Master's Thesis

Title

Proposal, Implementation, and Evaluation of a QoS-aware Routing Mechanism for Multi-Channel Multi-Interface Ad-Hoc Networks

Supervisor

Professor Masayuki Murata

Author

Shinsuke Kajioka

February 13th, 2008

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Proposal, Implementation, and Evaluation of a QoS-aware Routing Mechanism for Multi-Channel Multi-Interface Ad-Hoc Networks

Shinsuke Kajioka

Abstract

Wireless ad-hoc network is one of promising technologies to make our social life safe, secure, and comfortable. Since a wireless ad-hoc network does not require any fixed infrastructure, e.g. routers, switches, and cables, it can be deployed in a variety of regions, such as a building, a town, a historic landmark, and a disaster-affected area. Among typical applications of wireless ad-hoc networks, real-time multimedia applications involving real-time video and/or audio transmission, such as video conferencing, VoIP (Voice over IP), and remote monitoring, put much burden on a network for their constant generation of delay-sensitive traffic against the limited network capacity. Without any control mechanism, a wireless ad-hoc network is easily congested and the perceived quality of applications considerably deteriorates. Therefore, we need QoS (Quality of Service) control mechanisms to accommodate real-time multimedia application while satisfying application QoS requirements. In this thesis, 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. Then, a source node reactively determines a logical path on which application QoS requirements are satisfied and packets are encapsulated so that they traverse the logical path toward a destination node. Through experiments on a simulator and a prototype, we confirmed that our mechanism could achieve the packet delivery ratio of about 95% at the end-to-end delay of about 10 msec in a grid network of 100 nodes by assigning three dedicated channels to real-time traffic and conducting logical routing. In addition, real-time traffic was more evenly distributed over the whole network.

Keywords

Ad-Hoc Network

QoS Routing

Multi-channel

Multi-interface

Contents

List of Figures

List of Tables

1 Introduction

Wireless ad-hoc networks are an IP-based network which is built in an ad-hoc manner without any fixed communication infrastructure 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 or difficult as in a historic landmark or when conventional communication infrastructure is destroyed such as in catastrophic disasters like earthquake. Wireless ad-hoc network is one of the major research and development areas in these years [1, 2, 3].

Typical bands used for a wireless ad-hoc network are 2.4 GHz to 2.5 GHz (Bluetooth, IEEE 802.11/11b/11g, and ZigBee) and 5.725 GHz to 5.875 GHz (IEEE 802.11a) and their transmission rate ranges from 250 kbps (ZigBee at 2.45 GHz) to 54 Mbps (IEEE 802.11g/11a) [4, 5, 6]. Packets transmitted over a wireless ad-hoc network include both of best-effort traffic generated by, for example, file transfer, e-mail, and Web and real-time traffic generated by, for example, video conferencing, VoIP (Voice over IP), and remote monitoring. Since the capacity of wireless link is limited and the effective bandwidth is much smaller for contention among nodes [7, 8], it is not trivial to accommodate real-time multimedia traffic in a wireless ad-hoc network. Especially, they require certain level of QoS (Quality of Service) guarantee or control in terms of delay, delay jitter, and packet loss.

Over the past several years, many studies have been devoted to QoS control in wireless ad-hoc networks [9, 10, 11, 12]. 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 (QoS-enhanced Optimized Link State Routing) decreases the packet loss rate about a half of the conventional OLSR by considering more appropriate metrics, i.e. delay and bandwidth, than the hop distance used in OLSR in route calculation [11]. The modification of IEEE 802.11 standard MAC protocol supports frame transmission over a multi-channel and multi-interface wireless ad-hoc network. A node switches wireless channels [13] or both of channels and interfaces [14, 15, 16, 17, 18, 19] in a hop-by-hop manner or a time-based manner, to reduce the packet loss and improve the network throughput. Another modification of IEEE 802.11 standard MAC protocol working on a single-hop, multi-channel, multi-interface, and multi-rate wireless networks associate an interface and the data rate [20]. According to the modification, it overcomes the performance degradation caused by rate adaptation where a node always uses the lowest data rate among the available data rate in communicating with neighbor nodes. It has also been recognized that the network capacity of a single-channel and multi-hop wireless network using the normal IEEE 802.11 standard MAC is $O(n/\sqrt{n})$ times as low as the channel bandwidth of the node [7], where n is the number of nodes using the same channel in the network. Although having multiple channels and multiple interfaces contributes to avoidance of competition and collision for a wireless channel, in [21], they noticed that channel switching at an interface introduced the switching delay and proposed a new method which classified interfaces to "fixed" interfaces and "switchable" interfaces to avoid the influence of switching delay. Fixed interfaces stay on certain channels for a longer period than switchable interfaces and they are used when a switchable interface is switching a channel.

Among QoS control methods, we focus on QoS-aware routing with a multi-channel and multi-interface technology for higher wireless capacity in this thesis. QoS-aware routing methods can be categorized into two types, i.e. proactive QoS-aware routing and reactive QoS-aware routing. A proactive QoS-aware routing method such as QOLSR [11, 22] collects and maintains up-to-date bandwidth information including delay and link utilization by exchanging control messages. Based on the bandwidth information, all nodes prepare a routing table for the whole or a part of a wireless ad-hoc network taking into account application QoS requirements. Then, a packet generated by an application is immediately transmitted to a destination on the prepared path with QoS guarantee or control. This mechanism implies that the transmission delay for the first packet can be minimized and QoS can be guaranteed as far as a corresponding path for the destination is prepared, but it requires much overhead in maintenance of bandwidth information.

On the contrary, in a reactive QoS-aware routing method, a path is established on demand in accordance with application QoS requirements and the current condition of a wireless ad-hoc network when the first packet is generated. Although it does not require any advance preparation which introduces much control overhead, the first packet must wait for path establishment to be sent to the destination node. In addition, the optimal path cannot always be established due to its ad-hoc and rather greedy mechanism. In general, proactive QoS-aware routing methods are preferred in the case that a wireless ad-hoc network consists of immobile nodes, where prepared paths are valid for a longer period. On the other hand, reactive QoS-aware routing methods are better suited to a dynamic or mobile wireless ad-hoc network where nodes move and a path dynamically changes.

In this thesis, we propose a QoS-aware routing mechanism for wireless ad-hoc networks, especially used for temporal communication vehicle at a festival or disaster-affected area. Our mechanism assumes a node equipped with multiple network interfaces, each of which a different wireless channel can be assigned to. More specifically, we consider that the number of interfaces and the number of available wireless channels are identical and channels are assigned to interfaces without overlap. 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, our mechanism employs OLSRv2 (OLSR version 2) [23] as the underlying physical routing protocol. It means that our mechanism works above the IP routing layer, i.e. network layer. The bandwidth information is embedded in control messages of OLSRv2 and propagated in the whole network in an efficient and effective way. On receiving a request for packet transmission from an application, a node determines a path to the destination. Therefore, our mechanism can be categorized into reactive routing from the timing that a path is established. Since a node has the complete information about the whole network, it is able to compute the optimal path to a destination node to satisfy application QoS requirements. However, the derived path, called a logical path, is different from the physical path to the destination established by the underlying OLSRv2. Therefore, packets are encapsulated so that it traverses the optimal path, called a logical path, virtually built on the physical routing network. In addition to this logical routing, each intermediate node chooses or switches a wireless channel for packet transmission in a hop-by-hop manner for efficient use of wireless channels and collision avoidance.

We first perform simulation experiments to evaluate the effectiveness of our proposal from viewpoints of packet delivery ratio, end-to-end delay, channel utilization, and node utilization, and then, we build a prototype and conduct practical experiments to verify the practicality. Through simulation experiments, we confirmed that our mechanism could achieve the packet delivery ratio of about 95% at the end-to-end delay of about 10 msec in a grid network of 100 nodes by assigning three dedicated channels to real-time traffic and conducting logical routing. In addition, we confirmed that real-time traffic was more evenly distributed over the whole network.

The rest of this thesis is organized as follows. First, we describe our proposal in Section 2. Next, we introduce system architecture of our proposal in Section 3. Then, we present simulation results in Section 4 and practical results in Section 5. Finally, we summarize the thesis and describe some future work in Section 6.

2 QoS-aware routing mechanism on wireless ad-hoc networks

In this section, we give an overview of our proposed mechanism and its target system for QoS-aware routing on wireless ad-hoc networks. There are three key points in our proposed mechanism. The first point is estimation of the available bandwidth; each node estimates the available bandwidth on its assigned wireless channels by checking the amount of transmitted and received packets. The second point is exchange of the bandwidth information by using OLSRv2; the bandwidth information estimated at a node is embedded into control messages of OLSRv2 and disseminated over the whole network by OLSRv2, so that all nodes have the bandwidth information about the other nodes. The third point is logical routing based on the information about the network topology and the available bandwidth; each node determines a logical path over a physical wireless ad-hoc network, so that the bandwidth can be efficiently used in the whole network.

2.1 Overview of our proposed mechanism and its target system and application

We consider a wireless ad-hoc network consisting of nodes equipped with K $(2 \leq K)$ wireless network interfaces. The same number K of wireless channels are 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. Because of its application scenario, we assume that nodes are immobile. At least, no node in the network moves while there is packet transmission from one node to another. Nevertheless, condition of wireless communication dynamically changes. In our proposed mechanism, one channel numbered 0 from K channels is reserved for best-effort traffic and called best-effort channel, 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 mechanism operates for proactive physical routing and bandwidth information dissemination.

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. The wireless network interfaces are numbered as 0, 1, and 2, respectively. There are three channels, 1, 6, and 11 of IEEE 802.11g available for the network. They are also numbered as 0, 1, and 2, respectively. The interface wlan0 is assigned to channel 1,

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

interface	channel	IP address–node 1	IP address–node 2	IP address–node 3
0 (wlan0)	0 (1(2.412 GHz))	192.168.0.1/24	192.168.0.2/24	192.168.0.3/24
1 (wlan1)	1 (6(2.437 GHz))	192.168.1.1/24	192.168.1.2/24	192.168.1.3/24
2 (wlan2)	2 (11(2.462 GHz))	192.168.2.1/24	192.168.2.2/24	192.168.2.3/24

wlan1 to channel 6, and wlan2 to channel 11. 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. Since wireless channels are assigned to different networks, channel switching can be easily done by changing a network address of a packet at a source node and intermediate nodes. It might be noticed that third octet of a network address is the same as the channel number assigned to the network interface for the network for easier operation in this example.

Each node always probes the usage of real-time channels and estimates the available bandwidth. The estimated available bandwidth is disseminated over the whole network by embedded on control messages, i.e. HELLO messages and TC (Topology Control) messages of OLSRv2 operating on the best-effort channel. In OLSRv2, all nodes obtain and maintain the complete view of the network topology by exchanging control messages. In our mechanism, all nodes additionally obtain and maintain the complete information about the available bandwidth on all nodes 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 traversing a so-called logical path. A logical path consists of logical links connecting a pair of nodes, which are not necessarily physically neighboring each other. Therefore, 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. When a packet to a new destination is generated by a real-time application, a node determines a logical path to its destination for the session. Since the node has the complete information about the network condition, it can derive the optimal path to satisfy application QoS requirements. For example in Figure 1, the node S is a source node and the node D is a destination node. The node S first considers the logical mesh topology on the physical network (Figure 1(b)). Each of logical links in the logical mesh topology is related to the available bandwidth on the physical path connecting the two ends of the logical link. Then, the node S finds the optimal path which satisfies application QoS requirements and some other metric if needed. In this example, the logical path S-B-D is chosen (Figure 1(c)). Since the physical path from the node S to the node D can be different from the optimal logical path as for example S-E-F-D, all packets belonging to the session for this destination are encapsulated indicating the first destination node B and the last destination node D, as shown in Figure 2 (in the node S). Encapsulated packets are first sent to the first destination node B through the physical path from the source node S to the node B, and then sent to the final destination node D from the node B (Figure 1(d)). In this case, the logical next hop node at the node S is the node B while the physical next hop node at the node S is the node A, i.e. the node S sent a packet to the node A, then the node A forwarded the packet to the node B. The intermediate node A of the logical path S-B only relays a received packet to the node B, which is regarded as the destination of the packet from the physical routing view point (Figure 2, node A). Real-time packets are transmitted on real-time channels. For efficient use of wireless bandwidth, each node chooses the real-time channel with the largest available bandwidth in forwarding a packet to a physical next hop node, which is determined in accordance with the OLSRv2 routing table. When a packet arrives at the logical intermediate node, it is encapsulated with a new header indicating the next logical hop node. Then it is sent out to the logical next hop node (Figure 2, node B).

In the following sections, we give detailed description about estimation of the available bandwidth, exchange of bandwidth information on OLSRv2, and logical routing.

2.2 Estimation of available bandwidth

There have been some studies on estimation of the available bandwidth on a wireless network. One way is to subtract the measured throughput in the MAC layer from the typical or ideal channel

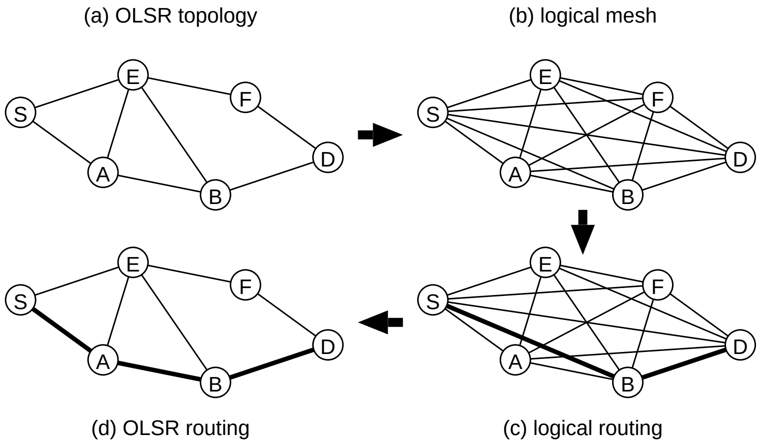


Figure 1: QoS-aware routing by our proposed mechanism.

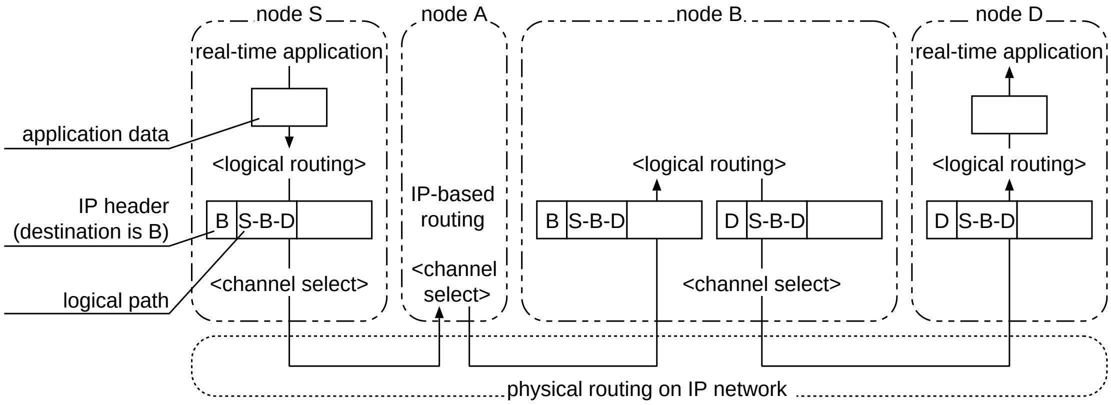


Figure 2: Packet processing in our proposed mechanism.

bandwidth [8], that is, (available bandwidth) = (channel bandwidth) − (measured throughput in MAC layer). Another way is to multiply the typical or ideal channel bandwidth by the ratio of the channel free time to the estimation interval [24], that is, (available bandwidth) = (ideal channel bandwidth) × (channel free time / estimation interval). In our proposal, for the sake of simplicity, the available bandwidth $B_k(c)$ of the channel c ($1 \leq c \leq K - 1$) on the node k is estimated by Equation 1, where $W(c)$ corresponds to the ideal capacity of the channel c, e.g. 54 Mbps for IEEE 802.11g, T is the estimation interval, and $B(c)$ is the total amount of data transmitted and received on the channel c at the node k in the estimation interval.

$$B_k(c) = W(c) - B(c)/T. \tag{1}$$

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

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

We should note here that we do not limit to the above estimation, but a node can use any of the above-mentioned estimations or any other in literatures.

2.3 Exchange of bandwidth information on OLSRv2

OLSRv2 is a proactive link-state routing protocol [23]. OLSRv2 reduces the overhead in flooding control messages for information dissemination by limiting nodes participating in flooding. They are called MPR (MultiPoint Relay). A control message generated at an MPR is broadcast in its range of radio signals. Among nodes receiving this message, only MPRs rebroadcast the message. A set of MPRs are chosen in a distributed manner aiming to have the smallest number of MPRs while keeping the connectivity.

Nodes in a wireless ad-hoc network exchange HELLO messages with neighboring nodes in the range of radio signals at regular HELLO intervals. A HELLO message consists of its validity time, originator address of the message, neighbor addresses of the originator, and some optional information. Based on the information in HELLO messages, a node maintains several tables, i.e. the Link Set consisting of the status of links with all neighbor nodes, the Neighbor Address Association Set consists of neighbor address list and validity time, the 2-Hop Neighbor Set consisting of next hop addresses for its two hop neighbors, and the MPR Selector Set consisting of

MPR selectors, which are nodes selecting the node as an MPR. In addition, an MPR generates and disseminates TC (Topology Control) messages at regular intervals. A TC message contains its validity time, originator address of the message, and addresses of its MPR selectors. On receiving a TC message, a node builds or updates the Topology Set consisting of MPRs, its MPR selectors, sequence number, and validity time, and the Attached Network Set consisting of network address, prefix length, gateway address, sequence number, and validity time. The routing table, called Routing Set, is built and maintained when any of the Link Set, the Neighbor Address Association Set, the 2-hop Neighbor Set, or the Topology Set changes. The Routing Set consists of destination address, next hop address for 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 entrained in HELLO and TC messages by adding the extended field in the form of TLV (Type Length Value) block. On receiving control messages, a node builds or updates a new table, called the Extended Topology Set to maintain the bandwidth information in addition to the above-mentioned tables.

2.4 Logical routing based on bandwidth information

As explained in section 2.1, a source node determines a logical path on a logical mesh network taking into account the network condition. On receiving the first packet to a new destination, the node generates a logical mesh network. The logical link between the node i and the node j in the logical mesh network is associated with the available bandwidth $B(i, j)$. The available bandwidth $B(i, j)$ is given as the minium of the available bandwidth among all physical links on the shortest path between the node i and the node j. The available bandwidth of the 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 $B(S, B) = \min(\min(B_S, B_A), \min(B_A, B_B))$. 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 which satisfies application QoS requirements. There have been several studies for derivation of the optimal path [25, 26]. Although they are also useful in our mechanism, in this thesis we use a simple heuristic algorithm described in the following. 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 to generate an unnecessarily long path and shorten the calculation time. Then, for each of logical paths in the set, a node derives the available bandwidth 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.

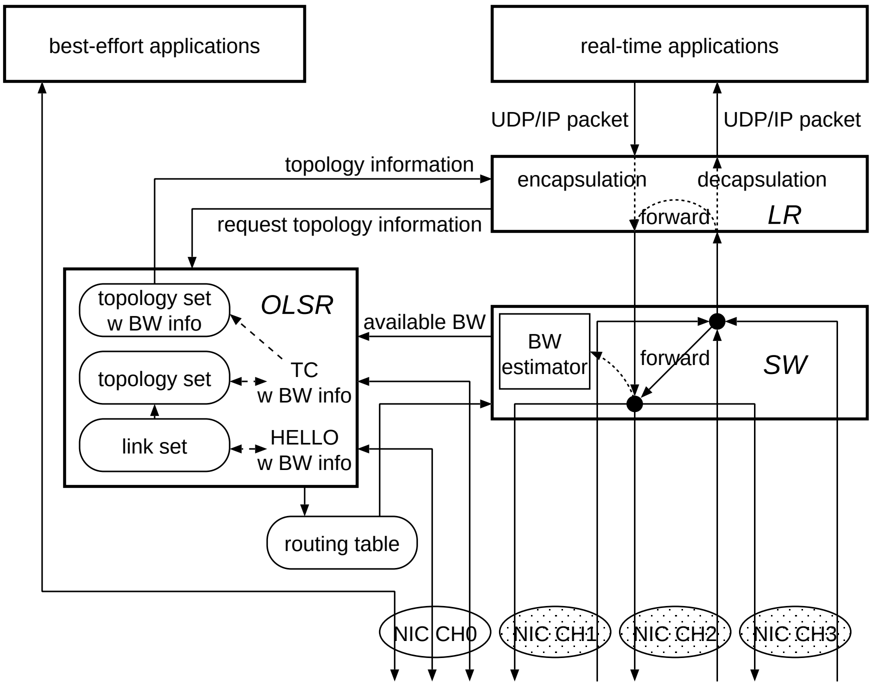


Figure 3: Module components of proposed mechanism.

3 System architecture

In this section, we describe how our QoS-aware routing mechanism proposed in the previous section is implemented on a wireless ad-hoc network system. We first give the overview of the system and then explain details of three key modules, i.e. the logical QoS-aware routing module (LR), the switching module (SW), and the OLSR module (OLSR).

3.1 Overview of the system

Figure 3 shows the module components of our proposed mechanism. In this figure, a node has four network interfaces and we assign channel 0 for best-effort traffic and channel 1, 2, and 3 for real-time traffic.

UDP/IP packets generated by a real-time application are first processed by the LR. The LR builds a logical path for the first packet of a new session. Packets are encapsulated by an LR header indicating addresses of intermediate nodes of the logical path, so that it traverses the logical path on the physical network operated by OLSRv2. Finally, the LR passes the encapsulated packet to the SW. On receiving a packet from the LR, the SW determines the physical next hop node in the physical network for the logical next hop node. Then, the SW emits the packet destined to the logical next hop node though a network interface for a channel with the maximum available bandwidth among network interfaces for real-time traffic.

When the SW receives a packet from the network, the SW investigates an LR header to identify the logical next hop node of the packet. If the logical next hop node is the node itself, the SW forwards the packet to the LR. Otherwise, the SW sends the packet to the physical next hop node on the physical path toward the logical next hop node. 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 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, estimates the available bandwidth of each channel by using Equation 1, determines the available bandwidth of the node by using Equation 2, and reports the result to the OLSR.

The OLSR manages the physical network by exchanging HELLO and TC messages on the best-effort channel. The OLSR obtains the information about the available bandwidth of the node from the SW. The OLSR generates and exchanges HELLO messages embedded the information about its available bandwidth with neighboring nodes. On receiving a HELLO message, the OLSR builds and updates the Link Set. 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 to the whole network. On receiving a TC message, a node builds and updates the Topology Set, the Attached Network Set, the Routing Set, and the Extended Topology Set. Then, the OLSR updates the IP routing table if the Routing Set is updated. On receiving a request, the OLSR provides the LR with the Extended Topology Set.

3.2 Logical QoS routing module (LR)

The logical QoS routing module (LR) on a source node determines a QoS-aware logical path to a destination node.

On receiving a UDP/IP packet from a real-time application, the LR first investigates its IP header and identifies a session from the source-destination IP addresses and ports or the session

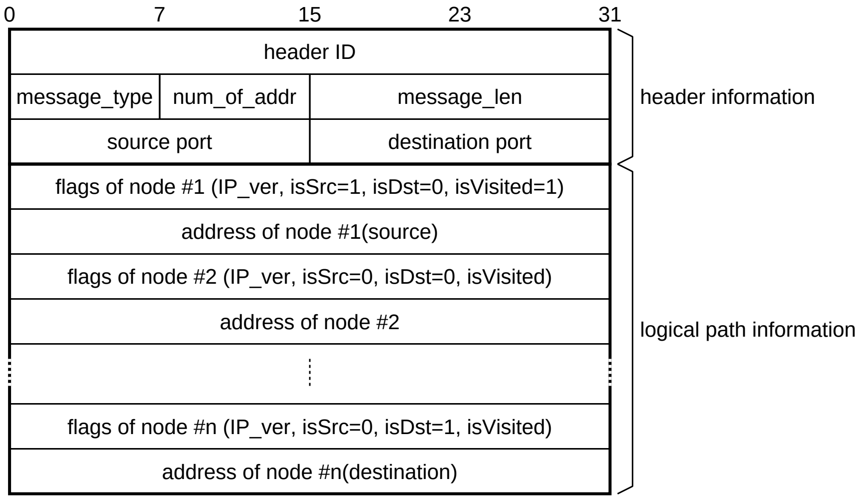


Figure 4: LR header format.

identifier specified by the application. 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 that is updated when the entry is made or referred to, and the corresponding LR header information. An entry is added to the table when the first packet for a new session arrives. If the packet is for a new session or the corresponding entry with the timestamp more than 30 seconds old exists in the table, the LR determines the logical path for the session based on the topology information and the bandwidth information that the LR periodically retrieves from the OLSR at regular intervals of, for example, 5 seconds. This interval is the same as the interval of broadcasting TC messages in OLSRv2. The LR constructs an LR header for the packet. The constructed LR header is stored in the session management table. On the other hand, if there has been already existed an entry for the session in the session management table and less than 30 seconds have passed since it is made or referred to, the LR header is obtained from the entry. Then, the packet is encapsulated by an LR header, so that it traverses the logical path by being relayed through the physical network based on the physical routing. Finally, the LR passes the encapsulated packet to the SW.

The structure of an LR header is illustrated in Figure 4. An LR header consists of header information and logical path information. The header information field contains header identifier, message type for the IP version and the LR version, the number of addresses in the logical path information field, message length including the LR header, UDP/IP header, and payload, application port number of the source, and application port number of the destination. The logical path information field contains pairs of the flag field indicating whether the following address is for the source node, the destination node, or an intermediate node and whether the node is visited or not, and IP address. The logical path information field starts from the source node, intermediate nodes on the logical path, to the destination node.

When the LR receives a packet from the SW, the LR checks the flag field for its address in the LR header. If it is marked as the destination, the LR removes the LR header from the packet and passes it to the corresponding real-time application. Otherwise, the LR sets the flag field as visited and sends the packet back to the SW.

3.3 Switching module (SW)

The switching module (SW) observes the channel usage, transmits real-time packets on the real-time channel with the most available bandwidth, and reports the bandwidth information to the OLSR.

The BW estimator module of the SW evaluates the amount of transmitted and received data through real-time interfaces. The BW estimator has the Channel Information table which consists of sets of the ideal channel bandwidth, the total amount of transmitted/received bytes since the SW and the BW estimator started, and the amount of transmitted/received bytes since the last estimation. From the Channel Information table, it estimates the available bandwidth B_k of the node k by using Equations 1 and 2 and informs the OLSR of the estimation at regular intervals of 2 seconds, which is the same as the HELLO interval of OLSRv2.

When the SW receives a packet from the LR, it first identifies the logical next hop node from an LR header. The top node whose flag field is not set as visited in the logical path information of the LR header is the logical next hop node of the packet. Next, the SW investigates the routing table of the system, which is maintained by the OLSR, to identify the physical next hop node for the logical next hop node. To balance the usage of channels, the SW chooses the least used channel. The IP address of the physical next hop node can be obtained by combining the host address part of the physical next hop node in the routing table and the network address part of the wireless interface for the real-time channel with the maximum available bandwidth. Then, the SW emits the packet destined to the logical next hop node though the appropriate network interface.

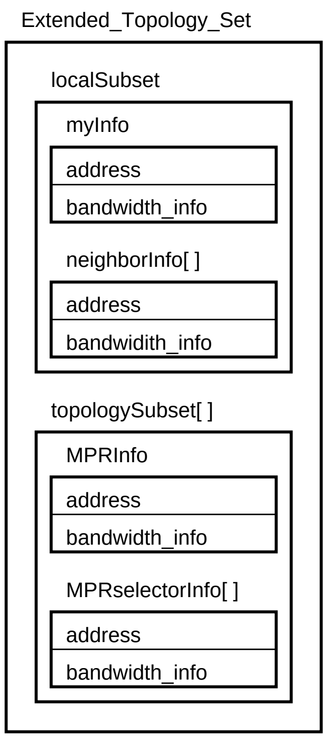


Figure 5: The structure of the Extended Topology Set.

On the other hand, when the SW receives a packet from a neighbor node through a network interface, it first identifies the logical next hop node of the packet as in the same way as the above. If the logical next hop node is the node itself, the SW passes the packet to the LR. Otherwise, it forwards the packet to the physical next hop node as in the same way as the above.

3.4 OLSR module (OLSR)

The OLSR module performs all functions of the OLSRv2 routing protocol with modification for our proposal. The OLSR maintains the Extended Topology Set, which deposits the original topology information of OLSRv2 and additionally the bandwidth information (Figure 5). The reason why we introduce a new table instead of adding a new field to the existing Topology Set is easier implementation.

The OLSR exchanges HELLO messages with neighboring nodes in the range of radio signals at regular HELLO intervals to construct and maintain bidirectional links with each other. When a node receives a HELLO message, the OLSR updates its local information described in Section

2.3. The OLSR selects MPRs for efficient communication with other nodes based on the collected information. The OLSR of a node chosen as an MPR generates TC messages and disseminates them to the whole network. When the OLSR receives a TC message, it updates its Topology Set that contains pairs of an address of MPR selector, as a destination address, and the address of its MPR, i.e. the originator of the TC message, as the last hop node to the destination. It also updates its Routing Set that contains pairs of an address of MPR selector as a destination address and an address of a neighbor node, i.e. the sender of the TC message, as the next hop node for the destination. Differently from the Topology Set, an address of a node is accompanied with the bandwidth information on the node.

The Figure 5 shows the structure of the Extended Topology Set, where the localSubset is a subset of a node and its neighbors, the topologySubset is subsets of MPRs in the Topology Set and its MPR selectors.

The OLSR maintains information about the available bandwidth of the node from SW. In generating a HELLO message, the OLSR adds the information about its available bandwidth to the Address Block TLV of the originator. On the other hand, when the OLSR receives a HELLO message, the address of the originator and its available bandwidth extracted from the Address Block TLV of the originator are stored at the Extended Topology Set. In generating a TC message, the OLSR add the information about its available bandwidth to the Address Block TLV of the originator and the information about the available bandwidth of its MPR selectors to the Address Block TLV of them. When the OLSR receives a TC message, the address of the originator and its available bandwidth extracted from the Address Block TLV of the originator are stored as the key at the topological subset of the Extended Topology Set and prepends the MPR selector and bandwidth information to the originator's subset.

4 Simulation experiments and discussions

In this section, we first evaluate the performance of our proposal through simulation experiments. We used QualNet 4.0 [27], which is a packet-based network simulator supporting many well-known, standard, and even experimental protocols and models in its libraries. QualNet 4.0 follows the OSI layer architecture in protocol implementation.

We based our OLSR module on the code provided by Niigata University OLSRv2 project [28] with some modifications for supporting our proposed mechanism. The modification includes reception of the bandwidth information from the SW, reception of a request for the topology information from the LR, transmission of the topology information to the LR, management of an Extended Topology Set, and management of control messages embedding the bandwidth information.

In implementing the LR and SW modules to QualNet 4.0, we needed to customize the socket API and the memory allocation API to communicate with other modules already implemented in QualNet 4.0.

4.1 Simulation settings

We built a grid network consisting of 100 nodes in the $1{,}000 \times 1{,}000\,\mathrm{m}^2$ region as shown in Figure 6. In the figure, open circles correspond to nodes, and nodes do not move. The range of radio signals is set at 289 m at a maximum. We assigned force wireless network interfaces with omni-directional antenna per node, ch1 (2.412 GHz), ch6 (2.437 GHz), and ch11 (2.462 GHz) assigned for real-time channels and ch14 (2.484 GHz) assigned for best-effort channel. A node can communicate with eight nodes in the diameter of 153 m at the rate of 54 Mbps as lines for bidirectional links show. Radio signals transmitted by a node can interfere 24 nodes in the diameter of 289 m. Since we set the link speed to 54 Mbps statically, a node cannot communicate with another node whose distance is more than 153 m. We used the free space path-loss model with no shadowing and no fading. IP version 4 was used in simulation experiments. A FIFO buffer at IP layer has the capacity of 50,000 bytes. For OLSRv2, we set intervals of HELLO and TC messages at 2 seconds and 6 seconds.

As an application, we assumed VoIP traffic which requires low delay, small delay jitter, and as high packet delivery ratio as possible. A source node generated packets of 172 bytes consisting of

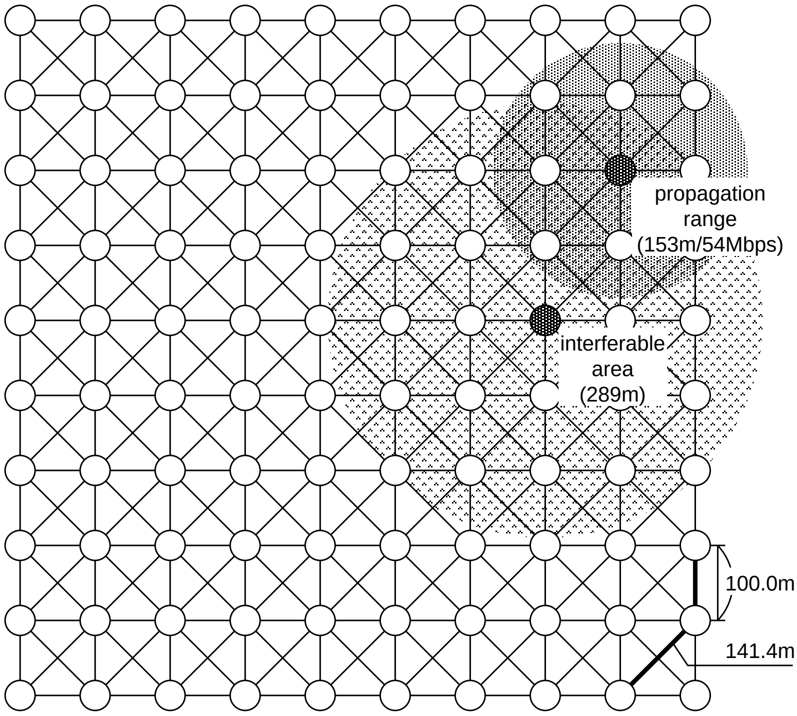


Figure 6: Network topology used in simulation experiments.

voice data of 160 bytes and RTP header of 12 bytes every 20 msec, i.e. 64 kbps CBR traffic. We measured the delay from a source to a destination and the delay jitter defined in RFC 3550 [29]. We initiated 80 sessions between randomly chosen pairs of a source node and a destination node at randomly chosen time from 30 to 90 seconds in simulation time. Each session lasts for 60 seconds. A simulation run was terminated at 155 seconds in simulation time after all packets had reached to the destination node. We considered 10 traffic patterns, i.e. 10 sets of 80 source-destination pairs and their starting time. For each of the traffic patterns, we conducted 10 simulation runs by changing a random seed. Therefore, the results shown in section 4.3 are averaged over 10 results for each traffic pattern.

4.2 Simulation scenarios

Our proposal constructs a logical path over the physical network operated by OLSRv2 and real-time packets are transmitted on channels dedicated to real-time traffic. To evaluate the effectiveness of our proposal, we consider five scenarios different in the number of available channels, their

Table 2: Simulation conditions.

scenario	the number of channels	logical routing	channel switching
1	1	N/A	N/A
2	2	off	at source
3	4	off	at source, at random
4	4	off	hop-by-hop
5	4	on	hop-by-hop

usage, and the way of channel selection. They are summarized in Table 2.

In the scenario 1, only one channel is available and our mechanism is not used. All of OLSRv2 control messages, best-effort traffic, and real-time traffic are accommodated into the single channel.

In the scenario 2, two channels are available. One channel is assigned to best-effort traffic and the other is to real-time traffic. The SW is used at a source node to send real-time packets on the real-time channel. By channel separation, the packet delivery ratio is expected to be higher than that of the scenario 1, since the decreased loss probability of OLSRv2 control packets reduces loss of real-time packets for missing a path to a destination.

In the scenario 3, four channels are available and three among them are assigned to real-time traffic. A source node transmits packet on a randomly chosen real-time channel. The channel assigned to the session does not change throughout the network and the session. Because of the increased network capacity for using multiple channels, QoS provided for real-time traffic will be enhanced in comparison to the scenario 2.

The scenario 4 is different from the scenario 3 by choosing the most unused real-time channel in sending real-time packets at a source node and intermediate nodes. Since real-time channels will be evenly used, we can expect further improvement of QoS, i.e. the packet delivery ratio, the delay, and the delay jitter. The scenarios 3 and 4 are similar to the proposal in [18].

Finally, in the scenario 5, the logical routing is performed to let real-time packets through the optimal path from the viewpoint of the bandwidth usage. The packet delivery ratio is expected to become much higher at the sacrifice of increase in the delay for taking a longer physical path to a destination node. We can also expect the number of sessions that a wireless ad-hoc network can accommodate increases for balancing the load among nodes by the logical routing.

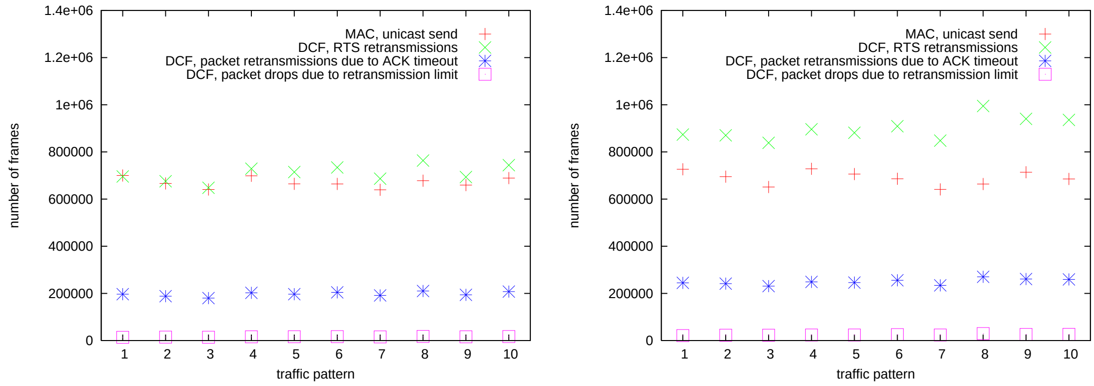
The size of the MAC frame is 214 bytes in the scenario 1, 262 bytes in the scenarios 2, 3, and 4, and 262–278 bytes in the scenario 5.

4.3 Simulation results and discussions

First, we compare the scenarios 1 and 2. All of control packets of OLSRv2 and data packets of real-time applications are accommodated in a single channel in the scenario 1, whereas they have separated channels in the scenario 2.

Figure 7 shows the simulation results related to the MAC layer. The horizontal axis corresponds to the traffic pattern numbered in order of the average hop counts of physical paths. The vertical axis indicates the total number of MAC frames averaged over 10 simulation experiments. In the figure, we consider four measures. The top most measure with a cross mark corresponds to the total number of MAC frames that are successfully sent to a next hop node by unicast transmission, i.e. data frames. The second measure with an x-mark shows the total number of retransmitted RTS (Ready To Send) frames of DCF (Distributed Coordination Function). Before packet emission, a sender broadcasts an RTS MAC frame to ask a receiver whether it is possible to send a data frame. When a receiver is ready to receive the frame, it broadcasts a CTS (Clear To Send) MAC frame to allow the sender to send the frame and inform the other nodes in the range of radio signals of frame transmission. By the RTS-CTS negotiation, collision can be avoided. An RTS frame is retransmitted when a sender does not receive a corresponding CTS from a receiver for some reasons, e.g. loss of RTS or CTS frame caused by radio congestion and the channel busy. Therefore, the number of RTS retransmissions indicates the radio congestion level. The third measure with a star mark is the number of retransmitted data frames due to ACK timeout caused by loss of data or ACK frame. Therefore, the number of retransmitted data frames also indicates the radio congestion level. The last measure with a square mark is the number of dropped data frames due to retransmission limit, 7 in this simulation, on MAC.

In Figure 7, although the number of RTS retransmissions increases by about 3% in the scenario 2 from the scenario 1, the number of data frames sent to a next hop node increases by only about 3%. The reason for this is that the total amount of real-time traffic, 80×64 kbps$=5.12$ Mbps exceeds the capacity of a wireless channel for competition among 100 nodes.



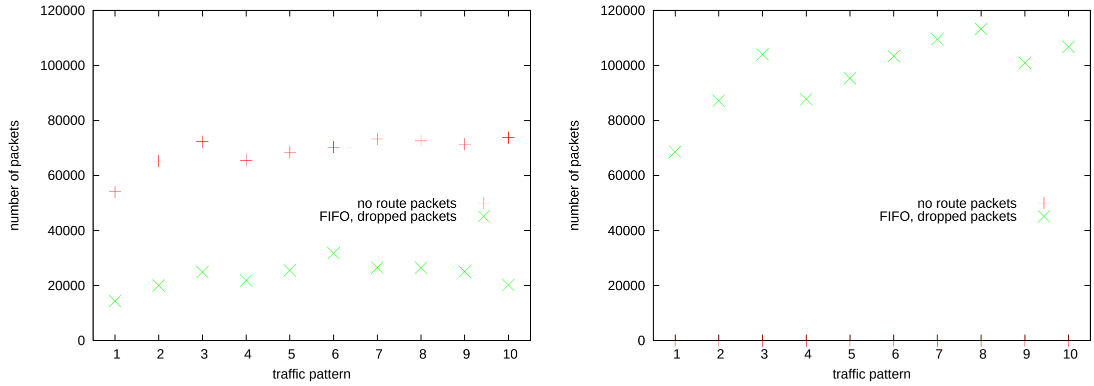
(a) Scenario 1: Best-effort and real-time share one channel.

(b) Scenario 2: 1 best-effort and 1 real-time channels.

Figure 7: Comparison of scenarios 1 and 2 in MAC-level performance.

Since channels for control packets of OLSRv2 and data packets of a real-time application are separated in the scenario 2, the probability of loss of control packets becomes small, and therefore, the probability that a path to a destination is successfully prepared increases. To see this effect, we show the total number of lost packets in Figure 8. The total number of data packets discarded at a source node or intermediate nodes for missing the routing information to a destination is shown by a cross mark and the total number of control and data packets dropped at a FIFO queue is shown by a x-mark. As shown in the figure, about 54,000 to 73,800 packets against 240,000 packets, i.e. 23% to 31% are lost in the scenario 1. Since control packets of OLSRv2 are transmitted by broadcasting, collision avoidance such as IEEE 802.11 DCF cannot be applied. In the scenario 1, control packets of OLSRv2 were lost for collisions with control packets and data packets and paths for destinations were not well prepared. On the contrary, by separating channels, the scenario 2 did not lose any data packets for missing path. However, the total number of packets dropped at a FIFO queue is larger in the scenario 2 than in the scenario 1.

Finally, results on the average end-to-end packet delivery ratio, the average delay, and the average delay jitter are shown in Figure 9. The left vertical axis is for the average end-to-end packet delivery ratio and the right one is for the average delay and the average delay jitter. The end-to-end packet delivery ratio can be defined in two ways. One is the end-to-end packet delivery ratio at the CBR application, shown by a cross mark and the term "sequencing at APP." The

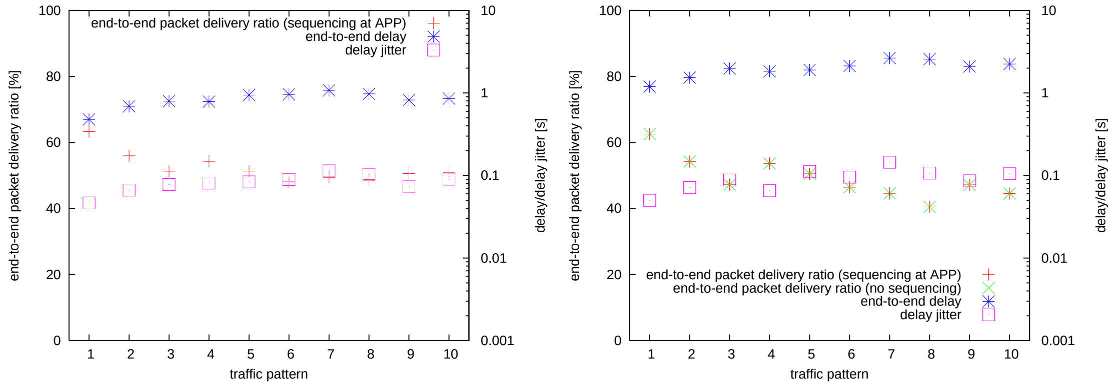


(a) Scenario 1: Best-effort and real-time share one channel.

(b) Scenario 2: 1 best-effort and 1 real-time channels.

Figure 8: Comparison of scenarios 1 and 2 in packet-level performance.

CBR application considers the sequence number of received packets. When the CBR application receives out-of-order packets, it discards them. The other one is the end-to-end packet delivery ratio at the LR module, shown by an x-mark and the term "no sequencing." Out-of-order packets are also taken into account in this measure. Since the scenario 1 did not have the LR, the end-to-end packet delivery ratio without sequencing is not shown in Figure 9. We can see that the average delay increases and the packet delivery ratio decreases by separating control and data channels.



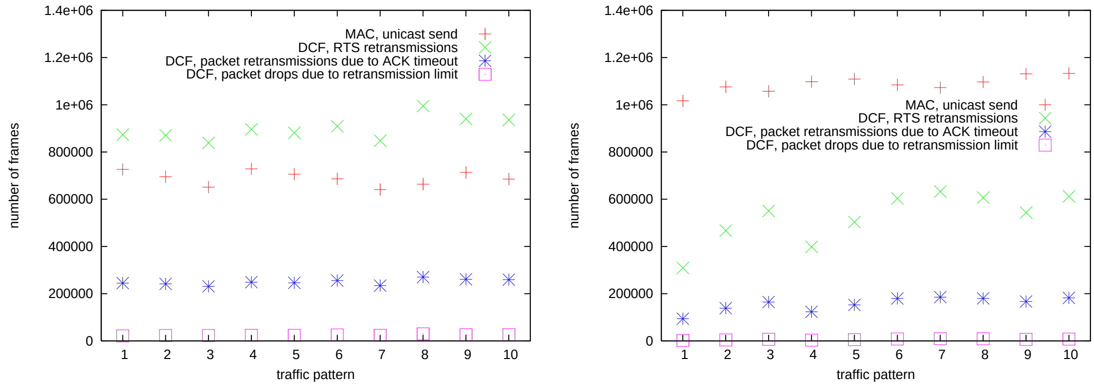
(a) Scenario 1: Best-effort and real-time share one channel.

(b) Scenario 2: 1 best-effort and 1 real-time channels.

Figure 9: Comparison of scenarios 1 and 2 in application-level performance.

Next, we compare the scenarios 2 and 3 in Figures 9, 10, and 11. In the scenario 3, the number of channels assigned to real-time traffic is increased from one to three. A real-time channel for a session is chosen at random at a source node. In Figure 9, it is shown that the total number of successfully sent MAC frames in the scenario 3 is 1.40 to 1.67 times larger than that in the scenario 2, the total number of data frame retransmissions in the scenario 3 is 0.35 to 0.66 times smaller than that in the scenario 2, and the total number of dropped frames in the scenario 3 is 0.38 to 0.79 times smaller than that in the scenario 2. In addition, we see the decrease in the total number of RTS retransmissions, the total number of data frame retransmissions, and the total number of dropped frames. These performance improvements are for having three channels for real-time traffic.

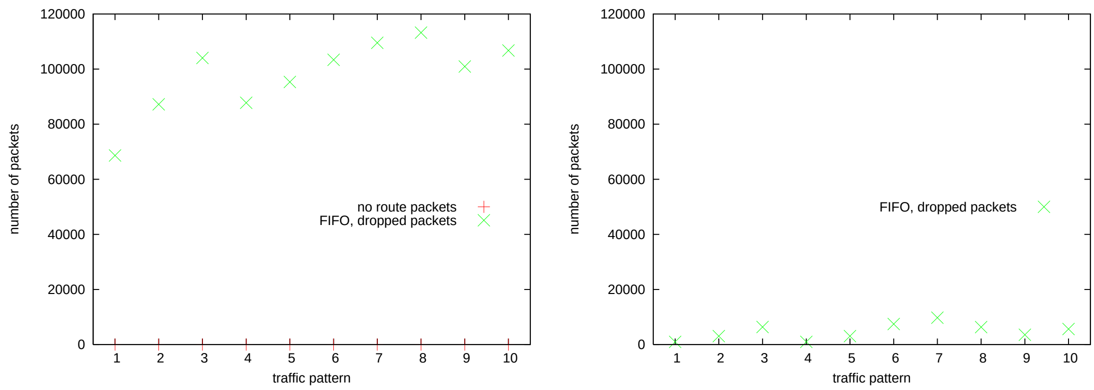
Also in Figure 11, we see that the total number of packets dropped at a FIFO queue considerably decreases to the loss probability of 0.4% to 4.1%. The improvement in the packet loss is more than 90% from the scenario 2. Furthermore, in Figure 12, the packet delivery ratio is as high as 92% to 99%. The delay and the delay jitter become ten times smaller than those in the scenario 2. The maximum delay in the scenario 3 was 210 msec, which satisfies the requirement of Class C quality of ITU-T Recommendation G.144, i.e. 400 msec. However, it is still larger than the requirement of Class B, i.e. 150 msec, and that of Class A, i.e. 100 msec.



(a) Scenario 2: 1 best-effort and 1 real-time channels.

(b) Scenario 3: 1 best-effort and 3 real-time channels (channel selection at source).

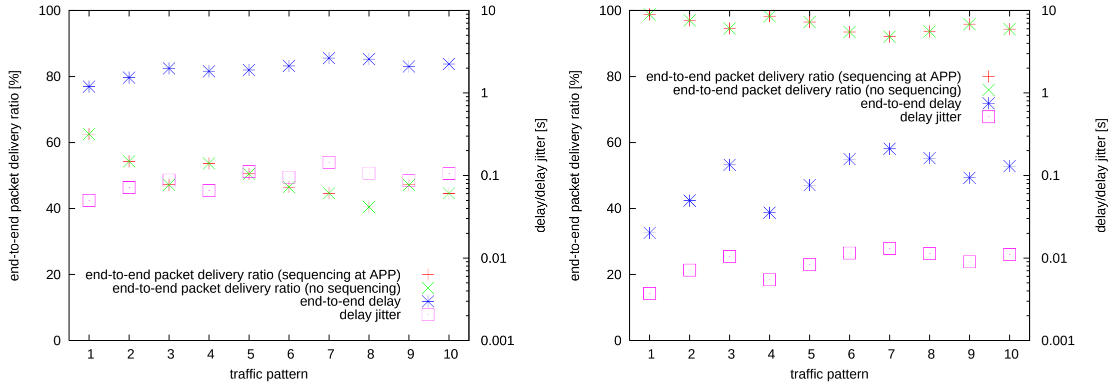
Figure 10: Comparison of scenarios 2 and 3 in MAC-level performance.



(a) Scenario 2: 1 best-effort and 1 real-time channels.

(b) Scenario 3: 1 best-effort and 3 real-time channels (channel selection at source).

Figure 11: Comparison of scenarios 2 and 3 in packet-level performance.

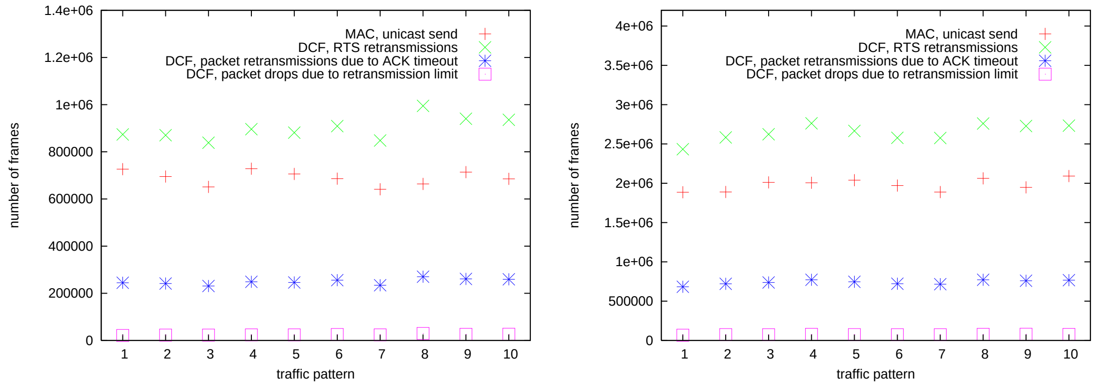


(a) Scenario 2: 1 best-effort and 1 real-time channels.

(b) Scenario 3: 1 best-effort and 3 real-time channels (channel selection at source).

Figure 12: Comparison of scenarios 2 and 3 in application-level performance.

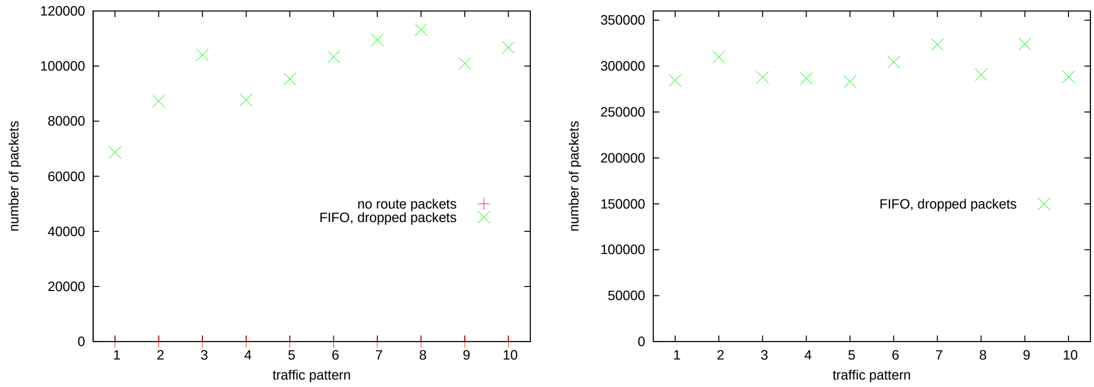
Now, we conduct additional experiments to compare the scenarios 2 and 3. We increased the number of sessions in the scenario 3, so that the MAC, packet, and application-level performance become in the same range of the scenario 1 as shown in Figures 13, 14, and 15. It might be noticed that the range of vertical axes in Figures 13(b) and 14(b) is three times as high as that in Figures 13(a) and 14(a), because a wireless ad-hoc network in the scenario 3 accommodated 240 sessions to reach the same level performance as in the scenario 2.



(a) Scenario 2: 1 best-effort and 1 real-time channels.

(b) Scenario 3: 1 best-effort and 3 real-time channels (channel selection at source, 240 sessions).

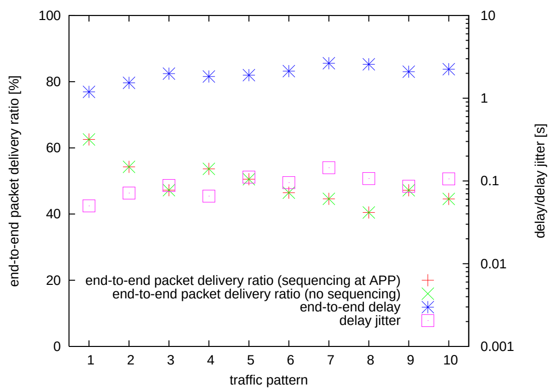
Figure 13: Comparison of scenarios 2 and 3 (240 sessions) in MAC-level performance.



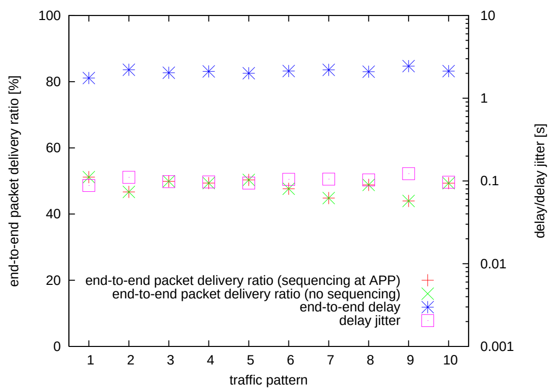
(a) Scenario 2: 1 best-effort and 1 real-time channels.

(b) Scenario 3: 1 best-effort and 3 real-time channels (channel selection at source, 240 sessions).

Figure 14: Comparison of scenarios 2 and 3 (240 sessions) in packet-level performance.



(a) Scenario 2: 1 best-effort and 1 real-time channels.



(b) Scenario 3: 1 best-effort and 3 real-time channels (channel selection at source, 240 sessions).

Figure 15: Comparison of scenarios 2 and 3 (240 sessions) in application-level performance.

Then, we compare the scenarios 3 and 4 in Figures 16, 17, 18, and 19. In these figures, the number of sessions was set at 80 again. The scenario 4 is different from the scenario 3 in choosing a real-time channel with the highest available bandwidth in sending a packet at a source node and intermediate nodes. In Figure 16, the total number of RTS retransmissions increases by about 14%, the total number of retransmitted data frames increases by about 20%, and the total number of dropped data frames increases by about 12% in the scenario 4 from the scenario 3, whereas the total number of data frames successfully sent increases by about only 1%. In Figure 17, we see the slight decrease in the total number of packets dropped at a FIFO. To explain this, we illustrate the distribution of channel usage, in Figure 18. Each cell in the figure corresponds to a node at that location. The sum of indexes on horizontal and vertical axes corresponds to the node identifier. A color of a cell corresponds the total number of MAC frames transmitted at the node. The lighter the color is, the larger the number is. It can be noticed that the color, or the lightness of cells, is different among channels in the scenario 3 in Figures 18(a), 18(c), and 18(e). Since a source node randomly chooses a real-time channel for a session, the channel usage becomes no uniform among channels. In addition, the channel usage in the whole network is also no uniform. On the contrary, in Figure 18(b), 18(d), and 18(f), the color looks similar among channels and the color is more homogeneous among cells in each of the figures. By evenly use wireless channels, the number of packets dropped at a FIFO queue decreases.

To quantify the balanced channel usage, we introduce following three metrics. The first one is the variance of transmitted data frames per channel, derived by the following equation.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(\bar{x} - x_i)^2, \tag{3}$$

where n is the number of data, $\bar{x}$ is the average of data, and x_i is the raw data. The smaller value of the variance indicates more balance use of wireless channels. The average value of variance of transmitted data frames per channel is 8,880,000 in the scenario 3 while it is 79,300 in the scenario 4.

The second one is the fairness index per channel derived by the following equation.

$$f = \frac{(\sum_{i=1}^{n} x_i)^2}{n\sum_{i=1}^{n} x_i^2}. \tag{4}$$

The fairness index of 1 means that wireless channels are used equally. The average value of fairness index of transmitted data frames per channel is 0.60 in the scenario 3 and it is 0.99 in the scenario 4.

The third one is PSNR (the phrase peak signal-to-noise ratio). PSNR is most commonly used measure for assessment of compressed image quality. For two images I and K of $m \times n$ pixels with the maximum pixel value of MAX_I, PSNR is derived by the following equations.

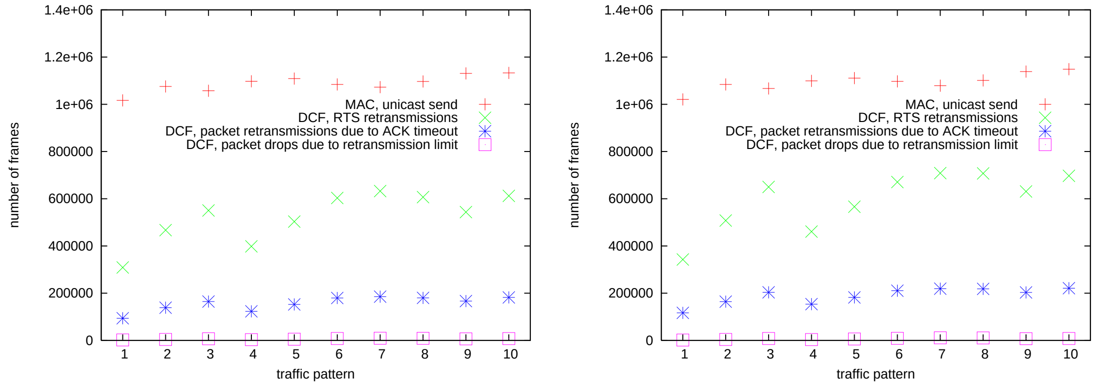
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}||I(i,j) - K(i,j)||^2 \tag{5}$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right). \tag{6}$$

The higher PSNR means the higher similarity between two images. PSNR between Figures 18(a) and 18(c) is 12.6, and that between Figures 18(a) and 18(e) is 12.7, while PSNR between the Figures 18(b) and 18(d) is 31.7, and that between the Figures 18(b) and 18(f) is 35.4.

From these results, we can conclude that the scenario 4 effectively uses wireless channels by distributing real-time traffic evenly among channels.

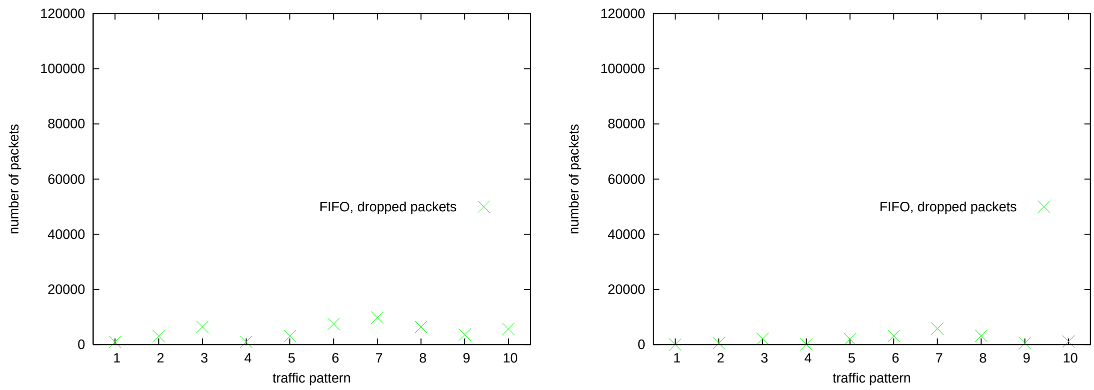
Finally in Figure 19, it is shown that the delay decreases in order of magnitude with the slight increase in the delay jitter in the scenario 4 from the scenario 3. The average of number of end-to-end packet delivery ratio in the scenario 4 decreases by 2% to 30%. The maximum delay in the



(a) Scenario 3: 1 best-effort and 3 real-time channels (source channel selection).

(b) Scenario 4: 1 best-effort and 3 real-time channels (hop-by-hop channel selection).

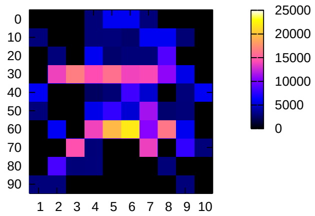
Figure 16: Comparison of scenarios 3 and 4 in MAC-level performance.



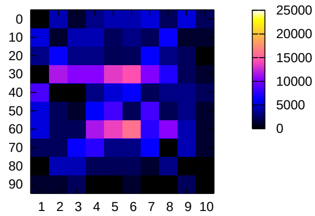
(a) Scenario 3: 1 best-effort and 3 real-time channels (source channel selection).

(b) Scenario 4: 1 best-effort and 3 real-time channels (hop-by-hop channel selection).

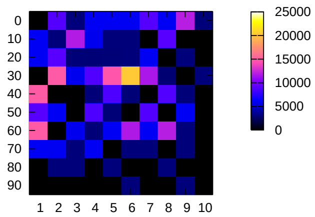
Figure 17: Comparison of scenarios 3 and 4 in packet-level performance.



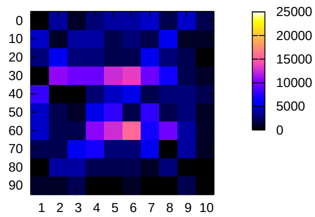
(a) Scenario 3: Source channel selection, channel 1.



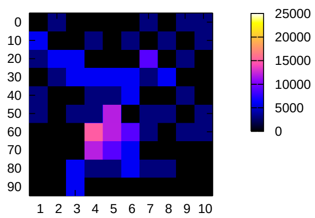
(b) Scenario 4: Hop-by-hop channel selection, channel 1.



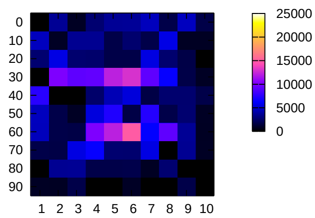
(c) Scenario 3: Source channel selection, channel 2.



(d) Scenario 4: Hop-by-hop channel selection, channel 2.

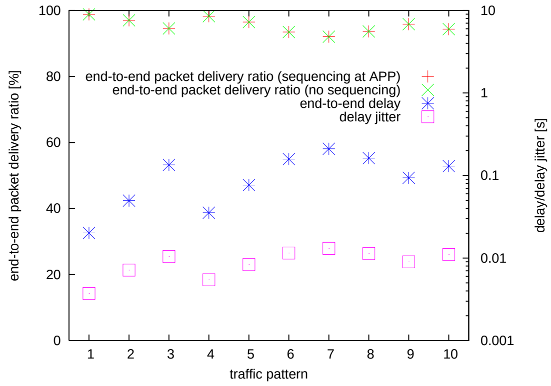


(e) Scenario 3: Source channel selection, channel 3.

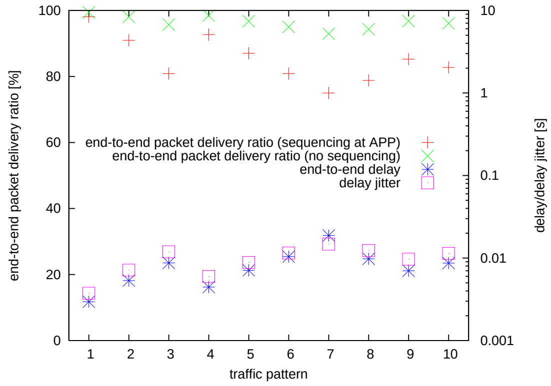


(f) Scenario 4: Hop-by-hop channel selection, channel 3.

Figure 18: Comparison of scenarios 3 and 4 in channel usage utilization.



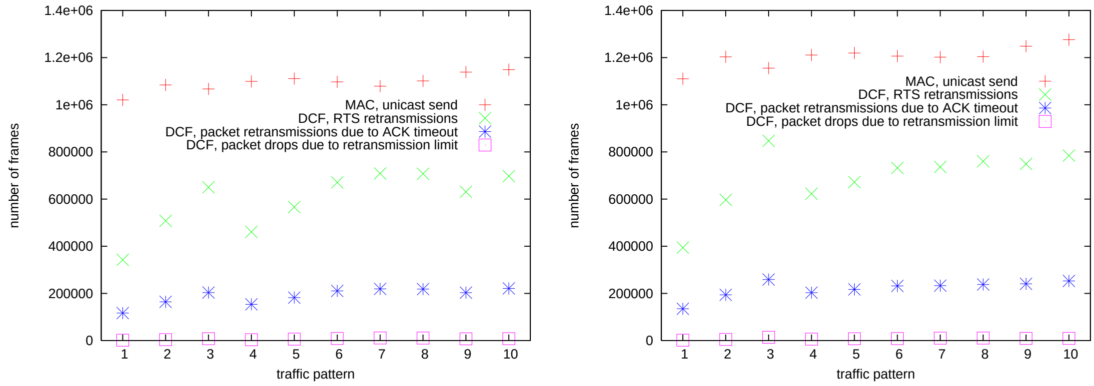
(a) Scenario 3: 1 best-effort and 3 real-time channels (source channel selection).



(b) Scenario 4: 1 best-effort and 3 real-time channels (hop-by-hop channel selection).

Figure 19: Comparison of scenarios 3 and 4 in application-level performance.

scenario 4 was 18 msec, which satisfies the requirement of Class A quality of ITU-T Recommendation G.144, i.e. 100 msec.



(a) Scenario 4: 1 best-effort and 3 real-time channels (hop-by-hop channel selection).

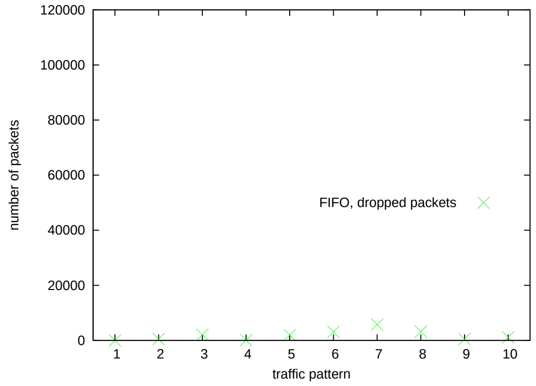
(b) Scenario 5: 1 best-effort and 3 real-time channels (logical routing).

Figure 20: Comparison of scenarios 4 and 5 in MAC-level performance.

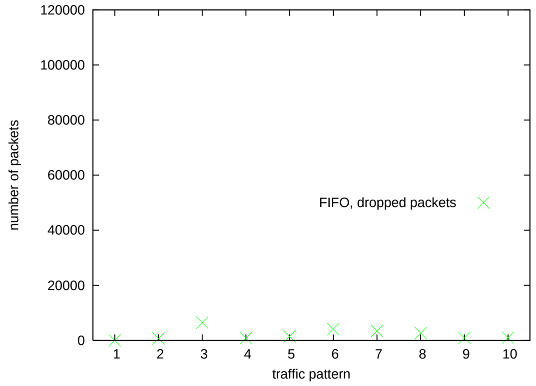
At last, we compare the scenarios 4 and 5 in Figures 20, 21, 22, and 23. The scenario 5 is different from the scenario 4 by logical routing. In Figure 20, the total number of transmitted data frames is 8% to 11% up, the total number of RTS retransmissions increases by 4% to 35% and the average is 17%, the total number of retransmitted data frames increases by 6% to 32% and the average is 17%, and the total number of dropped data frames is 0.89 to 1.63 times greater than that of in the scenario 4 and the average is 1.24. Finally, in Figure 21, the average of total number of packets dropped at a FIFO in the scenario 5 increases about 20%.

Such degradations are caused the increase in the packet length and the number of data frame transmission. Since an LR header in the scenario 5 sometimes contains more than three addresses for the source, destination, and intermediate nodes, the size of packet increases by 8 or 16 bytes, i.e. 3% to 6% in comparison with that in the scenarios 1 through 4. In addition, since the physical hop distance of a path established in the scenario 5 becomes longer than the physical shortest path used in the scenarios 1 through 4 for logical routing, the number of MAC frames increases. This increase for traffic in the scenario 5 results in the performance degradation.

In Figure 23, we illustrate the utilization of real-time channels in one simulation run. Comparing Figure 23(a) and 23(b), it is noticed that the nodes 32 and 65 are heavily loaded in the case without load balancing, whereas the load is relatively distributed in the case with load balancing. The average of variance of channel usage is 102,480,000 in the scenario 4 while it is 75,800,000

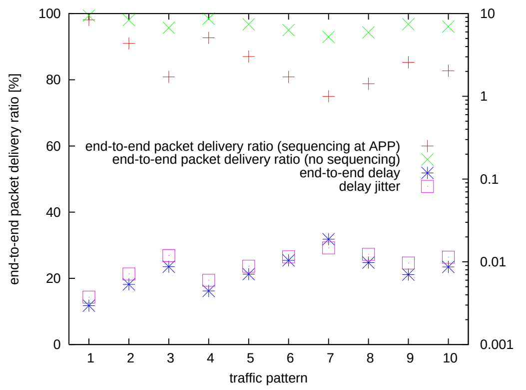


(a) Scenario 4: 1 best-effort and 3 real-time channels (hop-by-hop channel selection).

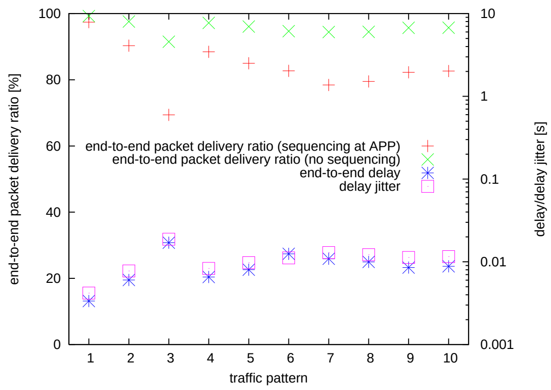


(b) Scenario 5: 1 best-effort and 3 real-time channels (logical routing)).

Figure 21: Comparison of scenarios 4 and 5 in packet-level performance.

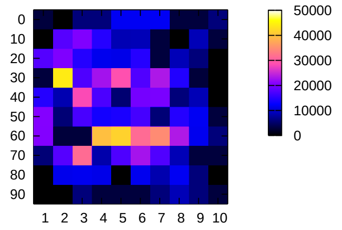


(a) Scenario 4: 1 best-effort and 3 real-time channels (hop-by-hop channel selection).

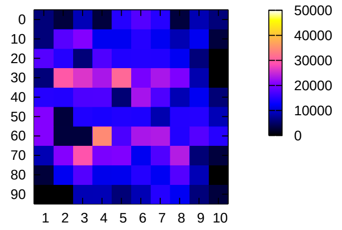


(b) Scenario 5: 1 best-effort and 3 real-time channels (logical routing).

Figure 22: Comparison of scenarios 4 and 5 in application-level performance.



(a) Scenario 4: 1 best-effort and 3 real-time channels (hop-by-hop channel selection).



(b) Scenario 5: 1 best-effort and 3 real-time channels (logical routing).

Figure 23: Comparison of scenarios 4 and 5 in logical routing efficacy.

in the scenario 5. In addition, the average fairness index is 0.53 in the scenario 4 and it is 0.66 in the scenario 5.

The average value of variance of transmitted data frames per node is 102,480,000 in the scenario 4 while it is 75,800,000 in the scenario 5. The average value of fairness index of transmitted data frames per node is 0.53 in the scenario 4 and it is 0.66 in the scenario 5. From these results, we can say that the scenario 5 effectively selects logical paths in order to avoid selecting busy nodes in the whole network.

Figure 24: The ad-hoc wireless relay node (Hitachi Information & Communication Engineering).

5 Practical experiments and discussions

In this section, we introduce the experimental system that implements our proposed mechanism and show results of practical experiments.

5.1 Experimental system

We used the equipmental system, the ad-hoc wireless relay node made by Hitachi Information & Communication Engineering as shown in Figure 24, to implement our proposed mechanism. The main part of specifications of the ad-hoc wireless relay node is shown in Table 3. Specifications are briefly summarized in Table 3.

The node has four wireless network interfaces which supports IEEE 802.11b/11g MAC protocols. One among the four operates in the infrastructure mode and the other three operate in the ad-hoc mode. We configured one interface among the three in the ad-hoc mode as a wireless network interface for best-effort channel and the other two for real-time channel. Since IEEE 802.11g has three orthogonal channels by being separated least 250 MHz to avoid inter-channel interference, we assigned 2.412 GHz (numbered as channel 0) for best-effort channel and 2.442 GHz (channel 1) and 2.472 GHz (channel 2) for real-time channels. In our preliminary experiments, we found that there was radio interference between electromagnetic waves emitted from antennas, but electromagnetic waves also emitted from antenna cables. Since the interference from antenna cables could be controlled by separating them by more than 20 cm, we built an antenna tower by

Table 3: Specifications of the ad-hoc wireless relay node.

Wireless interfaces	IEEE 802.11b/11g $\times$ 4
Ethernet interfaces	10Base-T/100Base-TX $\times$ 1
CPU	SH4
OS	Linux Kernel 2.6
RAM	DDR-SDRAM (64 MByte)
Serial ports	2 ports
Chassis and Size	Aluminum case, Not waterproof, $230 \times 210 \times 60$ mm
Manufacturer	Hitachi Information & Communication Engineering

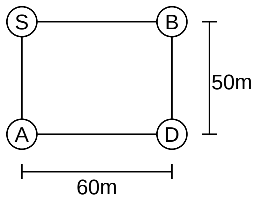
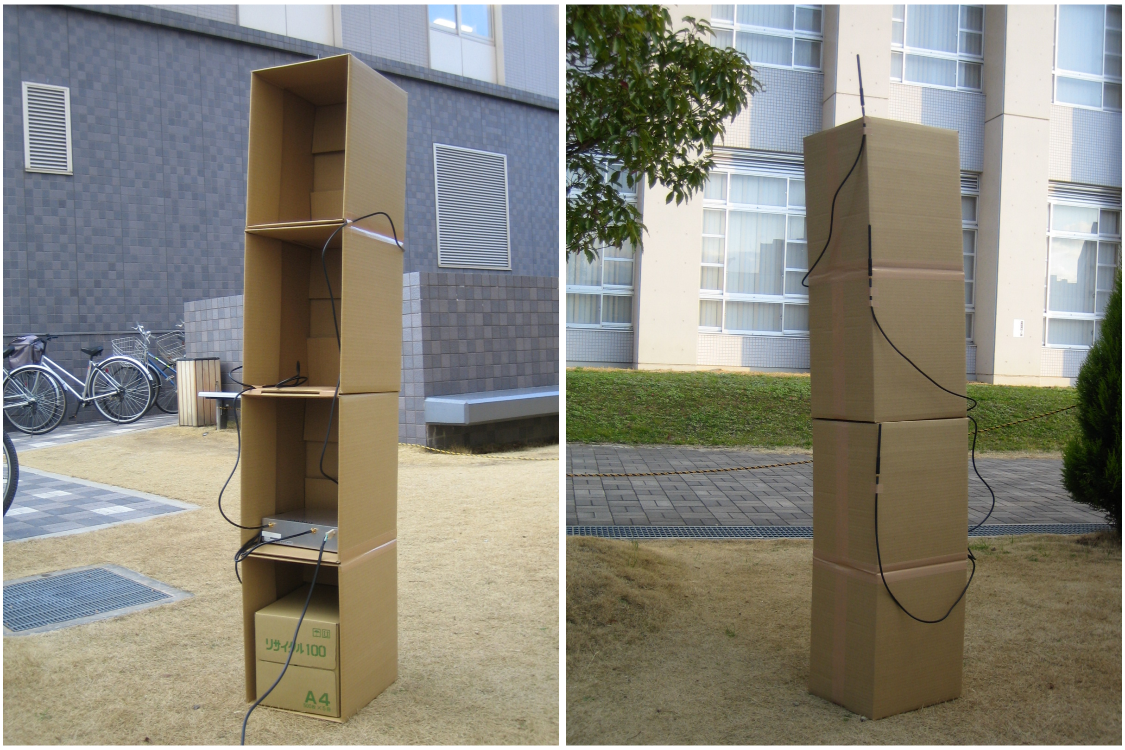


Figure 25: Experimental topology.

cardboard boxes as shown in Figure 26 to achieve the separation.

A node was put in the second lowest box as shown in Figure 26(a) and antennas were fixed on the side at the different height as shown in Figure 26(b). The OLSRv2 extended for our proposal used the best-effort channel. The intervals for HELLO message and TC message were set at 2 seconds and 5 seconds, respectively. So that all nodes could become a MPR, we set the value of the willingness, which determined the possibility that a node became an MPR, at 7 (`WILL_ALWAYS`).

On the node, Linux OS version 2.6.10 is running with GNU C library version glibc-2.3.3. We implemented our proposed mechanism onto three modules, the LR and SW, as a user-space application, and the OLSR, as both user-space and kernel-space application. Unlike the simulation environment which supports many useful APIs, only common APIs supported by common libraries are available on the relay node. Therefore, so we developed some sub modules to wrap APIs of both the QualNet 4.0 simulator and the practical system. In addition, we rebuild the kernel of the relay nodes to support semaphore functions required for asynchronous access to shared



(a) Front view of antenna tower.

(b) Rear view of antenna tower.

Figure 26: Antenna tower.

memory.

5.2 Experimental environment and discussions

Since we only had four relay nodes, we organized a simple square topology illustrated in Figure 25, where the node S is a source node, the node D is a destination node, and the nodes A and B are intermediate nodes. The nodes are placed around a building as Figures 27 through 30 show. The distance between two neighboring nodes, i.e. S-A, S-B, A-D, and B-D were about 50–60 m. The nodes had possibilities to connect links between S-D or A-B at the good condition of best-effort channel, but the nodes might not construct a link between S-D or A-B at our experiments. To generate real-time traffic as in VoIP applications, we used a CBR session which generated UDP/IP packets containing 172 bytes of data every 20 msec, i.e. 64 kbps UDP/IP traffic. In the practical experiments, the source node S generated a new session every 5 seconds and sessions kept sending packets until the end of the experiment. At the beginning of measurement, the network interfaces were already up and OLSRv2 was fully functional. For evaluation, we gathered information about the data transmission rate, the data reception rate, and the CPU usage per second, and the received data rate per session. The data transmission rate is defined as the total amount of MAC frames that are successfully sent to a next hop node in one second. The data reception rate is defined as the total amount of MAC frames that are received from a preceding node in one second. The CPU usage includes the percentages of CPU resource used by a user program, i.e. our proposed mechanism, and a system, and the percentage of unused CPU resource. The received data rate is defined as the per session amount of real-time application packets that are received in one second.

Results of one time experiment is shown in Figures 31 through 34 for the data transmission rate and the data reception rate per channel on each node.

As shown in Figure 31(a), the node S received only few bytes per second through the experiment, because the node S was always a source of packets. In contrast, we can see that the node S actively transmitted packets on channel 1 and channel 2 in Figure 31(b). In addition, lines for the data transmission rate on channels 1 and 3 overlap with each other. This implies that the node S evenly distributed real-time packets among these two real-time channels by observing the channel usage. The nodes A and B, i.e. intermediate nodes, also used real-time channels in a balanced manner as shown in Figures 32(b) and 33(b). On the contrary, there is difference in the data reception rate between the real-time channels on the nodes A and B in Figures 32(a) and 33(a). This

(a) View from S to A.

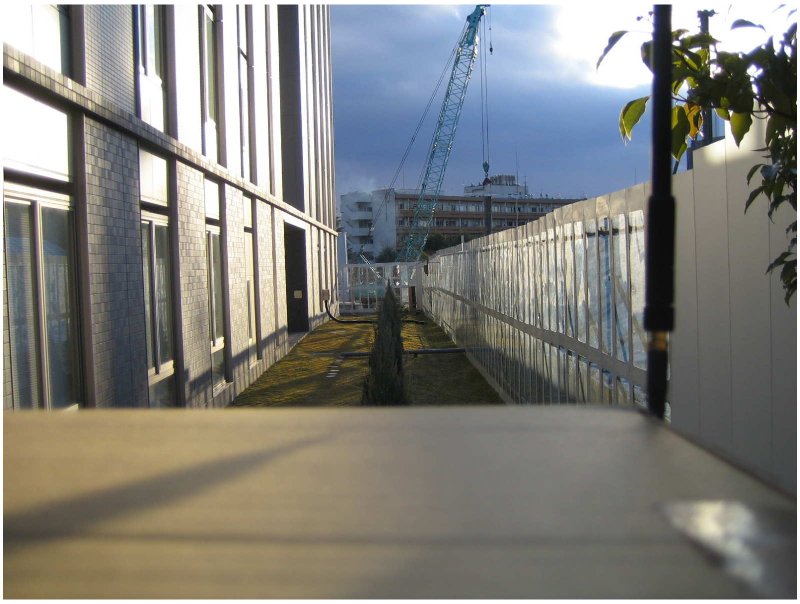
(b) View from S to B.

Figure 27: View from node S on the experimental environment.

(a) View from A to S.

(b) View from A to D.

Figure 28: View from node A on the experimental environment.



(a) View from B to S.

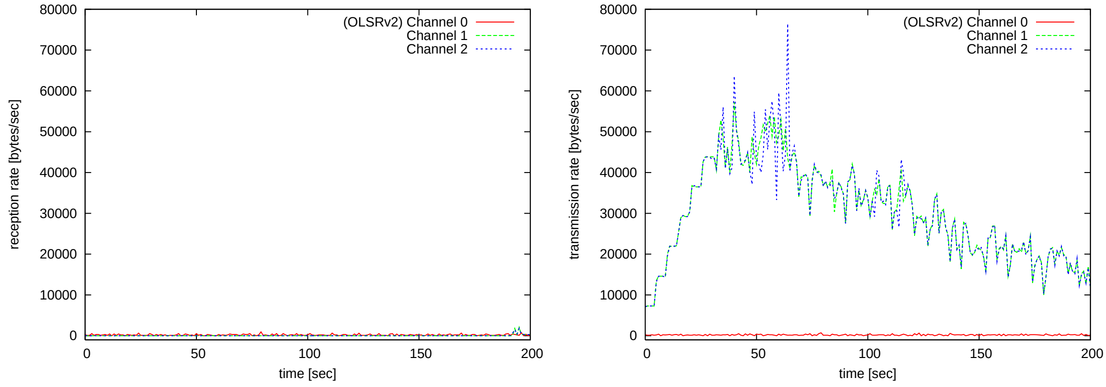
(b) View from B to D.

Figure 29: View from node B on the experimental environment.

(a) View from D to A.

(b) View from D to B.

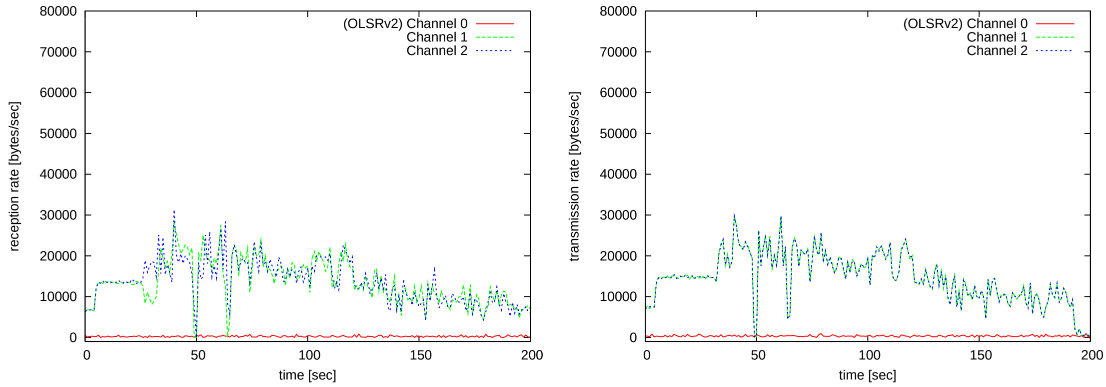
Figure 30: View from node D on the experimental environment.



(a) Per channel reception data rates.

(b) Per channel transmission data rates.

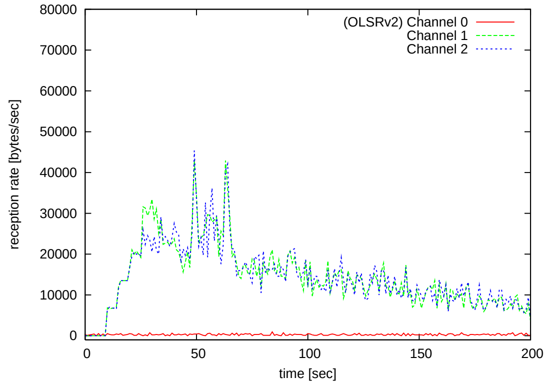
Figure 31: Device information on node S.



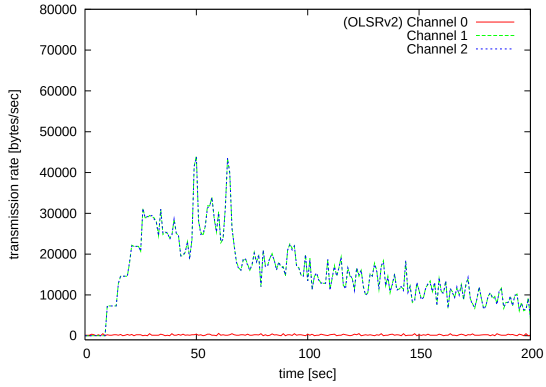
(a) Per channel reception data rates.

(b) Per channel transmission data rates.

Figure 32: Device information on node A.

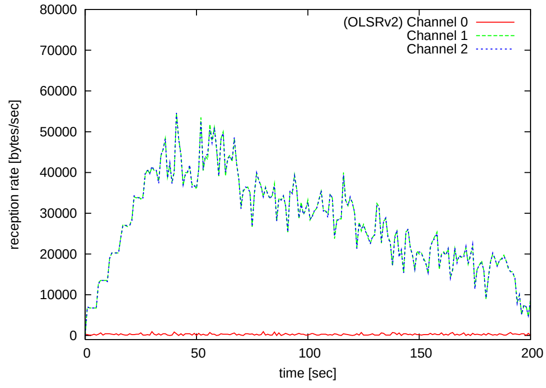


(a) Per channel reception data rates.

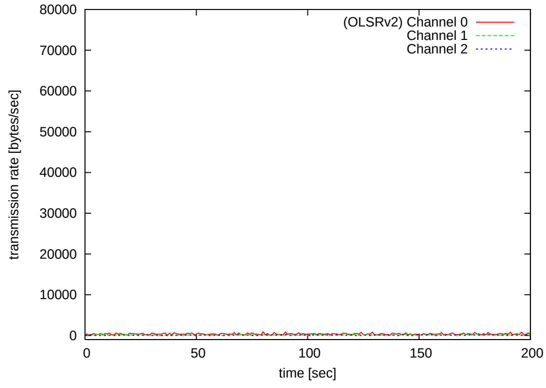


(b) Per channel transmission data rates.

Figure 33: Device information on node B.



(a) Per channel reception data rates.



(b) Per channel transmission data rates.

Figure 34: Device information on node D.

difference was caused by a hop-by-hop channel selection manner working at the node S. At the node S, the SW selected the most available real-time channel to send a packet based on hop-by-hop manner, but the SW could not consider a variation of next node. It could be explained by that the sum of the reception data rate of channel 1 at node A plus that of node B is the same as the transmission data rate of channel 1 at the source node S, and we could explain the same thing in channel 2. Furthermore, the sum of the data reception rate on channels 1 and 2 on a node is the same as the sum of the data transmission rate on channels 1 and 2 on the node in Figures 32 and 33, since they only relayed packets from the node S to the node D. Finally, Figure 34 shows results on the node D. Since the node D is a destination, it did not send much data as shown in Figure 34(b). The transition in the data reception rate on the node D looks similar to that in the data transmission rate on the node S (Figure 31(b)). However, there is difference.

The node S started a new session every 5 seconds. This leads to the stepwise increase in the data transmission rate (Figure 31(b)). From the timing of increase in the data reception rate in Figures 32(a) and 33(a), it can be said that the node S chose the path S-A-D for the first two sessions and then moved to the path S-B-D for the following four sessions. Since the advertising period of estimated channel information is 2 seconds and the propagation period of TC message is 5 seconds, there are few seconds of time lag for updated the Extended Topology Set that is used for logical routing.

As the number of sessions increased over 8 at 35 seconds, the data transmission rate decreased as shown in Figure 31(b). Figure 35 shows the data reception rate per session, the expected data reception rate per session, and the delay jitter per session. Until about 35 seconds, the data reception rate per session was as high as the expected data reception rate, which is defined as 8,600 bytes/sec. The delay jitter was also kept at the certain level of order. However, from 35 to 70 seconds, the data reception rate per session suddenly deteriorated and the delay jitter (determined in RFC 3550 [29]) exponentially increased.

The reason for this can be explained by Figure 36, where the transition of CPU usage is depicted.

As shown in Figure 36(a), the CPU idle ratio on the node S dropped to zero at 35 seconds and was kept zero since then. It implies that the drop of data transmission rate was caused by the full utilization of the poor CPU resource of the node S.

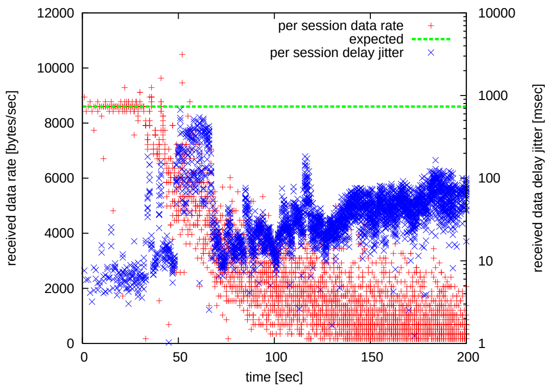
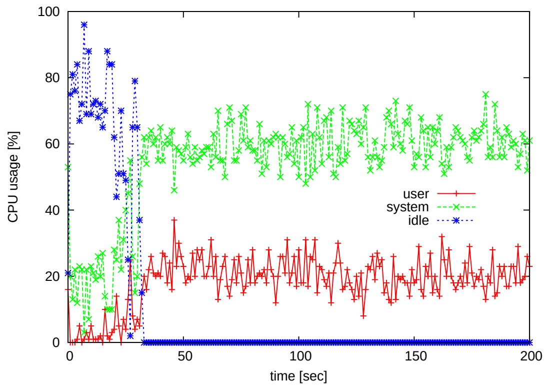


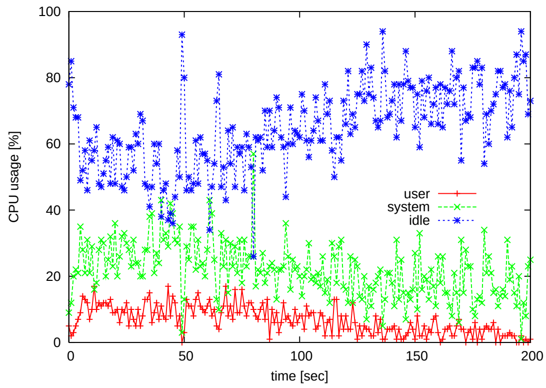
Figure 35: Received data rate per session at node D.

6 Conclusion

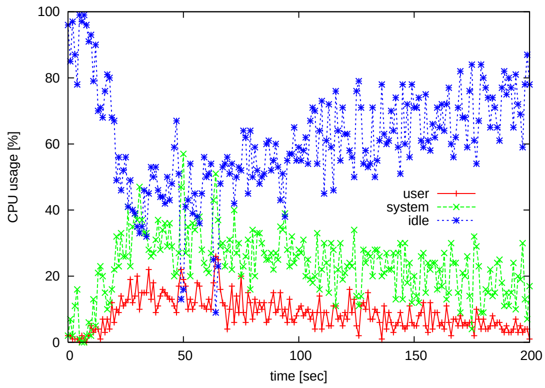
In this thesis, we proposed a QoS-aware routing mechanism for real-time applications. By embedding bandwidth information in control messages of OLSRv2, a source node can establish the logical path satisfying application QoS requirements with a view of topology and bandwidth of the whole network. Real-time packets are encapsulated so that traverses the logical path toward a destination node and sent on real-time channels. Through experiments on a simulator and a prototype, we confirmed that our mechanism could achieve the packet delivery ratio of about 95% at the end-to-end delay of about 10 msec in a grid network of 100 nodes by assigning three dedicated channels to real-time traffic and conducting logical routing. In addition, traffic was more evenly distributed over the whole network. However, some issues still remain. We need to reduce the load on a node to avoid the system-dependent bottleneck. In addition, we need to conduct extensive evaluation including the scalability.



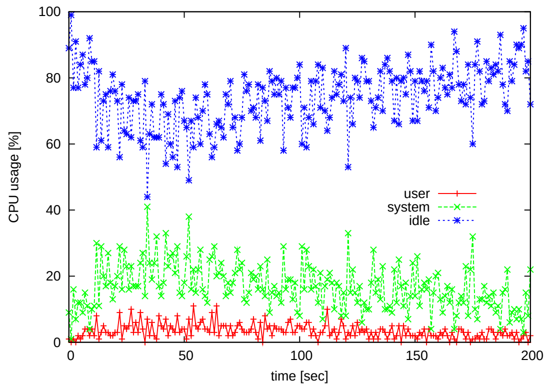
(a) Node S.



(b) Node A.



(c) Node B.



(d) Node D.

Figure 36: Transition of CPU usage.

Acknowledgments

I would like to appreciate my supervisor, Professor Masayuki Murata of Osaka University, for his extensive support on my career, invaluable comments on my studies, and strong encouragement on my path to doctoral course. I especially would like to express my appreciation to Associate Professor Naoki Wakamiya of Osaka University for his appropriate and elaborated guidance that inspired me a lot. My study would not have been achieved without his continuously support and encouragement.

I am most grateful to Professors Koso Murakami, Makoto Imase, Teruo Higashino, and Hirotaka Nakano, Associate Professor Go Hasegawa, Specially Appointed Associate Professor Kenji Leibnitz, Assistant Professors Shin'ichi Arakawa, Masahiro Sasabe, and Yuichi Ohsita of Osaka University, who gave me helpful comments and feedback.

I am very grateful to researchers of Hitachi, Ltd., System Development Laboratory for their valuable, specialistic and technical cooperation in my experiments.

Finally, I heartily thank my friends and colleagues in the Department of Information Networking, Graduate School of Information Science and Technology of Osaka University for their help.

References

[1] C. E. Perkins, ed., *AD HOC NETWORKING*. Addison-Wesley, first ed., Dec. 2000.

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, “Next century challenges: Mobile networking for “smart dust”,” in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom 1999)*, pp. 271–278, Aug. 1999.

[3] R. Rajaraman, “Topology control and routing in ad hoc networks: A survey,” *ACM SIGACT News*, vol. 33, pp. 60–73, June 2002.

[4] Bluetooth SIG, “Bluetooth special interest group.” `https://www.bluetooth.org/`.

[5] IEEE 802.11 Working Group, “IEEE 802.11 wireless local area networks.” `http://grouper.ieee.org/groups/802/11/index.html`.

[6] ZigBee Alliance, “ZigBee alliance.” `http://www.zigbee.org/en/`.

[7] J. Li, C. Blake, D. S. D. Couto, H. I. Lee, and R. Morris, “Capacity of ad hoc wireless networks,” in *Proceedings of 7th Annual International Conference on Mobile Computing and Networking*, pp. 61–69, July 2001.

[8] S. H. Shah, K. Chen, and K. Nahrstedt, “Available bandwidth estimation in IEEE 802.11-based wireless networks,” in *Proceedings of 1st ISMA/CAIDA Workshop on Bandwidth Estimation (BEst 2003)*, Dec. 2003.

[9] L. Hanzo and R. Tafazolli, “A survey of QoS routing solutions for mobile ad hoc networks,” *IEEE Communications Surveys and Tutorials*, vol. 9, pp. 50–70, 2nd Quarter 2007.

[10] R. Guimaraes, J. Morillo, L. Cerda, J.-M. Barcelo, and J. Garcıa, “Quality of service for mobile ad-hoc networks: an overview,” *Polytechnic University of Catalonia*, July 2004.

[11] H. Badis, A. Munaretto, K. A. Agha, and G. Pujolle, “QoS for ad hoc networking based on multiple metrics: Bandwidth and delay,” in *Proceedings of the 5th IFIP-TC6 International Conference on Mobile and Wireless Communications Networks*, pp. 15–18, Oct. 2003.

[12] S. Chakrabarti and A. Mishra, “Qos issues in ad hoc wireless networks,” *Communications Magazine, IEEE*, vol. 39, pp. 142–148, Feb. 2001.

[13] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A multi-radio unification protocol for IEEE 802.11 wireless networks,” in *Proceedings of the First International Conference on Broadband Networks (BROADNETS’04)*, pp. 344–354, July 2004.

[14] A. Nasipuri, J. Zhuang, and S. R. Das, “A multichannel CSMA MAC protocol for multi-hop wireless networks,” in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 1999)*, pp. 1402–1406, Sept. 1999.

[15] A. Tzamaloukas and J. Garcia-Luna-Aceves, “A receiver-initiated collision-avoidance protocol for multi-channel networks,” in *Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, vol. 1, pp. 189–198, Apr. 2001.

[16] Y. Liu and E. Knightly, “Opportunistic fair scheduling over multiple wireless channels,” in *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, vol. 2, pp. 1106–1115, Apr. 2003.

[17] J. So and N. H. Vaidya, “Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver,” in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc 2004)*, pp. 222–233, May 2004.

[18] P. Kyasanur and N. H. Vaidya, “Routing in multi-channel multi-interface ad hoc wireless networks,” tech. rep., University of Illinois at Urbana-Champaign, Dec. 2004.

[19] A. Raniwala and T. cker Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” in *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 2223–2234, Mar. 2005.

[20] N. Niranjan, S. Pandey, and A. Ganz, “Design and evaluation of multichannel multirate wireless networks,” *Mobile Networks and Applications*, vol. 11, pp. 697–709, Oct. 2006.

[21] P. Kyasanur and N. H. Vaidya, “Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks,” in *Proceedings of ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 10, pp. 31–43, Jan. 2006.

[22] H. Badis and K. A. Agha, “Quality of service for ad hoc optimized link state routing protocol (QOLSR),” *Mobile Ad hoc Networking (MANET) Internet-Draft*, Sept. 2007.

[23] T. H. Clausen, C. M. Dearlove, and P. Jacquet, "The optimized link state routing protocol version 2 (OLSRv2)," *Mobile Ad hoc Networking (MANET) Internet-Draft*, Jan. 2008.

[24] K. Xu, K. Tang, R. Bagrodia, M. Gerla, and M. Bereschinsky, "Adaptive bandwidth management and QoS provisioning in large scale ad hoc networks," in *Proceedings of IEEE Military Communications Conference*, vol. 2, pp. 1018–1023, Oct. 2003.

[25] W. Xiao, B. H. Soong, C. L. Law, and Y. L. Guan, "Evaluation of heuristic path selection algorithms for multi-constrained QoS routing," in *Proceedings of IEEE International Conference on Networking, Sensing and Control*, vol. 1, pp. 112–116, Mar. 2004.

[26] S. Licking, W.-P. Chen, M. Moh, and M. Nguyen, "A load-balanced channel assignment scheme for wireless mesh networks," in *Proceedings of the 18th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2006)*, Nov. 2006.

[27] Scalable Network Technologies, Inc., "QualNet 4.0." `http://www.scalable-networks.com/`.

[28] Niigata Universitiy, "OLSRv2 Wiki." `http://www.net.ie.niigata-u.ac.jp/~yowada/v2devel/`.

[29] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," July 2003.

Appendix

The following document is the main part of the config file for QualNet 4.0.

```
VERSION 4.0

################################################################################
# General simulation properties                                                #
################################################################################
SIMULATION-TIME   155S
COORDINATE-SYSTEM    CARTESIAN
TERRAIN-DIMENSIONS   (1000, 1000)
TERRAIN-DATA-BOUNDARY-CHECK NO

################################################################################
# Topology                                                                     #
################################################################################
NODE-PLACEMENT   GRID
GRID-UNIT        100
MOBILITY   NONE
MOBILITY-POSITION-GRANULARITY   1.0

################################################################################
# Channel properties                                                           #
################################################################################
PROPAGATION-CHANNEL-FREQUENCY[0] 2.484e9
PROPAGATION-CHANNEL-FREQUENCY[1] 2.412e9
PROPAGATION-CHANNEL-FREQUENCY[2] 2.437e9
PROPAGATION-CHANNEL-FREQUENCY[3] 2.462e9
PROPAGATION-LIMIT   -74.5
PROPAGATION-PATHLOSS-MODEL  FREE-SPACE
PROPAGATION-SHADOWING-MODEL NONE
PROPAGATION-FADING-MODEL NONE

################################################################################
# Phy layer                                                                    #
################################################################################
PHY-MODEL                   PHY802.11a
[N8-192.168.0.0] PHY-LISTENABLE-CHANNEL-MASK 1000
[N8-192.168.0.0] PHY-LISTENING-CHANNEL-MASK  1000
[N8-192.168.1.0] PHY-LISTENABLE-CHANNEL-MASK 0100
[N8-192.168.1.0] PHY-LISTENING-CHANNEL-MASK  0100
[N8-192.168.2.0] PHY-LISTENABLE-CHANNEL-MASK 0010
[N8-192.168.2.0] PHY-LISTENING-CHANNEL-MASK  0010
```

```
[N8-192.168.3.0] PHY-LISTENABLE-CHANNEL-MASK 0001
[N8-192.168.3.0] PHY-LISTENING-CHANNEL-MASK  0001
PHY-TEMPERATURE             290
PHY-NOISE-FACTOR            7.0
PHY-RX-MODEL                PHY802.11a
PHY802.11-AUTO-RATE-FALLBACK NO
PHY802.11-DATA-RATE                54000000
PHY802.11-DATA-RATE-FOR-BROADCAST  54000000
# 15.0 at 108-120m, 16.0 at 121-135m, 18.0 at 153-170m
PHY802.11a-TX-POWER--6MBPS  18.0
PHY802.11a-TX-POWER--9MBPS  18.0
PHY802.11a-TX-POWER-12MBPS  18.0
PHY802.11a-TX-POWER-18MBPS  18.0
PHY802.11a-TX-POWER-24MBPS  18.0
PHY802.11a-TX-POWER-36MBPS  18.0
PHY802.11a-TX-POWER-48MBPS  18.0
PHY802.11a-TX-POWER-54MBPS  18.0
# 14.0 at 500m
PHY802.11b-TX-POWER--1MBPS  14.0
PHY802.11b-TX-POWER--2MBPS  14.0
PHY802.11b-TX-POWER--6MBPS  14.0
PHY802.11b-TX-POWER-11MBPS  14.0

PHY802.11a-RX-SENSITIVITY--6MBPS  -69.9
PHY802.11a-RX-SENSITIVITY--9MBPS  -69.9
PHY802.11a-RX-SENSITIVITY-12MBPS  -69.6
PHY802.11a-RX-SENSITIVITY-18MBPS  -69.6
PHY802.11a-RX-SENSITIVITY-24MBPS  -69.3
PHY802.11a-RX-SENSITIVITY-36MBPS  -69.3
PHY802.11a-RX-SENSITIVITY-48MBPS  -69.0
PHY802.11a-RX-SENSITIVITY-54MBPS  -69.0

PHY802.11b-RX-SENSITIVITY--1MBPS  -83.9
PHY802.11b-RX-SENSITIVITY--2MBPS  -83.6
PHY802.11b-RX-SENSITIVITY--6MBPS  -83.3
PHY802.11b-RX-SENSITIVITY-11MBPS  -83.0

###############################################################
#  ANTENNA_CONFIGURATION
###############################################################
ANTENNA-GAIN             0.0
ANTENNA-EFFICIENCY       0.8
ANTENNA-MISMATCH-LOSS    0.3
ANTENNA-CABLE-LOSS       0.0
```

```
ANTENNA-CONNECTION-LOSS  0.2
ANTENNA-HEIGHT  1.5
ANTENNA-MODEL   OMNIDIRECTIONAL

###############################################################################
# MAC layer                                                                   #
###############################################################################
ARP-ENABLED    NO
MAC-PROTOCOL MACDOT11
MAC-DOT11-ASSOCIATION    NONE
MAC-DOT11-DIRECTIONAL-ANTENNA-MODE                NO
MAC-DOT11-SHORT-PACKET-TRANSMIT-LIMIT    7
MAC-DOT11-LONG-PACKET-TRANSMIT-LIMIT    4
MAC-DOT11-RTS-THRESHOLD    0
MAC-PROPAGATION-DELAY 1US
PROMISCUOUS-MODE   NO

###############################################################################
# Network layer                                                               #
###############################################################################
NETWORK-PROTOCOL   IP
DUAL-IP NO
IP-FRAGMENTATION-UNIT 2048
IP-ENABLE-LOOPBACK YES
IP-LOOPBACK-ADDRESS 127.0.0.1
IP-QUEUE-NUM-PRIORITIES   3
DUMMY-PRIORITY-QUEUE-SIZE NO
IP-QUEUE-PRIORITY-QUEUE-SIZE   50000
DUMMY-PRIORITY-WISE-IP-QUEUE-TYPE NO
IP-QUEUE-TYPE   FIFO
GREEN-PROFILE-MIN-THRESHOLD        10
GREEN-PROFILE-MAX-THRESHOLD        20
GREEN-PROFILE-MAX-PROBABILITY      0.02
YELLOW-PROFILE-MIN-THRESHOLD       5
YELLOW-PROFILE-MAX-THRESHOLD       10
YELLOW-PROFILE-MAX-PROBABILITY     0.02
RED-PROFILE-MIN-THRESHOLD          2
RED-PROFILE-MAX-THRESHOLD          5
RED-PROFILE-MAX-PROBABILITY        0.02
ECN NO
IP-QUEUE-SCHEDULER   STRICT-PRIORITY

###############################################################################
# Routing - forwarding, static, default routes                                #
```

```
################################################################################
IP-FORWARDING YES
STATIC-ROUTE NO

################################################################################
# Unicast routing - wireless ad hoc                                            #
################################################################################
[N8-192.168.0.0] ROUTING-PROTOCOL OLSRv2-NIIGATA
[N8-192.168.1.0] ROUTING-PROTOCOL NONE
[N8-192.168.2.0] ROUTING-PROTOCOL NONE
[N8-192.168.3.0] ROUTING-PROTOCOL NONE
OLSRv2-HELLO-INTERVAL 2S
OLSRv2-TC-INTERVAL 6S
OLSRv2-TIMEOUT-INTERVAL 2S
OLSRv2-START-HELLO 1S
OLSRv2-START-TC 1S
OLSRv2-START-TIMEOUT 1S
OLSRv2-LINK-LAYER-NOTIFICATION NO
OLSRv2-PACKET-RESTORATION NO
HSRP-PROTOCOL NO

################################################################################
# Transport layer                                                              #
################################################################################
TCP LITE
TCP-MSS                               512
TCP-SEND-BUFFER                       16384
TCP-RECEIVE-BUFFER                    16384
TCP-USE-RFC1323                       NO
TCP-DELAY-ACKS                        YES
TCP-DELAY-SHORT-PACKETS-ACKS NO
TCP-USE-NAGLE-ALGORITHM               YES
TCP-USE-KEEPALIVE-PROBES              YES
TCP-USE-PUSH                          YES

################################################################################
# Scheduler                                                                    #
################################################################################
SCHEDULER-QUEUE-TYPE          SPLAYTREE

#----------------Default Subnet -----------------
SUBNET N8-192.168.0.0 { 1 thru 100 } Default
SUBNET N8-192.168.1.0 { 1 thru 100 } Default
SUBNET N8-192.168.2.0 { 1 thru 100 } Default
```

```
SUBNET N8-192.168.3.0 { 1 thru 100 } Default

SEED   1
```