies of the RREQ reach the destination. The destination is responsible for computing the disjointness. The destination sends a *Route REPly* (RREP), one for each node-disjoint path. An RREP contains the following information <*SourceAddress, DestinationAddress, SourceSeqNo, Path, OtherPaths*>. Note that *Path* contains the path from the source to the destination, albeit in reverse direction. The field *OtherPaths* contains the information about other computed node-disjoint paths. When an RREP travels upstream, an intermediate node unicasts it to the next hop node along the path towards the source. The information contained in *OtherPaths* has to be stored by all intermediate nodes that receive and unicast the RREP. For that purpose, all nodes in the network maintain a *PathInfoCache* in which information about node-disjoint paths for a pair of communicating nodes is stored.

Following this way, all intermediate nodes that are lying on node-disjoint paths from the given source to the destination have information about all other node-disjoint paths from the source to the destination. Also, a node knows about its neighbors either through a beacon mechanism or through MAC layer contentions. An intermediate node sorts the node-id's of all its neighbors, and each node-disjoint path. It then computes which of its neighbors are parts of other node-disjoint paths.

3.2. Route Maintenance

In the *route maintenance* phase, when a node detects a link failure, it marks the failed outgoing link to be invalid. It, then, looks its *route cache* to find what are paths that are using the failed link. For each path, it sends a *RREQ-Maintenance* (RREQM) to its neighbors. Its neighbors reply by sending *RREP-Maintenance* (RREPM) packet. Upon receiving RREPM, the intermediate node checks whether it can repair the path such that node-disjointness with other paths that have not yet failed is maintained. If it is able to find such a neighbor, it generates a *Route rePAIR* (RPAIR) packet towards the source. Every node that receives RPAIR packet modifies the path in their corresponding *RouteCache*s for the particular source and destination. If the intermediate node is unable to find such a neighbor, it marks the path to the destination as invalid. It, then, generates a *Route ERRor* (RERR) message and sends it towards the source. When an intermediate node receives an RERR, it marks the path invalid and unicasts the RERR upstream. Upon receiving an RERR, the source marks the path invalid and looks for an available path to the destination in its route cache. If it finds a valid path in its route cache, it starts sending data packets towards the destination. Otherwise, it initiates a new route discovery.

Note that an intermediate node along a path that detects the link failure is allowed to repair the routes through its immediate neighbours. The node is able to do so because it is aware of the complete node-disjoint paths from the source to the destination. If the path is repaired, there will be changes only at one place in the routing entries of the upstream nodes when RPAIR packet travels upstream. However, an intermediate node would not like to repair the path beyond one hop because doing so will put an additional burden on the intermediate node as it may be relaying packets for other source-destination pairs besides its own traffic. Remember that doing otherwise may involve paths to be identified till the destination, and no body would like to discover paths on behalf of others, if one has its own traffic; and the source node also would not like it to be done. Therefore, we stick to the repair of paths by an intermediate node only upto one hop.

There is a question. How the nodes along the node-disjoint paths come to know that a particular path is amended due to link or node failures? To that end, whenever there is a modification among paths that were currently in use, the source assumes the responsibility to propagate the modification to nodes along currently used node-disjoint paths to the destination. For that, the source unicasts a *Route MODification* (RMOD) packet to its neighboring nodes that lie along node-disjoint paths to the destination. The RMOD packet contains the following information <*SourceAddress, DestinationAddress, RREQSeqNo, NewPath*(s)>, where the <*SourceAddress*, *DestinationAddress*, *RREQSeqNo*> contain the source address, destination address, and sequence number of the RREQ which was used to identify the route. The field *NewPath*(s) contains the newly computed paths to the destination that was repaired and reported by a node that sensed the link failure to the source. When an intermediate node receives an RMOD packet, it makes corresponding change in its *RouteCache* and unicasts it to the next hop node along the path. After that the intermediate node computes again who are its neighbors that lie along other node-disjoint paths from the source to the destination. This is done so as to update the information about the correlation at nodes along the modified set of node-disjoint paths.

4. Analysis

In this section, we analyze the probabilistic guarantees for provision of QoS. First, we discuss the adjustment made in the available bandwidth due to correlation and then we shall discuss the probabilistic guarantees.

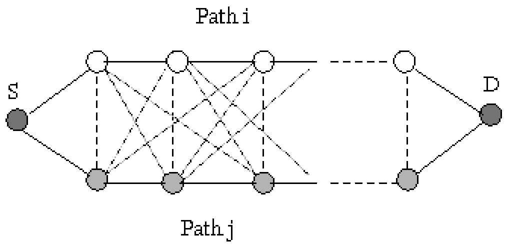

Figure 3. The maximum correlation between path *i* and path *j*

4.1. Adjustment for Path Correlation

Let there be m_i intermediate nodes in path i and m_j intermediate nodes in path j, then maximum number of links that may be common to path i and path j are

$$\begin{aligned} l_{\max} &= m_i m_j \\ &= (h_i - 1)(h_j - 1) \end{aligned} \tag{1}$$

where h_i and h_j are number of hops in path i and j, respectively. Note that the l and $l_{\max}$ have the units number of links (i.e. the product of $(h_i - 1)$ and $(h_j - 1)$ should not be confused to have a unit in square of the number of hops, it is simply a number). The proof of the above formula is simple and can be understood as follows. There are m_i intermediate nodes at one end and m_j intermediate nodes on the other end. As shown in Figure 3, one node out of m_i nodes can be selected in m_i number of ways. Similarly, one node of m_j nodes can be selected in m_j number of ways. Total number of ways in which two nodes are selected such that one of them is from m_i nodes and the other is from m_j nodes are $m_i m_j$ which is given by (1).

Let l be the number of links that are common in path i and j. We define a term *path correlation factor*, a , as follows.

$$\begin{aligned} \mathrm{a} &= \frac{l}{l_{\max}} \\ &= \frac{l}{(h_i - 1)(h_j - 1)} \end{aligned} \tag{2}$$

Let the raw bandwidth of outgoing links from a node be W. Due to path correlation, the bandwidth is reduced because multiple packets cannot be sent along node-disjoint paths that pass through neighboring nodes. We, heuristically, propose that the effective bandwidth of an outgoing link can be written as

$$W_{eff} = W\left(1 - \mathrm{a}^{\mathrm{g}}\right) \tag{3}$$

where, g denotes an offset for the number of source-destination pairs whose node-disjoint paths are passing through the node. For a single source-destination pair, $\mathrm{g} = 1$. For multiple source-destination pairs, $\mathrm{g} = 1 + \mathrm{b}$, where b depends upon how many neighbors of a node are common among different path sets of multiple source-destination pairs out of the total number of neighbors of the node.

4.2. Probabilistic Guarantees

We analyze probabilistic guarantees that may be provided using the awareness about the correlation among multiple node-disjoint paths. We assume that all data packets have been stamped for the time of start of their journey from the source and their deadlines. Let Δ be the maximum tolerable end-to-end delay. In other words, a packet sent by the source should reach the destination during the time Δ. Let δ be the time that a packet may spend at an intermediate node. Let h_i be the number of intermediate nodes along a path i. Then, the time that a packet is allowed to spend over an intermediate node is

$$\delta = \frac{\mathrm{D}}{h_i}.$$

In practice, some of the nodes that are lightly loaded may transmit the packet earlier while others that might be heavily loaded may forward the packet after relatively large delays. In other words, δ may vary from node to node and from packet to packet. Therefore, while forwarding a packet towards the destination one should take into account what service is received by the packet at upstream nodes and one should consider the residual time required and the remaining number of nodes to be traversed by the packet so as to reach the destination. Let $\hat{\Delta}$ be the remaining time and $\hat{h}_i$ be the remaining number of intermediate nodes that need to be traversed by the packet. Then, the average time that the packet may spend on an intermediate node downstream is

$$\hat{\delta} = \frac{\hat{\Delta}}{\hat{h}_i}.$$

A packet is said to be delayed at a node if it is not able to relay the packet on to its outgoing link within the stipulated time. The excess of amount time incurred at a node other than the packet was supposed to spend may or may not be recovered till the packet arrives at the destination. Also, whether the packet will be able to make up till the destination depends on the extent of delay. A large amount of time that a packet spends at a node is queueing delay which depends upon the queue length or number of occupied buffers. If the queue length is below a threshold value, the packets are not delayed. If the queue length grows beyond the threshold, the packets will start to be delayed. In general, the probability that a packet is delayed depends upon the queue length. However, whenever a node has both realtime packets and best effort packets, priority is given to relay the realtime packets. The priorities are nonpreemptive. The excess delay, therefore, will not only depend upon the queue length, it will also depend upon how many packets of higher priority are there before the realtime packet that arrives at the node.

On the other hand, the probability that a delayed packet will be able to make up the excess delay incurred at a node till it reaches the destination will depend upon at what hop the packet is delayed and the amount of excess delay incurred. In other words, the probability of making up for the extra delay by a packet that is still near the destination is larger than the situation when a packet is delayed and is near the destination. We assume that the probability that a delayed packet that is currently at an intermediate node i will be able to make up till it arrives at the destination is heuristically given by

$$P_i = \left(1 - \psi^{\frac{1}{i}}\right)\left(1 - \frac{\varepsilon}{\delta}\right) \tag{4}$$

where, ε is the value by which the delay incurred by a packet at a node exceeds the delay, δ, that the packet was normally supposed to spend. The variable ψ can be written as

$$\psi = \frac{1}{h_k C} \tag{5}$$

where, h_k is the number of hops in the path k, and C is a variable. The variable C depends on a number of factors e.g. node-density, node distribution, etc. For example, C is chosen in such a fashion so that $\psi \approx 0.01$ for 100 nodes distributed uniformly randomly in a region of area 1000*m* × 1000*m*.

Let n nodes be distributed uniformly randomly in a region of area A with a node density $\rho = \frac{n}{A}$. Let the transmission range of each node be r, then the average number of neighbors of a node is $\rho\pi r^2-1$. Let h_k in (5) represent the average number of hops along a path between any given source and a destination. For computation of average number of hops, we refer the readers to \cite{harb}. Then, we suggest the value of the variable C may be approximated using the following expression

$$C = \rho\pi r^2-1 \tag{6}$$

which is nothing but the average number of neighbors of a node in the network.

Let B_i be the probability that an intermediate node has enough bandwidth to relay the packet despite the correlation. The probability B_i can be related to (3) such that

$$B_i = \frac{W_{eff}}{W}. \tag{7}$$

Then, the probability that the packet will be able to arrive at the destination before the deadline is

$$P_D = \prod_{i}^{h_k}\left\{\left(1-\psi^{\frac{1}{i}}\right)\left(1-\frac{\varepsilon_i}{\delta_i}\right)B_i\right\} \tag{8}$$

where ψ is defined as in (5).

As mentioned earlier, ε_i is an amount of delay by which the delay caused by a node i, δ_i', exceeds the normal delay, δ_i, that a packet is supposed to spend at node i of path k. In other words, $\varepsilon_i = \delta_i' \times \delta_i$. Note that the delay, δ_i', consists of all delays such as queueing delays, MAC scheduling delays, transmission delays at node i.

$f = \frac{\varepsilon}{\delta} = 0.0$ and for $f = \frac{\varepsilon}{\delta} = 0.1$.

Algorithm 1: Actions taken at an intermediate node i for a delayed packet to enable it to catch the deadline

if $\{(p_i > p^*)$ **then**
$\quad Count\text{PrioClass}_j < ThrClassCount_j$
Increase the priority of the packet
else if $\{(BW_i^{avail} \geq MinimumAvailBW)$ &&
$\quad (Utilization_i \leq MaxUtilization_i)\}$ **then**
Increase bandwidth allocated to flow of packets
end if

Note that a node may increase the priority of the data packet if it is delayed less than a threshold value provided that the node has not forwarded too many packets of the same priority level within a time window. A packet which is delayed too much is less likely to reach at the destination in time. Similarly, forwarding too many packets with high priorities may not have a significant effect in terms of meeting the deadline. Further, a node may increase the bandwidth beyond the bandwidth reserved so as to speed up the packet till next hop provided that the enough additional bandwidth is available and the bandwidth utilization is fairly low.

The actions taken at an intermediate node, say i are summarized in Algorithm 1. In Algorithm 1, the variable p^* denotes the probability threshold which may generally be taken greater than or equal to 0.5. Otherwise, it is less likely that the packet will be able to meet its deadline finally at the destination. The variable $Count\text{Prio}Class_j$ denotes the number of packets of priority class j that node i has already forwarded, and the variable $ThrClassCount_j$ denotes the threshold on the number of packets of the same priority class that a node is allowed to forward. The variable BW_i^{avail} denotes the bandwidth available at node i, and the variable $MinimumAvailBW$ denotes the minimum bandwidth that should always be available at a node. On the same lines, the variable $Utilization_i$ denotes the utilization of bandwidth at node i, and the variable $MaxUtilization_i$ denotes the maximum value of the utilization. If the condition $\{(P_i > p^*)$ && $(Count\text{PrioClass}_j < ThrClassCount_j)\}$ comes out to be true, then node i increases the priority of the packet. Otherwise, node i examines whether the condition $\{(BW_i^{avail} \geq MinimumAvailBW)$ && $(Utilization_i \leq MaxUtilization_i)\}$ is satisfied or not. If yes, then node i allocates little more bandwidth to the stream of packets so as to enable them to catch the deadlines at the destination.

5. Results and Discussion

In this section, we present some empirical results. As mentioned earlier, we assumed that the probability that a packet (which is delayed at an intermediate node i along a node-disjoint path) is able to make up till it reaches the destination decreases as the packet travels downstream. We pointed out that this assumption is somewhat realistic because a packet can make up for the delay near the source as compared to the situation when it is about to arrive at the destination. Figure 4 shows the probability P_i at i th intermediate node along a path from the given source to the destination. Here,

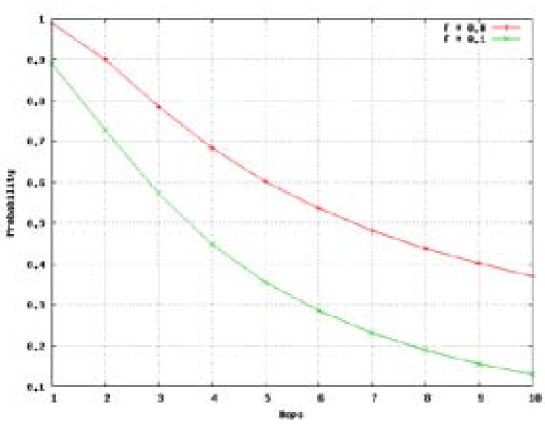

Figure 4. The probability P_i as a function of number of hops for

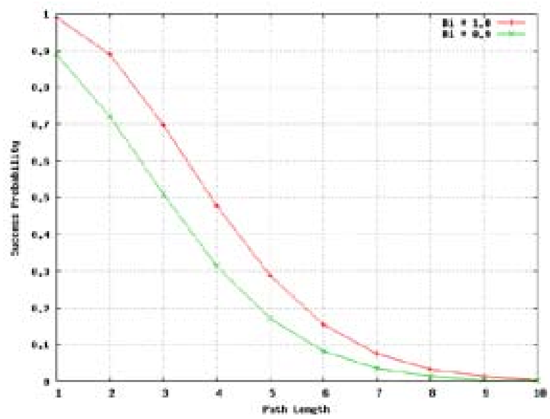

Figure 5. The probability of success, P_D, as a function of path length for $B_i = 1.0$, and $B_i = 0.9$

the number of intermediate nodes in the path, h_k, is 10. The probability P_i is shown for $f = \frac{\varepsilon}{\delta} = 0.0$ and for $f = \frac{\varepsilon}{\delta} = 0.1$. As the packet travels downstream, the probability that a delayed packet will be able make up for the delays till the destination decreases. If the packet is delayed more, the probability that the packet will be able to reach at the destination in time decreases faster as compared to the less delayed situation. Figure 5 shows the probability that the packets sent by the source will be able to reach the destination on or before the deadline, P_D, as a function of the path length. In this case, the probability that sufficient bandwidth is available at an intermediate node is assumed to be $B_i = 1.0$, and $B_i = 0.9$. As the length of path is increased, the probability that a packet sent by the source will be able to meet the deadline decreases. For smaller values of the probability that enough bandwidth is available, the success probability of meeting the deadlines is smaller.

6. Conclusion

Provision of QoS in an ad hoc network is a challenging task. In this paper, we presented a protocol for the provision of QoS in mobile ad hoc networks. In our protocol, intermediate nodes are aware of the correlation between multiple node-disjoint paths. The awareness about the dependencies or correlation enables nodes along the paths from the source to the destination to take remedial actions so as to provide QoS to packets flowing between the given source to the destination. We have shown that by using multiple node-disjoint paths in which nodes are aware of the correlation, one may have probabilistic guarantees for packets that are required to arrive at the destination before their respective deadlines. Further validation of the protocol forms our future work.

Acknowledgement

This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. A preliminary version of this paper appeared in the proceedings of the First IEEE International Conference on Networked Digital Technologies (NDT), July 29-31, 2009, Ostrava, Czech Republic [Abbas and Kure, 2009].

References

[1] Abbas, A. M., Alam, M., Jain, B. N. (2007). Alleviating Path Correlation Using Directional Antenna in Node-Disjoint Multipath Routing for Mobile Ad hoc Networks. *In:* Proceedings of 3rd ACM International Wireless Communication and Mobile Computing Conference (IWCMC), p. 405-410, Honolulu, Hawaii, August 12-14.

[2] Abbas, A. M., Jain, B. N. (2006). Path Diminution is Unavoidable in Node-Disjoint Multipath Routing with Single Route Discovery. *In:* Proceedings of 1st IEEE International Conference on *COMmunication SoftWAre and MiddlewaRE* (*COMSWARE)*, p. 1-6, New Delhi, January.

[3] Abbas, A. M., Kure, Ø. (2009). A Probabilistic Quality of Service Routing for Mobile Ad hoc Networks. *In:* Proceedings of First IEEE International Conference on *Networked Digital Technologies (NDT),* p. 269-273, Ostrava, July.

[4] Bienstock, D. (1991). On the Complexity of Testing for Odd Holes and Induced Odd Paths. *Discrete Mathematics,* 90(1) 85-92, June.

[5] Haan, R. D., Boucherie, R. J., Ommeren, J. K. V. (2007). The Impact of Interference on Optimal Multi-path Routing in Ad Hoc Networks. *In:* Proceedings of 20th International Teletraffic Congress (ITC), LNCS 4516, p. 803-815, Ottawa, June.

[6] Harb, S. M., McNair, J. (2008). Analytical Study of the Expected Number of Hops in Wireless Ad Hoc Networks. *In:* Proceedings of 3rd International Conference on *Wireless Algorithms, Systems, and Applications (WASA).* LNCS 5258, p. 63-71, Dallas, Texas, October.

[7] Huang, X., Wang, J., Fang, Y. (2007). Achieving Maximum Flow in Interference-Aware Wireless Sensor Networks with Smart Antennas, *Elsevier Journal on Ad hoc Networks*, 5(6) 885-896, February.

[8] Mohanoor, A. B., Radhakrishnan, S., Sarangan, V. (2008). Interference Aware Multi-path Routing in Wireless Networks. *In:* Proceedings of 5th IEEE International Conference on *Mobile Ad Hoc and Sensor Systems (MASS),* p. 516-518, Atlanta, Georgia, October.

[9] Sheriff, I., Royer, E. M. B. (2006). Multipath Selection in Multi-Radio Mesh Networks. *In:* Proceedings of 3rd IEEE International Conference on Broadband Communications (BROADNETS), San Jose, California, p. 1-11, October.

[10] Teo, J. Y., Ha, Y., Tham, C. K. (2008). Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High-Rate Streaming. *IEEE Transactions on Mobile Computing*, 7(9) 1124-1137, September.

[11] Tsai, J. W., Moors, T. (2007). Interference-Aware Multipath Selection for Reliable Routing in Wireless Mesh Networks. *In:* Proceedings of 3rd IEEE International Conference on *Mobile Ad hoc and Sensor Systems (MASS),* p. 1-6, Pisa, October.

[12] Waharte, S., Boutaba, R. (2008). On the Probability of Finding Non-Interfering Paths in Wireless Multihop Networks. *In:* Proceedings of International Conference on Networking (NETWORKING), pages 914-921, Singapore, May.

[13] Wang, J. L., Silvester, J. A. (1993). Maximum Number of Independent Paths and Radio Connectivity. *IEEE Transactions on Communications,* 41(10) 1482-1493, October.

Author Biographies

Ash Mohammad Abbas is a Post-Doctoral Research Fellow under European Research Consortium on Informatics and Mathematics (ERCIM) at Centre for Quantifiable Quality of Service in Communication Systems (Q2S), Norwegian University of Science and Technology (NTNU), Norway. He is a Reader at Department of Computer Engineering, Aligarh Muslim Univertsity, India. He received his PhD from Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India. His current research interests include mobile ad hoc and sensor networks, quality of service, and routing.

Øivind Kure is a Professor at Centre for Quantifiable Quality of Service (Q2S) and Department of Telematics, Norwegian University of Science and Technology (NTNU), Norway. He obtained his PhD from University of California, Berkeley in 1988. His current research interests include quality of service, multicast protocols, and ad hoc networks.