A Quality of Service Aware Routing Using Multiple Paths for TDMA-Based Ad hoc Networks

Khaled Abdullah Mohd Al Soufy[1], Ash Mohammad Abbas[2]
[1]Department of Computer Engineering
Zakir Husain College of Engineering and Technology
Aligarh Muslim University
Aligarh, India
kalsoufy@gmail.com

[2]Center for Quantifiable Quality of Service in Communication Systems
Centre of Excellence
Norwegian University of Science and Technology
Trondheim, Norway
ash@q2s.ntnu.no

The corresponding author is Ash Mohammad Abbas whose email address is ash@q2s.ntnu.no.

ABSTRACT: *Provision of quality of service (QoS) in an ad hoc network is a challenging task. In this paper, we propose a QoS aware routing for Time Division Multiple Access (TDMA) based ad hoc networks. Our protocol tries to identify multiple paths each of which is capable of providing the QoS in terms of the number of time slots at its own or by combining it to a group of paths. Our protocol incorporates a procedure to determine the available time slots in a localized and distributed fashion. We evaluate the performance of the protocol through simulations. While evaluating the performance, we focus on the number of paths that are able to satisfy the QoS and the QoS success ratio.*

Keywords: Ad hoc networks, Quality of service, Time division multiple access, Multipath routing.

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

1. Introduction

An ad hoc network can be formed spontaneously without actually needing a centralized infrastructure support. An ad hoc network is a self organizing and self healing in nature. Such a network finds applications in battlefield communications, emergency rescue missions, law enforcement, riot control and online conferences or classrooms. There are many peculiar characteristics of an ad hoc network. There are no separate routers, therefore, participating nodes have to relay the packets to their ultimate destinations. The devices used in such a network possess limited transmission range that generally results in multiple hops from a given source to a destination. The devices used are often operated through batteries whose powers are limited and their power depletions may cause node and associated link failures. Also, nodes may move about randomly, and therefore, links may break frequently. As a result, the topology of the network is highly dynamic. The absence of a centralized infrastructure implies that the participating nodes cooperate among themselves so as to communicate with one another.

In order to enable an ad hoc network to run some applications such as multimedia streaming, it is desirable that the network supports some form of *quality of service* (QoS). However, the provision of QoS is not an easy task in case of an ad hoc network. The dynamic topology of the network may not let any hard guarantees to be made about the QoS to be provided by the network. Further, a protocol or scheme should be able to work with partial topology information in a localized and distributed manner as the information about the whole topology of the network is not available at any centralized node. Specifically, a node knows only about its neighbors. Therefore, a protocol should be able to use only localized topology information and should not require the information about the entire topology of the network.

Recently, *time-division multiple access* (TDMA) is studied as a medium access control (MAC) layer protocol to satisfy the QoS requirements of an application [Jawhar and Wu, 2005], [Li et al, 2006], [Wu et al, 2004], [Sriram et al, 2004]. A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network is proposed in [Jawhar and Wu, 2005]. A multipath QoS routing with power control in TDMA-based mobile ad hoc networks is proposed in [Li et al, 2006]. QoS routing in *time division multiple access/code division multiple access* (TDMA/CDMA) based ad hoc networks has also been investigated in [Wu et al, 2004]. A framework for the influence of QoS routing on the achievable capacity of TDMA-based ad hoc networks is described in [Sriram et al, 2004]. A distributed QoS routing protocol that is based on ticket based probing is proposed in [Chen and Nahrstedt, 1999]. The protocol in [Chen and Nahrstedt, 1999] may provide multiple paths between the given source and destination.

In this paper, we present a protocol for TDMA-based ad hoc networks. In our protocol, the source tries to identify multiple node-disjoint paths that are able to satisfy the bandwidth (expressed in terms of number of time slots) required by an application. Our protocol is distributed and is able to work with partial and localized topology information. Our approach is different from the approach presented in the [Liao et al, 2002] as per the following. Firstly, in our approach, multiple node-disjoint paths are discovered between the source and the destination. This is in contrast to the approach used in [Liao et al, 2002], where only one path that may satisfy the bandwidth requirements is discovered. Secondly, in our approach, if a single path is not able to satisfy the bandwidth requirement of a flow of packets, then the bandwidth is reserved along multiple node-disjoint paths in the ratio of the bandwidth along the paths so selected. Further, a flow will be discarded only when the bandwidth required by the flow exceeds the total available bandwidth provided by all available node-disjoint paths between the source to the destination. However, as per the approach described in [Liao et al, 2002], the flow has to be discarded if the single path fails to provide enough bandwidth.

The rest of this paper is organized as follows. Section 2 contains the related work. In Section 3, we present the proposed protocol. Section 4 contains results and discussion. Finally, the last section is for conclusion and future work.

2. Related Work

In an ad hoc network that is based on TDMA, the notion of bandwidth is related to the free time slots available along a link for transmission. The number of free slots is decided by the neighbors and their transmission/reception activities. Therefore, reservation of bandwidth from a source to a destination implies reservation of time slots that are free and are available for sending packets. The methods of determining what time slots are free and available for transmission can be different and define the approach taken by a protocol.

Figure 1 shows that a TDMA frame consists of a number of time slots. The number of time slots in a frame is called the frame length. A time slot *i* is divided into a *control slot* (CS) and a *data slot* (DS). A control slot may occur at both ends of the data slot i.e. as a header and a trailer. Control slots are used for synchronization of data slots.

A QoS routing (QoSR) protocol based on Destination Sequence Distance Vector (DSDV) routing that tries to provide bandwidth guarantees is proposed in [Lin and Liu, 1999]. The protocol QoSR is based on TDMA. As mentioned earlier, bandwidth is defined in terms of the number of time slots available for transmission. The protocol is based on heuristics that may provide an approximate solution because finding a schedule of free slots that maximizes the bandwidth is an NP complete problem.

A protocol called Distributed Slot Reservation Protocol (DSRP) that comprises of several strategies for dynamic bandwidth allocation to be used in QoS routing for TDMA based ad hoc networks is proposed in [Shih et al, 2006]. In DSRP, QoS routing depends on the information gathered only from one-hop neighbors and thus making the protocol localized and distributed. An on-demand QoS routing protocol that is based on Ad hoc On-demand Distance Vector (AODV) routing for TDMA-based ad hoc networks is presented in [Zhu and Corson, 2002]. The protocol identifies a route from a given source to a destination and reserves bandwidth in terms of time slots for QoS routing. The protocol is able to restore a route when it fails due to changes in the topology of the network and thus has a support for mobility. An analytical model for computation of the bandwidth along a path is also discussed in [Zhu and Corson, 2002].

A protocol that reserves the bandwidth for QoS routing in networks that utilize TDMA at the MAC layer is proposed in [Liao et al, 2002]. The protocol takes into account the hidden terminal problem and the exposed terminal problem during the route establishment phase. A QoS routing protocol that identifies multiple paths using ticket based probing is proposed in [Chen and Nahrstedt, 1999]. A QoS routing protocol that is based on reservation pool is presented in [Pyo et al, 2007].

In this paper, we wish to present a protocol that has a provision of QoS for TDMA based ad hoc networks. As opposed to [Liao et al, 2002], the protocol described in the present paper is designed in such a manner so that there are multiple node-disjoint paths from the given source to the destination. If a single path is not able to provide enough bandwidth, then the destination examines whether a group of multiple node-disjoint paths can provide the required bandwidth. If yes, the bandwidth is reserved along the paths in the ratio of their available bandwidth. If the bandwidth required by the flow of packets is larger than the total bandwidth available along all paths from the given source to the destination, then the bandwidth is not reserved by the destination. The source is then informed about the unavailability of the required bandwidth. The source then discards the flow or asks it to wait till enough bandwidth is available. A flow will be discarded only when the bandwidth required by the flow exceeds the total available bandwidth provided by all available node-disjoint paths between the source to the destination. However, as per the approach described in [Liao et al, 2002], the flow has to be discarded if the single path fails to provide enough bandwidth.

Note that TDMA requires the slot assignment and that presumes some kind of centralized infrastructure support. However, in an ad hoc network, it is generally assumed that a centralized infrastructure is not present. The absence of a centralized infrastructure implies that a scheme or a procedure to be used for an ad hoc network should be able to work in a localized and distributed manner. Therefore, an issue that is to be addressed is devising a scheme for the assignment of time slots in a localized and distributed manner. We describe a procedure for determining time slots for transmission/reception by nodes in the network. The procedure is localized as it uses only the local topology information available at a node. The protocol is distributed in the sense that it does not require the computations to be performed at a centralized node. In what follows, we describe the proposed protocol.

3. Proposed Protocol

Our protocol is based on the request-reply cycle. Our protocol has three phases: *slot determination*, *route discovery*, and *route maintenance*. We briefly describe each of these phases.

3.1 Slot Determination

In the slot determination phase, every node in the network tries to determine how many and which slots are available to it for transmission and reception. Each node may have slots out of the maximum number of slots called *MaxSlots* depending upon what slots are occupied by its one-hop neighbors and two-hop neighbors.

Figure 2 shows an ad hoc network. Note that the slot used by a node *i* should not be used by its 1-hop neighbors. The reason is that when a node is transmitting, its neighbor should not transmit so as to avoid collision. When a node is transmitting, its 2-hop neighbors should not transmit simultaneously (or in the same slot) so as to avoid hidden node problem. Further, the same slot should not be used by a node which is exactly 3-hops away from a transmitting node so as to avoid exposed node

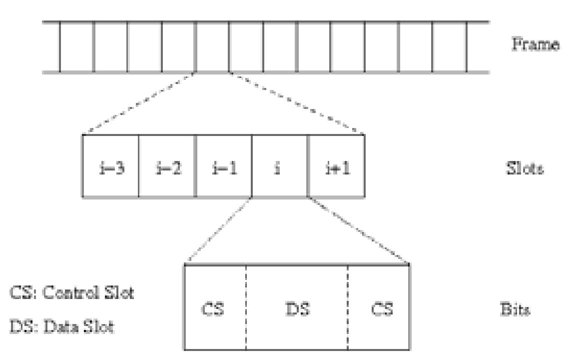

Figure 1. A TDMA frame that is divided into slots. A slot is further divided into *control slots* (CS) and *data slot* (DS)

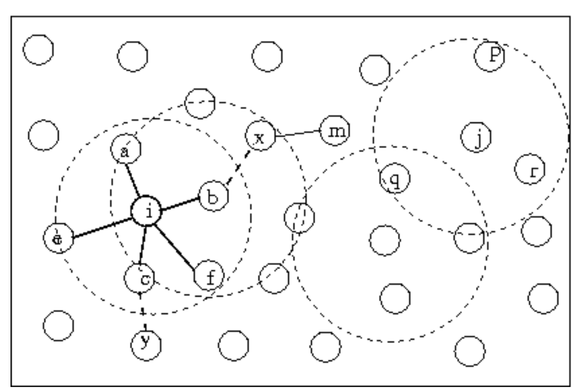

Figure 2. Bold lines represent 1-hop neighbors of a node *i* and dashed lines represent the 2-hop neighbors of node *i*. Thin regular lines represent the 3-hop neighbor of node *i*. Beyond three hops, slot numbers may be reused

problem. Beyond three hops, slot numbers may be reused for simultaneous transmissions as that shall not be harmful due to either hidden node problem or exposed node problem.

Initially (when a node does not have information about how many and what slots are taken by its neighbors), a node selects a number of slots randomly and enquires to its neighbors whether any one is using those slots or not by sending a *Slot ENQuiry* (SENQ). The SENQ contains the following.

<OriginatorAddress, EnqSequenceNo, SlotsCount, SlotsPicked, HopCount>.

The *HopCount* field contains a value 3 and is decremented whenever the SENQ is forwarded by a node. If a node receives a SENQ packet with *HopCount* equals to 0, it simply discards the SENQ packet.

Nodes that are 1-hop neighbors of the originator node examine whether their 1-hop neighbors (i.e. 2-hop neighbors of the originator node) and 2-hop neighbors (i.e. 3-hop neighbors of the originator node) are using the slots contained in the *SlotsPicked* field of the SENQ message. A *j*-hop neighbor (where *j* ranges from 1 to 3) of the originator node replies if it is using any of those slots picked by the originator node by sending a *Slot REPly* (SREP) message, otherwise they simply ignore the message. The SREP message contains the following information.

<OriginatorAddress, ReplyingNeighAddress, NeighHopCount, EnqSequenceNo, SlotsUsed, SlotsUsedByNeigh, NeighAddresses, SlotsUsedByTwoHopNeigh, TwoHopNeigh Addresses>.

Note that the field *SlotsUsedByNeigh* is used in replies sent by 1-hop and 2-hop neighbors only. It is insignificant if the SREP is sent by a node which is three or more hops away from the originator node. Similarly, the field *SlotsUsedByTwoHopNeigh* is used in replies sent by 1-hop neighbors only and it is insignificant if the SREP is sent by a node which is two or more hops away from the originator node.

After a timeout, the node which sent SREQ messages examines the SREP messages sent by the neighboring nodes. It marks the slots reported by its neighbors to be invalid. The slots that are not used by any of its neighbors are placed on the *AvailableSlots* list. The node repeats this step until and unless it has a required number of slots. Algorithm 1 summarizes the steps in the slot determination procedure. The variable *MySlots* represents a tentative set of slots picked by a node randomly. The variable *ReqdSlots* represents the number of slots required for satisfying the bandwidth requirements of a flow of packets. As mentioned above, a node starts with a random selection of slots and enquires its neighboring nodes upto three hops whether any of them is using the slots picked by it by a sequence of SREQ and SREP packets. The node crosses out the slots reported by the neighbors to be in use through SREPs. A node tries to determine slots in next iterations if the number of slots is not enough to satisfy the bandwidth requirements. Each and every node in the network is supposed to run the slot determination procedure. Using the above procedure, each node may determine the slots for communication in a distributed and localized manner.

Algorithm 1: Slot determination at a node

```
Select a set of slots MySlots
Generate SENQ packet with the following information <OriginatorAddress, ReplyingNeighAddress,
NeighHopCount, EnqSequenceNo, SlotsUsed, SlotsUsedByNeigh, NeighAddresses, SlotsUsedByTwoHopNeigh,
TwoHopNeighAddresses>.
Wait for SREP for a timeout
if PacketRecvd == SREP then
  Examine the SREP.NeighHopCount
  Cross out the slots from MySlots that are used by k-hop neighbors, k=1 to 3
  if |MySlots|<|ReqdSlots| then
    AvailSlots = MySlots
    Go to Step 1
  else if PacketRecvd == SREQ then
    if PacketRecvd.HopCount ≥0 then
      Append slots used to appropriate field of RREQ
      SREQ.HopCount= SREQ.HopCount+1
      Forward the SREQ to the neighbors
    else
    Discard the SREQ
    end if
  end if
end if
```

3.2 Route Discovery

In the route discovery phase, a source node that wants to communicate to a destination node generates a *Route REQuest* (RREQ) provided that it does not have a valid route to the destination. An RREQ contains the following information.

<*SourceAddress, DestinationAddress, SequenceNumber, TraversedHopList, ReqSlots, AvailableSlotsPath*>.

Note that the tuple <*SourceAddress, DestinationAddress, SequenceNumber*> uniquely identifies an RREQ. Two RREQs with the same values for the fields of the tuple are known as copies of one another. Let us call a node, which is neither the source of the RREQ nor is the destination of the RREQ, as an intermediate node.

When an intermediate node receives an RREQ, it examines whether its own address is already present in the *TraversedHopList* field of the RREQ. If yes, it discards the RREQ. Otherwise, it examines whether the *TraversedHopList* of the RREQ is disjoint with the *TraversedHopList*s of the RREQs already forwarded. If yes, it decides to forward the copy of the RREQ, otherwise, it discards the copy of the RREQ. Before forwarding the RREQ to its neighbors, the intermediate node appends its own address to the *TraversedHopList* of the RREQ.

Eventually, the copies of the RREQ reach the destination. The destination node collects the copies of the RREQ received until a timeout. It then heuristically computes the disjointness among the *TraversedHopList*s of the copies of RREQ. Actually, the problem of computing a *maximal set* of node-disjoint paths using the *TraversedHopLists* of copies of an RREQ received by the destination node is proved to be an NP-complete problem in [Abbas and Jain, 2006]. Therein, a heuristic for possibly computing a maximal set of node-disjoint paths is proposed. We use the same heuristic to possibly compute a maximal set of node-disjoint paths, and therefore, we use the word *heuristically*. The destination sends a *Route REPly* (RREP) against each copy of the RREQ with disjoint *TraversedHopList*. The RREP is sent towards the source along the reverse path brought by the copy as *TraversedHopList*. Note that this RREQ forwarding policy is called *All Disjoint Copies* (ADC) of an RREQ and is presented in [Abbas and Jain, 2006]. It is used to enhance the diversity among the *TraversedHopList*s of the RREQ that reach the destination and simultaneously limits the number of copies to be forwarded by an intermediate node.

Note that the field *ReqSlots* represents the number of slots required by the flow of packets. Further, the field *AvailableSlotsPath* represents the number of slots available along the path contained in *TraversedHopList* of the copy of the RREQ. If the number of available slots at node *i* (say *AvailableSlots*[*i*] is less than the *AvailableSlotsPath*, then node *i* replaces the value of *AvailableSlotsPath* by the number of slots contained in *AvailableSlots*[*i*]) before forwarding the RREQ to its neighbors. Otherwise, it leaves the field, *AvailableSlotsPath*, unchanged. It is clear that when the RREQ reaches the destination, the field *AvailableSlotsPath* contains the minimum number of slots that are available along any node along the path represented by the *TraversedHopList* of the RREQ. In other words,

$$AvailableSlotsPath = \min_{i=1}^{h}\left(AvailableSlots[i]\right) \tag{1}$$

where, h is the number of intermediate nodes along the path. It is assumed that the number of slots available at the source and the destination is larger than those available at any intermediate node lying along a path between the given source and the destination.

The destination has to examine whether a path satisfies the QoS requirement. If yes, the required bandwidth is reserved along the path when the RREP travels upstream. If a single path cannot satisfy the bandwidth requirement, the destination examines whether a group of node-disjoint paths can satisfy the bandwidth requirement. If yes, then the bandwidth is reserved along the group of paths in the ratio of their available bandwidth. If all paths taken together are not able to satisfy the bandwidth requirement, then the destination informs the source and does not reserve the bandwidth along paths. The source, then, discards the flow or asks the flow to wait till enough bandwidth is available. Putting it another way, let $|S_R|$ be the number of slots required to satisfy the bandwidth required by the flow. Let $|S_1|, |S_2|, ...|S_k|$ be the number of slots available along paths between the given source and the destination. Let $|S_1^{'}|, |S_2^{'}|, ...|S_k^{'}|$ be the number of slots reserved along these paths. Then, the following conditions should hold.

- $|S_R| \le \sum_{i=1}^{k} |S_i|$: This condition has to be satisfied to accept or reject a flow.
- $|S_R| = \sum_{i=1}^{q} |S_i^{'}|$: The total bandwidth reserved for the flow is the summation of the proportional bandwidths reserved along each path among the q selected paths.

The proportional bandwidth is given by the following expression

$$\left|S_j^{'}\right| \approx \frac{\left|S_j\right|}{\sum_{i=1}^{k}\left|S_i\right|}\left|S_R\right|, \forall j = 1, q \qquad \textbf{.(2)}$$

The approximation here means that the number of slots reserved along each path should be made whole numbers. In what follows, we describe the route maintenance phase.

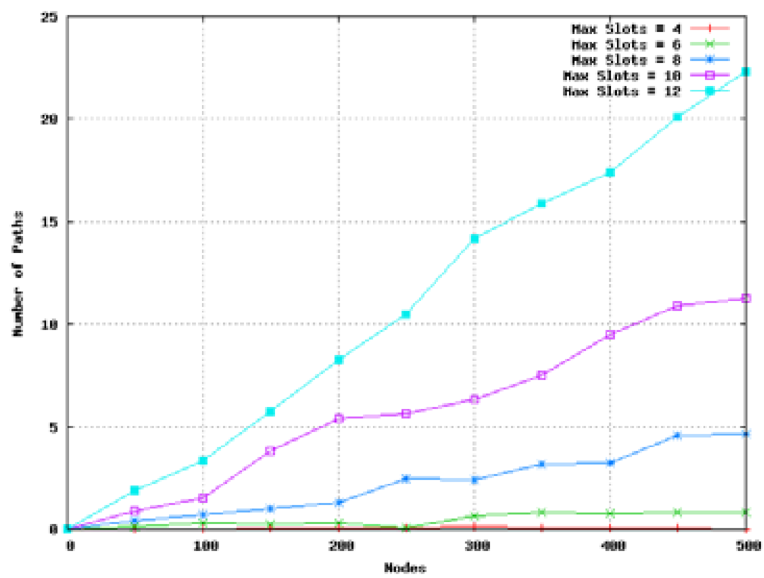

Figure 3. Average number of paths as a function of number of nodes for different number of *MaxSlots*

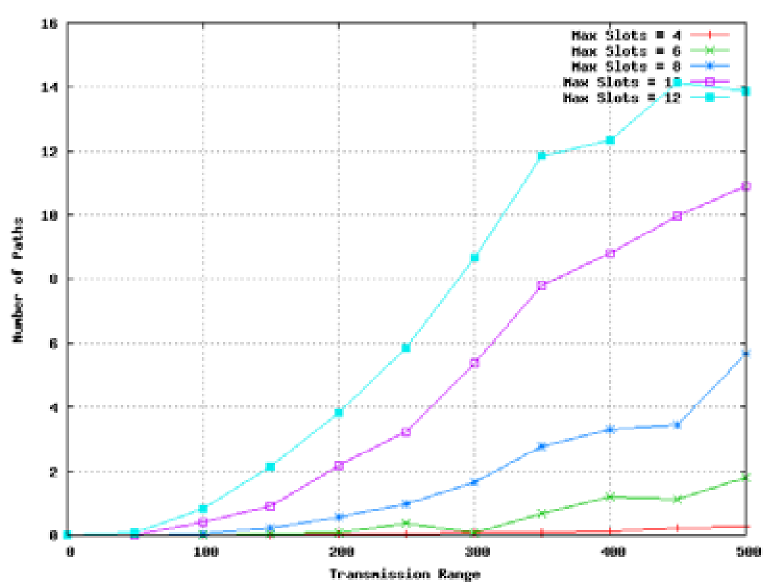

Figure 4. Average number of paths as a function of transmission range for different number of *MaxSlots*

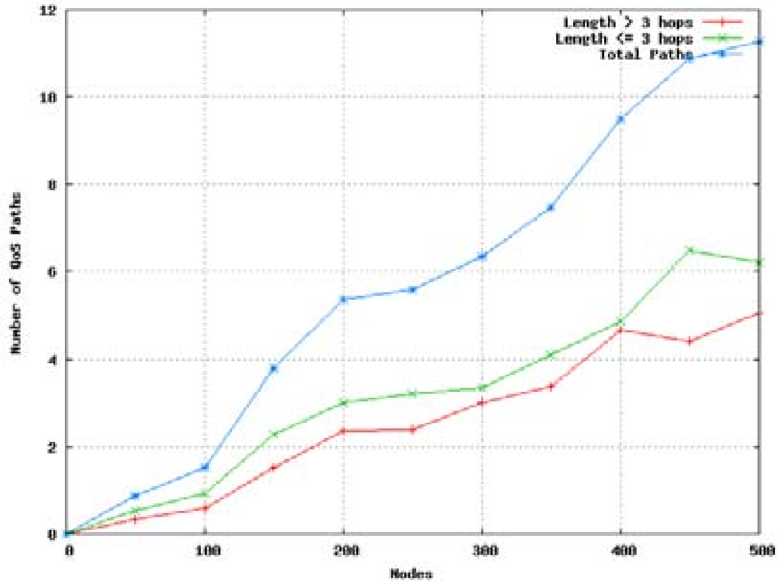

Figure 5. Average number of QoS paths as a function number of nodes for different path lengths

3.3 Route Maintenance

If a node senses a link failure, it informs upstream nodes along all those paths whose part the failed link was by unicasting *Route ERRor* (RERR) messages, one for each failed path. Every node that receives an RERR message marks the path invalid and unicasts the RERR upstream. Eventually, the RERR arrives at the source. When the source receives an RERR message it marks the failed path invalid. The source then looks for an alternate path, in the *route cache*. If it finds a path to the destination that is not yet failed and that satisfies the QoS requirements of the flow of packets, it starts sending data packets along the path. Otherwise, it initiates a new route discovery.

In what follows, we present the results and discussion.

4. Results and Discussion

We carried out simulations using our own simulator in C++. In our results, we have assumed that 200 nodes (unless and otherwise stated explicitly) are distributed uniformly randomly in a region of area 1000m × 1000m. The transmission range of each node is assumed to be 250m. The default value of the maximum number of time slots that may be available (if not occupied by neighboring nodes) is 10.

Figure 3 shows average number of paths as a function of number of nodes for different number of *MaxSlots*. As the number of nodes are increased the number of paths available increases. This is due to the fact that increasing the number of nodes increases the node density and as a result the number of neighbors of a node increases, thereby increasing the number of node-disjoint paths from a given source to a destination. Further, an increase in the *MaxSlots* increases the number of node-disjoint paths. The reason is that an increase in *MaxSlots* implies that the node has more available slots and therefore it is likely to find paths with a required number of minimum slots to support the desired QoS. This accounts for the observed behaviour.

Figure 4 shows average number of paths as a function of transmission range of nodes for different number of *MaxSlots*. As the transmission range of nodes is increased, the number of neighboring nodes increases, thereby, increasing the number of paths that may provide a given number of slots from the given source to the destination. The reason for an increase in number of paths with an increase in the number of *MaxSlots* is similar to that in Figure 3.

Figure 5 shows average number of QoS paths as a function number of nodes for different path lengths. As the number of nodes increases, the number of paths (total paths as well as hop constrained paths) increases. The reason for the increase can be attributed to the increase in the number of neighbors as in Figure 3. One may ask a question: What is the difference between the number of paths represented by Figure 5 and that of Figure 3? The answer is that Figure 5 shows the number of paths satisfying the bandwidth requirement in terms of the number of time slots while Figure 3 shows paths that are available

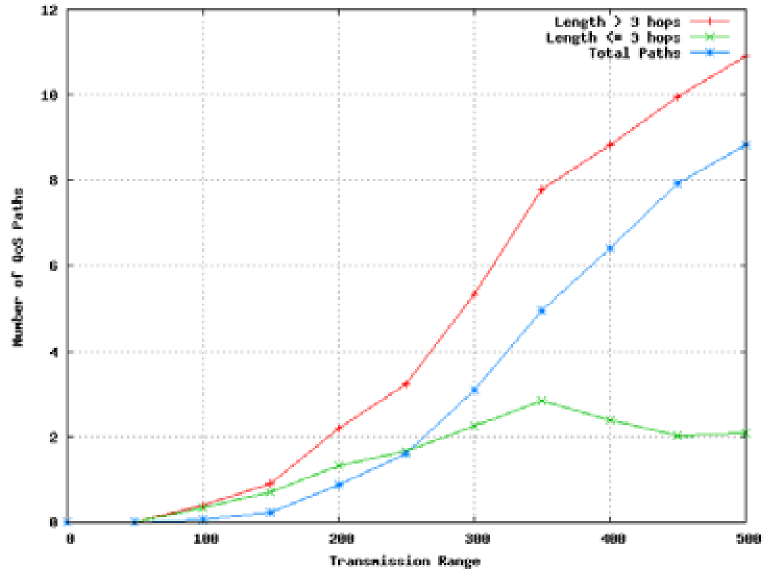

Figure 6. Average number of QoS paths as a function of transmission range for different path lengths.

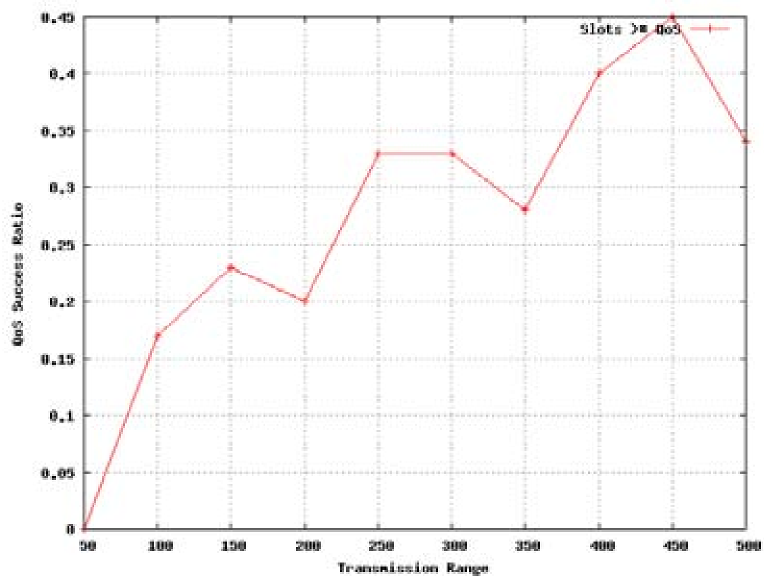

Figure 7. QoS success ratio as a function of transmission range of nodes in the network.

between a given source to a destination irrespective of the fact that they may or may not satisfy the QoS.

Figure 6 shows average number of QoS paths as a function of the transmission range of nodes for different path lengths. We observe that as the transmission range is increased, the average number of QoS paths increases. The reason is that an increase in the transmission range increases the average number of paths that may exist between a pair of nodes in the network. Therefore, it is more likely to find paths that satisfy the QoS.

Figure 7 shows the ratio between the number of cases when enough slots were available to satisfy the bandwidth requirement and the total number of cases examined as a function of the transmission range of participating nodes. It is observed that as the transmission range is increased, the QoS-success ratio is increased. This is due to the fact that an increase in the transmission range increases the number of paths and thus increases the chances of availability of paths with the desired number of time slots to satisfy the QoS requirements of an application.

5. Conclusion

In this paper, we presented a protocol for QoS provisioning in TDMA-based ad hoc network. In our protocol, the source tries to find multiple node-disjoint paths. These multiple node-disjoint paths can be utilized if a single path is not able to provide the required bandwidth, and multiple node-disjoint paths are able to satisfy bandwidth requirements of a flow of packets. The bandwidth requirements are expressed in terms of number of time slots. We evaluated the performance of our protocol by carrying out simulations. In the performance evaluation, we focused on the number of paths that are able to satisfy the QoS requirements in terms of the bandwidth and the QoS success ratio. However, we did not take into account the effect of mobility. Further validation incorporating the effect of mobility forms our future work.

6. Acknowledgements

A preliminary version of this paper appeared in the proceedings of First IEEE International Conference on Networked Digital Technologies (NDT), July 29-31, 2009, Ostrava, Czeck Republic [AlSoufy and Abbas, 2009]. At the time of writing of this paper, Khaled Abdullah Mohd Al Soufy is pursuing for his PhD at Department of Computer Engineering, Aligarh Muslim University, India. Ash Mohammad Abbas is a postdoctoral research fellow under European Research Consortium on Informatics and Mathematics (ERCIM) “Alain Bensoussan” Fellowship Programme. He is on leave for academic pursuit from Department of Computer Engineering, Aligarh Muslim University, India.

References

[1] 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, Bangalore, January 8-12.

[2] AlSoufy, K. A. M., Abbas, A. M. (2009). A Quality of Service Aware Routing for TDMA-Based Ad hoc Networks. *In: Proceedings of First IEEE International Conference on Networked Digital Technologies (NDT)*, p. 347-351, Ostrava, July.

[3] Chen, S., Nahrstedt, L. (1999). Distributed Quality of Service Routing in Ad Hoc Networks. In: *Proceedings of IEEE Journal on Selected Areas in Communications*, 17 (8) 1488-1505, August.

[4] Jawhar, I., Wu, J. (2005). QoS Support in TDMA-Based Mobile Ad Hoc Networks. *Journal of Computer Science and Technology*, 20 (6) 797-810, November.

[5] Li, Y., Chan, X. M., Yu, D. (2006). Multi-Path QoS Routing with Power Control in TDMA-Based Mobile Ad Hoc Networks. *In: Proceedings of First International Conference on Innovative Computing, Information and Control (ICICIC)*, p. 410-414, Beijing, September.

[6] Liao, W. H., Tseng, Y. C., Shih, K. P. (2002). A TDMA-Based Bandwidth Reservation Protocol for QoS Routing in a Mobile Ad Hoc Network. *IEEE International Conference on Communications (ICC)*, Anchorage, Alaska p. 3186-3190, May.

[7] Lin, C. R., Liu, J. S. (1999). QoS Routing in Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17 (8) 1426-1438, August.

[8] Pyo, D., Lee, S., Lee, M. G. (2007). A QoS Routing Protocol for Mobile Ad Hoc Networks Based on a Reservation Pool. *In: Proceedings of International Conference on Software Technologies for Embedded and Ubiquitous Systems (SUES)*. LNCS 4761, p. 253-262, Santorini Island, Greece, September.

[9] Shih, K. P., Chang, C. Y., Chen, Y. D., Chuang, T. H. (2006). Dynamic Bandwidth Allocation for QoS Routing on TDMA Based Mobile Ad Hoc Networks, *Elsevier Journal on Computer Communications*, 29 (9) 1316-1329, May.

[10] Sriram, S., Reddy, T. B., Manoj, B. S., Murthy, C. S. R. (2004). The Influence of QoS Routing on the Achievable Capacity in TDMA-Based Ad Hoc Wireless Networks. *In: Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, p. 2909-2913, Dallas, Texas, December.

[11] Wu, H., Huang, C., Jia, X., He, X. (2004). Multi-path QoS Routing in TDMA/CDMA Ad Hoc Wireless Networks. *In: Proceedings of International Workshop on Agents and Autonomic Computing and Grid Enabled Virtual Organizations (AAC-GEVO)*, Springer Verlag Lecture Notes in Computer Science (LNCS), 3252, p. 609-616, Wuhan, September.

[12] Zhu, C., Corson, M. S. (2002). QoS Routing for Mobile Ad Hoc Networks. *In: Proceedings of 21st Annual Joint Conference of IEEE Computer and Communication Societies (INFOCOM)*, p. 958-967, New York, June.

Author Biographies

Khaled Abduulah Mohammad AlSoufy received his Bachelor of Technology in Computer and Control Engineering from Sana'a University, Yemen, in 1998 and his Master of Technology in Computer Science and Engineering from Osmania University, Hyderabad, India in 2006. Currently, he is pursuing for PhD at Department of Computer Engineering, Aligarh Muslim University, India. His current research interests include ad hoc networks, quality of service, and routing.

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. Currently, he is on leave for academic pursuit from AMU Aligarh. 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.