

A QoS-aware Routing Mechanism for Multi-Channel Multi-Interface Ad-Hoc Networks

Shinsuke Kajioka^{*a}, Naoki Wakamiya^a, Hiroki Satoh^b, Kazuya Monden^b, Masato Hayashi^c, Susumu Matsui^b, Masayuki Murata^a

^aGraduate School of Information Science and Technology, Osaka University, 1-5, Yamadaoka, Suita, Osaka, Japan 565-0871

^bSystems Development Laboratory, Hitachi Ltd., 292, Yoshida-cho, Totsuka, Yokohama, Kanagawa, Japan 244-0817

^cRenesas Technology Europe GmbH, Gothaer Straße 18, 40880 Ratingen, Germany

Abstract

To accommodate real-time multimedia application while satisfying application QoS requirements in a wireless ad-hoc network, we need QoS control mechanisms. In this paper, we propose a new routing mechanism to support real-time multimedia communication by efficiently utilize the limited wireless network capacity. Our mechanism considers a wireless ad-hoc network composed of nodes equipped with multiple network interfaces to each of which a different wireless channel can be assigned. By embedding information about channel usage in control messages of OLSRv2, each node obtains a view of topology and bandwidth information of the whole network. Based on the obtained information, a source node determines a logical path with the maximum available bandwidth to satisfy application QoS requirements. Through simulation experiments, we confirmed that our proposal effectively routed multimedia packets over a logical path avoiding congested links. As a result, the load on a network is well distributed and the network can accommodate more sessions than OLSRv2 and QOLSR. We also conducted practical experiments using wireless ad-hoc relay nodes with four network interfaces and verified the practicality of our proposal.

Key words: ad-hoc network, QoS routing, multi-channel, multi-interface

1. Introduction

Wireless ad-hoc networks need no fixed communication infrastructures such as routers, switches, access points, and cables. Nodes communicate with each other through radio signals to organize a network and transmit data from one node to another. For its infra-less feature, wireless ad-hoc networks are considered the promising technology to establish a means of communication where installation of network equipment and cables is not allowed, difficult, or expensive as in a historic landmark or a festival site or when conventional communication infrastructures are destroyed such as in catastrophic disasters like earthquake. In such situa-

tions, wireless ad-hoc networks are expected to accommodate real-time multimedia traffic for remote monitoring, video conferencing, and VoIP (Voice over IP) communications.

Packets which are transmitted over a wireless ad-hoc network include both of best-effort traffic (file transfer, e-mail, and Web) and real-time traffic (remote monitoring, video conferencing, and VoIP). It has been recognized that the effective network capacity of a single-channel and multi-hop wireless network using the normal IEEE 802.11 standard MAC is not $n \times (\text{per-channel-throughput})$, but $O(n/\sqrt{n}) \times (\text{per-channel-throughput})$ [1], where n is the number of nodes using the same channel in the network. In [2], they further took into account that the phenomena, such as medium contention, channel fading, and radio interference, causing the degradation of the effective bandwidth. Since the capacity of wireless link is limited and the effective bandwidth is much smaller for contention among nodes [1, 2], it is not trivial to accommodate real-time multimedia traffic in a wireless ad-hoc network. Especially, the fact that real-time applications require

*Corresponding author

Email addresses: s-kajioka@ist.osaka-u.ac.jp (Shinsuke Kajioka), wakamiya@ist.osaka-u.ac.jp (Naoki Wakamiya), hiroki.satoh.yj@hitachi.com (Hiroki Satoh), kazuya.monden.vw@hitachi.com (Kazuya Monden), hayashi.masato@renesas.com (Masato Hayashi), susumu.matsui.cb@hitachi.com (Susumu Matsui), murata@ist.osaka-u.ac.jp (Masayuki Murata)

Preprint submitted to Ad Hoc Networks

certain level of QoS guarantee or control in terms of packet loss, delay, and delay jitter makes it challenging.

Over the past several years, many studies have been devoted to QoS control in wireless ad-hoc networks [3, 4, 5, 6]. There are several techniques or methods for controlling QoS in wireless ad-hoc networks, such as bandwidth reservation, channel switching, channel separation, and QoS-aware routing. For example, QOLSR [4, 7] is a QoS-aware routing protocol based on the conventional OLSR (RFC3626). In selecting MPRs (MultiPoint Relay), which are nodes designated to relay broadcast messages, QOLSR considers QoS-related metrics, i.e. bandwidth and delay, than the hop distance used in OLSR. At the MAC layer, the modification of IEEE 802.11 standard MAC protocol is considered to support frame transmission over a multi-channel and multi-interface wireless ad-hoc network. A node switches wireless channels [8] or both of channels and interfaces [9, 10, 11, 12, 13, 14] in a hop-by-hop manner or a time-based manner, to reduce the number of packet losses and improve the network throughput. In [15], they consider multi-channel, multi-interface, and multi-rate wireless network, but they do not consider multi-hop scenario. Their idea is to assign physical links having same or similar data rates on the same channel to minimize the waste of channel resources due to inconsistency among high and low data rate links. According to this modification, they overcome the performance degradation caused by rate adaptation. Although multiple channels and interfaces contribute to avoidance of competition and collision for a wireless channel, P. Kyasanur et al. showed that channel switching in the same frequency band on an interface introduced non-negligible switching delay in [14]. To tackle the problem, they proposed to classify interfaces on a node into “fixed” and “switchable” interfaces so that neighboring nodes can communicate with each other on their fixed channels to avoid the interface switching delay. In their proposal, fixed interfaces stay on their channels for a longer period than switchable interfaces. There are several proposals for controlling QoS as seen above. However, network layer protocols such as QOLSR need routing at all channels and MAC layer techniques need modification of MAC.

In this paper, we propose a QoS-aware routing mechanism for wireless ad-hoc networks, especially used for temporal communication vehicle at a historic landmark, a festival, or a disaster-affected area. Our mechanism assumes a node equipped with multiple network interfaces and to each of which a different wireless channel can be assigned. More specifically, we consider that the number of available wireless channels is equal

to or more than the number of interfaces and channels are assigned to interfaces without overlap. Our mechanism consists of three cooperative techniques; bandwidth estimation, efficient message distribution, and logical routing. One of interfaces is assigned to best-effort traffic and OLSRv2 (OLSR version 2) [16]. The remaining interfaces are devoted to real-time multimedia traffic. A node estimates the usage of its wireless channels and disseminates the information about the available bandwidth on the node, called the bandwidth information, to the other nodes in the whole network. For this purpose, the bandwidth information is embedded in control messages of OLSRv2 and propagated in the whole network in an efficient and effective way. In transmitting real-time packets, a source node tries to estimate the optimal path to its destination node to satisfy application QoS requirements using the obtained topology and bandwidth information. Since the derived path, called a logical path, is different from the physical path from the source to the destination established by the underlying OLSRv2, packets are encapsulated by destination addresses of logical next-hop nodes so that it traverses the logical path. Each intermediate node receiving an encapsulated real-time packet chooses the least used wireless channel on the node to transmit the packet for efficient use of wireless channels and collision avoidance. One of key advantages of our mechanism is that it can be implemented using off-the-shelf hardware.

In the rest of this paper, we first describe our proposal in Section 2 and explain implementation in Section 3. Next, we perform simulation experiments to evaluate the effectiveness of our proposal from viewpoints of end-to-end packet delivery ratio, delay, delay jitter, and node utilization in Section 4. Then, we further build a prototype and conduct practical experiments to verify the practicality in Section 5. Finally, we summarize the paper and describe some future work in Section 6.

2. QoS-aware routing mechanism for wireless ad-hoc networks

In this section, we first give an overview of our proposed mechanism and describe three key techniques in more details, i.e. estimation of the available bandwidth, distribution of bandwidth information, and logical routing.

2.1. Overview of our proposed mechanism

We consider a wireless ad-hoc network consisting of nodes equipped with K ($2 \leq K$) wireless network in-

Table 1: An example of wireless channel and IP address assignment.

IF	ch	IP addr-node 1	IP addr-node 2	IP addr-node 3
wlan0	1	192.168.0.1/24	192.168.0.2/24	192.168.0.3/24
wlan1	6	192.168.1.1/24	192.168.1.2/24	192.168.1.3/24
wlan2	11	192.168.2.1/24	192.168.2.2/24	192.168.2.3/24

terfaces. The same number K of wireless channels is available for wireless communication. Wireless channels are assigned to interfaces without overlap. Without loss of generality, we number channels and interfaces from 0 to $K - 1$, while assigning the same number to the coupled channel and interface and numbering is the same among nodes. In our proposal, one channel numbered 0, called “best-effort channel”, is reserved for best-effort traffic and the other $K - 1$ channels, called “real-time channels”, are used for real-time traffic such as voice or video data. On the best-effort channel, the OLSRv2 with extension for our proposed mechanism operates for proactive physical routing and bandwidth information dissemination. Because of the application scenarios, we assume that nodes are immobile. At least, the topology is stable and unchanged while a session is active. Nevertheless, condition of wireless communication can dynamically change by fading or some other environmental effects.

Table 1 shows an example of wireless channel and IP address assignment on our proposed mechanism. In this example, each of nodes 1, 2, and 3 has three wireless network interfaces named wlan0, wlan1, and wlan2. There are three available channels without interference, 1, 6, and 11. As seen, we assigned the interface wlan0 to channel 1, wlan1 to channel 6, and wlan2 to channel 11 on all of the three nodes. Each interface belongs to a different network, i.e. 192.168.0.0/24 for wlan0, 192.168.1.0/24 for wlan1, and 192.168.2.0/24 for wlan2. Each node has a unique host address, 1, 2, and 3. Therefore, node 1 for example has three IP addresses, 192.168.0.1, 192.168.1.1, and 192.168.2.1, on three network interfaces. By such channel and address assignment, channel switching can be easily done by changing network address of a packet at a source node and intermediate nodes.

Each node always evaluates the usage of real-time channels and estimates the available bandwidth. Information about the estimated available bandwidth is disseminated over the whole network by being embedded on control messages, i.e. HELLO messages and TC (Topology Control) messages of OLSRv2 operating on the best-effort channel. In our mechanism, with a help of OLSRv2, all nodes obtain and maintain the complete information about the available bandwidth on all nodes

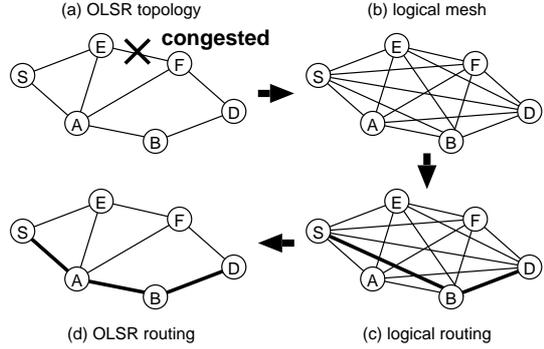


Figure 1: QoS-aware routing by proposed mechanism.

in the network.

Packets belonging to best-effort traffic are transmitted to a destination node on the best-effort channel. Intermediate nodes choose a next-hop node for the destination node of a received packet in accordance with the routing table maintained by OLSRv2. On the other hand, packets belonging to real-time traffic are transmitted to a destination on real-time channels traversing a so-called logical path. A logical path consists of one or more contiguous logical links. A logical link consists of one or more physical links from one end to the other. A logical path is determined taking into account the topology of a wireless ad-hoc network, the available bandwidth on all physical links, and application QoS requirements.

Figure 1 illustrates an example of logical path construction and packet forwarding. Figure 2 shows the way that a packet is processed in our system. We refer to a flow of traffic generated by a real-time application as a session. When a packet to a new destination is generated by a real-time application, a source node determines a logical path to its destination for the session. To determine a logical path, source node S first considers a logical mesh topology on a physical network (Figure 1(b)). Each of logical links in the logical mesh topology is related to a physical path connecting the two ends of the logical link. Then, source node S tries to find an optimal path with respect to application QoS requirements and some other metric if needed, to destination node D. In this example, logical path S-B-D is chosen (Figure 1(c)) and a physical path determined by OLSRv2 could be S-E-F-D. The purpose of the logical routing is to avoid traversing a physical path containing any congested links, which deteriorate QoS provided to an application. So that packets travel the logical path, all packets belonging to the session for node D are encapsulated by a logical routing header, which in-

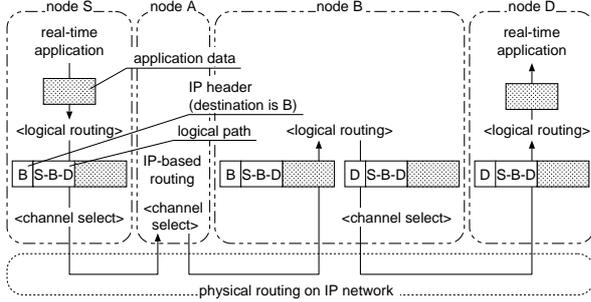


Figure 2: Packet processing in proposed mechanism.

indicates the first destination node B and the last destination node D, as shown in Figure 2. Encapsulated packets are sent to the first destination node B through the physical path from source node S to node B, and then sent to the final destination node D from node B (Figure 1(d)). In this case, the logical next-hop node at node S is node B while the physical next-hop node at node S is node A, based on OLSRv2 physical routing. Therefore, node S sends a packet to node A, then the node A forwards the packet to node B. The intermediate node A only relays a received packet to node B, which is regarded as the destination of the packet from the physical routing view point. For efficient use of wireless bandwidth, each node chooses one real-time channel with the largest available bandwidth among real-time channels in forwarding a packet to a physical next-hop node. When a packet arrives at a logical intermediate node, it is encapsulated with a new header indicating the next logical hop node (Figure 2, node B). Then the packet is forwarded to the logical next-hop node. In this way, real-time packets traverse a logical path over a network maintained by a physical routing protocol, i.e. OLSRv2.

2.2. Estimation of available bandwidth

There have been some studies on estimation of the available bandwidth in a wireless network [2, 17]. The easiest way is to subtract the measured throughput in the MAC layer from the typical or ideal channel bandwidth [2], that is, (available bandwidth) = (channel bandwidth) – (measured throughput in MAC layer). In our proposal, for the sake of simplicity, the available bandwidth $B_k(c)$ of channel c ($1 \leq c \leq K-1$) on node k is estimated by Equation 1, where $W_k(c)$ corresponds to the ideal capacity of channel c on node k , e.g. 54 Mb/s for IEEE 802.11g, T is the estimation interval, and $D_k(c)$ is the total amount of data transmitted and received on channel c at node k in the estimation interval.

$$B_k(c) = W_k(c) - D_k(c)/T. \quad (1)$$

Then, the total available bandwidth B_k for real-time traffic on node k is derived by the following equation.

$$B_k = \sum_{c=1}^{K-1} B_k(c). \quad (2)$$

We should note here that we do not limit to the above estimation and a node can use any of useful estimation mechanisms proposed in literatures.

2.3. Distribution of bandwidth information on OLSRv2

On OLSRv2 (and OLSR) protocol, nodes to forward TC messages are limited to avoid the loss of bandwidth for control packets. They are called MPRs (MultiPoint Relay). Among nodes receiving TC message, only MPRs rebroadcast the message. MPRs are chosen in a distributed manner to keep the connectivity with the smallest number of MPRs. Nodes which select other nodes as MPR are called MPR selectors. Please refer to the standard for selection of MPR (RFC3626 1.4. or [18]).

Nodes exchange HELLO messages with neighboring nodes in the range of radio signals at regular HELLO intervals, e.g. 2 seconds. A HELLO message consists of validity time, originator address of the message, neighbor list of the originator, and some optional information. In addition to HELLO messages, an MPR generates and disseminates TC messages at regular intervals, e.g. 5 seconds. A TC message contains validity time, originator address of the message, and addresses of its MPR selectors. On receiving a TC message, a node builds or updates a table called Topology Set consisting of MPRs, their MPR selectors, sequence number, and validity time. A node also builds or updates another table called Attached Network Set consisting of OLSRv2 gateway address, network address which may be reachable via the gateway, prefix length of the network address, sequence number, and validity time. A routing table, called Routing Set, is built and maintained when any of Link Set, Neighbor Address Association Set, 2-hop Neighbor Set, or Topology Set changes. The Routing Set consists of destination address, next-hop address to the destination, number of hops to the destination, and interface address. Entries of the Routing Set are copied to the IP routing table in the system.

In our proposal, the bandwidth information is also entrusted in HELLO and TC messages by adding the extended field in the form of TLV (Type Length Value) block. On receiving them, a node builds or updates the Extended Topology Set, newly introduced for the proposal, to maintain the bandwidth information.

2.4. Logical routing based on bandwidth information

On receiving a first packet to a new destination, a node generates a logical full-mesh topology. The current physical network topology and bandwidth information of each node are obtained from OLSRv2 with our extension. A logical link between node i and node j in a logical full-mesh topology is associated with the available bandwidth $B(i, j)$. The available bandwidth $B(i, j)$ is given as the minimum among the available bandwidth of all physical links on the shortest path between node i and node j . The available bandwidth of physical link is defined as the minimum of the available bandwidth on nodes at the both edges. For example, the available bandwidth $B(S, B)$ in Figure 1 is given as $B(S, B) = \min(\min(B_S, B_A), \min(B_A, B_B))$, where B_S , B_A , and B_B are the total available bandwidth for real-time traffic on each node derived by Equation 2. When there are two or more shortest paths for a logical link, one with the minimum available bandwidth is chosen.

Once a logical mesh network is constructed, a source node begins to find the optimal path with the maximum available bandwidth. First, a set of logical paths with a logical hop count of less than H are obtained from the logical mesh network. The upper bound H is introduced to avoid generating an unnecessarily long path and shorten the calculation time. Then, a source node derives the available bandwidth of each of logical paths in the set as the minimum available bandwidth of logical links constituting the logical path. Finally, the logical path with the largest available bandwidth in the set is chosen for the session. When there are two or more logical paths with the same largest available bandwidth, the logical path that has the smallest physical hop count is chosen for the session to minimize end-to-end delay. When there are two or more logical paths with the same largest available bandwidth and the smallest physical hop count, the logical path found the earliest is chosen for avoidance of overhead in memory copy.

3. Implementation of QoS-aware routing mechanism

In this section, we describe how our QoS-aware routing mechanism is implemented on a wireless ad-hoc network system. Figure 3 shows module components of our proposed mechanism. In the figure, a node has four network interfaces and four wireless channels. We assign channel 0 for best-effort traffic and channels 1, 2, and 3 for real-time traffic.

Packets generated by a best-effort application are transmitted through channel 0. They are sent to a destination following physical routing maintained by the

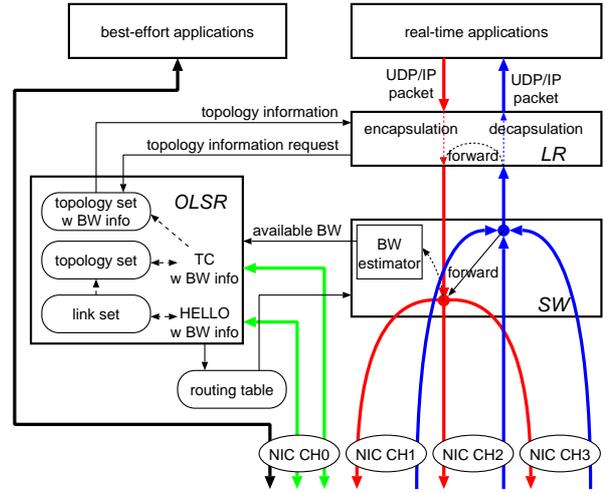


Figure 3: Module components of proposed mechanism (red: outgoing, blue: incoming, green: control flow).

OLSR module, which implements standard OLSR with our extension.

Packets generated by a real-time application are first processed by the logical routing module (LR). On receiving the first packet of a new session, the LR determines a logical path based on topology and bandwidth information maintained by the OLSR module. Packets are encapsulated by an LR header indicating addresses of intermediate nodes of the logical path as shown in Figure 4, so that it traverses the logical path on the physical network maintained by the OLSR module. Encapsulated packets are passed to the switching (SW) module. The LR header consists of two parts, i.e. the header information part and the logical path information part. The header information part consists of header identifier, message type, number of addresses in the logical path information part, message length, source port number, and destination port number. The logical path information part consists of pairs of flags (IP version, source, destination, and visited bit) and an IP address, from the source node to the destination node on the logical path. The LR maintains a table of existing sessions, called the session management table, consisting of destination IP address, source port number, destination port number, timestamp, and the corresponding LR header information. Timestamp in the table is updated when the entry is made or referred to. The structure of the session management table, written in C language, is shown below.

```
struct session_management_table {
    InetAddr dstAddr; /* destination IP address */
    uint16_t srcPort, /* source port number */
    dstPort; /* destination port number */
    clocktype lastTime; /* made/referred timestamp */
}
```

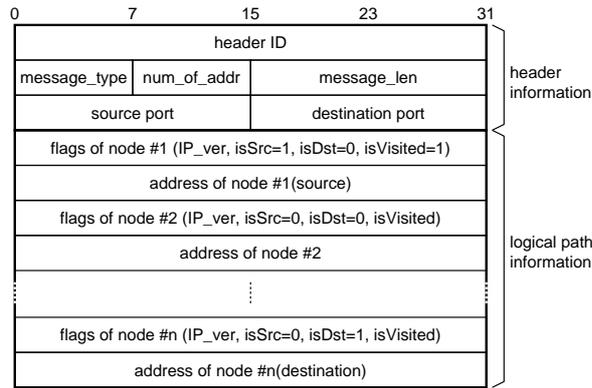


Figure 4: LR header format.

```

void *lr_info; /* LR header information */
}

```

Constructed LR headers are stored in a memory, and the LR header information in the table is a pointer to the corresponding one of them prepared to avoid reconstruction overhead. If the session management table already has an entry for the session and less than 30 seconds have passed since the entry was made or referred to, the LR header is obtained from the LR header information of the entry.

On receiving a packet from the LR, the SW looks up the logical next-hop node of the packet. Next, the SW determines a physical next-hop node for the logical next-hop node based on the routing table. Then, the SW selects one real-time interface with the maximum available bandwidth. Finally, the SW emits the packet destined to the logical next-hop node through the network interface.

On the contrary, when the SW receives a packet from a network, it searches the logical path information part in the LR header. The topmost node with unset ‘visited flag’ is the logical next-hop node. If the logical next-hop node is not node itself, the SW sends the packet to the physical next-hop node on the physical path toward the logical next-hop node. Otherwise, the SW forwards the packet to its own LR. On receiving a packet from the SW, the LR investigates the LR header to check whether it is the final destination or not. If the node is the final destination, the LR removes the LR header from the packet and passes it to the corresponding real-time application.

The SW is also responsible for estimation of the available bandwidth. The BW estimator module in the SW observes packet transmission and reception on real-time channels, estimates the available bandwidth by Equation 1, derives the available bandwidth of node by

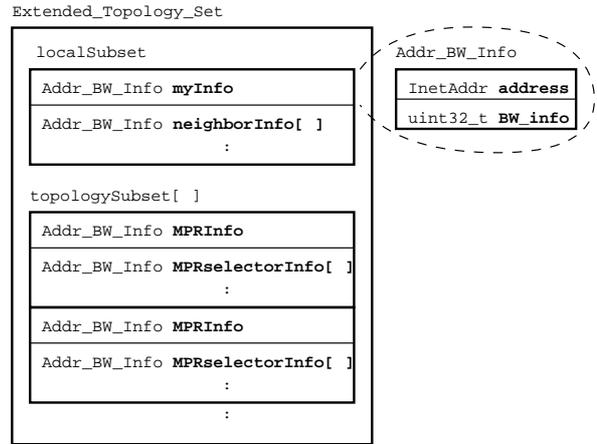


Figure 5: The structure of the Extended Topology Set.

Equation 2, and reports the result to the OLSR module (OLSR). By using the message counter, the BW estimator module counts the number of packets the SW sent and received (all the while the SW is active) and their total size in bytes including the actual size of UDP/IP packet and estimated size of MAC header and MAC trailer. The structure of message counter is shown below.

```

struct message_counter {
    uint64_t TxByte, /* number of sent bytes */
    TxPacket, /* number of sent packets */
    RxByte, /* number of received bytes */
    RxPacket; /* number of received packets */
}

```

The OLSR manages a physical network by exchanging HELLO and TC messages on a best-effort channel. The OLSR obtains information about the available bandwidth of node from the SW. The obtained information is stored in the myInfo field of the localSubset in the Extended Topology Set, which deposits the original topology information of OLSRv2 and additionally the bandwidth information (Figure 5). Addr_BW_Info consists of IP address and bandwidth information as shown below.

```

struct addr_bw_info {
    InetAddr address; /* IP address */
    uint32_t bandwidth_info; /* bandwidth information */
}

```

The OLSR embeds the information about its available bandwidth in HELLO messages and sends them to neighboring nodes. Once the OLSR receives a HELLO message from a neighboring node, it also embeds the information about the neighboring node’s available bandwidth in HELLO messages. In addition to HELLO messages, the OLSR of an MPR generates and disseminates

TC messages embedded with the information about its available bandwidth and the available bandwidth of its MPR selectors. On receiving HELLO or TC message, the OLSR builds or updates the Extended Topology Set. The structure of the Extended Topology Set is shown below.

```

struct extended_topology_set {
    struct localSubset {
        struct addr_bw_info myInfo; /* obtained from the SW */
        struct addr_bw_info neighborInfo[]; /* from HELLO msgs */
    }
    struct topologySubset[] {
        struct addr_bw_info MPRInfo; /* obtained from TC msgs */
        struct addr_bw_info MPRselectorInfo[]; /* from TC msgs */
    }
}

```

On receiving a request from the LR, the OLSR provides the LR with the Extended Topology Set.

4. Simulation experiments and discussions

In this section, we evaluate the performance of our proposal through simulation experiments. We used QualNet 4.0 simulator [19]. We based our OLSR module on the nOLSRv2 [20] with some modifications for supporting our proposed mechanism. Although the normal nOLSRv2 supports multiple interfaces, we modified the nOLSRv2 to operate on the best-effort channel only, thus, no routing protocol operated on the real-time channels.

4.1. Fundamental settings

We built a network consisting of 100 nodes randomly distributed in the $6000 \times 6000 \text{ m}^2$ region. Each node has four wireless network interfaces with Omnidirectional antenna, to each of which ch1 (2.412 GHz), ch6 (2.437 GHz), and ch11 (2.462 GHz) for real-time channels and ch14 (2.484 GHz) for best-effort channel are assigned respectively. Many researches of ad-hoc or mesh networks consider a simple connection model, e.g. a node can communicate with other nodes in the diameter of 125 m at the rate of 54 Mb/s and radio signals can cause interference so the node cannot communicate in the diameter from 126 m to 250 m. In our simulation, different from these researches, a node can communicate with other nodes in the diameter of up to 1219 m at the rate ranging from 6 to 54 Mb/s depending on the distance shown in Table 2. Broadcasting data rate was set to 6.0 Mb/s, which is the lowest rate of IEEE 802.11g with OFDM, to keep network connectivity. Radio signals transmitted by a node can cause SNR (Signal to Noise Ratio) deterioration of other nodes in the diameter of 2237 m (-111.0 dB). We used the free space path-loss model with no shadowing and no fading. A FIFO

Table 2: Transmitting data rates and maximum communication range.

transmitting data rate	maximum communication range
6.0 – 9.0 Mb/s	1219 m
12.0 – 18.0 Mb/s	863 m
24.0 – 36.0 Mb/s	432 m
48.0 – 54.0 Mb/s	122 m

buffer at IP layer has the capacity of 50000 bytes. For OLSRv2, we set intervals of HELLO and TC messages at 2 seconds and 6 seconds, respectively.

As a real-time application, we assumed video streaming traffic. A pair of source and destination nodes was chosen at random without overlapping between two nodes. A source node generated UDP packets of 1292 bytes every 20 ms, i.e. 512 kb/s CBR traffic. We measured the packet delivery ratio, the delay, and the delay jitter averaged over all packets of all sessions. After first 60 s for initialization of network, we started sessions one by one from 60 to 120 s in simulation time. Each session kept sending packets for 60 s. To keep the number of active sessions from 120 to 540 s in the experiments, we initiated a new session between a newly selected node pair as soon as any of existing session was finished. A simulation run was terminated at 606 s in simulation time after all packets had reached to destination nodes.

4.2. Comparison with QOLSR

To evaluate the effectiveness of our proposal, we consider QOLSR [7], one of the QoS-aware routing protocol based on the OLSR, for comparison. The behavior of QOLSR in our evaluation is as follows. Although the original QOLSR takes into account bandwidth and delay as routing criteria, we modified it to consider only bandwidth, which can be measured by our SW. We also extended QOLSR to handle multiple interfaces and channels. From now on, we refer to the modified QOLSR simply as QOLSR. In the QOLSR evaluation, each node operating on QOLSR also has four channels. Ch1 (2.412 GHz), ch6 (2.437 GHz), and ch11 (2.462 GHz) are assigned to real-time channels and ch14 (2.484 GHz) to best-effort channel. Real-time channels accommodate real-time traffic and best-effort channel carries best-effort traffic and control messages. Each node operating on QOLSR selects MPRs in the descending order of total available bandwidth on all real-time channels. In case of a tie, a node with maximum number of uncovered 2-hop neighbors is chosen as an MPR. Each node propagates control messages via the selected MPRs on the best-effort channel to manage network topology. As in our proposal, the phys-

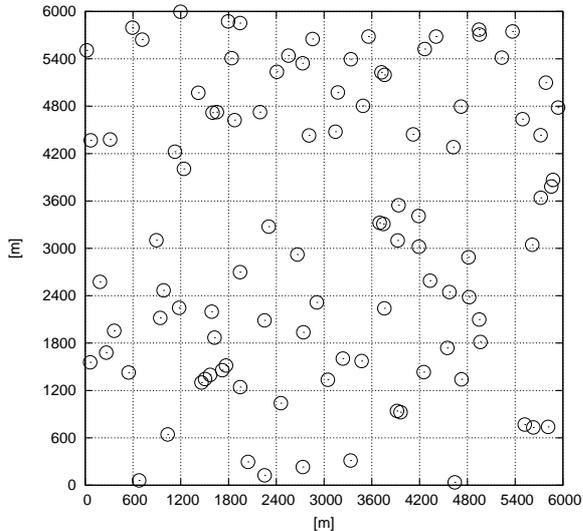


Figure 6: Node placement of one case of the simulation I.

ical topology for routing real-time traffic in QOLSR is the same as that constructed based on control messages exchanged on the best-effort channel. A real-time packet is sent to one of neighboring MPRs on a real-time channel with the largest available bandwidth. We had disabled our LR (QoS-aware logical routing) when QOLSR (QoS-aware physical routing) was running while our SW was kept active to obtain bandwidth information used for MPR and channel selection. As a physical routing protocol, QOLSR was running instead of our OLSR in the QOLSR evaluation.

4.3. Simulation I—General topology

We accommodated 10 random seeds. An example of node placement is shown in Figure 6. Because of the density, all nodes have at least one node in the range of radio signals and a constructed network is connected. Results on the average end-to-end packet delivery ratio, the average end-to-end delay, and the average delay jitter are shown in Figure 7. We can see from Figure 7(a) that the average end-to-end packet delivery ratio is similar between our proposal and QOLSR or a little lower in our proposal, e.g. 90% with our proposal and 92% with QOLSR for 10 sessions. From Figures 7(b) and 7(c), our proposal is also slightly inferior to QOLSR in terms of the average delay and the average delay jitter. A reason for these slight performance degradations are for establishing about 1.3 times as long path in the number of physical hops as the shortest path established by QOLSR in order to detour congested links.

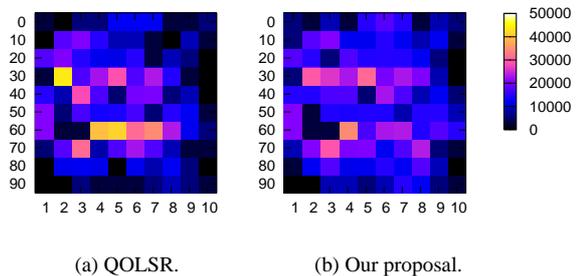


Figure 8: Comparison of total number of transmitted MAC frames.

While the performance in terms of packet delivery ratio, delay, and delay jitter is almost identical among the proposal and QOLSR, the proposal has an advantage in load distribution. In Figure 8, the total numbers of transmitted MAC frames at nodes on a random seed are illustrated. Each of cells corresponds to a node. The sum of values on the x and y axes indicates the node identifier. We should note that nodes are arranged following their identifiers, not the location. It is noticed that nodes 32 and 65 are heavily loaded with QOLSR (Figure 8(a)), whereas the load is relatively distributed over the whole network with our proposal (Figure 8(b)). From quantitative viewpoints, the average variance of transmitted MAC frames per node is 1.02×10^8 in Figure 8(a) while it is 0.76×10^8 in Figure 8(b). The average fairness index is 0.53 in Figure 8(a) and it is 0.66 in Figure 8(b). The fairness index f of 100 nodes is derived by the following equation.

$$f = \frac{\left(\sum_{i=1}^{100} x_i\right)^2}{100 \sum_{i=1}^{100} x_i^2} \quad (3)$$

where x_i is the value of transmitted MAC frames at node i . The fairness index 1 means that nodes are used equally. From these results, we can say that our proposal compensates the performance degradation caused by taking a longer physical path with avoiding congested links and balancing the load over the whole network.

4.4. Simulation II—Uniform topology

In the second simulation scenario, considering rather regular placement of nodes as in the actual environment where nodes are placed keeping a certain distance, we first divided the region into 100 cells and placed nodes at random location one per cell. We accommodated other 10 random seeds. Furthermore, taking into account the

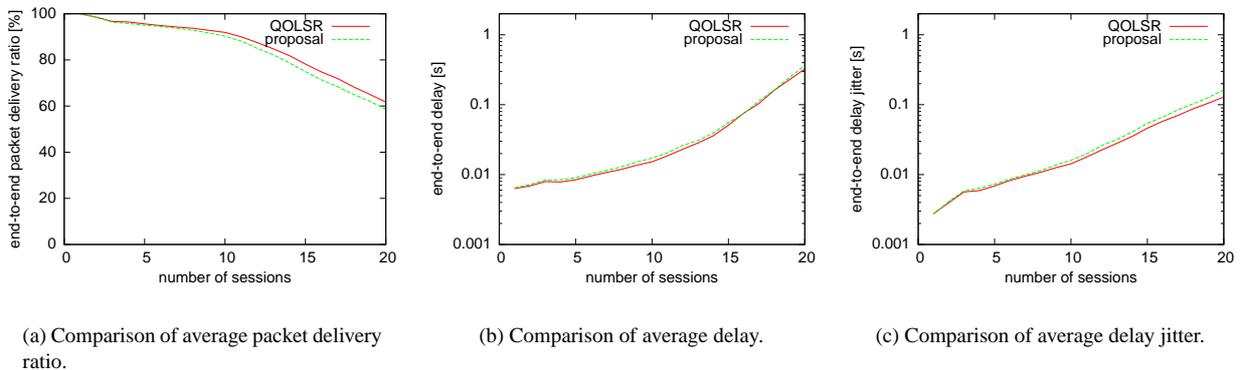


Figure 7: Results of the simulation I (simulation area: 6000×6000 m, number of nodes: 100, broadcast rate: 6 Mb/s, packet size: 1292 bytes).

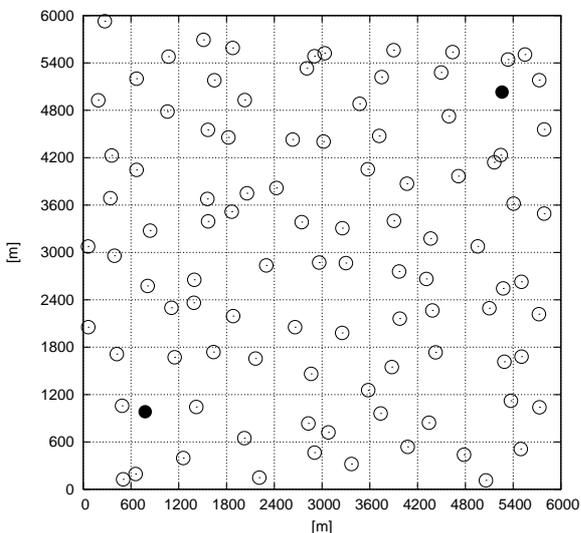


Figure 9: Node placement of one case of the simulation II.

fact that video sessions are not established among arbitrary pairs of nodes, but between a specific pair of nodes, we fixed source and destination nodes during a simulation run. An example of node placement is shown in Figure 9, where filled circle at lower left cell indicates the source node and one at upper right cell indicates the destination node. Because of the regularity of node placement, a node has at least one neighbor within the distance of 863 m and thus we set the broadcasting data rate at 12.0 Mb/s. Results are shown in Figure 10.

We can see that the proposal can accommodate more sessions than QOLSR keeping the high packet delivery ratio in Figure 10(a). Up to 5 sessions, both of the proposal and QOLSR could achieve the packet delivery ra-

tio of about 97%. However, when the amount of traffic further increases, the performance of QOLSR deteriorates more rapidly than the proposal for the concentration of traffic. For example, with 7 sessions, the packet delivery ratio is about 89% with the proposal and about 83% with QOLSR. When we see the delay and delay jitter in Figures 10(b) and 10(c), there is a crossing point at around 5 sessions. In a lightly loaded network, i.e. less than 5 sessions, the delay and delay jitter of the proposal is slightly higher than that of QOLSR for the overhead of LR header and longer physical paths. However, when the load further increases, the proposal outperforms QOLSR by distributing traffic over the whole network by the logical routing. The number of sessions satisfying the delay requirement for interactive voice communication, i.e. 150 ms (ITU-T G.114 about one-way transmission time) increases from 7 with QOLSR to 9 with the proposal. From these results, we consider that our proposal is effective especially for real-time multimedia applications which may exhaust the capacity of particular wireless links, such as high-quality P2P video conferencing and remote monitoring with multiple cameras and single monitoring point.

5. Practical experiments and discussions

We implemented the proposal on a real wireless ad-hoc network and conducted practical experiments to verify the practicality and applicability of the proposal. In this section, we describe the experimental system and the obtained results.

5.1. Experimental system

We used ad-hoc wireless relay nodes made by Hitachi Information & Communication Engineering shown in

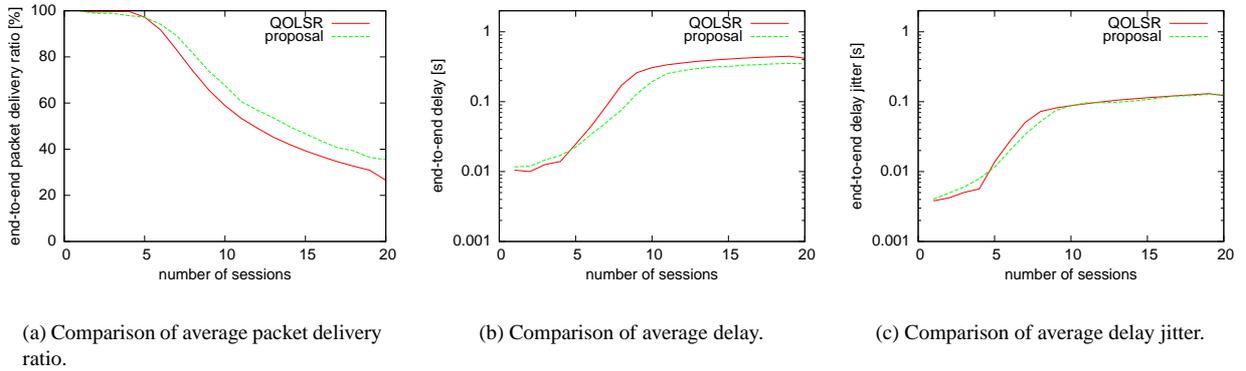


Figure 10: Results of the simulation II (simulation area: 6000×6000 m, number of nodes: 100, broadcast rate: 12 Mb/s, packet size: 1292 bytes).



Figure 11: The ad-hoc wireless relay node.

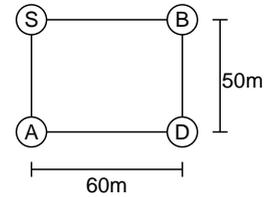


Figure 12: Experimental topology.

Figure 11, in implementing our proposed mechanism. A node has four wireless network interfaces which support IEEE 802.11b/11g MAC protocols. We set three interfaces to the ad-hoc mode and configured one interface among them as best-effort channel and the other two as real-time channels. Since IEEE 802.11g has three orthogonal channels by being separated by at least 25 MHz to avoid inter-channel interference, we assigned 2.412 GHz (numbered as channel 0) to best-effort channel and 2.442 GHz (channel 1) and 2.472 GHz (channel 2) to real-time channels. So that several modules running on a node could share the information, such as topology, on an embedded system with very limited memory space, we developed a semaphore module to realize a shared memory mechanism. To avoid the performance degradation for exclusive memory access, we should carefully determine the locking duration of each access. Because of this severe limitation on the architecture, the obtained performance was not as high as expected as will be shown later. However, we think that we could successfully confirm the behavior of our proposal in an actual operating environment.

5.2. Experimental environment and discussions

Since we had only four available nodes, we organized a simple square topology as illustrated in Figure 12. Nodes are placed at corners of a building (Figure 13). Although each channel was separated by 30 MHz to avoid inter-channel interference, a channel might be affected by other channels. To avoid the interference as much as possible, we separated antennas at least 20 cm from each other. Nodes S and D are source and destination node, respectively. The distance between two neighboring nodes, i.e. S-A, S-B, A-D, and B-D were about 50–60 m. Solid lines indicate physical links. Assuming VoIP traffic, we configured the source node to generate 64 kb/s CBR traffic per session. In practical experiments, source node S generated a new session every 5 s and sessions kept alive until the end of each experiment. At the beginning of measurement, network interfaces were operating and OLSRv2 was fully functional.

Figure 14 shows the data reception rate and the expected data rate, which is equal to 8600 bytes per second (64 kb *data traffic* + *IP header*). Figure 15 shows the delay jitter per session. Figures 16 and 17 show channel usage in terms of the transmission data rate at interme-



Figure 13: Node placement on practical experiments.

intermediate nodes A and B, respectively. Until about 35 s, the data reception rate per session was as high as the expected data reception rate and the packet delivery ratio was higher than 98%. The delay jitter was as small as 20 ms. A physical path from node S to D established by OLSRv2 was S-A-D. However, as can be seen in Figures 16 and 17, nodes A and B were almost equally used by load balancing of logical routing. The LR selected node A as an intermediate node for the first two sessions and node B for the next four sessions. Moreover, it can be noticed that real-time channels on each intermediate node were evenly used by the SW.

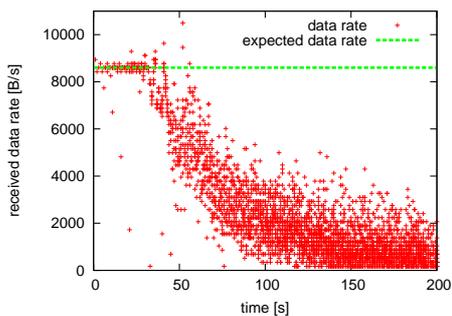


Figure 14: Data reception rate per session at node D.

From 35 s, however, the data reception rate per session suddenly deteriorated and the delay jitter exponentially increased. The reason for this can be explained by Figure 18, where the transition of CPU usage on node S is depicted. The ratio of idle CPU dropped to zero at 35 s and was kept zero since then. It implies that the drop of data reception rate was caused by full utilization

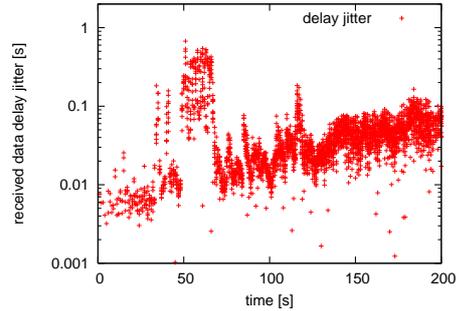


Figure 15: Delay jitter per session at node D.

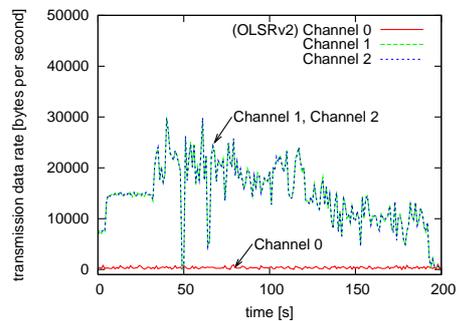


Figure 16: Transmission data rate per channel at node A.

of poor CPU resource of node S.

6. Conclusion

In this paper, we proposed a QoS-aware routing mechanism for real-time applications. By embedding bandwidth information in control messages of OLSRv2, a source node establishes the logical path with the maximum available bandwidth. Through experiments on a simulator, we confirmed that our proposal could achieve almost the same packet delivery ratio, the end-to-end delay, and the delay jitter as QOLSR in general topology. In addition, our proposal more evenly distributed traffic over the whole network than QOLSR. When we considered more regular node placement, our proposal could achieve better performance than QOLSR. We implemented our proposal to the experimental system and confirmed that our proposal worked at an actual environment. Since the logical routing is done at a source, the proposal can be extended to deal with other QoS measurements than the available bandwidth as far as the required information is locally obtained by the SW module.

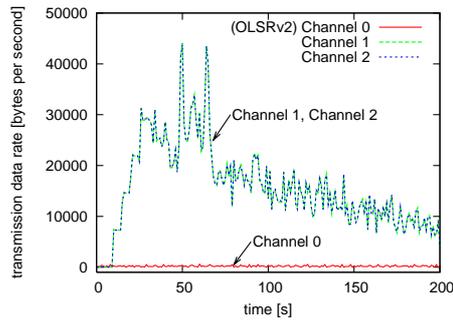


Figure 17: Transmission data rate per channel at node B.

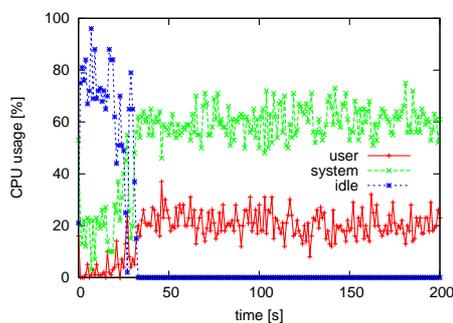


Figure 18: Transition of CPU usage at node S.

As future research work, we are going to conduct large scale experiments and for this purpose we need to improve the structure and program to reduce the load on node.

Acknowledgment

This research was supported in part by “Special Coordination Funds for Promoting Science and Technology: Yuragi Project” of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

- [1] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, R. Morris, Capacity of ad hoc wireless networks, in: Proc. MobiCom 2001, 2001, pp. 61–69.
- [2] S. H. Shah, K. Chen, K. Nahrstedt, Available bandwidth estimation in IEEE 802.11-based wireless networks, in: Proc. ISMA/CAIDA Workshop BEst 2003, 2003.
- [3] S. Chakrabarti, A. Mishra, QoS issues in ad hoc wireless networks, IEEE Communications Magazine 39 (2) (2001) 142–148.
- [4] H. Badis, A. Munaretto, K. A. Agha, G. Pujolle, QoS for ad hoc networking based on multiple metrics: Bandwidth and delay, in: Proc. IFIP-TC6 MWCN 2003, 2003, pp. 15–18.
- [5] R. Guimarães, J. Morillo, L. Cerdà, J. Barceló, J. García, Quality of service for mobile ad-hoc networks: an overview, Polytechnic University of Catalonia.
- [6] L. Hanzo, R. Tafazolli, A survey of QoS routing solutions for mobile ad hoc networks, IEEE Communications Surveys and Tutorials 9 (2) (2007) 50–70.
- [7] H. Badis, K. A. Agha, Quality of service for ad hoc optimized link state routing protocol (QOLSR), Mobile Ad hoc Networking (MANET) Internet-Draft.
- [8] A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks, in: Proc. BROADNETS 2004, 2004, pp. 344–354.
- [9] A. Nasipuri, J. Zhuang, S. R. Das, A multichannel CSMA MAC protocol for multihop wireless networks, in: Proc. IEEE WCNC 1999, 1999, pp. 1402–1406.
- [10] A. Tzamaloukas, J. J. Garcia-Luna-Aceves, A receiver-initiated collision-avoidance protocol for multi-channel networks, in: Proc. IEEE INFOCOM 2001, 2001, pp. 189–198.
- [11] Y. Liu, E. W. Knightly, Opportunistic fair scheduling over multiple wireless channels, in: Proc. IEEE INFOCOM 2003, 2003, pp. 1106–1115.
- [12] J. So, N. H. Vaidya, Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver, in: Proc. ACM MobiHoc 2004, 2004, pp. 222–233.
- [13] A. Raniwala, T. Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in: Proc. IEEE INFOCOM 2005, 2005, pp. 2223–2234.
- [14] P. Kyasanur, N. H. Vaidya, Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks, ACM SIGMOBILE Mobile Computing and Communications Review 10 (2006) 31–43.
- [15] N. Niranjana, S. Pandey, A. Ganz, Design and evaluation of multichannel multirate wireless networks, Mobile Networks and Applications 11 (5) (2006) 697–709.
- [16] T. H. Clausen, C. M. Dearlove, P. Jacquet, The optimized link state routing protocol version 2 (OLSRv2), Mobile Ad hoc Networking (MANET) Internet-Draft.
- [17] K. Xu, K. Tang, R. Bagrodia, M. Gerla, M. Bereschinsky, Adaptive bandwidth management and QoS provisioning in large scale ad hoc networks, in: Proc. IEEE MILCOM 2003, 2003, pp. 1018–1023.
- [18] A. Qayyum, L. Viennot, A. Laouiti, Multipoint relaying for flooding broadcast messages in mobile wireless networks, in: Proc. HICSS 2002, 2002, pp. 3866–3875.
- [19] Scalable Network Technologies, Inc., QualNet 4.0, <http://www.scalable-networks.com/>.
- [20] Niigata University, OLSRV2 Wiki, <http://www2.net.ie.niigata-u.ac.jp/nOLSRv2/v2devel/>.