

Admission Control and Simple Class Based QoS Provisioning for Mobile Ad hoc Network

Mohammad Aminul Haq
and Mitsuji Matsumoto
GITS, Waseda University
Tokyo 169-0051, Japan

Email: mdaminul@asagi.waseda.jp
mmatsumoto@waseda.jp

Jacir Luiz Bordim
Masakatsu Kosuga
and Shinsuke Tanaka
ATR-Adaptive Communications Research
Kyoto 619-0288, Japan
Email: {bordim,kosuga,shinsuke}@atr.jp

Abstract—In this paper we present a network layer based admission control and simple class based service differentiation model to support QoS in mobile ad hoc network. Our admission control procedure works along with the route finding phase of reactive routing protocols for mobile ad hoc network (AODV, DSR etc). We also propose a simple class based service differentiation system to support QoS once a traffic is admitted by our admission controller. The proposed service differentiation is based on DiffServ model and includes modifications like configuration of each node with edge and core functionality, dynamic selection of edge/core functionality, use of minimal and simple classes. Simulation results show that our system allows seven times more real time traffic in the network than the proposed QoS for AODV model while satisfying the demanded end-to-end delay and providing low jitter.

I. INTRODUCTION

In contrast with a cellular network that has the support of a robust infrastructure, a Mobile Ad hoc NETWORK (MANET) is a distributed system of mobile hosts which can communicate among themselves via multi-hop in an infrastructure-less environment. Such networks find its applications in special environments to support temporary and urgent tasks. Since the deployment of an ad hoc network does not rely on the existence of any infrastructure, they are well suited for communication in rural areas, disaster-and-relief operations, search-and-rescue, law enforcement, among other special applications. The need to support real time and multimedia communication (voice, video) over MANET has increased with the advancement of mobile devices, mobile applications and deployment of ad hoc networks in emergency rescue and military battlefields. These types of communication require specific requirements to be satisfied like minimum bandwidth, maximum end-to-end delay, tolerable jitter etc.

Providing the required quality to these applications over MANET is a challenging task. An ad hoc network is a very complex distributed network where nodes are free to move and hence topology of the network is dynamic. Movement of the nodes does not only change the topology of the network but also causes traffic load condition to change dynamically. Moreover, a node in MANET is not only a host in the network but also a router and has to perform processing on behalf of other nodes. Besides, wireless characteristics of links

pose additional challenges. We need to consider dynamic link capacity, bandwidth, mobile device limitations like battery power and processing power before going for any QoS system for MANET. Again, such QoS provisioning system should not put much load on the nodes and should not increase the volume of information to be exchanged or maintained to support its operation.

Considering all these problems, we present here a very simple QoS system which directly fits into any reactive routing protocol, does not put much processing load on the nodes and reduce the state information to be maintained.

II. MOTIVATION

Proposals concerning providing QoS in ad hoc networks so far are broadly based on either per flow QoS provisioning or a variation of DiffServ Model [2] which is built on service differentiation. Per flow based QoS provisioning is inspired by the IntServ Model [3] which differentiates each flow in the network and performs reservation for each of them. It suffers from the flaw that it increases state information and put much load on the mobile nodes. Moreover it does not suit with the dynamic nature of mobile ad hoc network. A reservation based services in ad hoc networks is presented in [12]. DiffServ variant, on the other hand, seems more promising because it neither maintains any state information nor relies on any reservation procedure and it is lightweight. Even though DiffServ is used in fixed Internet, to use it in MANET we need to make some modification to cope with several challenges like mobility of the nodes, dynamic topology and parameter (like bandwidth, delay) of the network. Most of the previous works [12] [1] related to service differentiation or class based QoS provisioning either aimed at a layer between network and MAC [1] or relied on feedback from MAC layer or both. In this paper we describe a very simple class based QoS provisioning system and admission control procedure that works in the Network Layer. Moreover our approach does not depend on any special type of routing protocol. It can be adapted in any reactive routing protocols. A preliminary version of this work appeared in [10]. Before going into details of our model, let us take a look at major previous works.

In [4] a QoS provisioning extension to the Ad hoc On Demand Distance Vector Routing (AODV) [16] is proposed. The source node initiates a route request with required QoS values (maximum end-to-end delay, minimum bandwidth). Each intermediate node subtracts its `NODE_TRAVERSAL_TIME` (NTT) from the required end-to-end delay. If any node finds that the remaining end-to-end delay is less than the NTT then it drops the route request. Here, a constant NTT is used to check the actual delay. However in dynamic network like MANET, this value may be different from node to node and also from time to time.

SWAN [1] is a QoS provisioning system that treats UDP traffic as real time and TCP traffic as best effort. It uses admission control based only on bandwidth measured along the path of communication by sending a probe message. If the bandwidth is greater than the required bandwidth plus a threshold then the traffic is allowed otherwise it is marked as best effort. The admission controller promiscuously listens to all packet transmissions within its range to gather information about the status of outgoing link in terms of bandwidth and congestion. The best effort traffic passes through a rate controller that shapes the traffic according to the rate based on the feedback from the MAC layer.

QoS for mobile ad hoc networks has become an active research topic where researchers are participating in a growing number and numerous works have been done and are being done. A complete review of these works is out of limit of this paper. Here we briefly discussed about the previous works that are related to our work. Comprehensive references on QoS routing in ad hoc networks can be found in [6]. The problems of QoS routing in MANET are discussed in [7]. For general understanding of QoS routing in mobile ad hoc networks, we refer the reader to [14], [20]. MAC protocols that support QoS are presented in [15] and [9]. For more on QoS aware services please see [13]. Some other notable works in QoS provisioning in mobile ad hoc networks include [8], [19], [5], [18]. Performance issues of QoS routing in mobile ad hoc network is addressed in [11].

The rest of the document is organized as follows. Section III and Section IV illustrates the two major part of our system: proposed admission control and simple class based QoS provisioning, Section V depicts the simulation results and analysis and finally section VI concludes the paper.

III. ADMISSION CONTROL

Since we can not over provision the network and also can not accommodate flows beyond the network capacity, it is better to control the number of flows so that we can at least provide quality to some of the flows. The decision of whether to accept or reject a flow is done by admission control procedure based on resource availability basis. In reactive routing protocols (like AODV, DSR), when a node wants to communicate to another node and does not know a route to the destination, it sends out a route request (RREQ) message. We extend this route request message to include not only the destination to which a route is wanted but also the quality of

such route in terms of QoS parameters like maximum end-to-end delay, minimum bandwidth etc. We denote the traditional route request message by $RREQ_{Reactive}$, and the extended RREQ by $RREQ_{Ext}$. So we can write

$$RREQ_{Ext} = RREQ_{Reactive} \cup \{D_{max}, B_{min}, B_{max}\}$$

where D_{max} is the required maximum end-to-end delay, B_{min} is the minimum bandwidth that the application will ever use, B_{max} is the maximum limit of bandwidth the application will use or require. So the route to the destination must have available bandwidth greater than or equal to B_{max} and if the flow's bandwidth falls below B_{min} then we decide that it is due to degradation of network condition rather than application reducing the bandwidth usage.

In our model the source network layer uses this information to check whether there is any route to satisfy this requirement. The source node's network layer gets a request in the format of $RREQ_{Ext}$ from its application. It removes D_{max} from $RREQ_{Ext}$, stores D_{max} locally, starts a timer with value $2 * D_{max}$, and sends out the route request (RREQ) with the remaining parameters (only the bandwidth constraint). We refer to this RREQ by $RREQ_{QoS}$. So $RREQ_{QoS}$ has the format

$$RREQ_{QoS} = RREQ_{Reactive} \cup \{B_{min}, B_{max}\}.$$

When an intermediate node gets this $RREQ_{QoS}$, it uses B_{max} to determine whether to forward the route request or drop it. As will be explained in Section IV, each node allocates its available bandwidth to the two classes of our model namely QoS class and Best Effort class. (Please see Section IV for more on this). An intermediate node compares the requested maximum bandwidth with the available bandwidth of the outgoing link that is allocated for QoS traffic. If the available bandwidth is higher, the node reduces the available bandwidth for QoS traffic by the amount of requested maximum bandwidth, stores the required minimum and maximum bandwidth and rebroadcasts the route request to its neighbors. If the available bandwidth in the QoS portion of allocated bandwidth is less than B_{max} , the node drops the route request. For example node C in Figure 1 dropped the $RREQ_{QoS}$ assuming that it does not have the requested bandwidth available.

By this way if the route request reaches the destination by satisfying the bandwidth constraint then the destination records the required bandwidth contained in the Route Request message (like other intermediate nodes) and replies with a route reply message. Now the route reply message travels back to the source node as mentioned in the routing protocol. So a round trip time T_{round} is required to find a route to the destination. Thus if $T_{round} \leq 2 * D_{max}$, the route reply reaches the source before the timer of the source node expires. Otherwise the timer expires first meaning that required end-to-end delay is not guaranteed. This way we check for both minimum bandwidth and maximum delay to admit traffic. The destination node, once got the route request, records the required end-to-end delay and bandwidth.

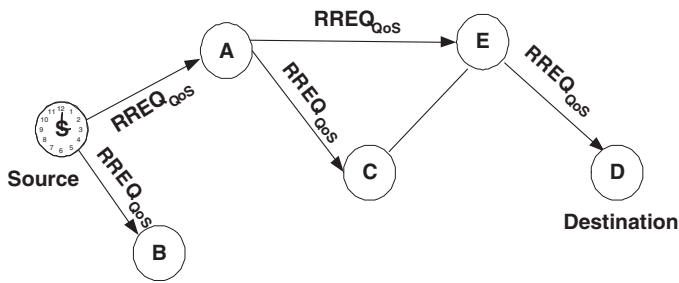


Fig. 1. Admission Control in RREQ Phase

QoS for AODV [4] also checks bandwidth in every node during the route request phase. However our admission control procedure is distinct. Specifically, in [4], the source node broadcast the required end-to-end delay with the route request (which we do not) and each node subtracts its `NODE.TRAVERSAL.TIME` (NTT) and if the result is positive it forwards the route request otherwise drops the request. The drawback of such calculation is that, NTT is a constant in [4] which is 10ms by default. With the dynamic nature of mobile ad hoc networks, the actual NTT may vary among the nodes as such it can not be relied upon. In our system, the source node measures the real end-to-end delay and uses this value to admit traffic in the network. Moreover once the traffic is admitted the measured end-to-end delay is verified in the source side and if the admitted traffic is not getting the initial required end-to-end delay then the route is dropped and new route discovery is initiated by the source. The verification process will be explained in Section III-A. This feature is also different from that of [4] where each intermediate node has to maintain the delay to the destination and has to check this value every time it forwards a packet. This approach does not only increase state information to be maintained but also make the intermediate nodes' operation very complex. In our system we keep the intermediate network simple which is consistent with our use of DiffServ once the traffic is admitted. Let us now take a look at our verification process.

A. Verification

Our verification process verifies that the initial measurement during traffic admission is still valid. Our verification process is also distributed over the route. Each intermediate node keeps track of the bandwidth at which it is receiving data from the previous hop. If this bandwidth is smaller than B_{min} then this node sends a Route Error message to the source to inform that the route is not valid for QoS traffic. B_{max} is used to rate control at the edge router so that no node can exceed its initial request as will be explained in Section IV. To verify the end-to-end delay of current traffic, the source sends a special type of *Route Reply* to the destination which needs to be acknowledged by the destination. The source sends this Route Request and waits for the acknowledgement to come. If the time interval is still within $2 * D_{max}$ then the traffic is continued otherwise dropped. In reactive routing protocols, there are several ways to generate a route request that must

be acknowledged by destination. For example in AODV, the sender can set the *A* field of the Route Reply packet which means that the destination of the route reply message must acknowledge the reception. The source node performs this verification every TTL/c seconds where *TTL* is the validity period of the route and *c* is a constant such that $1 \leq c \leq TTL$. The source node can configure the value of *c*. In our model we used $c = 2$.

We adapt modified class based service differentiation once the traffic is admitted for QoS provisioning during data transfer. Section IV will explain the system in detail.

B. Route reply by an intermediate node

In reactive routing protocols, during a route discovery phase, if any intermediate node has a valid route to the destination it can reply to a route request rather than the destination replying everytime. In our model we change the scenario a little bit. We have two classes of traffic in our model namely QoS class and Best Effort (BE) Class. QoS class has some specific QoS requirements to be fulfilled whereas BE class does not have any QoS requirement i.e., traditional traffic. See Section IV for more on this. For the best effort traffic there is no change to this feature. Any intermediate node can reply to a route request if it has a valid up-to-date route to the destination. But for the QoS traffic, the route reply must come from the destination. Since enabling the intermediate node to reply to a QoS route request requires a good number of state information (delay to the destination, minimum bandwidth available up to the destination etc) to be gathered and maintained in the intermediate nodes of the network, it is thus preferable to take the reply from the destination to know the network situation that exist at that moment. On demand routing protocols allow some mechanism by which we can force the route reply to come from the destination. For example in AODV we can set the *D* field of the route request packet which if set means that the route reply must come from the destination. So the source node of the route request can set this field to force the route reply to come from the destination or each intermediate node can distinguish between route request for a QoS traffic and route request for a best effort traffic by looking for the required bandwidth field in the route request message whose presence imply that route request is for QoS traffic and thus intermediate node should refrain from replying to that. Either of the above two methods can be used for our purpose.

IV. SIMPLE CLASS BASED QOS PROVISIONING

Here we introduce a simple class based service differentiation to provide QoS provisioning during data transmission. This method is inspired by the DiffServ model because this also keeps the intermediate nodes' activity very simple and does not require maintaining any state information. However we have incorporated several modification to DiffServ's service differentiation to make it suitable for mobile ad hoc network as mentioned in the following section.

Unlike traditional DiffServ, we define only two classes: QoS class and Best Effort (BE) class. QoS traffic needs quality of

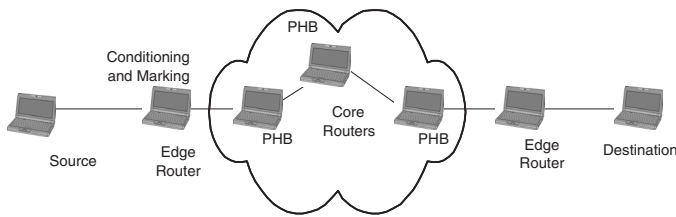


Fig. 2. Differential Services Model for Mobile Ad hoc Network

service guarantee and BE is the traditional best effort traffic. QoS class gets better forwarding assurance from all the nodes which forward packets according to a per hop behavior. In our model, control packets are classified as QoS packets and receive the same type of processing (queueing and scheduling) as QoS packets. This special feature in our model enables us to measure end to end delay using control packets during admission control procedure as mentioned in Section III. The queueing and scheduling used in our model will be explained shortly. We include only two classes of traffic to keep the system simple and not to put much load on the mobile nodes by creating more classes that need more queueing facilities and processing to be implemented.

The architecture of DiffServ distinguishes two parts: the core network and the access network. The core network is composed of one or several ISPs. Packet forwarding is done by core routers whereas the access networks connect end hosts to an edge router (Figure 2). Performance agreements between administrative domains (SLA-Service Level Agreements) allow to statically reserve sufficient resources to support statistical performance guarantees according to different BA (Behavior Aggregates)-QoS classes that group flows of similar properties. Performance perceived by each class depends on the type of processing at core routers specified in a PHB (Per Hop Behavior). Edge routers perform classification of the incoming traffic and marking according to application types, source and destination addresses or ports etc. Incoming traffic is checked against a TCC (Traffic Conditioning Configuration), a profile of the traffic defined in the SLA. Traffic exceeding a given TCC can be dropped, marked as out of profile or marked with a lower priority class.

Considering a dynamic topology network like MANET, we change the scenario again. Each node configures its own PHB and TCC and each node has to have the capability to perform as an edge and core router. As mentioned in the previous paragraph, in DiffServ architecture, the nodes are either core node or edge node. The edge nodes perform traffic conditioning, marking of packets according to the Type Of Service (TOS) field of the IP packet while the core routers just look for the TOS field and forward the packet according to their per hop behavior which is assumed to favor QoS traffic. Each node reserves its major proportion of available bandwidth for QoS traffic and minor proportion for BE traffic. This allocation is configurable at each node. During admission control of QoS traffic, each node makes sure that it has the requested bandwidth available in the QoS portion of the

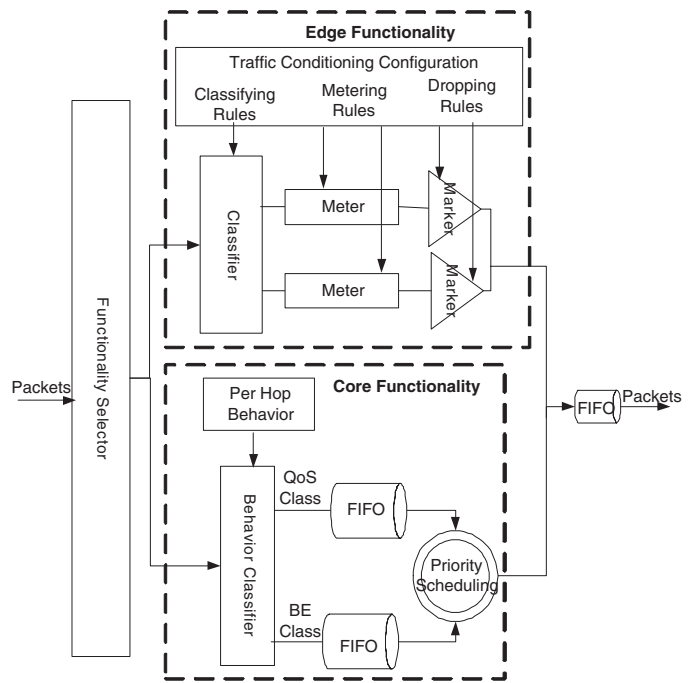


Fig. 3. Configuration of one node showing details of edge and core functionality

bandwidth.

A. Dynamic Edge-Core Functionality

In a static network the classification of edge router and core routers is easy and simple. However in a dynamic network this classification does not hold. Assigning core and edge router functionality in a dynamic mobile network where all nodes act as a terminal as well as router is another issue that needs to be solved before using any DiffServ based model. In our model, we consider edge router and core router as two *functionality* rather than two separate kind of routers. Each node has these two functionalities (Figure 2). Whenever a packet reaches a node, the packet either pass through the edge router functionality or the core router functionality depending on the following criteria. A *Functionality Selector* (FS) in each node looks at the packet header, carries out the following procedure and forwards the packet to either edge router functionality or the core router functionality. If we take a look at Figure 3, we find that the first hop from the source should be the edge router. So FS receives a packet from the source node then it forwards the packet to Edge Router functionality. Again, the previous hop to destination should also perform as an edge router. In this case, the checking criterion is that the destination is the neighbor of current node. If FS finds that the received packet's destination is this node's neighbor or destination is the next hop, then FS forwards the packet to Edge Router Functionality. Except for the above mentioned two cases, FS forwards the packet to core router functionality. The procedure is depicted in Figure 4. Here (S1, D1) and (S2, D2) are the communicating pairs. The flow of packets from S1 to D1 is shown in solid line while that of S2 and D2

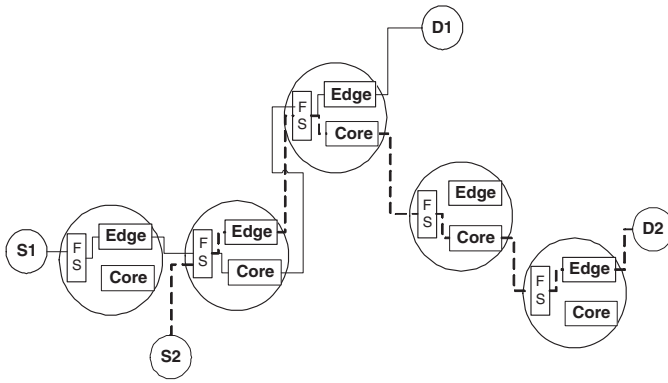


Fig. 4. Selection of Edge and Core router functionality

is shown in dotted line. All the intermediate nodes' internal structure (edge, core functionality and functionality selector) is shown except for the communicating nodes S1, D1 and S2, D2. The solid and dotted lines depict the functionality selected by FS at each node. As can be understood from the figure, for a certain traffic a node may act as a core router but for another traffic it may be an edge router as such the resulting processing done on the traffic is same as that of traditional DiffServ model in a static network. So our model provides the static DiffServ architecture in a dynamic network. In what follows, we describe the operation performed by edge and core router.

B. Edge Functionality

If the packet is forwarded to edge router functionality from FS then the packet is first classified either as QoS class or BE class in the edge router's *Classifier* based on the rules specified in Traffic Conditioning Configuration (TCC). The classifying rules are expressed in term of source and destination address, source and destination port etc. A DSCP (Differential Services Code Point) is inserted into the packets TOS (Type Of Service) field. TCC specifies rules for classification and metering. TCC allocates bandwidth to both QoS and BE traffic such that QoS traffic gets better service than BE traffic. As aforementioned, the proportion of bandwidth allocated to the QoS and Best Effort class is based on majority-minority proportion. Both QoS and BE traffic rate is controlled to match this bandwidth allocation. Although our TCC drops all the packets that are out of this bandwidth allocation, a more adaptive TCC is desirable where the allocation is flexible, i.e., one class of traffic is benefited from the absence of other class of traffic. We are currently extending our model to include this adaptive TCC. Our current TCC allocates 60% of the bandwidth to QoS class and 40% to BE class. Marker drops the out-of-profile packets (that are out of this allocation). The dropping (or marking) rules as specified in TCC. In our model out-of-profile packets of both BE and QoS classes are dropped.

C. Core Functionality

If a packet is forwarded to core router functionality from FS then the packets TOS field is checked and the packet is put

into corresponding queues of BE and QoS traffic. A priority scheduler takes the packets out the queues. The QoS traffic has a static priority higher than best effort so that QoS traffic gets better forwarding than BE traffic. The output traffic from both core router and edge router functionality goes into a FIFO queue (Figure 3) which ultimately serves the outgoing link.

V. SIMULATION AND ANALYSIS

We have performed several simulations to evaluate the performance of our model and compare it with QoS for AODV [4] and traditional system with different routing protocols. In all cases we measure and compare the number of real time traffic getting required quality, average end-to-end delay, and average jitter. The results are provided in Section V-C.

A. Simulation Environment

The simulations have been performed using the QualNet simulation package version 3.6 [17]. The mobility model used in each of the simulations is the random direction model. The pause time used during the simulations is 30 seconds. Nodes' mobility speed is 10 meter/sec. The MAC layer protocol used in the simulations is the IEEE standard 802.11 Distributed Coordination Function (DCF). In the simulations the data rate is 2 Mb/sec and the data packet size is 64 bytes.

AODV and DSR are used as routing protocols in different scenarios. Broadcast of Hello packets is enabled and number of retries in case of no router reply against a router request is 2. Control packets and QoS packets go to the same queue whereas BE packets go to a different queue. This is to ensure the correctness of measurement of end-to-end delay using control packets to reflect the end-to-end delay faced by QoS packets. As mentioned in previous sections, route reply of intermediate nodes in AODV and DSR is disabled for QoS traffic to have the actual end-to-end delay measurement during call admission.

We used Voice over IP (VoIP) as the application which needs QoS support i.e., VoIP traffic is classified as QoS class traffic in edge routers. In QualNet, VoIP is used to simulate IP telephony. The initiator and receiver generate real-time traffic with an exponential distribution function. It simulates a real life telephone conversation. Randomly selected source/ destination pairs starts a VoIP session at a randomly selected time and continues the talk for an *average talk time* randomly selected between 1 and 300 seconds. The QoS requirement of VoIP traffic is that maximum end-to-end delay should be at most 200 millisecond to have an understandable conversation. The bandwidth requirement for VoIP is dependent on many factors like coding technique, sampling rate etc. In our experiment we used 48Kbps as the bandwidth required by VoIP. FTP is used as the application that does not require QoS support i.e., FTP traffic is classified as BE traffic in edge routers. Here also, randomly selected source and destination pairs initiate FTP session with randomly selected parameters. Both the VoIP and FTP traffic are active simultaneously. The network consists of 50 nodes randomly distributed over a 1000m×1000m area. Same initial position of nodes was used in the same set of

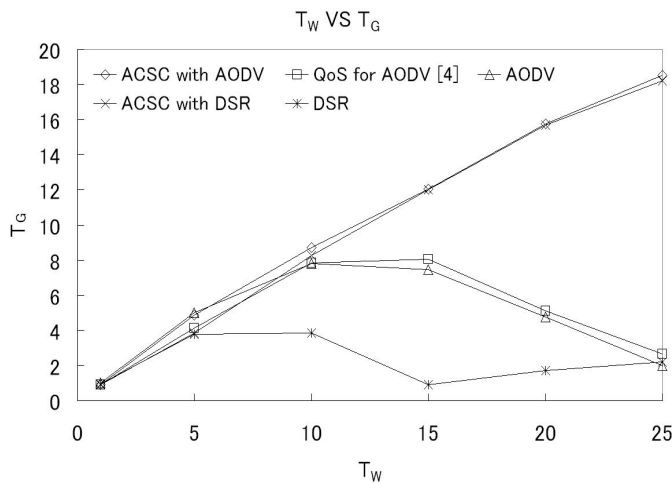


Fig. 5. Number of QoS flows getting required QoS in different models

simulation so that different node placement does not affect the simulation result. Each simulation models 5 minutes of real time. Each simulation was run for 20 seeds making an overall 600 runs and the results were averaged.

B. Simulation Parameters

We will use the following symbols with the given meaning throughout the presentation and analysis of simulation results.

- T_W =Total Number of QoS Traffic Willing to go into the network.
- T_A =Number of QoS Traffic Admitted into the network by the admission control mechanism.
- T_G =Number of admitted QoS Traffic that actually Got their required QoS.
- D_{E-E} =Average End-to-End Delay of QoS traffic.
- J_R =Average Jitter at Receiver side of QoS traffic.

From the definition of Admission Control, it is obvious that,

$$T_W \geq T_A \geq T_G.$$

C. Comparison of Proposed Model

In this section, we will use ACSC (Admission Control and Simple Class based QoS System) to refer to our model. We implemented ACSC on AODV as well as DSR and refer to them by ACSC with AODV and ACSC with DSR respectively. By traditional system we refer to conventional mobile ad hoc network without any quality of service provisioning. We used two traditional system in the simulation: one that uses AODV as routing protocol and one that uses DSR and we refer to them by AODV and DSR respectively. These four mentioned systems and QoS for AODV[4] are compared on the basis of number of QoS traffic getting their required QoS, average end-to-end delay and average jitter.

Figure 5 shows T_W VS T_G in these models. As can be seen from the figure, for a very low T_W , there is little difference in T_G 's of these systems. But as T_W increases, the difference between T_G 's of these systems also increases.

ACSC with AODV or with DSR has the highest number of traffics satisfied over whole the simulation time with the former performing better than the later. Traditional system with DSR has the lowest number of flows satisfied. The number of flows getting their requested QoS in QoS for AODV has slightly greater value than that of a traditional system with AODV. This arises from the fact that QoS for AODV [4] does not provide any QoS provisioning mechanism once a traffic is admitted. Moreover in admitting a traffic into the network, [4] uses a constant NODE_TRAVERSAL_TIME at each node as a measure of end-to-end delay which may be different from node to node and time to time in reality. In the AODV implementation of Qualnet, the source node waits $2 * TTL * AODV_NODE_TRAVERSAL_TIME$ for the route reply to come which prevents for waiting too long and as such similar to a admission controlling procedure producing better performance than DSR which lacks both delay and bandwidth checks.

At $T_W = 25$, T_G in ACSC is 18.5 whereas T_G in [4] and all other mentioned models is around 2.6. This means that our model is able to accommodate almost 7 times more real time traffic than the all other models. This arises from different admission control mechanism in our model than [4] and also because our model incorporates a simple class based service differentiation system which is absent in all other models. Our admission controller measures the actual end to end delay before admitting a traffic into the network and verifies the initial measurement during data transmission. Admission control based on real measurement along with simple service differentiation enables our model to provide required QoS to more traffic than the others.

Figure 6 shows T_W VS D_{E-E} in ACSC and other mentioned systems. For VoIP conversation to go smoothly, $D_{E-E} \leq 200$ millisecond should be satisfied. As is evident from the figure, $\forall T_W, D_{E-E} \leq 200$ millisecond in our system. However, in QoS for AODV and traditional system with AODV, D_{E-E} goes well over 200 millisecond with a limited number of T_W . D_{E-E} in traditional system with DSR rises even sharply. This is because, before admitting traffic admission control mechanism in ACSC measured the delay and bandwidth that the traffic will have in the present network condition. If this measurement indicated the feasibility of admitting the traffic only then the system permitted the traffic. After that service differentiation and edge-core router functionality also provided better support for real time traffic which is absent in all the other systems. Next we take a look at T_W VS J_R in the systems mentioned. Figure 7 shows the graph. As can be seen the average jitter experienced by real time traffic in our model is very small as compared with that of other systems.

VI. CONCLUSION

In this paper we have presented an admission control and simple class based service differentiation mechanism. In our admission control mechanism, the source node checks the end-to-end delay and all the intermediate nodes perform

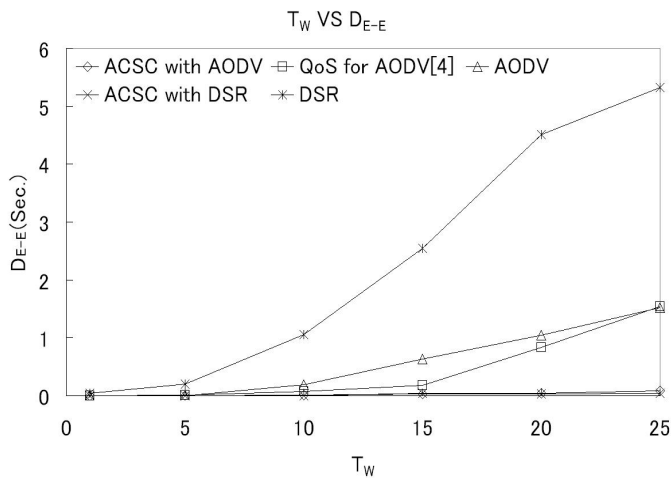


Fig. 6. Average end-to-end delay in different models

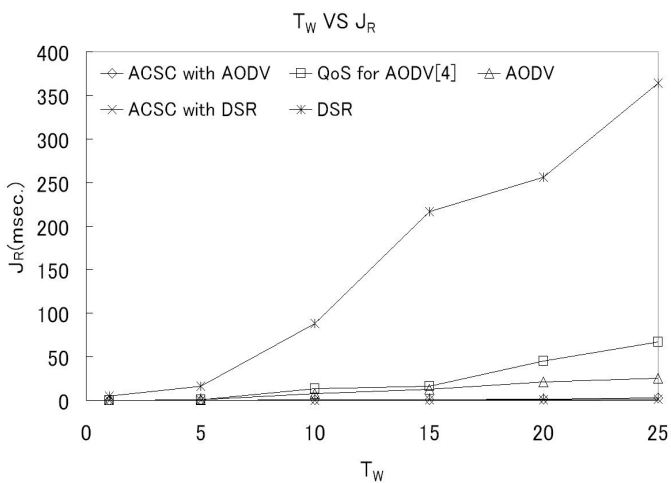


Fig. 7. Average Jitter in different models

bandwidth check. If both of these tests are successful, only then a QoS traffic is permitted in the network. To employ service differentiation through edge-core router functionality we have shown how a service differentiation network can be maintained with the use of Functionality Selector and what the configuration of each node should be in such a case. The evaluation of this system shows great difference in number of traffic getting required QoS between our system and other proposed and traditional system. It is seen from the simulation results that our system can accommodate more real time traffic in the network while satisfying the demanded end-to-end delay and providing low jitter.

ACKNOWLEDGEMENTS

The research work presented here was partly supported by the National Institute of Information and Communications

Technology (NICT) of Japan. The authors would like to thank Dr. Komiyama Bokuji for his effective direction and the technical program committee for their valuable comments.

REFERENCES

- [1] G.-S. Ahn et al. *Supporting Service Differentiation for Real-Time and Best Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN)*, IEEE Transactions on Mobile Computing, (Special Issue of Best Wireless Papers from IEEE INFOCOM 2002), Vol. 1, No. 3, pp. 192-207, July-September 2002.
- [2] S. Blake et al. *An Architecture for Differentiated Services*, Internet RFC 2475, 1998.
- [3] R. Braden et al., *Integrated Services in the Internet Architecture: an Overview*, Internet RFC 1633, 1994.
- [4] C. E. Perkins, E. M. Royer, S. R. Das, *Quality of Service for Ad hoc On-Demand Distance Vector Routing*, Internet Draft, 14 July 2000, draft-ietf-manet-aodvqos-00.txt
- [5] DH. Cansever, AM. Michelson, AH. Levesque, *Quality of service support in mobile ad-hoc IP networks*, Proceedings of MILCOM 99, October 1999.
- [6] S. Chakrabarti, A. Mishra, *Quality of service challenges for wireless mobile ad hoc networks*, Wireless Communications and Mobile Computing Journal, Volume 4, Issue 2, March 2004, pp 129-153.
- [7] S. Chen, N. Klara, *An overview of quality-of-service routing for the next generation high-speed networks: problems and solutions*, IEEE Network, Volume 12, Issue 6, 1998, pp 64-79.
- [8] D. Dharmaraju, A. Roy-Chowdhury, P. Hovareshti, J.S. Baras, *INORA-a unified signaling and routing mechanism for QoS support in mobile ad hoc networks*, Proceedings of International Conference on Parallel Processing, August 18-21, 2002.
- [9] M. Gerla, T. Jack, *Multicluster; mobile, multimedia radio network*, ACM/Baltzer Wireless Networks Journal, Volume 1, Issue 3, 1995, pp 255-265.
- [10] M. A. Haq, J. Bordim, M. Kosuga, S. Tanaka, M. Matsumoto, *Admission Control and service differentiation based qos provisioning for mobile ad hoc network*, Proceedings of Third Annual Mediterranean Ad Hoc Networking Workshop, June 27-30, 2004, Turkey.
- [11] T-C T. Jack, T. Chen, M. Gerla, *QoS routing performance in multihop, multimedia wireless network*, Proceedings of IEEE ICUPC 97, 1997.
- [12] S.B. Lee et al. *INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks*, Journal of Parallel and Distributed Computing (Academic Press), Special issue on Wireless and Mobile Computing and Communications, Vol. 60, No. 4, pp. 374-406, April 2000.
- [13] B. Li, *QoS-aware adaptive services in mobile ad-hoc networks*, Proceedings of the 9th IEEE International Workshop on Quality of Service, June 6-8, Germany.
- [14] CR. Linn, JS. Liu, *Qos routing in ad hoc wireless networks*, IEEE Journal on Selected Areas on Communications, 1998, Volume 17, Issue 8, pp 1426-1438.
- [15] CR. Linn, M. Gerla, *MACA/PR: an asynchronous multimedia multihop wireless network*, Proceedings of the 16th Annual Joint Conference of IEEE Computer and Communications Societies, INFOCOM 1997.
- [16] C. E. Perkins, E. M. Royer, I. D. Chakers, *Ad hoc On-Demand Distance Vector (AODV) routing*. Internet Draft, 19 October 2003.
- [17] *Qualnet Simulator of Scalable Network Technologies*, www.scalable-networks.com
- [18] JL. Sobrinho, AS. Krishnakumar, *Quality-of-service in ad hoc carrier sense multiple access wireless networks*, IEEE Journal Selected Areas in Communications, 1999, Volume 17, Issue 8, pp 1353-1414.
- [19] K.C. Wang, P. Ramanathan, *End-to-end throughput and delay assurances in multihop wireless hotspots*, Proceedings of the 1st ACM international workshop on wireless mobile applications and services on WLAN hotspots, San Diego, 2003.
- [20] K. Wu, J. Harms, *QoS support in mobile ad hoc networks*, Crossing Boundaries, 2001, Volume 1, Issue 1, pp 92-106.