Cognitive Radio-Aware Transport Protocol for Mobile Ad Hoc Networks

Kazuya Tsukamoto, *Member, IEEE,* Shun Koba, *Non member, IEEE,* Masato Tsuru, *Member, IEEE* and Yuji Oie, *Member, IEEE*

Abstract—With the proliferation of new wireless service, scarce wireless resources is expected to become a critical issue. For this reason, cognitive radio mobile ad hoc networks (CogMANET) are being developed as a promising solution to this problem. However, in CogMANET, channel switching is inherently necessary whenever a primary user with a license appears on the channel. Allowing secondary users to choose an available channel from among a wide spectrum range thus enables reliable communication in this context, but communication characteristics such as bottleneck bandwidth and RTT will change with channel switch. In response to this change, TCP has to adaptively update its congestion window (*cwnd*) to make an efficient use of the available resources. For this purpose, TCP CRAHN was proposed for CogMANET. In this paper, TCP CRAHN is first evaluated in cases where bottleneck bandwidth and RTT drastically change. Based on these results, TCP CoBA is proposed to further improve the throughput of the above use cases. TCP CoBA updates the *cwnd* based upon the available buffer space in the relay node upon channel switch, as well as other communication characteristics. Through simulations, we show that compared with TCP CRAHN, TCP CoBA improves the throughput by up to 200%.

Index Terms—Transport Protocol, Cognitive Radio, Mobile ad hoc network, Multi-hop communication

1 Introduction

Cognitive radio technology has the potential to ameliorate the scarcity of wireless resources because unlicensed users (secondary users: SUs) can use wireless resources only if they have no impact on the operations of licensed users (primary users: PUs). In the future, cognitive radio mobile ad hoc networks (CogMANET) will be constructed from many mobile SUs connected to each other in a distributed manner, which can be deployed for various applications, including intelligent transport systems (ITS).

A promising way to improve not only the survivability but also the reliability of communication in CogMANET is to allow SUs to select a communication channel (channel) satisfying their application requirements from a wide range of spectrum. However, since SUs always need to guarantee no impact on PU performance, they have to engage in periodical sensing to detect PUs, and then switch channels whenever a new PU appears. Hence, communication in CogMANET is likely to experience changes in characteristics in terms of bottleneck bandwidth and round-trip time (RTT) due to channel switching. In such case, in response to channel change, TCP has to adaptively update its window size (wnd) to achieve an efficient use of available wireless resources. It is assumed that the wnd is determined by the congestion window ($cwnd$) only because of a large advertised window ($awnd$).

- *K. Tsukamoto is with the Dept. of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka, Fukuoka, Japan, 820-8502. E-mail: tsukamoto@cse.kyutech.ac.jp*
- *S. Koba, M. Tsuru and Y. Oie are with Kyushu Institute of Technology.*

This study was supported in part by the JSPS KAKENHI (25330107).

In this context, TCP CRAHN was proposed as a TCP variant for CogMANET [12]. Except for the features of channel switching and periodical sensing, TCP CRAHN's congestion control is completely same with those of existing TCP (TCP NewReno). When a channel is changed, TCP CRAHN's congestion control uses information sent by relay nodes (see Table 2), as is the case of XCP [11]. If the relay node changes its channel, it will inform the TCP sender of when to begin and finish changing as well as the bandwidth and link delay of two neighboring nodes (i.e., a new channel link). After receiving this information, the TCP sender updates $cwnd$ appropriately. This is the key function required in CogMANET. However, this scheme assumes that each node engages in periodical sensing, the timing and duration of periodical sensing are totally controlled by the TCP sender. For that purpose, the TCP sender sends a message to the nodes on the routing path. This may not be scalable, as the nodes have to accommodate multiple flows from different TCP senders.

Furthermore, it is assumed that SUs are allowed to use some wide range of the spectrum, such as 400 MHz to 6 GHz [18], in a future CogMANET, for spectrum efficiency. However, this can cause drastic changes in bottleneck bandwidth and RTT when a node changes its channels. Therefore, TCP CRAHN is first evaluated when bottleneck bandwidth and RTT drastically change in CogMANET, exposing some issues to degrade throughput performance. Next, a new TCP that solves the related issues identified above is developed. Our contributions are described below.

- Performance degradation, caused by drastic change in bottleneck bandwidth and/or RTT when channel switching, can be avoided effectively by updating the *cwnd* appropriately through collaborative work with relay nodes.
- Each of multiple TCP CoBA flows can achieve a fair share of the network resources.

2 Target Network

The target network is multihop communication in CogMANET. It is assumed that different channels are used in each hop so that communication of each hop does not interfere with each other. Communication in CogMANET is likely to suffer from changes in characteristics due to channel switching and interruptions caused by periodical sensing. In this study, channel characteristics, such as bandwidth and transmission coverage area of channel, are changed as a result of channel switching, because it is assumed that the nodes select a channel from a wide range of the spectrum, such as 400 MHz to 6 GHz. Therefore, end-to-end communication characteristics such as bottleneck bandwidth and RTT may change. The evaluation focuses on TCP performance when bottleneck bandwidth and RTT are changed drastically.

In practice, SUs have to sense whether PUs are communicating on their licensed channel, where SUs are assumed to use energy detection (ED) as the sensing technology. ED provides high speed processing but less sensitivity. Since ED cannot distinguish between PUs and SUs, each SU in CogMANET synchronously executes sensing and data transmission in sequence periodically in a time-division manner. That is, the signal received during the sensing period indicates PUs' signal. To achieve this in TCP CRAHN, the TCP sender sends a message to adjust the timing and duration of periodical sensing to nodes on the routing path. On the other hand, if the Global Positioning System (GPS) function is assumed to be available for each SU, the SU can synchronously execute periodical sensing. This can result in frequent interruptions in ongoing communication. Conversely, each node cannot execute sensing during the data transmission period, which has the potential to interfere with PU communication. Furthermore, to switch a channel used between two adjacent nodes, exchange of some type of control messages is necessary. Thus, data transmission is interrupted during channel switching as well. As can be seen above, communication in CogMANET suffers from frequent interruptions due to periodical sensing and channel switching. Note that, in this paper, packet losses on wireless links do not occur since the goal is to focus on the impact of cognitive radio only.

3 Related Work

A large number of TCP variants has been developed to achieve high performance in various contexts. This section will summarize the main state-of-the-art TCP from the perspective of multihop cognitive networks.

3.1 Transport protocol for a 1-hop cognitive radio

In [1], [2], [3], [4], the performance of conventional TCPs in a cognitive radio environment was found to be significantly degraded because of frequent interruptions due to sensing and channel switching.

On the other hand, TCP Westwood [5] prevents performance degradation due to random losses in the last 1-hop wireless environment. When packet losses are detected, the sender updates *ssthresh* to the calculated bandwidth-delay product (BDP) based on the estimated bandwidth and the minimum RTT in ongoing communications, thereby avoiding excess reduction of the *cwnd*. Although in [5], the interval of receiving time of two consecutive ACKs is utilized to estimate bandwidth, the bandwidth is overestimated when the ACK interval is reduced due to ACK filtering and so on. To solve the problem, TCP Westwood+ [6] was proposed. In [6], the bandwidth is estimated based on both the amount of data acknowledged by receiving the latest ACK and the calculated RTT. Moreover, moving average is employed to smooth the estimated bandwidth, thereby improving the estimation accuracy of Westwood+. Since these TCPs were not developed for use in cognitive radio environments, it underestimates the bottleneck bandwidth because it ignores the interruptions caused by periodical sensing. To resolve this issue, cogTCPE [9], which is based on Westwood(+), was proposed for the last 1-hop cognitive radio environment. With this TCP, the estimation of bandwidth by Westwood is enhanced. The *cwnd* is also computed using messages exchanged between end nodes only. However, when cogTCPE is used for multihop communication in CogMANET, it cannot detect channel switching at the relay nodes. In this case, the Layer 2 (L2) in switching nodes successfully switches a channel, but cogTCPE continues to send segments even during channel switching, thereby causing buffer overflow in the switching nodes. For this reason, cogTCPE cannot efficiently use wireless resources, because *cwnd* is not updated after channel switching at the relay nodes.

In addition, [10] targets multihop communication in mesh networks with cognitive radio. The authors propose a multichannel MAC protocol that can improve the TCP throughput, without modifying TCP.

3.2 Transport protocol for MANET

ATP [7] and ATCP [8] are proposed for multihop communication in MANET. ATP [7] is not a variant of TCP, but a rate-based transport protocol. ATP obtains network congestion information, which is first attached to each ATP packet by the relay nodes and is then received and sent back to the sender by the ATP receiver. By using this information, the ATP sender can adjust its transmission rate. In contrast, ATCP [8] is fully compatible with the traditional TCP. ATCP relies on a network layer feedback to make the TCP sender aware of the status of network path, so that support of ATP is required for the sender only. Indeed, among the feedback information, explicit congestion notification (ECN) flags and ICMP destination unreachable messages enable the TCP sender to detect network congestion and path disruption, respectively.

In this way, these TCP variants use specialized control messages transmitted from relay nodes, for example, feedback control. Feedback control in ATP and ATCP is used to determine the cause of packet loss and avoid performance degradation as a result of inappropriate congestion control. However, these approaches do not detect channel switching in the CogMANET environment. Therefore, they continue to send segments during channel switching, leading to buffer overflow in the switching node. As a result, they cannot efficiently use wireless resources due to inappropriate $cwnd$ sizing.

3.3 Transport protocol for CogMANET

To the best of our knowledge, existing transport protocols that support multihop communication in CogMANET are TCP CRAHN [12] and TFRC-CR [13] only. TFRC-CR maintains the end-to-end principle, which does not need both of feedback information from relay nodes and cross-layer collaboration with lower layers, whereas TCP CRAHN needs both of them, that is, the end-to-end principle is broken.

In [13], since all nodes in CogMANET could experience drastic changes in communication performance due to PU appearance and channel switching, the authors pointed out that handling the frequent changes is difficult when using a RTT-based self-clocking mechanism (as in the regular TCP) and the increase in the amount of feedback information could cause performance degradation by 20%. To solve these issues, the authors propose a new transport protocol based on TFRC (TFRC-CR), which shows stable performance in CogMANET. Note that the sender of TFRC-CR is assumed to be able to obtain various information such as which channel on the en-to-end path is used by PU, through the whitespace Database (DB) authorized by FCC.

So far, the standardization process of whitespace DB has not been finalized yet. In particular, in terms of the DB access, its radio frequency and access procedures are not fixed yet. Furthermore, access latency may become large due to traffic congestion and/or variation of wireless link quality, so that quick and appropriate control of data transmission while considering PU signal will be extremely difficult.

On the other hand, since sensing only devices still remain in FCC regulations, this type of devices would be deployed and could not be negligible. Moreover, we believe that a dedicated control channel shared by neighbor nodes can avoid performance degradation effectively, even when access to the DB is not available due to the congested traffic of access messages and/or the variation of wireless condition [14]. Therefore, in this paper, we focus on TCP CRAHN in which each node executes periodical sensing without the use of DB. Note that TCP CRAHN is based on TCP NewReno, so that the existing applications employing conventional TCP can be used.

3.4 TCP CRAHN

TCP CRAHN [12] was proposed as a transport protocol for multihop communication in CogMANET. This TCP variant adaptively updates $cwnd$ in response to changing communication characteristics due to channel switching at relay nodes.

3.4.1 *Sensing method*

CRAHN uses ED as the sensing technology. Each SU performs sensing and data transmission processes in an asynchronous time-division manner. Therefore, CRAHN modifies the three-way handshake process in NewReno such that the TCP sender can obtain the sensing schedules of all nodes on the routing path. Furthermore, the TCP sender sends messages to adjust the timing and duration of sensing to the nodes on the routing path during TCP communication. However, it may happen that a node receives messages from more than one TCP sender, which will cause the node to require different schedules for sensing. Such a situation is not considered in [12], which limits the scalability of this solution.

3.4.2 *Response to interruptions*

As noted in Section 2, end-to-end communication may suffer from frequent interruptions because the nodes cannot send data packets during periodical sensing. In addition, once the node begins to switch channel, it stops data transmission until the completion of channel switching. Packet arrivals to the node during the channel switching can result in buffer overflow in the node. Hence, the relay node informs the TCP sender of the start and the end of channel switching. The TCP sender stops sending TCP segments once the start of channel switching is notified. After that, CRAHN freezes the retransmission timeout (RTO) timer so as to avoid timeout during channel switching. The TCP sender restarts the RTO timer after receiving notification of end of channel switching.

3.4.3 *Response to changes in communication characteristics*

In CRAHN, nodes switching their channels measure communication characteristics (link bandwidth W, link delay L^T) and send back related information to the TCP sender. When the TCP sender receives the

TABLE 1
Communication parameters

Simulation time	50 s
Number of trials	30
Segments size	1460 bytes
L4 buffer size	1000 packets
L2 buffer size	100 packets
Data transmission period	1.0 s
Sensing period	0.5 s
Avg. of PU com. period	3.0 s
P_{on}	0.5
channel switching time	0.5 s

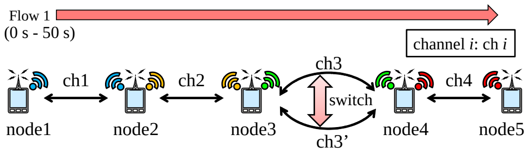


Fig. 1. Simulation topology

information, the bottleneck bandwidth (W'_b) is calculated using this information. When W'_b is changed by over 20% relative to the value before channel switching, the sender updates $cwnd$ and RTT from Eqs. (1) and (2). L'^T is the link delay before channel switching. In addition, α in Eq. (2) is 1 or less[1] because the updated $cwnd$ is intentionally limited to less than the $cwnd$ limit [15].

$$RTT_{new} = RTT_{old} + L^T - L'^T \quad (1)$$

$$\text{cwnd} = \alpha \cdot W'_b \cdot RTT_{new} \quad (2)$$

4 Evaluation of existing schemes

Some of TCP variants described in the previous section, TCP NewReno, Westwood+, and CRAHN, will be examined below in terms of their throughput performance in CogMANET.

4.1 Simulation Model

Simulations are performed using Qualnet 4.5, with parameters given in Table 1. In Qualnet 4.5, we used standard codes for NewReno. However, since no official support on Westwood+ and CRAHN is provided, we implemented these protocols and used them. L4 buffer size of end nodes is 1000 packets, so that it is considerably larger than BDP, while L2 buffer size of each node is set to 100 packets [2]. Furthermore, the sensing period is fixed to 0.5 s using Eq. (3) in [17] to prevent the probability of PU misdetection P_f from becoming larger than 5%. W is the channel spectrum bandwidth [Hz] and γ is the external signal to noise ratio ($\text{SNR}_{db}[\text{dB}] = 10\log_{10}\gamma$). Furthermore, P_{on} is the probability that a PU is in a communication period; $P_{off} = 1 - P_{on}$, and Q is the standard Q function.

$$t_s = \frac{1}{W \cdot \gamma^2}\left[Q^{-1}(P_f) + (\gamma+1)Q^{-1}\left(\frac{P_{off}P_f}{P_{on}}\right)\right]^2 \quad (3)$$

W was set to 2 MHz, and SNR_{dB} was set to -25 dB [12] since signal quality deteriorates due to node mobility. Moreover, P_{on} was set to 0.5, i.e., $P_{off} = 0.5$.

We assume communication on CogVANET in which PU appears frequently. Thus, we set the average of PU communication period to 3.0 s and set P_{on} to 0.5.

The simulation network is shown in Fig. 1, in which node 1 sends bulk data to node 5. Channels 3 and 3′ are available between nodes 3 and 4, one of which is chosen in a way to avoid interference to PU communication. Channel 3′ is assumed not to be used for a single hop, but to be composed of several channels for multihop communication between nodes 3 and 4. Therefore, link delay of Channel 3′ is set to be larger than that of Channel 3. In this paper, we assume that channel switching at some link causes the change in the number of hops between end nodes, which in turn changes the link delay of multiple hops, so that we emulate it by changing link delay. Furthermore, since the bandwidth over multiple hops may be changed, we varied the bandwidth from 20 Mb/s to 60 Mb/s.

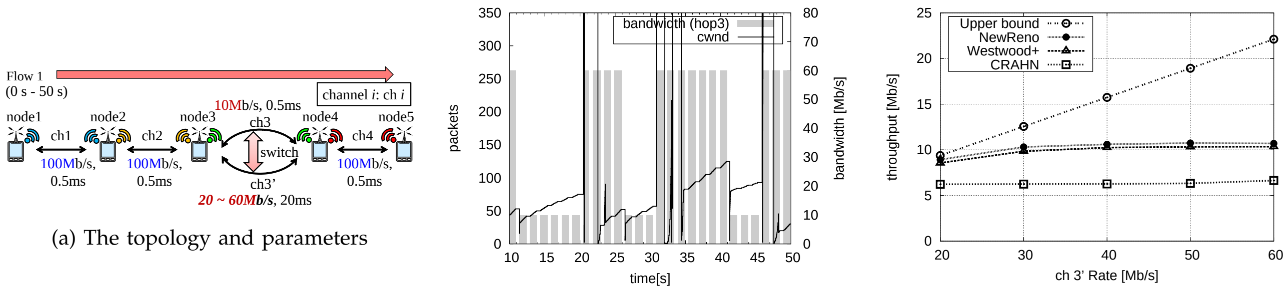
The focus of the simulations is on the status of PU communication. When a PU is present on both Channels 3 and 3′ simultaneously, communication between nodes 3 and 4 is disconnected. Therefore, it is assumed that the PU uses Channel 3 only. These nodes switch from Channel 3 to 3′ quickly after detecting the PU on Channel 3. On the other hand, these nodes switch back from Channel 3′ to 3 when the PU on Channel 3 ceases communication. The PU communication period follows an exponential distribution with a mean 3 s. Furthermore, communication performance of both the PU and the SU inevitably degrades when the SU interferes with the PU. However, this problem is out of consideration as we focus on how the changes in communication characteristics impact on throughput performance of the SU. Furthermore, $cwnd$ and the average throughput are used as a performance measure. The average throughput is defined as the time-averaged TCP goodput (from 10 s after communication has begun) averaged over 30 runs.

4.2 Impact of changes in both the bottleneck bandwidth and RTT caused by channel switching

In this section, the case where both RTT and bandwidth change due to channel switching is considered.

1. In Ref. [12], α is 0.8 because each node stores packets from multiple flows. In the present study, α is set to 1 since only a single flow is considered.

2. Reference [16] investigated L2 buffer size on several types of wireless APs. It was shown that commercial APs have a buffer whose size varies from 50 to 350 packets and residential APs vary from 40 to 100 packets. In the present paper, it is assumed that mobile nodes, which are not superior in their performance to commercial APs, are used as nodes. Therefore, L2 buffer size on each node is set to 100 packets in consideration of future performance improvement of mobile node.



(a) The topology and parameters

(b) *cwnd* (BW of Channel 3′ is 60 Mb/s)

(c) Averaged throughput

Fig. 2. Impact of the changes on both the bottleneck bandwidth and RTT by channel switching

As shown in Fig. 2a, link delay of Channel 3 is 0.5 ms, and that of Channel 3′ is 20 ms, so that RTT is significantly changed after channel switching. In addition, it is assumed that SUs select a channel from a wide spectrum, so that bandwidth of Channel 3′ varies from 20 Mb/s to 60 Mb/s.

Fig. 2b illustrates the *cwnd* of CRAHN when the bandwidth of Channel 3′ is 60 Mb/s. The *cwnd* was updated to a large value when node 3 switched from low-rate Channel 3 to high-rate Channel 3′. After that, a too large *cwnd* immediately after channel switching causes buffer overflow at the relay node, thereby resulting in a drastic decrease in the *cwnd*. In CRAHN, TCP sender calculates *cwnd* based on BDP (Eq. (2)).

Fig. 2c shows the average throughput of all TCP variants when the bandwidth of Channel 3′ is changed. The performance difference between an upper bound and throughput of all TCP variants increases with the bandwidth of Channel 3′, which is the bottleneck bandwidth. Note that the upper bound is an achievable maximum throughput obtained by transmitting UDP packets. As mentioned earlier, CRAHN increases the *cwnd* up to a too large value by Eq. (2) compared with the available buffer space of the relay node when both link delay and bandwidth are increased, thereby causing buffer overflow. On the other hand, both NewReno and Westwood do not update *cwnd* at all after channel switching. Thus, NewReno and Westwood cannot make an efficient use of available bandwidth if the bandwidth is increased due to channel switching. CRAHN dynamically changes its *cwnd* by using changing BDP. However, Fig. 2b shows that it can excessively increase its *cwnd* compared with available buffer space of relay node when the bandwidth increases drastically. This results in throughput performance degradation as shown in Fig. 2c.

The solution to this issue is to adjust appropriately the *cwnd* based upon the available buffer space of the bottleneck, if necessary, in addition to BDP. Furthermore, the change in RTT should also be considered for updating BDP, although CRAHN does not. Therefore, we will propose a new TCP which solves this issue in the following section.

5 TCP CoBA

Based on the evaluation presented in Section 4, this section proposes TCP CoBA as a new transport protocol. Behavior of TCP CoBA is completely same with that of TCP NewReno except for the period of channel switching, as in the TCP CRAHN. Note that we assume that each SU can synchronously execute periodical sensing by exploiting GPS function.

CoBA is proposed to achieve high performance by updating the *cwnd* appropriately in response to the change in the bottleneck bandwidth (W_b) and RTT. Hence, CoBA also updates the *cwnd* when the RTT is changed by over 20% due to channel switching, which is different from CRAHN. CoBA freezes data transmission and RTO timer during the channel switching as in CRAHN.

5.1 Feedback information

First, a sender of TCP CoBA can get various information from each relay node. Relay nodes send the information in the following four cases: (1) three-way handshake, (2) forwarding of data packet, (3) start of channel switching, and (4) end of channel switching. The information sent in each case is listed in Table 2, where BW_i is the bandwidth on link i, $L^T_{i,i+1}$ is the 1-hop RTT between neighboring two nodes (nodes i and $i+1$), and BF_i is the remaining buffer space in node i. These parameters are obtained from layer 2 [3]. Since the information of bandwidth (BW_i) and latency ($L^T_{i,i+1}$) for each link are inserted in the TCP header of all relaying packets, these information can be known by TCP sender. Therefore, TCP sender also grasps the bottleneck bandwidth (W_b) by BW_i of all links.

The relay node returns a layer 3 packet including feedback information of new link bandwidth and new latency immediately after channel switching, so that the TCP sender on layer 4 can quickly recognize a new latency ($L'^T_{i,i+1}$) and a new bandwidth (BW'_i) as well as where the bottleneck node is located and how much its bandwidth is (W'_b). In addition, CoBA

3. BW_i and BF_i can be obtained from driver software of radio interface. Furthermore, we assume that $L^T_{i,i+1}$ can be obtained by measurement packet exchange between neighboring two nodes $(i, i+1)$, as in the TCP CRAHN.

uses feedback information on the remaining buffer space transmitted from a relay node having changed its channel in order to prevent buffer overflow. Note that, for ease of implementation, it is assumed that the feedback information is available only from a node changing its channel.

When multiple TCP flows coexist, $\frac{BW_i}{(Num.of flows)}$ and $\frac{BF_i}{(Num.of flows)}$ are informed by a node experiencing channel switching. TCP sender updates the amount of transmitted data to $min(\frac{BW_i}{(Num.of flows)}, \frac{BF_i}{(Num.of flows)})$, thereby preventing buffer overflow even under a case of multiple TCP flows. Moreover, when bi-directional flows coexist, each of the two neighboring nodes experiencing channel switching sends back the feedback information to the TCP sender, so that each of the TCP senders adjusts the transmission rate to the appropriate value based on the updated $cwnd$ (described later). As a result, we can say that the CoBA can provide good communication performance better than that for CRAHN, irrespective of direction of multiple flows. Note that, for ease of mathematical expressions, the information in case of single flow are hereinafter assumed.

5.2 Discussion for feedback-based method

Since the sender node of TCP CoBA receives feedback information transmitted from relay nodes after channel switching, the end-to-end principle cannot be applied to CoBA, as in the CRAHN. Collaborative control between sender node and relay nodes requires an exchange of control message, thereby increasing not only computational load but also message overhead in the network. As a result, communication performance may degrade.

In CogMANET, all nodes could experience drastic changes in communication performance due to appearance of PU signal and channel switching, so that transmission control with the end-to-end principle is extremely difficult. In such case, this collaborative control provides relatively stable communication in spite of possible drawbacks mentioned earlier. Therefore, we propose a new transport protocol based on collaborative control, as in CRAHN.

It may happen that the feedback information gets lost in real wireless environment. To prevent this from causing severe performance degradation, we design the control of CoBA to be completely same with that of NewReno, except for in the channel switching. That is, CoBA ensures that negative impact by introducing CoBA can be limited extremely.

5.3 Procedures for cwnd updates

The method used to update $cwnd$ and $ssthresh$ when node i changes its channel (see Fig. 3) is outlined in Fig. 4 and explained below.

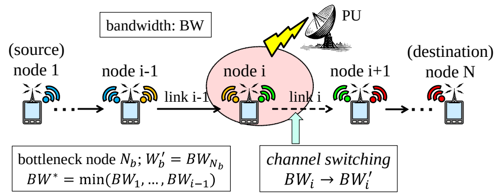


Fig. 3. Scenario of channel switching

TABLE 2
Feedback information from relay nodes

Timing	CRAHN	CoBA
(1) 3 way handshake	timing of sensing	
	duration of sensing	
(2) relaying of data/ack	BW_i, $L^T_{i,i+1}$	
(3) switching start	time at start of switching	
(4) switching end	time at end of switching	
	BW'_i: BW_i after swiching	
	$L'^T_{i,i+1}$: $L^T_{i,i+1}$ after switching	
		BF_i

When a relay node changes its channel if PU communication is detected, its bandwidth and link delay ($L^T_{i,i+1}$) can also be changed. This change is drastic when the bottleneck bandwidth or RTT changes. Therefore, in response to a change of more than 20% in the bottleneck bandwidth or RTT, CoBA appropriately updates its congestion control parameters, $cwnd$ and $ssthresh$, using the feedback information received from the relay node[4].

We can categorize cases of where the bottleneck node is located into two cases, depending on whether the remaining buffer space at the switching node should be considered or not. The first case is that the bottleneck node is located on a path from the TCP sender to node i, in which even in the bottleneck node, the buffer will be empty by the end of channel switching because the TCP sender stops sending packets during channel switching. In this case, BDP is thus good enough for updating $cwnd$. The second case is that the bottleneck node is just the switching node or located forward of it. In this case, $cwnd$ should be limited to not allow excessive consecutive packet transmissions by using the remaining buffer space at the switching node, as explained later. $cwnd$ and $ssthresh$ are updated as shown in Fig. 4.

a) The switching node sends information described in Section 5.1 to the TCP sender. When the TCP sender receives the feedback message after channel switching, it calculates the RTT_{new} by Eq. (4), which corresponds to Eq. (1), and bottleneck bandwidth W'_b.

$$RTT_{new} = L^T_{1,2} + \cdots + L^T_{i-1,i} + L'^T_{i,i+1} + \cdots \quad (4)$$

4. Cross-layer control that can share information obtained from layer 2, 3, and 4, is essential. In [20], cross-layer collaboration from layer 2 to layer 4 can be realized by an independent management module implemented in the Linux kernel. Therefore, we assume that this kind of implementation can be used for TCP CoBA.

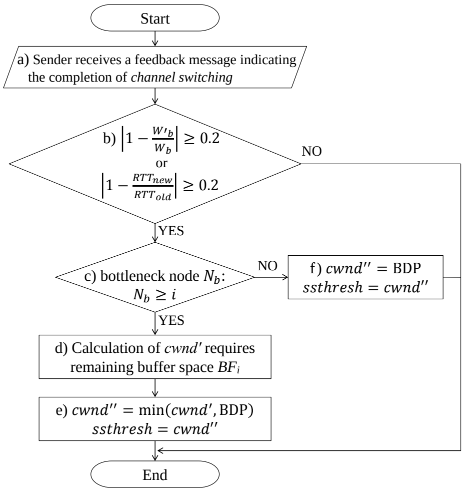


Fig. 4. Procedure for $cwnd$ updates

Note that CRAHN includes queueing delay in $L^T_{i,i+1}$, so that its resulting $cwnd$ assumes some packets are buffered in relay nodes. This can congest the buffer, while achieving high throughput if the available buffer space is large enough. Since the queuing delay can be estimated by exploiting packet length, bandwidth, and buffer length, CoBA employs $L^T_{i,i+1}$ excluding queuing delay. However, in our simulation, we pre-set the parameters of bandwidth and 1-hop RTT for each channel. On the other hand, the buffer length is obtained from layer 2.

b) The W_b and RTT_{old} are the bottleneck bandwidth and RTT before channel switching, respectively. When either the W'_b or RTT_{new} is changed by over 20% from the previous value before channel switching (i.e.,W_b and RTT_{old}), the sender updates $cwnd$.

c) The following procedures depend on where the bottleneck node N_b is located. It is important to carefully deal with the case that the bottleneck node is just node i or some node located between node i and node N after channel switching, i.e., $N_b \geq i$ (see (d)). Otherwise, the $cwnd$ and $ssthresh$ are determined by just BDP (see (f)).

d) Here the $cwnd'$ that prevents buffer overflow in the bottleneck node N_b based upon the remaining buffer space given by node N_b is derived. First, suppose that $cwnd$ is increased to $cwnd'$. In this case, some packets of $cwnd' - cwnd$, which is denoted by R [packets], are now allowed to be sent. This can then result in R consecutive packet arrivals at node N_b. On the other hand, let BW^* be the minimum bandwidth between node 1 and $i-1$. Note that the BW^* can be obtained through the feedback information. Then, the transmission time of a packet, t_t, is $\frac{PS}{W'_b}$ (PS: packet size), which is larger than the packet arrival interval, t_a, equal to $\frac{PS}{BW^*}$. Thus, a new packet arrives before a packet is entirely transmitted, and so part of the packet is buffered at every packet arrival, which is given by $1-\frac{t_a}{t_t}$. Therefore, R consecutive packet arrivals cause the queue of packets to increase by BL_{N_b}, which is given by Eq. (6).

$$R[\text{pkts}] = (cwnd'[\text{pkts}] - cwnd[\text{pkts}]) \tag{5}$$

$$\begin{aligned} BL_{N_b}[\text{pkts}] &= R\left(1-\frac{W'_b}{BW^*}\right) \\ &= (cwnd' - cwnd)\frac{BW^* - W'_b}{BW^*} \end{aligned} \tag{6}$$

In order to prevent buffer overflow in node N_b, BL_{N_b} has to satisfy $BF_{N_b} \geq BL_{N_b}$. On the other hand, during the channel switching time, node N_b receives no packets from node i, while transmitting packets stored there, so that the buffer of node N_b will be empty by the end of channel switching. After the end of channel switching, node i begins to transmit packets stored back-to-back, which will arrive at node N_b and then be further forwarded by node N_b. In addition, the bandwidth of node N_b is smaller than that of node i. Thus, BF_{N_b} is expected not to be smaller than BF_i. To be on the safe side, assume that $BF_{N_b} = BF_i$ and BL_{N_b} satisfies inequality (7).

$$BF_i[\text{pkts}] \geq BL_{N_b} = (cwnd' - cwnd)\frac{BW^* - W'_b}{BW^*} \tag{7}$$

Therefore, $cwnd'$ considering BF_i is calculated by

$$cwnd'[\text{pkts}] = BF_i\frac{BW^*}{BW^* - W'_b} + cwnd \tag{8}$$

e) The $cwnd'$ can avoid buffer overflow in the switching node, but it does not take into account BDP which is known to be suitable for $cwnd$. Therefore, it could lead to congestion when $cwnd$ is updated to $cwnd'$. To solve this problem, the TCP sender of CoBA also calculates BDP using Eq. (9), and it updates $cwnd$ using Eq. (10), which corresponds to Eq. (2). Furthermore, the TCP sender of CRAHN can change its mode to slow start immediately after channel switching because of the non-updated $ssthresh$. As a result, buffer overflow could occur. Therefore, the TCP sender of CoBA updates $ssthresh$ using Eq. (11).

$$BDP[\text{pkts}] = \frac{W'_b[\text{b/s}] \cdot RTT_{new}[\text{s}]}{PS[\text{b/pkt}]} \tag{9}$$

$$cwnd''[\text{pkts}] = \min(BDP, cwnd') \tag{10}$$

$$ssthresh[\text{pkts}] = cwnd'' \tag{11}$$

f) When bottleneck node N_b is located on a path between TCP sender and node i, $cwnd$ and $ssthresh$ are determined by BDP only Eqs. (12), which corresponds to Eq. (2), and (13).

$$cwnd''[\text{pkts}] = \frac{W'_b[\text{b/s}] \cdot RTT_{new}[\text{s}]}{PS[\text{b/pkt}]} \tag{12}$$

$$ssthresh[\text{pkt}] = cwnd'' \tag{13}$$

6 Performance Evaluation

In this section, we implemented TCP CoBA into Qualnet 4.5 and examine how the change of bottleneck bandwidth and RTT due to channel switching affects single flow and multi flows of NewReno, Westwood+, CRAHN, and CoBA. As described in Fig. 4, the $cwnd$ is updated in different ways depending upon the location of the bottleneck node. Thus, in order to examine whether that works well, we also consider the case where the bottleneck node moves. The topology used is the same as that shown in Fig. 1. Unless otherwise stated, the same parameter values as in Table 1 are used. Note that we set the L2 buffer size to 100 packets in all the experiments, and $cwnd$ and the average throughput are used as a performance measure.

6.1 Case of Single TCP Flow

Here, we examine the impact of the change in bottleneck bandwidth and RTT on the throughput performance of single TCP flow.

6.1.1 Case where a node experiencing channel switching is a bottleneck node

As noted in Section 5, the sender of TCP CoBA takes into account both *BDP and the remaining buffer size in the switching node*. This section evaluates whether this control can prevent buffer overflow in the switching node. The network topology and simulation parameters are listed in Fig. 2a.

Fig. 5a shows $cwnd$ for CoBA, and Fig. 5b shows the queue length of node 3, which is not only switching node but also bottleneck node. In these figures, the bandwidth of Channel 3′ is 60 Mb/s. The sender of CoBA can increase the $cwnd$ to a suitable value when node 3 switches from low-rate Channel 3 to high-rate Channel 3′, while preventing buffer overflow in the switching node. Furthermore, CoBA can decrease the $cwnd$ to a suitable value when node 3 switches from high-rate Channel 3′ to low-rate Channel 3. As a result, CoBA can limit the queue length to an appropriate value.

When node 3 switches from low-rate Channel 3 to high-rate Channel 3′, the $cwnd$ of CRAHN is drastically increased and then immediately reduced. This is because the buffer in the relay nodes overflowed due to an excessive large $cwnd$ (See in Fig. 5b). As a result, throughput of CRAHN is degraded.

Fig. 5c shows the average throughput of all TCP variants when the bandwidth of Channel 3′ is changed from 20 Mb/s to 60 Mb/s. This figure indicates that the throughput of CoBA is higher than that of the other TCP variants. The advantage of CoBA becomes clear with an increasing gap of changes in the communication characteristics for each *channel switch*. In particular, when Channel 3′ is 60 Mb/s, CoBA can achieve up to a 200% improvement in performance compared to CRAHN.

6.1.2 Case where bottleneck node moves between the channel switching node and upstream node

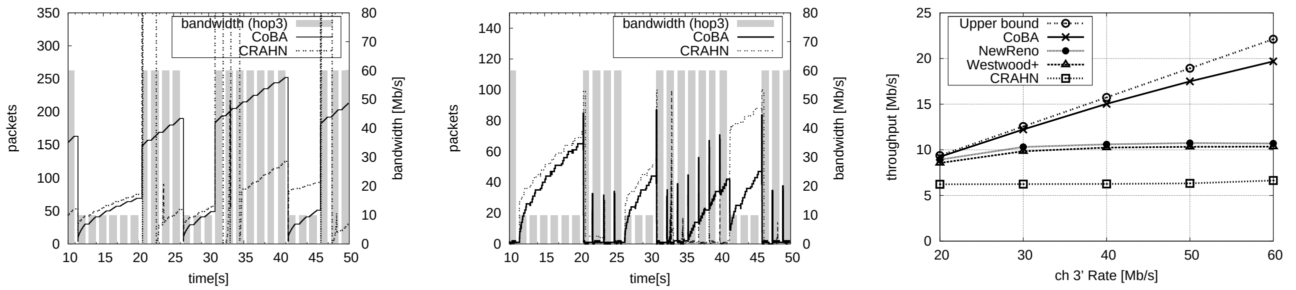
In our method, when some node experiences channel switching, TCP CoBA suspends data transmission to avoid buffer overflow. Therefore, nodes located between the channel switching node and the sender node, which will be called as "upstream nodes" from now on, can transmit packets queued in L2 buffer even during the period of channel switching. Consequently, a length of L2 buffer tends to be small immediately after channel switching. CoBA, hence, updates the $cwnd$ based on BDP when the bottleneck node is located at upstream nodes from the channel switching node (See f) in Fig. 4). Here, to examine how this works well, we consider the case where the bottleneck node moves between the channel switching node and its upstream node due to channel switching. Fig. 6a indicates bandwidth and link delay of each channel. In this evaluation, the bottleneck node is changed from node 2 to node 3 when channel switching node switches from high rate Channel 3 to low-rate Channel 3′, and vice versa.

Fig. 6b shows $cwnd$ for CoBA and CRAHN. Then, Figs. 6c and 6d show the queue length of node 2 and node 3, respectively. From these figures, the bandwidth of Channel 2 is 60 Mb/s. The sender of both TCP variants update $cwnd$ to the BDP when the node changes from low-rate Channel 3 to high-rate Channel 3′. However, the sender of CoBA is able to increase the $cwnd$ to a suitable value, while preventing buffer overflow in node 2 (see Fig. 6c). On the other hand, $cwnd$ of CRAHN increases drastically and thus buffer overflow at relay node occurs, which is the same as in Fig. 5a. This is because CoBA is totally different with CRAHN in a way of measuring RTT_{new}; i.e., CoBA excludes the queuing delay at relay nodes, while CRAHN includes it in RTT_{new}. As a result, RTT_{new} measured by CoBA is much smaller than that of CRAHN; thereby avoiding buffer overflow due to the small BDP successfully (see Figs. 6c and 6d).

Fig. 6e shows the average throughput of all TCP variants when the bandwidth of Channel 2 is changed from 20 Mb/s to 60 Mb/s. The BDP is drastically changed due to channel switching to the large bandwidth of Channel 2, i.e., 60 Mb/s. CoBA updates the $cwnd$ to suitable value in response to the drastic change in the BDP. In particular, when Channel 2 is 60 Mb/s, CoBA can achieve up to 200% improvement in performance, compared to CRAHN.

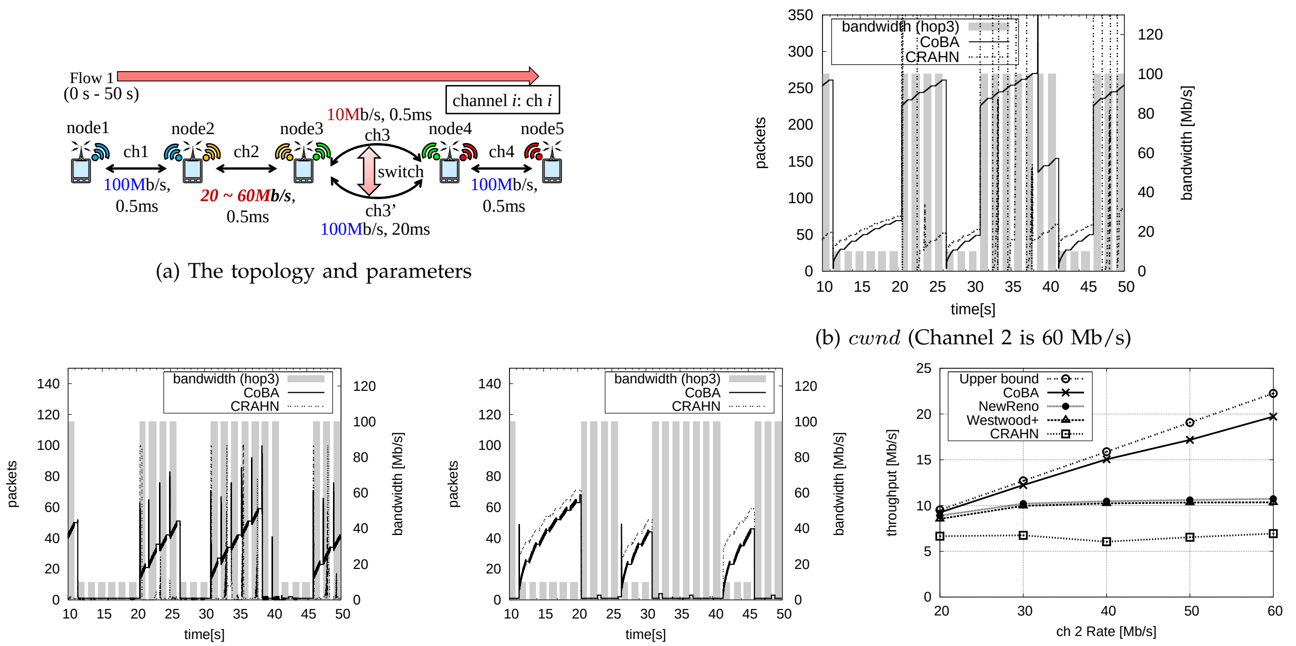
6.2 Case of Multiple TCP Flows

Here, we examine how the change in bottleneck bandwidth and RTT affects the performance of multiple TCP flows. The simulation network is illustrated in Fig. 7, in which node 1 and node 2 send bulk data to node 4 from 0 s to 40 s (flow 1), and node 5 from 10 s to 50 s (flow 2), respectively.



(a) $cwnd$ (Channel 3′ is 60 Mb/s) (b) Queue length (node 3) (c) Averaged throughput

Fig. 5. Bottleneck node is switching node (The bandwidth of Channel 3’ set at 20 to 60 Mb/s)



(a) The topology and parameters (b) $cwnd$ (Channel 2 is 60 Mb/s)

(c) Queue length (node 2) (d) Queue length (node 3) (e) Averaged throughput

Fig. 6. Bottleneck node is switching node or upstream node (The bandwidth of Channel 2 set at 20 to 60 Mb/s)

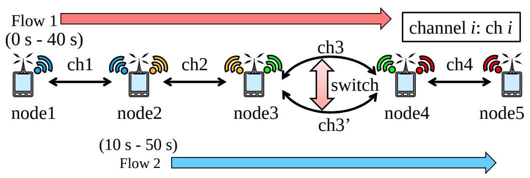


Fig. 7. Simulation model

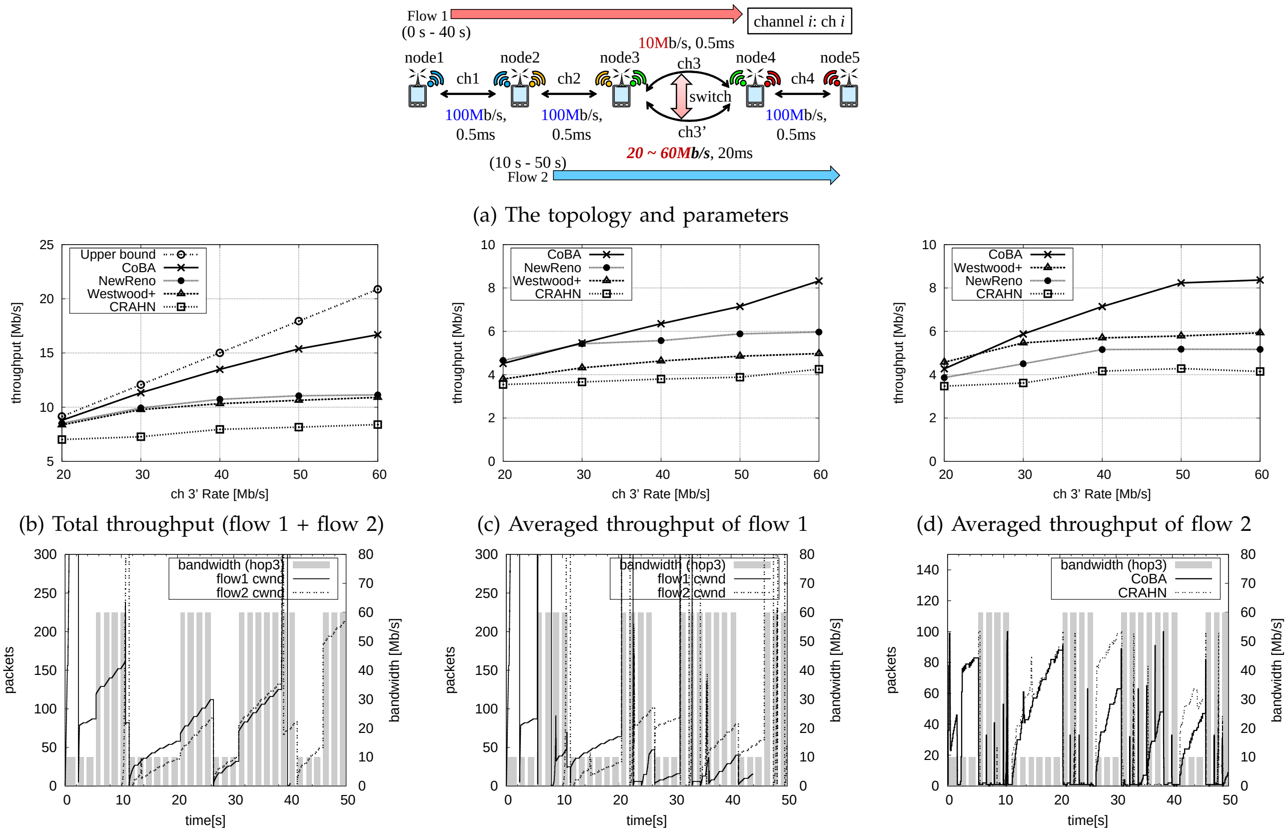
6.2.1 *Case where the shared bottleneck node of both flows experiences channel switching*

This section evaluates the performance of each of two TCP flows. The network topology and simulation parameters are listed in Fig. 8a. Note that the bottleneck node of two TCP flows is fixed to node 3, irrespective of channel switching.

Fig. 8b shows the total throughput of all TCP variants when the bandwidth of Channel 3′ is changed from 20 Mb/s to 60 Mb/s. This figure indicates that the throughput of CoBA is obviously higher than that of the other TCP variants. The advantage of CoBA becomes clear with an increasing gap of changes in the communication characteristics for each channel switch. In particular, when Channel 3′ is 60 Mb/s, CoBA can achieve up to a 110% improvement in performance compared to CRAHN.

Fig. 8c and Fig. 8d show the average throughput of each TCP flow when the bandwidth of Channel 3′ is changed from 20 Mb/s to 60 Mb/s. These figures indicate that the throughput of both flow 1 and flow 2 obtained by CoBA is obviously higher than that of the other TCP variants. On the other hand, the throughput of both flows of the CRAHN is the lowest.

Fig. 8e and Fig. 8f show how the $cwnd$ changes in CoBA and CRAHN and Fig. 8g shows the queue length of node 3, when the bandwidth of Channel 3′ is 60 Mb/s. From Figs. 8e and 8g, the senders of CoBA increase their $cwnd$ to a suitable value when switching from low-rate Channel 3 to high-rate Channel 3′, while preventing buffer overflow in the switching node. Furthermore, CoBA is able to decrease the $cwnd$



(a) The topology and parameters

(b) Total throughput (flow 1 + flow 2)

(c) Averaged throughput of flow 1

(d) Averaged throughput of flow 2

(e) Behavior of $cwnd$ for TCP CoBA

(f) Behavior of $cwnd$ for TCP CRAHN

(g) Queue length (node 3)

Fig. 8. 2 TCP flows share the bottleneck node experiencing channel switching.

of two flows to a suitable value when switching from high-rate Channel 3′ to low-rate Channel 3.

In contrast, from Figs. 8f and 8g, we can see that the $cwnd$s of two flows in CRAHN change in a different way. Namely, these values are drastically increased and then immediately reduced when the nodes switch from low-rate Channel 3 to high-rate Channel 3′. This is because the buffer in the relay nodes overflows due to an excessively large $cwnd$. As a result, the throughput of CRAHN is degraded.

When the bottleneck node experiencing channel switching is shared by two TCP flows, each of TCP flows achieves almost the same throughput performance. Therefore, in such case, we can say that fairness can be achieved, irrespective of the type of TCP variants, but this evaluation indicates that, compared with CRAHN, CoBA improves the throughput of multiple TCP flows.

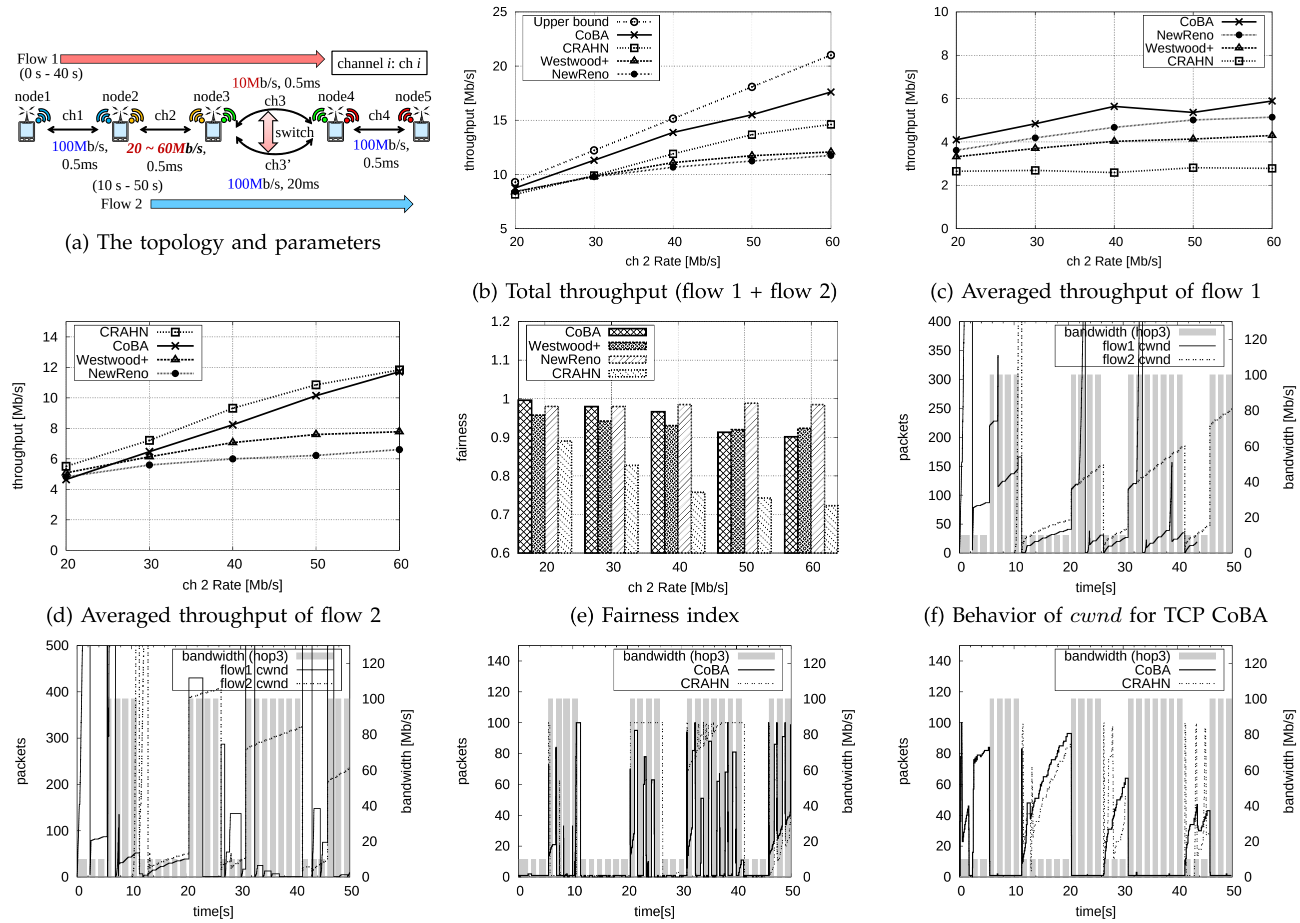
6.2.2 Case where the bottleneck node changes between the switching node and upstream node at every channel switching

Next, we investigate how the change in bottleneck nodes impacts the throughput performance achieved by TCP CoBA. Topology and parameter settings are listed in Fig. 9a. As shown in this figure, the bottleneck node swaps between node 3 (when using Channel 3) and node 2 (when using Channel 3′).

Fig. 9b shows the sum of throughput of two TCP flows achieved by all TCP variants when the bandwidth of Channel 2 is changed from 20 Mb/s to 60 Mb/s. This figure indicates that the total throughput of CoBA is clearly higher than that of the other TCP variants. On the other hand, the throughput of CRAHN is the second highest in all TCP variants.

Fig. 9c and Fig. 9d show the mean throughput of each flow. These figures indicate that for CRAHN, the throughput of flow 2 is higher than that of the flow 1. This throughput gap of each flow in CRAHN is the largest, especially the throughput of flow 2, being 4 times as high as that of flow 1. On the other hand, the gap in CoBA is obviously limited. Fig. 9e shows the Jain's fairness index [19] of all TCP variants. From this figure, we can remark that the fairness index achieved by CoBA is significantly higher than that of CRAHN, while providing excellent throughput performance (see Fig. 9b). Next, we focus on the behavior of $cwnd$ for each flow in order to clarify the reason of the gap.

When the bandwidth of Channel 3′ is 60 Mb/s, Figs. 9f and 9g show behavior of $cwnd$ controlled by CoBA and CRAHN, and Figs. 9h and 9i show the queue length of node 2 and node 3. From Fig. 9f, CoBA of flow 1 increases the $cwnd$ and then immediately reduces when node 3 and node 4 switch from low-rate Channel 3 to high-rate Channel 3′. When Channel 3′ is used, the bottleneck node is node 2, which is also the



(a) The topology and parameters

(b) Total throughput (flow 1 + flow 2)

(c) Averaged throughput of flow 1

(d) Averaged throughput of flow 2

(e) Fairness index

(f) Behavior of *cwnd* for TCP CoBA

(g) Behavior of *cwnd* for TCP CRAHN

(h) Queue length (node 2)

(i) Queue length (node 3)

Fig. 9. Bottleneck node is swapped between the switching node and its upstream node by channel switching.

sender of flow 2. Hence, data transmission from node 2 (i.e., flow 2) is stopped successfully, whereas data transmission from node 1 (i.e., flow 1) is not, thereby causing buffer overflow at node 2 (See Fig. 9h).

From Fig. 9g, we can see that the *cwnd* of flow 2 in CRAHN is larger than that of CoBA. On the other hand, the *cwnd* of flow 1 in CRAHN is smaller than that of CoBA. This is because the buffer overflows of flow 1 are drastically increased at CRAHN because *cwnd* of flow 2 is too large.

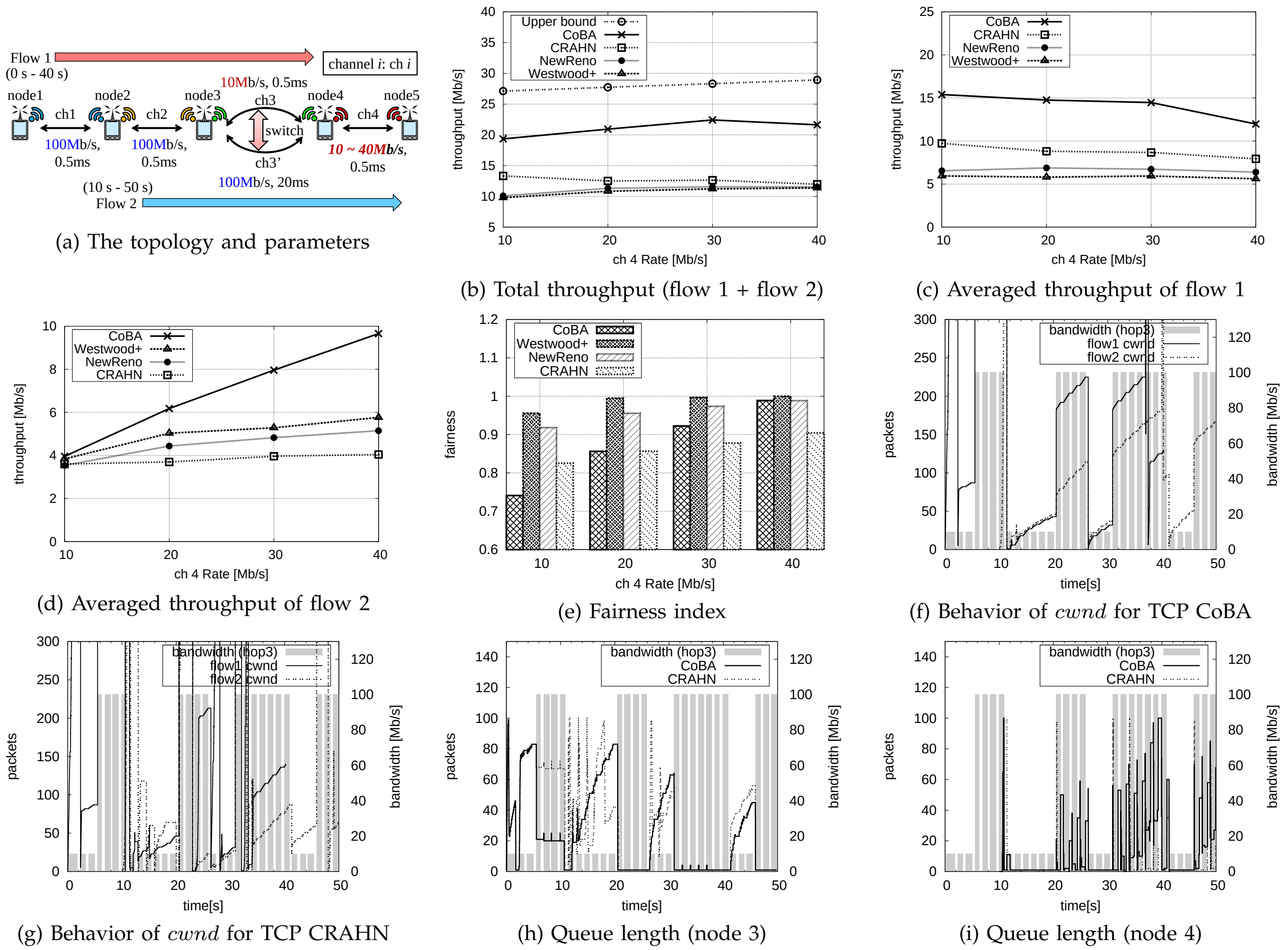
From these results, we can remark that when the bottleneck node is changed to upstream node due to channel switching, CoBA updates the *cwnd* based on BDP as in CRAHN, and thus packet losses cannot be avoided completely (see Figs. 9h and 9i). As a result, the fairness index value of CoBA is gradually decreased with the increase in the bandwidth of channel 2, but CoBA provides good throughput performance.

6.2.3 The bottleneck node of each flow is rotated between the shared and the isolated in response to channel switching

In the above sections, the bottleneck node of each flow is always shared. In this section, the network topology and parameters are illustrated in Fig. 10a. Hence, the bottleneck node of flow 1 is node 3, while that of flow 2 is node 4 when using Channel 3′ for hop 3. On the other hand, the bottleneck node of each flow is node 3 (i.e., shared) when using Channel 3 for hop 3. Furthermore, the bandwidth of Channel 4 is changed from 10 Mb/s to 40 Mb/s.

Fig. 10b shows the average throughput of two TCP flows achieved by all TCP variants when the bandwidth of Channel 4 is changed from 10 Mb/s to 40 Mb/s. This figure indicates that the throughput of CoBA clearly increases with an increasing bandwidth of Channel 4. In contrast, the throughput of CRAHN slightly decreases with an increasing bandwidth of Channel 4. To clarify the reason of this behavior, we next examine how the throughput and the *cwnd* of each flow are changed.

Fig. 10c and Fig. 10d show the average throughput of each flow when the bandwidth of Channel 4 is changed from 10 Mb/s to 40 Mb/s. These figures indicate that in CoBA, the throughput of flow 2 drastically increases with an increasing bandwidth of Channel 4. On the contrary, the throughput of flow 1 slightly decreases with an increasing bandwidth of Channel 4. Only flow 2 passes through Channel 4, so that the throughput of flow 2 is increased in proportion



(a) The topology and parameters

(b) Total throughput (flow 1 + flow 2)

(c) Averaged throughput of flow 1

(d) Averaged throughput of flow 2

(e) Fairness index

(f) Behavior of *cwnd* for TCP CoBA

(g) Behavior of *cwnd* for TCP CRAHN

(h) Queue length (node 3)

(i) Queue length (node 4)

Fig. 10. Bottleneck node is swapped between the shared and the isolated in response to channel switching.

to the increase in the bandwidth of Channel 4. As a result, as shown in Fig. 10e, the fairness index value of CoBA is increased at constant inclination with the increase in the bandwidth of Channel 4. Although other TCP variants generally show the high fairness index value, we can see that the throughput of flow 1 with larger bottleneck bandwidth (than flow 2) always shows relatively low value. On the other hand, CoBA can utilize its large bottleneck bandwidth efficiently, thereby increasing the throughput of flow 1 clearly.

Here, we focus on the *cwnd*s of each flow in CoBA (Fig. 10f). From Fig. 10f, the *cwnd* of flow 1 is decreased to 1 at 38 s because TCP timeout caused by buffer overflow. When Channel 3 is used for hop 3, the bottleneck nodes of both flows are shared at node 3. On the other hand, when Channel 3′ is used, the bottleneck node of flow 1 is node 3, and that of flow 2 is node 4. Hence, in this case, the *cwnd* of flow 2 is increased with an increasing bandwidth of Channel 4. As a result, the packets of flow 1 cause buffer overflow at node 3 with increasing of Channel 4's bandwidth, as shown in Fig. 10h.

From Fig. 10d, the throughput of flow 2 in CRAHN does not increase even if bandwidth of Channel 4 is increased. Fig. 10g shows the behavior of *cwnd* for CRAHN when bandwidth of Channel 4 is 40 Mb/s. The bottleneck node of flow 1 is node 3 and the bottleneck node of flow 2 is node 4 when Channel 3′ is used for hop 3. The *cwnd*s of both flows are drastically increased and then immediately reduced when the nodes switch from low-rate Channel 3 to high-rate Channel 3′. This is because the *cwnd*s of both flows are updated to a large value, the packets of both flows are overflowed at the buffers of each bottleneck node (see Figs. 10h and 10i).

This evaluation indicates that, compared with CRAHN, CoBA improves the throughput even if the bottleneck nodes of each flow are isolated.

6.3 Scalability evaluation in a large-scale network

Finally, to evaluate the scalability of TCP CoBA, we assume a large-scale network of chain topology consisting of 100 nodes in which five TCP flows are transmitted simultaneously (See Fig. 11a). Note that since five nodes are randomly selected from node 1 to 49 as the sender nodes, while another five nodes are randomly selected from node 51 to 100 as the receiver nodes, the number of hops is drastically different among coexisting flows. Then, to examine the

TABLE 3
Fairness index (5 flows).

TCP CoBA	0.9902	TCP Westwood	0.9793
TCP NewReno	0.9770	TCP CRAHN	0.9815

communication performance in the worst case where all five flows with different hop numbers experience channel switching at some node simultaneously, we assume that a node located at the center of network (node 50) experiences channel switching.

Suppose that one of two channels (Channel 1 and Channel 2) with different link bandwidth and delay is used at a time on a link between node 50 and node 51 to examine communication performance when the characteristics of channel used is drastically changed: Channel 1 (bandwidth 60 Mb/s, delay 5 ms) and Channel 2 (bandwidth 10 Mb/s, delay 0.01 ms). For other links, bandwidth is varied from 80 Mb/s to 100 Mb/s, while delay is varied from 0.01 ms to 5 ms. Other parameters employed here are listed in Table 1.

Figure 11b shows how the total throughput is changed, when the number of TCP flows is varied from 1 to 5. From this figure, we can see that with the increase in the number of TCP flows, the total throughput of CoBA is increased by more than 40%, compared with other TCP variants. In contrast, the total throughput of CRAHN indicates the worst performance and almost constant value, irrespective of the increase in the TCP flows.

Next, to evaluate the fairness for five TCP flows achieved by all TCP variants, we show the achievable fairness index in Table 3. From Table 3, we can remark that CoBA indicates the highest fairness index value, whereas NewReno shows the worst value. To clarify this reason, Fig. 11c shows how the throughput is varied with the change in the number of hops. From this figure, we can see that NewReno with small number of hops attains high throughput performance, while that with large number of hops suffers low throughput performance. In contrast, flows of CoBA with small hop numbers intentionally limit the amount of transmitted data for flows with large hop numbers. Therefore, CoBA can achieve the highest fairness index value among multiple flows.

6.4 Discussion on fairness of TCP CoBA

Here, we focus on fairness issue between TCP CoBA and the existing TCP. First, as mentioned before, we design CoBA to be exactly same with NewReno except for the periods of channel switching. From this, fair use of network resources can be guaranteed between them unless channel switching occurs.

When the channel is switched, the existing TCP determines the amount of transmitted data without awareness of the change in communication characteristics. Therefore, burst packet losses could occur due to buffer overflow and thus the $cwnd$ is reduced drastically. If multiple TCP flows coexist, it will become more severe due to the interference with each other.

In contrast, CoBA updates $cwnd$ appropriately based on the feedback information including the number of coexisting flows, immediately after the channel switching. This indicates that each CoBA flow does not affect the coexisting TCP flows' performance, regardless of the types of coexisting TCP, e.g., CoBA and the conventional NewReno. Therefore, we can say that negative impact by introducing CoBA can be limited extremely.

7 Conclusions

This paper focused on transport protocols in cognitive radio network which select a channel from a wide spectrum range. We then examined how the transport protocol and the relay node should be redesigned to make an efficient use of available wireless resource.

First, the TCP performance of existing TCP variants such as TCP CRAHN was compared. Simulation results showed that CRAHN outperforms all the other TCP variants considered, but cannot attain good performance when communication characteristics drastically change due to channel switching. The issue arises from excessively increasing window size after the channel switching in the above context, which leads to many consecutive arrivals to relay nodes and eventual buffer overflow. Thus, TCP sender should consider where the bottleneck node is located and how much buffer resource is available in the bottleneck node, in addition to BDP.

Next, to resolve this issue, this paper proposed TCP CoBA as a new transport protocol. Each SU is equipped with a GPS function and thus can synchronously execute periodical sensing. The sender in TCP CoBA updates the $cwnd$ when either the bottleneck bandwidth or RTT is changed by over 20% after channel switching. Furthermore, it also takes into account both the remaining buffer space and BDP. Through simulation experiments, it was shown that, compared with TCP CRAHN, TCP CoBA drastically improves throughput performance.

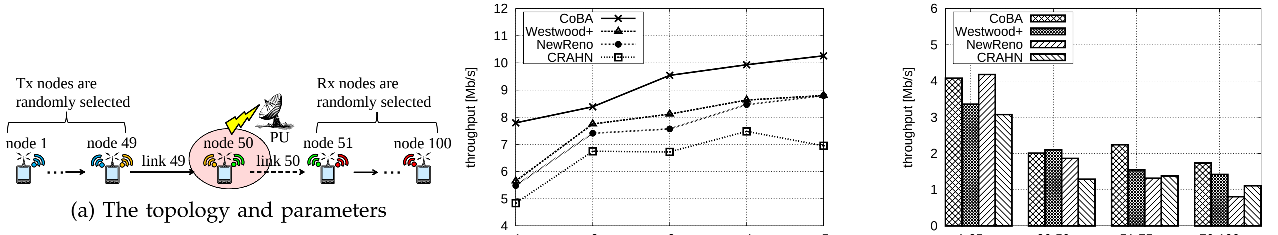
References

[1] Y. R. Kondareddy and P. Agrawal, "Effect of Dynamic Spectrum Access on Transport Control Protocol Performance," in *Proc. of IEEE GLOBECOM*, 2009, pp. 1-6.

[2] A. M. R. Slingerland, P. Pawelczak, R. V. Prasad, A. Lo, and R. Hekmat, "Performance of Transport Control Protocol Over Dynamic Spectrum Access Links," in *Proc. of IEEE DySPAN*, 2007, pp. 486-495.

[3] W. Kim, A. Kassler, and M. Gerla, "TCP Performance in Cognitive Multi-Radio Mesh Networks," in *Proc. of CogArt*, 2011, p. 44.

[4] M. D. Felice, K. R. Chowdhury, W. Kim, A. Kassler, L. Bononi, "End-to-end Protocols for Cognitive Radio Ad Hoc Networks: An Evaluation Study," *Elsevier Performance Evaluation*, Vol. 68, No. 9, pp. 859-875, 2011.



(a) The topology and parameters (b) Total throughput (5flows) (c) Throughput vs. Hop counts (5flows)

Fig. 11. Scalability evaluation (5 flows).

[5] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," in *Proc. of ACM MOBICOM*, 2001, pp. 287-297.

[6] L. A. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno and Vegas TCP congestion control," *ACM Computer Communication Review*, Vol. 34, No. 2, pp. 25-38. 2004.

[7] K. Sundaresan, V. Anantharaman, H. Y. Hsieh, and R. Sivakumar, "ATP: A Reliable Transport Protocol for Ad Hoc Networks," *IEEE Trans. On Mobile Computing*, Vol. 4, No. 6, pp. 588-603, 2005.

[8] J. Liu and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks," *IEEE Journal on Sel. Areas of Comm.*, Vol. 19, No. 7, pp. 1300-1315, 2001.

[9] D. Sarkar and H. Narayan, "Transport Layer Protocols for Cognitive Networks," in *Proc. of IEEE INFOCOM Computer Communication Workshop*, 2010, pp. 1-6.

[10] C. Luo, F. R. Yu, H. Ji, and V. C. Leung, "Optimal Channel Access for TCP Performance Improvement in Cognitive Radio Networks," *Springer Wireless Networks*, Vol. 17, No. 2, pp. 479-492, 2011.

[11] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. of ACM SIGCOMM*, 2002, pp.89-102.

[12] K. R. Chowdhury, M. D. Felice, I. F. Akyildiz, "TCP CRAHN: A Transport Control Protocol for Cognitive Radio Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Vol. 12, No. 4, pp. 790-803, 2013.

[13] A. Al-Ali and K. R. Chowdhury, "TFRC-CR: An Equation-based Transport Protocol for Cognitive Radio Networks," *Elsevier Ad Hoc Networks Journal*, Vol. 11, No. 6, pp. 1836-1847, 2013.

[14] K. Tsukamoto, S. Matsuoka, O. Altintas, M. Tsuru, Y. Oie, "Distributed Channel Coordination in Cognitive Wireless Vehicle-to-Vehicle Communications," in *Proc. of WAVE*, 2008, CD-ROM.

[15] K. Chen, Y. Xue, and K. Nahrstedt, "On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks," in *Proc. of IEEE ICC*, pp.1080-1084, 2003.

[16] F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki, "Measuring queue capacities of IEEE 802.11 wireless access points," in *Proc. of IEEE BROADNETS*, pp. 846-853, 2007.

[17] W. Y. Lee and I. F. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks," *IEEE Trans. on Wireless Communications*, Vol. 7, No. 10, pp. 3845-3857, 2008.

[18] M. Wellens, A. Baynast, and P. Mahonen, "Performance of dynamic spectrum access based on spectrum occupancy statistics," *IET Communications*, Vol. 2, No. 6, pp. 772-782, 2008.

[19] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation and shared computer system," Tech. Rep. DEC-TR-301, 1984.

[20] W. Lim, D. Kim, Y. Suh, J. Won, "Implementation and performance study of IEEE 802.21 in integrated IEEE 802.11/802.16e networks," *Elsevier Computer Communications*, Vol. 32, No. 1, pp. 134-143, 2009.

Kazuya Tsukamoto received his D.E. degree in computer science from Kyushu Institute of Technology (KIT), Japan, in 2006. From April 2006 to March 2007, he was a Japan Society for the Promotion of Science (JSPS) Research Fellow at KIT. In 2007, he was an assistant professor in the Department of Computer Science and Electronics, KIT, and then has been an associate professor in the same department since April 2013. His research interests include performance evaluation of computer networks and wireless networks. He is a member of the ACM, the IPSJ, and the IEICE.

Shun Koba received the B.E., and M.E. degrees from Kyushu Institute of Technology in 2011, 2013, respectively. Since April 2013, he has been employee in KDDI CORPORATION. His research interests include performance evaluation of wireless networks, especially cognitive radio networks. He is a member of the IEICE.

Masato Tsuru received B.E. and M.E. degrees from Kyoto University, Japan in 1983 and 1985, respectively, and then received his D.E. degree from Kyushu Institute of Technology, Japan in 2002. He worked at Oki Electric Industry Co., Ltd., Nagasaki University, and Telecommunications Advancement Organization of Japan. In 2003, he joined the Department of Computer Science and Electronics, Kyushu Institute of Technology as an Associate Professor, and then has been a Professor in the same department since April 2006. His research interests include performance measurement, modeling, and management of computer communication networks. He is a member of the ACM, IEICE, IPSJ, and JSSST.

Yuji Oie received the B.E., M.E., and D.E. degrees from Kyoto University in 1978, 1980, and 1987, respectively. From 1980 to 1983 he worked at Nippon Denso Company Ltd., Kariya, Japan. From 1983 to 1990 he was with the Department of Electrical Engineering, Sasebo College of Technology. From 1990 to 1995 he was an associate professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, KIT. From 1995 to 1997 he was a professor at the Information Technology Center, NAIST. Since April 1997 he has been a professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, KIT. His research interests include performance evaluation of computer communication networks, high-speed networks, and queuing systems. He is a fellow of the IPSJ and the IEICE, and a member of the the ACM.