IPSJ Journal Vol. 50 No. 1 2–13 (Jan. 2009)

Regular Paper

A New QoS Routing Scheme Based on Bandwidth Consumption for MANETs

Toshiaki Osada,[†1,†2] Gen Kitagata,[†1,†2]
Debasish Chakraborty,[†1] Takuo Suganuma[†1,†2]
and Norio Shiratori[†1,†2]

Quality of Service (QoS) routing in Mobile Ad hoc NETworks (MANETs) is hard to achieve because the network topology tends to change constantly in a dynamic network. In this paper, we propose an effective QoS routing scheme to satisfy the required service demand and adapt to the dynamic changes in network resources. This novel approach of Application-Level QoS Routing Scheme (ALRS) comprises three significant features: (1) Estimation of consumed bandwidth on each node by using 2-hop neighbors' traffic information, (2) Route construction combined with admission control on each application session with (1), and (3) Route maintenance based on (1) and (2). By using our proposed ALRS scheme the availability of bandwidth is greatly improved for high-load communications, such as multimedia applications. By computer simulation we confirmed that ALRS increases the delivery ratio by up to 50% and decreases delay by less than one-fourth of the existing traditional method.

1. Introduction

Satisfying QoS and the effective management of high traffic communication, such as multimedia applications in Mobile Ad hoc NETworks (MANETs) is a challenging issue due to dynamic topology and time-varying link capacity. Several works have been done in the related fields[1].

In our proposed Application-level QoS Routing Scheme (ALRS) we focus on providing multimedia applications for MANETs satisfying desirable end-to-end QoS in a stable manner, by efficiently coping with the dynamic changes in the network resources.

ALRS includes the following three significant features: (1) Finding the traffic information of 2-hop neighbors by using a modified HELLO message in order to know traffic information within the Carrier Sensing Range. (2) Constructing an on-demand route on each application session combined with admission control based on the acquired information, (3) Deleting unused routes to release the bandwidth and propagating the request for route reconstruction in the case of QoS violation.

In this scheme, 'QoS satisfaction' is defined as satisfying the delivery ratio of end-to-end data and at the same time minimizing the delay for each application session. We consider the total volume of bandwidth consumed by each node as the existing traffic. By introducing the above-mentioned route construction and maintenance scheme, based on the precise bandwidth estimation, it is possible to construct a route that satisfies the QoS requirements and effectively to control the amount of traffic. With the significant improvement of available bandwidth by using this scheme, high-load communications, such as multimedia applications on MANET will be greatly benefited.

From simulation results, we confirmed that the proposed scheme increased delivery ratio by up to 50% and decreased delay by less than a quarter, by effective suppression of traffic, without any deterioration of throughput compared to the traditional AODV[2] method. The results show that the proposed scheme enables the provision of the QoS-aware services in a MANET environment.

The paper is organized as follows. We give a brief account of related work in Section 2. In Section 3 we present our scheme, ALRS. Simulation results and related discussion are presented in Section 4. Finally we draw conclusions in Section 5.

2. Related Work

Existing routing protocols in MANETs are basically classified into two categories; table-driven[3),4)] and on-demand[2),5)].

On-demand routing protocols have a higher latency to start data transmissions, while the overhead can be suppressed. Since QoS provisioning after route-establishment is more significant for relatively longer-term data communication such as streaming than initial cost, i.e., the establishment latency, the on-demand

[†1] Research Institute of Electrical Communication, Tohoku University
[†2] Graduate School of Information Sciences, Tohoku University

approach is feasible for realizing our desired QoS-aware routing. Our method acts as a proactive protocol, as it collects two hop neighbor's information; however, even if the source and destination are the same, the route can be different depending on bandwidth requirement. Therefore, it is more appropriate to use our scheme for on-demand routing protocol, such as AODV.

For the construction of a QoS routing scheme which satisfies the required QoS on every session, reliable admission control is expected.

Many QoS routing protocols in MANET based on CDMA/TDMA MAC layer are proposed [6)–8)]. However, it is difficult to realize such a centralized MAC scheme in a dynamic MANET environment. Therefore, in this work, our aim is to realize a QoS routing scheme based on IEEE 802.11 DCF [9)] with distributed MAC protocol with collision avoidance (CSMA/CA) which is widely used in the wireless LAN environment.

Some of the QoS routing schemes have been investigated based on IEEE 802.11 DCF. Xue, et al. [10)] proposed AQOR, which estimates the consumed bandwidth of self-node based on neighboring nodes' traffic. This scheme incorporates admission control and resource reservation into routing. However, AQOR is not sufficient to construct a highly-reliable QoS-aware route, because it does not consider bandwidth consumption caused by Interference Range (IR), as discussed in the later part of this paper. Sivakumar, et al. [11)] proposed the core-extraction distributed routing algorithm (CEDAR), which can effectively set up the QoS-aware route. However, if the core node moves out of the selected route, the rerouting is very costly. Moreover, it is not enough to verify the effects of the scheme, because a static network environment without nodes' mobilities is used for the simulation.

3. ALRS: An Effective Application-Level QoS Routing Scheme

3.1 Overview of ALRS

We propose an effective Application-level QoS Routing Scheme for MANETs (ALRS) to satisfy the required QoS to the dynamic changes of network resources.

For constructing efficient QoS routing in MANETs an accurate estimation of bandwidth consumption is essential for admission control considering not only TR but also CSR. At first, our view on bandwidth consumption for MANETs is described before explaining the details of our proposed scheme.

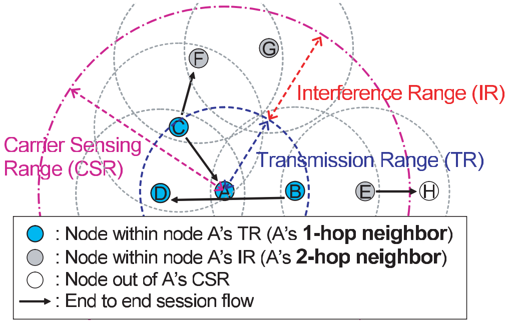

G
F
Interference Range (IR)
C
Carrier Sensing
Range (CSR)
Transmission Range (TR)
D
A
B
E
H
● : Node within node A's TR (A's **1-hop neighbor**)
● : Node within node A's IR (A's **2-hop neighbor**)
○ : Node out of A's CSR
→ : End to end session flow

Fig. 1 Bandwidth consumption on node A.

In this paper, 'self-traffic' is defined as the accumulated traffic, generated by the node itself, as a source and also when acting as a forwarding node.

Here, let $B_{self}(X)$ denote self-traffic per unit time on node 'X'. If we denote a set of nodes within the Carrier Sensing Range (CSR) of node 'X' by N_X, consumed bandwidth on node 'X' $B_{cons}(X)$ is at least

$$B_{cons}(X) = \sum_{i \in N_X} B_{self}(i). \tag{1}$$

For example, in **Fig. 1**, if T_{XY} is defined as traffic volume from node 'X' to node 'Y', $B_{self}(A) = T_{AD}$ and $B_{cons}(A)$ is as

$$B_{cons}(A) = T_{BA} + T_{AD} + T_{CA} + T_{CF} + T_{EH}. \tag{2}$$

Basically, IEEE 802.11 DCF with RTS/CTS handshake mechanism is used to avoid collision free transmission, where each node makes an effort to transmit each set of data without collision. However, each self-traffic cannot be transmitted in time and in an error free manner if $B_{cons}(X)$ exceeds the maximum available bandwidth supported by the wireless device (denoted by B_{max}), i.e., raw data rate.

When the traffic is excessive, it may exceed the available bandwidth. It will cause transmission failure and as a result the QoS, i.e., the delay and delivery ratio will be adversely affected.

Therefore, to satisfy the QoS it is necessary to control the generation of traffic effectively, and respond to the dynamic traffic changes caused by node mobility.

Proposal:

Our proposed scheme, ALRS includes the following three significant features: (1) Finding the traffic information of 2-hop neighbors by using a modified HELLO message in order to know the traffic information within CSR. Based on the acquired information, we estimate the consumed bandwidth for each node by using the neighbors' information tree, (2) Constructing an on-demand route on each application session combined with admission control based on the acquired information, to avoid performance degradation caused by the bottlenecks due to excessive traffic, (3) Deleting unused routes to release the bandwidth and propagating the request for route reconstruction and information of changed volume of traffic immediately and effectively by using two types of messaging, in the event of QoS violation.

The main innovative point of our scheme as described in Section 3.2 is where we use traffic information within CSR in order to realize reliable admission control.

In our approach, to control traffic generation, explained in Section 3.3, a new route will be constructed only if it does not degrade the overall performance of both the newly established and the other existing routes.

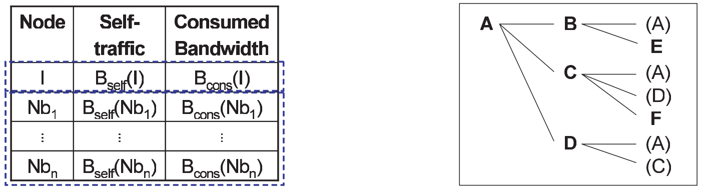
So we expect that our scheme can increase the proportion of the number of sessions with a high delivery ratio and low delay compared to traditional routing methods; however, the number of established sessions perhaps might be restricted.

3.2 Bandwidth Estimation within CSR

For QoS-aware routing, an appropriate estimation of bandwidth consumption is needed to realize both reliable admission control in route construction and QoS violation detection. So, firstly, we propose a method to dynamically acquire and update both the existence of nodes and traffic information within CSR by using HELLO messages and Neighbors' Information Tree (NIT).

In general, the Transmission Range (TR) is set to, at most, half of CSR. So, we extend the HELLO structure as shown in **Fig. 2** (a) in order to know the information within CSR. It includes two fields. The first field is the information about self-node and uses only the first line. The second field is the information about 1-hop neighbors and uses from the second line to the last.

When node 'X' receives the HELLO message from its 1-hop neighbor 'Y', from the acquired information node 'X' can recognize its 2-hop neighbors, directly linked with 'Y'.



Node	Self-traffic	Consumed Bandwidth
I	$B_{self}(I)$	$B_{cons}(I)$
Nb_1	$B_{self}(Nb_1)$	$B_{cons}(Nb_1)$
⋮	⋮	⋮
Nb_n	$B_{self}(Nb_n)$	$B_{cons}(Nb_n)$

(a) (b)

Fig. 2 (a) HELLO structure and (b) NIT.

Additionally, in order to manage the traffic information about all neighboring nodes, each node has NIT as shown in Fig. 2 (b). In the tree, the root indicates the self-node and the vertex at a depth of 'd' indicates the *d*-hop neighbor of the root. And each vertex 'X' has $B_{self}(X)$ and $B_{cons}(X)$. So $B_{cons}(root)$ is the total self-traffic on every vertex from the root to a depth of two (if not overlapping). B_{cons} in each line of HELLO is used to update the value of B_{cons} in the vertex that has the same node ID with the node ID specified in the line. It is similar to the B_{self}. Suppose that the elements of the vertex 'x' in NIT are expressed as follows: $\langle v_x.nodeID, v_x.self, v_x.cons\rangle$. For example, when the node received HELLO, with $\langle$'a', 'b', 'c'$\rangle$ of the line in the first field, and $\langle$'i', 'j', 'k'$\rangle$ in the second field, B_{cons} and B_{self} are updated as: $v_x.self = b, v_x.cons = c$ for the vertex 'x' with $v_x.nodeID = a$. Moreover, for the vertex 'y' with $v_y.nodeID = i$ in the child of vertex 'x', they are updated as: $v_y.self = j, v_y.cons = k$. This is the basic procedure for NIT construction. The algorithm to construct NIT is shown in **Fig. 3**.

Here, each node has a routing table which is the list of route entries per session. Each route entry includes allocation bandwidth as the session's information, where allocation bandwidth is the volume of generated traffic by the session flow, which is equal to the requested bandwidth for the session. So, for each node 'X', $B_{self}(X)$ is equal to the total volume of allocated bandwidth in every entry in X's routing table.

The mechanism of acquisition of information of 2-hop neighbors' and updating bandwidth consumption is illustrated in **Fig. 4**, where the network situation

```
Procedure UpdateNIT($he\_nodeID$[], $he\_self$[], $he\_cons$[], $length$) {
/* $edge(x,y)$: combination of node $x$ and node $y$ ($edge(x,y) \neq edge(y,x)$) */
/* $\boldsymbol{C}$: set of combination $edge(x,y)$ */
/* $selftraff(x,y)$: value of self-traffic on $x$'s children $y$ */
/* $consbw(x,y)$: value of consumed bandwidth on $x$'s children $y$ */

/* The following three parameters show 1st element, 2nd one, and 3rd one
   shown in $i$th line (in HELLO) */
/* $he\_nodeID[i]$: (neighbor) node (ID) shown in $i$th line (in HELLO),
   i.e. one of neighbor node (ID) of the node which sent the HELLO */
/* $he\_self[i]$: value of self-traffic on $he\_nodeID[i]$ */
/* $he\_cons[i]$: value of consumed bandwidth on $he\_nodeID[i]$ */
/* $length$: the number of lines in the HELLO */

   $root$:= root on NIT, i.e. which received the HELLO;
   $x$:= the node (ID) which sent the HELLO;
   if ($edge(root,x) \notin \boldsymbol{C}$) $\boldsymbol{C} \leftarrow \boldsymbol{C} + \{edge(root,x)\}$;
   extend $edge(root,x)$'s lifetime on $\boldsymbol{C}$;
   $i$: = 1; $\boldsymbol{TEMP} = \phi$;
   while($i < length$){
      $y$:= $he\_nodeID[i]$;
      if ($edge(x,y) \notin \boldsymbol{C}$){
         $\boldsymbol{C} \leftarrow \boldsymbol{C} + \{edge(x,y)\}$;
         $\boldsymbol{TEMP} \leftarrow \boldsymbol{TEMP} + \{y\}$;
      }
      extend $edge(x,y)$'s lifetime on $\boldsymbol{C}$;
      $selftraff(x,y)$:= $he\_self[i]$; $consbw(x,y)$:= $he\_cons[i]$;
      $i++$;
   }
   while($\boldsymbol{TEMP}! = \phi$){
      $y$:= extracted one from $\boldsymbol{TEMP}$;
      if ($edge(x,y)$ has expired) remove $edge(x,y)$ from $\boldsymbol{C}$;
      remove $y$ from $\boldsymbol{TEMP}$;
   }
}
```

Fig. 3 Algorithm pseudo-code for NIT construction and update.

follows Fig. 1 and $t_n < t_{n+1}$. In Fig. 4 (a), at first, node B is still unaware of the existence of node E. In this state, even if node A receives the HELLO sent by node B, node A can neither find node E nor can it add E into its own NIT. Therefore, $B_{cons}(A)$ and $B_{cons}(B)$ are calculated lower than the real value on node A. As shown in Fig. 4 (b), when B receives the HELLO sent from E at t_1, B updates existing nodes, i.e., adds node E and node H, and updates traffic information. After that, as shown in Fig. 4 (c), B can create a renewed HELLO by using this information. By receiving the HELLO sent from B at t_2, A finds E as one of A's 2-hop neighbors and adds it to A's tree. And then A correctly updates $B_{cons}(A)$ and $B_{cons}(B)$ to $T_{BA} + T_{AD} + T_{CA} + T_{CF} + T_{EH}$.

Thus, each node can see other existing nodes and existing traffic within CSR and then it can appropriately estimate the minimum consumed bandwidth.

3.3 Route Discovery Combined with Admission Control

In this section, we introduce a route construction method per session combined with admission control.

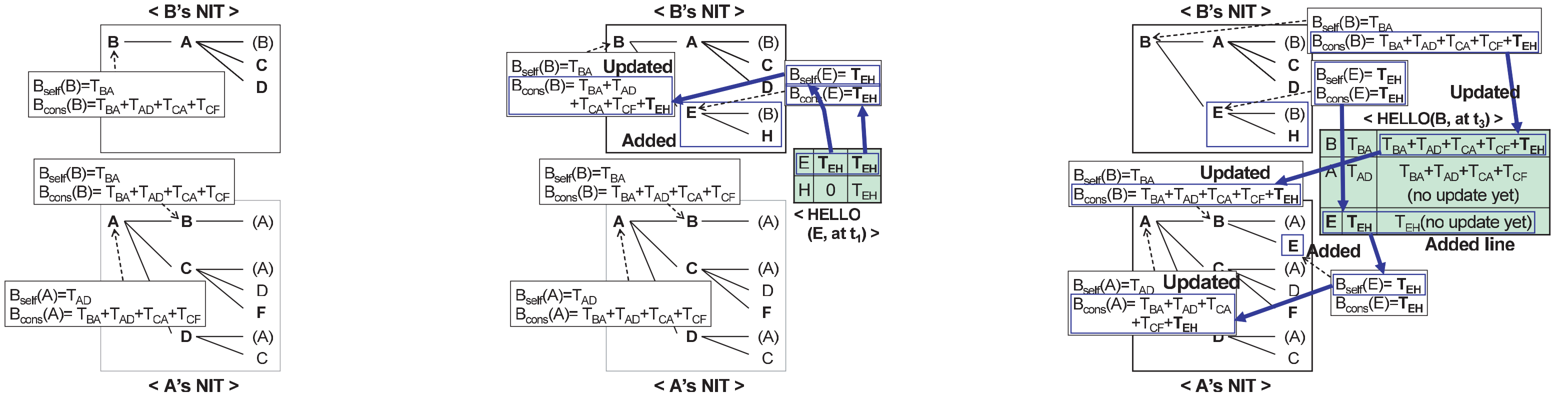
To realize the QoS-aware route discovery, we modify commonly used RREQ/RREP messaging. Route discovery for one session works as shown in **Fig. 5**. For route selection strategy, the earliest found routes will be selected from the possible available multiple routes that fulfill the bandwidth requirement. This implicitly implies that the route with the shortest delay will be selected. Details have been discussed in our earlier work [12].

RREQ with the addition of the requested bandwidth and "route list" is broadcast from the source to the destination while adding each intermediate node's ID to the "route list".

When the intermediate node receives RREP, it determines whether to accept acting as a forwarding node by comparing its own current traffic information with that of RREP. This check means admission control and is realized by the "Final Check" procedure as shown in **Fig. 6**. If the node accepts, it becomes one of the candidate nodes and forwards the RREP.

Similarly, when the source receives the RREP and accepts, it begins to transmit data packets. At the same time, "Immediate" HELLO (Imm-HELLO) is broadcast so that neighboring nodes can immediately update the traffic information. When the candidate node receives a data packet for the first time, it becomes an actual data forwarding node, and starts forwarding data packets.

Here, the process of "Final Check" is shown in Fig. 6. The node checks the increased bandwidth consumption not only for its self but also for every other node existing within the node's cover area. Therefore, it is possible to avoid performance degradation of existing routes, even if the traffic increases due to the newly established routes.


< B's NIT >
< A's NIT >
B
A
C
D
E
F
H
(A)
(B)
B_self(B)=T_BA
B_cons(B)=T_BA+T_AD+T_CA+T_CF
B_self(A)=T_AD
B_cons(A)=T_BA+T_AD+T_CA+T_CF
Updated
Added
B_cons(B)=T_BA+T_AD+T_CA+T_CF+T_EH
B_self(E)=T_EH
B_cons(E)=T_EH
E T_EH T_EH
H 0 T_EH
< HELLO (E, at t1) >
< HELLO(B, at t3) >
B T_BA T_BA+T_AD+T_CA+T_CF+T_EH
A T_AD T_BA+T_AD+T_CA+T_CF (no update yet)
E T_EH T_EH(no update yet)
Added line
B_cons(A)=T_BA+T_AD+T_CA+T_CF+T_EH

(a) Node B has never received HELLO from node E before.

(b) At t_2, node B updates NIT by receiving $HELLO(E, t_1)$.

(c) At t_4, node A updates NIT by receiving updated $HELLO(B, t_3)$.

Fig. 4 Acquisition mechanism of neighbors' information within CSR. Where $HELLO(X, t_n)$ means HELLO message sent from node 'X' at t_n.

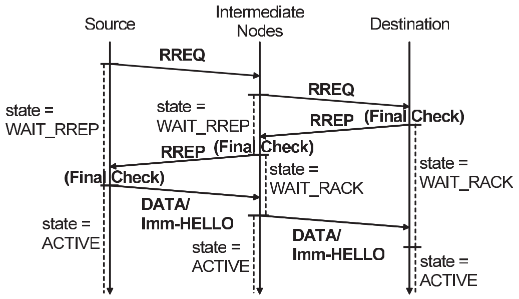


Fig. 5 Message sequence for route discovery.

Thus, the QoS-aware route can be established if one exists.

3.4 Route Maintenance

In this section, we briefly introduce a method to realize route release and updating of traffic information quickly and reliably by using both RERR and Immediate HELLO messages in the event of QoS violation.

Each node periodically updates neighbors' information within CSR by HELLO messaging.

If $B_{cons}(X) > B_{max}$, 'X' determines that it is impossible to satisfy QoS requirements for the sessions with the presently established routes, then the QoS violation recovery process is carried out on each node as follows.

The node that has detected the QoS violation extracts one entry from its own routing table and then sends RERR and Imm-HELLO messages for the session, where RERR is used to request route rediscovery and sent by unicast.

When the node in the route receives the RERR, the node forwards the RERR and broadcasts Imm-HELLO and then removes the entry for the session.

Nodes may overestimate consumed bandwidth caused by the delay of release of unused bandwidth. As a result, there is a possibility that other nodes may consider it as a QoS violation. Imm-HELLO can solve this problem.

Moreover, decrease of self-traffic and consumed bandwidth is based on a single session at a time, when there is no physical link failure. In our scheme, we follow the policy that route maintenance tries to maintain a route of a single session when a QoS violation is detected. Because, if traffic is still excessive after the route maintenance for one session, available bandwidth on the node can be increased in stages due to the next QoS violation detection and route maintenance for other sessions by self-node or any other neighboring nodes.

```
Procedure FinalCheck($\boldsymbol{RL}$, $\boldsymbol{NIT}$, $\boldsymbol{SNIT(d)}$, $req$) {
/* $\boldsymbol{RL} := \{x \mid x$ is in route list except Destination$\}$ */
/* $\boldsymbol{SNIT(d)} := \{x \mid x \in \boldsymbol{NIT},\ x$ is node at a depth of $d\}$ */
/* $req :=$ requested bandwidth for the session */
  $n$:= number of nodes in $\boldsymbol{RL} \cap \boldsymbol{NIT}$;
  if ($B_{max} - B_{cons}(root) < n \times req$) return ("refusal"); /* check for self-node */
  copy $\boldsymbol{SNIT(1)}$ to $\boldsymbol{C1}$;
  while ($\boldsymbol{C1}$ != $\phi$) {
    $en$:= an extracted node from $\boldsymbol{C1}$;
    $n$:= number of nodes in $\boldsymbol{RL} \cap \boldsymbol{NIT}$;
    if ($B_{max} - B_{cons}(en) < n \times req$) return ("refusal"); /* check for 1-hops */
    else remove $en$ from $\boldsymbol{C1}$;
  }
  copy $\boldsymbol{SNIT(2)}$ to $\boldsymbol{C2}$;
  while ($\boldsymbol{C2}$ != $\phi$) {
    $en$:= an extracted node from $\boldsymbol{C2}$;
    $\boldsymbol{F} := \{x \mid x \in \boldsymbol{SNIT(1)},\ x$ has $en$ as its child node$\}$;
    $\boldsymbol{E} := \{x \mid x \in \boldsymbol{SNIT(2)},\ x$ is child of $y$ (for $y \in \boldsymbol{F}$)$\}$;
    $n$:= number of nodes in $\boldsymbol{RL} \cap \boldsymbol{E} \cup \boldsymbol{SNIT(0)}$;
    if ($B_{max} - B_{cons}(en) < n \times req$) return ("refusal"); /* check for 2-hops */
    else remove $en$ from $\boldsymbol{C2}$;
  }
  return ("acceptance");
}
```

Fig. 6 Algorithm pseudo-code for final check.

3.5 Discussion about Overhead Weight

We define "overhead weight", denoted by Ov_Wt, as the proportion of the additional cost to the total cost of sending data.

If we denote the minimal total time for the successful transmission of data by T_{total} and denote the size of a data packet, i.e., IP payload, by $DATA$, we can write

$$Ov\_Wt = T_{total} \times \frac{DataRate}{DATA} \qquad (3)$$

where $DataRate$, i.e., raw data rate, is used to send data frame except the physical header. Now, ignoring the uncertain factors such as backoff time, T_{total} can be calculated based on IEEE 802.11 with RTS/CTS handshake as

$$T_{total} = T_{data} + T_{control} + 4 \times T_{PHY} + 3 \times SIFS + DIFS \qquad (4)$$

where T_{data}, $T_{control}$, and T_{PHY} are the amount of time to transmit each data frame, control frames, and physical header, respectively.

Here, Ov_Wt is given by the equation (3) and specification of IEEE 802.11[9]. The actual value of Ov_Wt is 1.71, and we use this static value in the simulation. Suppose that the data rate of a session flow required for a particular application is $ReqBw$. When the route discovery starts, $ReqBw \times Ov\_Wt$ is calculated at the source node, and $ReqBw$ is updated to that value. Once the source node specifies $ReqBw$, the assigned bandwidth for the session in the route entry is also updated.

In our exploratory experiment, we confirmed that our computational value of Ov_Wt based on IEEE 802.11 specification is appropriate for smooth transmission, and provides operations as expected. So, for bandwidth estimation and route construction in our scheme, we add the overhead weight to each traffic e.g., flow and use it as substantive traffic volume.

4. Simulation and Evaluation

NS-2 simulator[13] is used for the performance evaluation of the ALRS scheme.

4.1 Network Configuration

In our simulation, the performance of each node is based on IEEE 802.11 DCF with a raw data rate of 2 Mbps using RTS/CTS handshake. All the nodes have the same TR of 250 m and CSR of 550 m. A network field of 1,500 m×1,500 m with 50 nodes is used.

Each node's initial position is randomly chosen. And its mobility is determined by the random way-point[14] with pause time set to 0 and with a uniform and equal speed in the network. Each pair of source and destination for each session is randomly chosen.

Constant Bit-Rate (CBR) traffic with fixed data packets of 512 bytes is used as stream media application and required bandwidth of the session.

Bit-rate of CBR traffic (represented as CBR-rate) is the same for every source. CBR-rate is denoted as x Kbps, by "CBR_x".

The total value of all CBR-rates is represented as "Load".

Each of the above three combinations, consisting of speed, Load, and CBR-rate, is used in 30 different scenarios.

For each scenario, we run the simulation for 500 sec.

4.2 Performance Metrics

For evaluating performance of our proposed scheme, ALRS, we have measured the following four performance metrics:

(1) Traffic admission ratio (*Ad*)

The ratio between the number of data packets sent to the network from the sources and the number of data packets generated by the sources during the simulation. This metric is closely related to the proportion of established routes in the network based on the policy of each scheme.

(2) End-to-end packet delivery ratio (*Deli*)

The ratio between the number of data packets received by the destinations and the number of data packets sent by the sources during the simulation. A high delivery ratio means that both admission control with high reliability for constructing the QoS-aware route and appropriate route maintenance have been conducted.

(3) Throughput (*Th*)

The amount of data received at the destinations during the simulation per unit time. It implies utilization efficiency of bandwidth. Control messages such as the HELLO message, RREQ, and RREP are not included in the throughput.

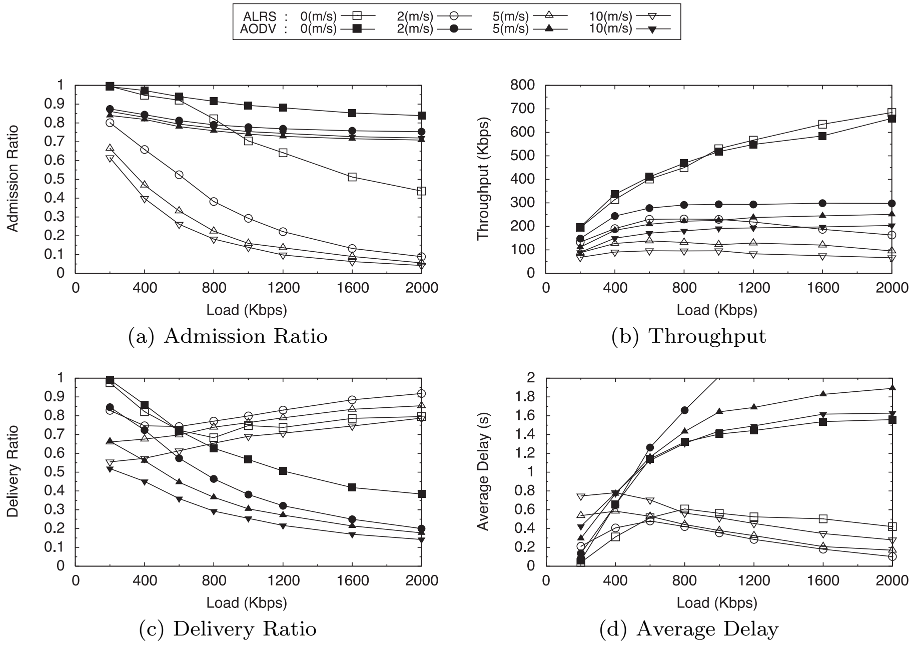
(4) Average end to end delay (*Delay*)

The latency incurred by one packet between the generated time by the source and the arrival time at the destination. The average delay of one session is defined by all the received packets at the destinations during the simulation.

Through experimental results, we evaluate the performance and make a comparison between ALRS and AODV [2], where AODV is a major non-QoS routing scheme for MANETs.

4.3 Results and Study on Static Topology Scenarios

Figure 7 and **Fig. 8** show results at CBR_{40} and at CBR_{200}, respectively. Each point in the graphs expresses the average of 30 simulations.



(a) Admission Ratio

(b) Throughput

(c) Delivery Ratio

(d) Average Delay

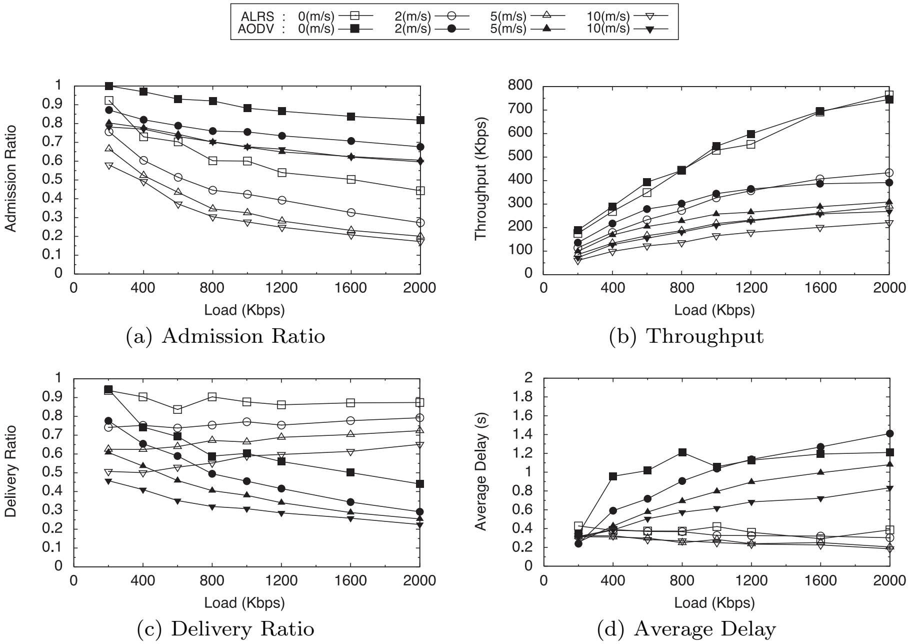
Fig. 7 Average of each metric among 30 scenarios at CBR_{40}.

As a general trend, with both schemes, when the Load is 200 Kbps, *Ad* maintains at 0.95 and above and *Th* is equal to the Load. *Delay* is also set to a very low value. This is because, with low Load, no bottleneck occurs in the network and traffic may not be high.

However with AODV, although *Th* increases, *Deli* decreases and *Delay* increases considerably with the increase in Load when Load is low at each CBR-rate. The trend continues until the Load is 800 Kbps at CBR_{40} and 400 Kbps at CBR_{200}, respectively.

We confirmed that although *Th* of the entire scenario with AODV is about the same as that of ALRS, both *Deli* and *Delay* of each session are low and high respectively since AODV creates excessive traffic in the network by sending many packets in a blind way as shown in Fig. 7 (a) and Fig. 8 (a).

On the other hand, ALRS keeps *Deli* at least 0.7 with any Load. In addition, ALRS achieves low *Delay*, having reduced it by up to 25% of AODV. And *Th* is nearly equal to AODV. Thus, ALRS achieves a high delivery ratio and low delay. And also it has the same level of throughput as AODV in Fig. 7 (b) and Fig. 8 (b) although ALRS suppresses the volume of sent packets as shown in Fig. 7 (a) and Fig. 8 (a). Thus, it can be assured that overheads caused by additional control



(a) Admission Ratio (b) Throughput

(c) Delivery Ratio (d) Average Delay

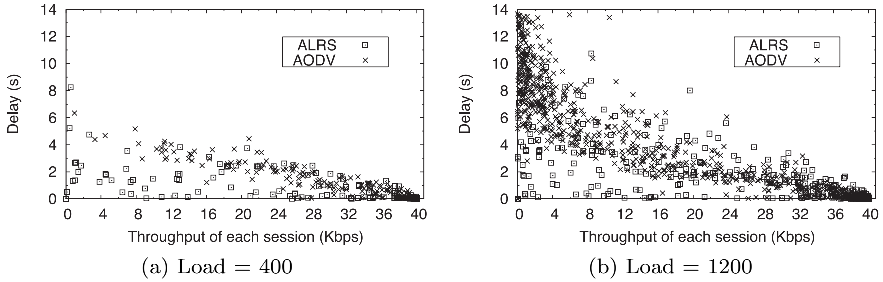
Fig. 8 Average of each metric among 30 scenarios at CBR_{200}.

messages such as a HELLO message, RREQ, and RREP have hardly any negative impact on the performance of our scheme. Now, in Fig. 7 and Fig. 8, with ALRS, the delay decreases and the delivery ratio increases with the increases in load. When the load increases, the requirement that needs multiple hops from source node to destination node will not be fulfilled. Therefore, the number of routes with smaller hop-count will increase with the increase in load.

We will discuss this in more detail in the following sections.

4.3.1 Performance for difference of the level of bottleneck

In each of the static topology scenarios, the possibility of bottlenecks occurring (in the case of route establishment based only on the existing link) depends on the outcome of the selection of the source-destination pair for each session.



(a) Load = 400 (b) Load = 1200

Fig. 9 *Th* vs. *Delay* for each session (all sessions of 30 scenarios in one graph at CBR_{40}).

As shown in Fig. 7 (d) and Fig. 8 (d), with AODV, there is little increase of *Delay* after the Load becomes 800 Kbps at CBR_{40} and 400 Kbps at CBR_{200}. And with ALRS, the *Delay* remains low with the increase in Load after Load becoming 800 Kbps at CBR_{40} and with any Load at CBR_{200}. These tendencies are understood by observing the following results.

Figure 9 shows *Th* vs. *Delay* of each session for all of 30 sessions at CBR_{40}. All graphs with a Load of more than 800 Kbps show the same tendency for both high *Th* and low *Delay* and also for low *Th* and high *Delay*.

First, with AODV, as shown in Fig. 9 (a), when the Load is relatively small (less than 800 Kbps), the levels of both excessive traffic for each bottleneck and the incidence of bottlenecks may be relatively low. So, the degree of increase of delay per packet is larger than the degree of increase of probability of packet drop.

However, as shown in Fig. 9 (b), the higher the Load the higher will be the levels of bottlenecks. So, for each packet which faces bottlenecks, both the delay and the probability of packet drop will dramatically increase.

Therefore, the proportion of the number of received packets with higher delay for each scenario becomes smaller compared to the total packets received. Thus, the average *Delay* of 30 different scenarios reached the saturation level (Fig. 7 (d) and Fig. 8 (d)).

On the other hand, with ALRS, few sessions have very low *Th* and high *Delay*.

Therefore, we can confirm that ALRS achieves a reliable admission control for any load, which determines the creation of a route for each target session based on the appropriate estimation of bandwidth consumption, i.e., to control the volume of sent packets in order to avoid bottlenecks.

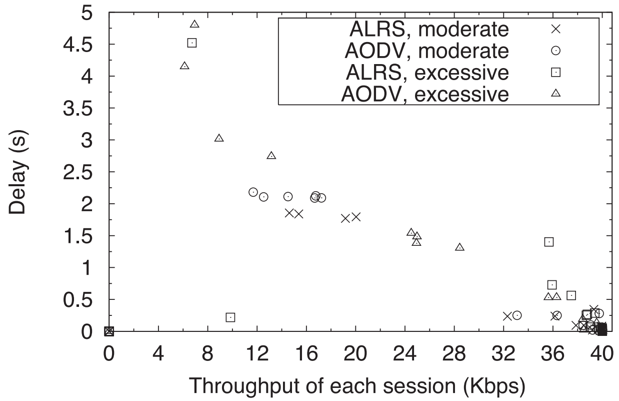
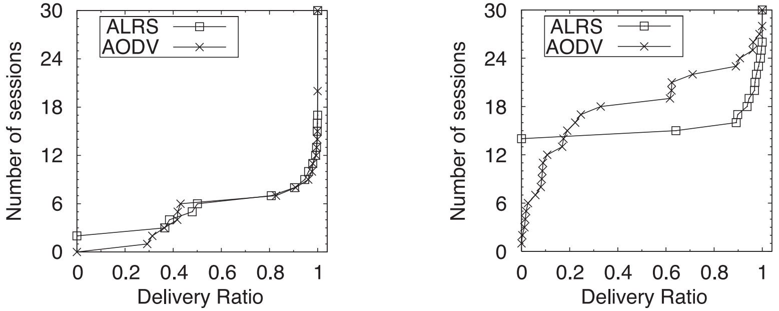


Fig. 10 Th vs. $Delay$ for each session of two scenarios with Load of 1200 Kbps at CBR_{40} for static topology scenarios.



(a) Moderate traffic

(b) Excessive traffic

Fig. 11 Cumulative frequency of sessions with respect to $Deli$ for AODV and ALRS.

The scatter plot of **Fig. 10** shows the relationship between Th and $Delay$, and **Fig. 11** represents the number of sessions with respect to $Deli$ with a Load of 1,200 Kbps at CBR_{40}, i.e., applications require total 30 sessions per scenario. Where in Fig. 11, the Delivery Ratio of the session that cannot establish the route, is set to zero. Delivery ratio with respect to the number of sessions is measured by using the same scenario considered in Fig. 10; moderate traffic in Fig. 11 (a) and excessive traffic in Fig. 11 (b).

The effectiveness of ALRS is signified when traffic is excessive (second scenario) compared to AODV, that does not consider QoS requirements.

In the case of the first scenario, when traffic is moderate, both schemes show a similar performance as shown in Fig. 10 and Fig. 11 (a).

In the second scenario, with AODV, when traffic is excessive, the proportion of sessions with both low Th and high $Delay$ increases as shown in Fig. 10, whereas, for ALRS only one incident appears to have a high delay. Also, for ALRS the number of sessions with high $Deli$ is larger than AODV as shown in Fig. 11 (b).

That is to say, ALRS keeps delivery ratios high and reduces the increase of delays for established routes with selective route establishment to avoid bottlenecks.

Depending on the network topology, in the case of insufficient neighboring nodes, some traffic information of 2-hop neighbors cannot be delivered. In our experiment, this loss of traffic information had occurred. In this situation, the node underestimates B_{cons}, and an unexpected route might be established. This miss-estimation originates bottlenecks in the session, and causes degradation of the delivery ratio and an increase in delay.

4.4 Impact of Nodes Mobility on Performance

Figure 7 and Fig. 8 show how node mobility affects the performance for both AODV and ALRS. We have evaluated our scheme along with AODV in both static and dynamic situation with varying node mobility and with CBR_{40} and CBR_{200}.

Naturally, as ALRS is trying to satisfy QoS, Ad is lower than AODV for both a static and a non-static situation. But when performance metrics are $Deli$ and $Delay$, the two most important parameters in QoS satisfaction, ALRS shows a better performance than AODV even when there is node mobility. Th is almost the same for both the schemes in a static situation and remains the same when the node mobility is low. Only when the node mobility increases to 5 m/sec or higher, does ALRS show slightly lower Th than AODV.

In a static situation when traffic increases, bottleneck occurs and as a result packets are dropped. So both Th and $Deli$ also decreases. When there is node mobility, the occurrence of link failure increases and until a new link is established, packets get dropped and as a consequence Th and $Deli$ decrease for both

ALRS and AODV.

And when node mobility is high, it takes time to update traffic information and therefore inaccurate bandwidth estimation tends to occur. This results in failure or delay of QoS violation detection, so *Deli* drops for ALRS.

4.5 Discussion

AQOR and CEDAR aim to improve the reliability of admission control according to the basic MANET model where TR and CSR have the same range. Therefore, these previous works cannot realize the expected performance improvement on the model that we considered in this paper. The reason could be the underestimation of consumed bandwidth by the existing schemes. This incorrect estimation causes bottlenecks in the established session flow, and it may initiate the degradation of delivery ratio and increase the end-to-end delay. Especially when the load is heavy, this tendency will be more prominent.

As for the applications with best-effort traffic on which the required bandwidth cannot be estimated beforehand, ALRS can work effectively by specifying a lower priority to the frame than applications that have explicit required bandwidth. This can be performed by using MAC protocol of IEEE 802.11e, that can control frame transmission priority. Meanwhile, in terms of such applications where the required bandwidth cannot be estimated beforehand, the bandwidth does not need to be ensured at the initial stage. Thus, ALRS does not have functions to inquire about the required bandwidth to the applications.

In terms of accuracy of the bandwidth estimation, a thorough discussion and evaluation are not completed yet. However, for a consideration of CSR, the related work [15] etc. pointed out the importance of the concept of the range. Moreover, the relationship between flow, traffic, and bandwidth is discussed in Ref. 10). From these works, we think that our bandwidth estimation is adequate. Experiments on real environments have not been conducted yet. In a future work, we would like to investigate the performance of our scheme in a real situation.

5. Conclusions

In this paper we proposed an Application-level QoS Routing Scheme for MANETs (ALRS) which acts adaptively to the dynamic changes of network resources, and described the detail of route construction and maintenance.

We have shown in our simulation based experiment, that ALRS achieved an improvement in delivery ratio by up to 50% and the delay is also reduced to less than one-fourth of AODV on all occasions, while the throughput remains similar to non-QoS routing schemes in a static situation. So it can be said that, ALRS can achieve a high ratio of QoS satisfaction without deterioration of bandwidth utilization. We also proved that our method performs better especially when mobility is moderately low.

In a future work, we are aiming to provide a higher reliability in admission control with more accurate information of route and available resources. Also we wish to evaluate our scheme by varying the node density.

We also wish to consider how to accommodate effective scheduling techniques and efficient use of channel utilization in MAC layer. Performance evaluation of our scheme in an actual application is also one of our future plans.

Acknowledgments This work was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grants-in-Aid for Scientific Research, 19200005 and Ministry of Internal Affairs and Communications in Japan, SCOPE project (071502003).

References

1) Internet Engineering Task Force (IETF) MANET Working Group: Mobile Ad-hoc Networks (manet) Charter (online), available from http://www.ietf.org/html.charters/manet-charter.html

2) Perkins, C.E., Belding-Royer, E.M. and Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561 (2003).

3) Perkins, C. and Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *Proc. ACM SIGCOMM'94*, pp.234–244 (1994).

4) Clausen, T. and Jacquet, P.: Optimized Link State Routing Protocol (OLSR), IETF RFC 3626 (2003).

5) Johnson, D.B. and Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks, *Mobile Computing*, Imielinski and Korth (Eds.), Kluwer Academic Publishers (1996).

6) Lin, C.R. and Liu, J.-S.: QoS routing in ad hoc wireless networks, *IEEE Journal on Selected Areas in Communications*, Vol.17, No.8, pp.1426–1438 (1999).

7) Chen, S. and Nahrstedt, K.: Distributed Quality-of-Service Routing in Ad Hoc Networks, *IEEE Journal on Selected Areas in Communications*, Vol.17, No.8,

pp.1488–1505 (1999).

8) Zhu, C. and Corson, M.S.: QoS routing for mobile ad hoc networks, *Proc. IEEE INFOCOM 2002*, Vol.2, IEEE, pp.958–967 (2002).

9) IEEE Standard Association: Wireless LAN Medium Access Control (MAC) and Phisical Layer (PHY) Specifications (IEEE Std 802.11, 1999 Edition) (Online), available from http://standards.ieee.org/getieee802/802.11.html

10) Xue, Q. and Ganz, A.: Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks, *Journal of Parallel and Distributed Computing*, Vol.63, pp.154–165 (2003).

11) Sivakumar, R., Sinha, P. and Bharghavan, V.: CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm, *IEEE Journal on Selected Areas in Communications*, Vol.17, No.8, pp.1454–1465 (1999).

12) Osada, T., Kitagata, G., Chakraborty, D., Suganuma, T. and Shiratori, N.: An Effective Application-level QoS Routing Scheme for MANETs, *Proc. SAINT 2007*, IEEE Computer Society, p.18 (2007).

13) The VINT Project: The Network Simulator — ns-2 (online), available from http://www.isi.edu/nsnam/ns/

14) Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.-C. and Jetcheva, J.: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proc. ACM/IEEE MobiCom'98*, pp.85–97 (1998).

15) Muqattash, A. and Krunz, M.: A distributed transmission power control protocol for Mobile ad hoc networks, *IEEE Trans. Mobile Computing*, Vol.3, No.2, pp.113–128 (2004).

(Received March 31, 2008)
(Accepted October 7, 2008)
(Original version of this article can be found in the Journal of Information Processing Vol.17, pp.14–25.)

Toshiaki Osada received his M.S. degree from the Graduate School of Information Sciences, Tohoku University, Japan in 2003. Presently he is a doctoral student of the Graduate School of Information Sciences, Tohoku University. His research interests include QoS routing for mobile ad hoc networks. He is a student member of IPSJ.

Gen Kitagata is an associate professor of the Research Institute of Electrical Communication of Tohoku University, Japan. He received a doctoral degree from the Graduate School of Information Sciences, Tohoku University in 2002. His research interests include agent-based computing, network middleware design, and symbiotic computing. He is a member of IEICE, IPSJ.

Debasish Chakraborty received his doctoral degree from the Graduate School of Information Sciences, Tohoku University, Japan in 1999. Presently he is working as a visiting Associate Professor in the Research Institute of Electrical Communication, Tohoku University. He was a Telecommunication Advancement Organization (TAO) research fellow and a National Institute of Information and Communication Technology (NiCT) foreign researcher at Tohoku Research Center. His main research interests are multicast routing algorithm, QoS, Internet traffic analysis, and wireless and ad hoc networking.

Takuo Suganuma is an associate professor of the Research Institute of Electrical Communication of Tohoku University, Japan. He received a doctor of Engineering degree from Chiba Institute of Technology. His research interests include agent-based computing, flexible network, and symbiotic computing. He is a member of IEICE, IPSJ and IEEE.

Norio Shiratori received his doctoral degree from Tohoku University, Japan in 1977. Presently he is a Professor of the Research Institute of Electrical Communication, Tohoku University. He has been engaged in research related to symbiotic computing paradigms between human and information technology. He was the recipient of the IPSJ Memorial Prize Winning Paper Award in 1985, the Telecommunication Advancement Foundation Incorporation Award in 1991, the Best Paper Award of ICOIN-9 in 1994, the IPSJ Best Paper Award in 1997, the IPSJ Contribution Award in 2007, and many others. He was the vice president of IPSJ in 2002. He is a fellow of IEEE, IPSJ, and IEICE.