Regular Paper

Combining Local Channel Selection with Routing Metrics in Multi-channel Wireless Mesh Networks

Hiromichi Kanaoka[1] Takuya Yoshihiro[2,a)]

Received: March 24, 2014, Accepted: December 3, 2014

Abstract: WMNs (Wireless Mesh Networks) using IEEE 802.11 have been deeply studied to extend the area of coverage of the Internet. A typical approach to implement this kind of WMNs is to use dynamic metrics (e.g., ETX) over link-state routing protocols (e.g., OLSR). Although studies have demonstrated clarified that the approach performs well, there is still room for improve. In this paper, we first point out that the dynamic metrics by nature cannot pursuit rapid change in link quality, which prevents routing protocols to choose the best forwarding paths at every moment. To complement this drawback of the dynamic metric, we propose a local switching mechanism of forwarding channels for multi-radio, multi-channel WMNs, which works in combination with dynamic metrics. Our evaluation showed that the proposed method improves throughput and stability of communications when it works with dynamic metrics.

Keywords: Wireless Mesh Network, IEEE802.11, dynamic metrics, Multi Channel, Local Channel Selection

1. Introduction

Recently, WMNs (Wireless Mesh Networks) have been deeply studied as a new wireless infrastructure that inexpensively expands the coverage area of the Internet using multi-hop wireless communications among stationary nodes. However, since WMNs use radio for communications, there is a problem with heavy interference that significantly degrades communications performance. To achieve acceptable communication speed in WMNs, technologies to avoid interference have a significant importance.

As an access control method in WMNs, we have two choices, STDMA (Spatial Time Division Multiple Access) and CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance). STDMA [1] is a kind of communication model where channels are divided into several small slots and radio transmissions are scheduled on slots in advance to avoid any collision between radios. Several scheduling algorithms [2] have been proposed from various optimization approaches to realize efficient communications. However, STDMA has several drawbacks; (i) it is not good at treating changing traffic demands, (ii) it requires accurate synchronization among nodes, and (iii) it cannot share radio resources with other popular access control methods such as IEEE 802.11.

Recently, to treat changing traffic demands, several CSMA algorithms have been developed such as Q-CSMA [3] and VMC-CSMA [4] for multi-hop wireless mesh networks. These algorithms achieve multiple accesses without any collision of transmissions in multi-hop mesh networks. However, because they also assume slotted systems like TDMA that requires precomputed slot schedules that do not include any collisions, problems (ii) and (iii) still remains.

In contrast, in spite of considerable number of studies [5], the performance of WMNs that works over the traditional non-slotted CSMA/CA algorithm is quite limitted due to severe interference coming from hidden terminal problems. Nevertheless, this kind of WMNs is still useful because it does not include drawbacks (i)-(iii) and also because IEEE 802.11 devices are widely used and easily obtained around the world. If reliable and high-throughput WMNs are realized over IEEE 802.11, we would obtain a flexible and easy-to-use wireless infrastructure that supports various useful services and applications.

One of the typical ways to construct WMNs on IEEE 802.11 is to use link-state routing protocols such as OLSR [6]. Namely, in such networks, a routing table works at each router to deliver packets to any destination in the network. Note that, since WMNs use a wireless medium, the quality of links significantly changes as time passes. Therefore, we typically use a dynamic metric such as ETX [7] to take the instability of wireless links into account. Dynamic metrics are the values that represent the quality of each wireless link. Because dynamic metrics are computed in real-time such that smaller values represent higher quality, routing protocols such as OLSR always try to choose the best paths by computing the shortest-paths over dynamic link metrics. Also for Multi-channel environments, several dynamic metrics such as WCETT [8] have been proposed, which further reduce interference in networks by taking advantage of multi-channel availability.

Although it is well-known that routing metrics improve communication performance, we have to point out that communications over WMNs are not sufficiently stable from the viewpoint of users' experience; in fact, their throughput fluctuates signifi-

[1] Graduate School of Systems Engineering, Wakayama University, Wakayama 640–8510, Japan
[2] Faculty of Systems Engineering, Wakayama University, Wakayama 640–8510, Japan
a) tac@sys.wakayama-u.ac.jp

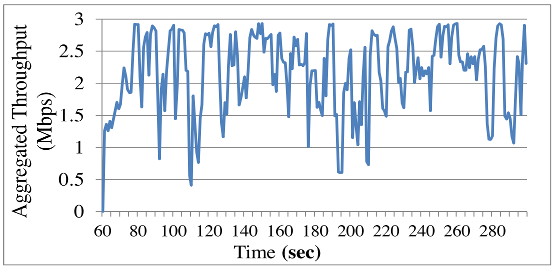

Fig. 1 Fluctuation of throughput over time in a multi-channel WMNs that employs a dynamic metric.

cantly. See **Fig. 1** for our simulation result. This result shows the transition of the total throughput of four CBR (Constant Bit Rate) flows measured every second under dynamic metric WCETT with 5 channels*1. Although the result is the sum of 4 flows, the throughput still fluctuates largely every second such that the performances between the best 10-seconds periods and the worst 10-seconds periods differ by more than double, showing that the communication performance is unstable.

The main cause of this instability can be explained by the delay of dynamic metrics when pursuing link quality. Note that, in order to avoid frequent changes of forwarding paths, link metrics are designed to be insensitive. In fact, dynamic metrics are typically computed through observing a link for a relatively long time period. Moreover, it takes considerable time to propagate metric values to every node in the network. Consequently, dynamic metrics represent the link quality of at least several seconds before. This means that dynamic metrics do not always lead to the best paths based on an up-to-date network state, which is a cause of the instability of networks.

In this paper, we introduce a local channel selection mechanism that works in combination with a dynamic metric. The proposed method (i.e., the local channel selection) helps with the selection of better forwarding paths by following up-to-date traffic state of networks. Specifically, in the proposed method, each node locally decides its forwarding channel (i.e., link) through quick local observations of the link quality, to reflect up-to-date link quality on metrics without delay. In contrast with dynamic metrics, which follows slow network-wide dynamics of network state, the proposed method follows quick local transitions of link quality. This local channel decision complements the drawback of dynamic metrics and improves stability as well as throughput performance in multi-channel WMNs. This paper is organized as follows. In Section 2, we describe related work and the problems with past work. In Section 3, we describe the proposed method in detail, and give the results from simulations in Section 4. Finally, we conclude this paper in Section 5.

2. Related Work and the Problem

Several approaches to multi-channel WMNs with CSMA/CA for improving communication performance have appeared in the literature. Several studies treat channel assignment algorithms that reduce interference between nodes as a way for efficient routing in multi-channel WMNs. These kinds of studies typically formulates the problem of assigning a channel for each link as the coloring problem over the conflict graph, and minimize the interference between links [9], [10], [11]. This optimization approach is very effective if a sufficient number of channels is available, e.g., if TDMA is employed in the MAC layer. The number of available channels, however, is quite limited as long as we assume WMNs with populated IEEE 802.11 MAC.

The dynamic routing metric quantifies the quality of links in order to compute forwarding paths using high-quality links. ETX, which computes the average retransmission count of each link from the successful reception ratio of periodically transmitted *Probe* packets on the link, is the best known dynamic metric. ETT [8] extends ETX by computing the average transmission time of packets on each link while taking link speed into account. WCETT [8] is an extension of ETT, designed for multi-channel networks, which consider intra-flow interference. Following these representative dynamic metrics, many routing metrics for multi-channel WMNs have been proposed from various approaches [12], [13], [14].

Although dynamic metrics actually improve the performance of networks, they have a drawback that they do not follow the rapid fluctuation of link quality. This results in degradation of network performance. In fact, this gap between the link metrics used in route computation and the actual link quality prevents us from useing the optimal paths at every instant. There are two reasons for this:

- Because a routing metric is estimated by observing the corresponding link for a certain period of time, naturally there is a gap between the computed metric and the actual link quality.
- Due to the nature of periodic transmission of control messages in the link-state routing protocols, we inevitably have a significant delay in notifying distant nodes of routing metrics. This also prevents nodes from using the newest link quality values for its shortest-path computation.

To make matters worse, this delay significantly degrades network performance because link quality (i.e., dynamic metrics) may transit rapidly in seconds. In dynamic metric environments, most dynamic metrics take traffic load and congestion into account. Thus, when dynamic metrics invoke path changes, the traffic load may dramatically change, resulting in a transition of dynamic metrics again. This circular relationship makes the transition of traffic rapid and dynamic metrics difficult to follow the traffic transition. Consequently, the performance of networks become worse. To follow the transition of traffic with less delay is essential in choosing better forwarding paths that improve communication performance.

In this study, we assume WMNs over IEEE 802.11 MAC, although in principle the throughput is very limited [15]. We focus on the instability of communications in WMNs with dynamic metrics, and try to improve it using local decisions on forwarding channels in combination with a conventional dynamic metric. Specifically, we propose a method that switches the forwarding channel based on short-period measurements of congestion, in order to complement the shortcoming of routing metrics, i.e., to

*1 More specifically, Fig. 1 is obtained from Qualnet ver. 5.0 simulator where we generated four 750 kbps CBR flows diagonally on the 5×5 grid topology presented in Fig. 5. Other parameters are the same as Scenario 1 described in Section 4.1.

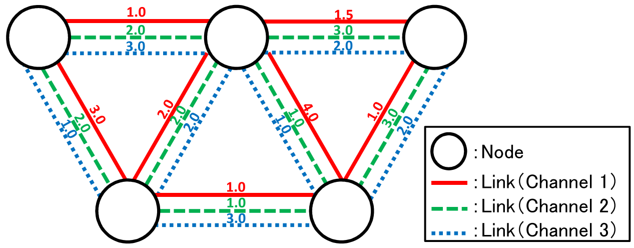

Fig. 2 An example of a topology with 3 channels (i.e., $k = 3$).

improve the response speed of paths selection against rapid transition of link quality.

3. Proposed Method

3.1 Assumptions in Our Multi-channel WMNs

We assume multi-radio, multi-channel networks deploying IEEE 802.11 MAC where the number of channels is the same as that of radio. Thus, we suppose that every node has k radios, each of which corresponds to each channel $c \in C$, where C is the set of available channels. We further assume that proactive routing protocols such as OLSR as well as a dynamic metric are deployed to compute forwarding paths. Therefore, to each neighbor, the channel used to transmit frames is selected from C based on the shortest-path computation.

For example, if $k = 3$, there are three links between each pair of neighbor nodes in the network. (**Fig. 2** shows an example topology of the case $k = 3$.) Furthermore, forwarding paths are chosen through the shortest-path computation over dynamic metrics, where an independent routing metric is computed for each of the k links. Our method works with any dynamic metric proposed so far, including ETX, ETT, and WCETT. Dynamic metrics generally reflect interference among links. Consequently, traffic fluctuation is reflected on dynamic metrics, routing paths are changed accordingly, and this in turn changes traffic patterns. Through this arrangement, nodes in a network always attempt to choose better paths that involve less interference.

3.2 The Idea

As described in Section 2, dynamic metrics inevitably include delay against the real link-quality values due to two reasons:

(1) metrics are measured by observing links during a certain period of time, and,

(2) metrics are propagated over networks via periodically forwarded messages.

Thus, we try to complement dynamic metrics by introducing local switching of forwarding channels. Consider a next-hop node for a destination computed by the deployed routing protocol. Thus, the node has several links to the next-hop node. (Since a link corresponds to a unique channel in C, we use the terminology "links" instead of "channels," hereafter.) In our method, by instantly checking the quality of those links connecting to the next-hop node, we would select the best one for the next-hop transmission. Since this local switching follows a rapid change in link quality, we can make more-optimal path selections by following both the rapid link-quality transition (with the proposed method) and the network-wide trend of communication quality (with dynamic metrics).

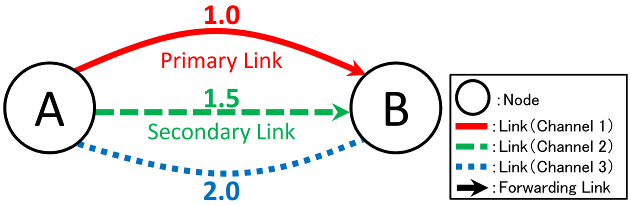

Fig. 3 An example of forwarding link selection.

3.3 Preparing the Secondary Link

The proposed method selects two links that are possibly used as the forwarding link. Every node u has a *primary link* for each neighbor v that is computed as the next-hop link through the shortest-path computation. Thus, the primary link (u, v, c_1), which is the link from u to v using channel c, has the lowest metric among all links from node u to v. For each primary link (u, v, c_1), we select the *secondary link* (u, v, c_2) such that it is the link from u to v with the second-lowest metric. We forward packets using these two links adaptively.

Figure 3 shows an example of a pair of neighboring nodes, each of which have 3 channels, i.e., $k = 3$. Here, node A chose the minimum-metric link as the primary link, and chose the second minimum-metric link as the secondary link in order to forward packets to node B. Each node holds the primary and the secondary links, and uses the secondary link to forward packets as soon as the node detects congestion on the primary link.

3.4 Detecting Congestions

We detect congestion by observing the link based on the mechanism of CSMA/CA, and perform the local selection of links to forward packets. Recall that, in CSMA/CA, a node re-transmits a data frame when the node does not receive the corresponding ACK frame. Naturally, the retransmission count becomes larger when congestion on the link grows larger. Thus, in our method, we take a threshold on retransmission counts to detect congestion. Specifically, every node measures the transmission counts of data frames on every connecting link l, and holds the results for the past n packets as well as their average r_l. If r_l exceeds the threshold T_r, the node regards that the link is congested, and for a certain period of time T_c the node assumes that the congestion on the link continues. Recall that the dynamic link metrics tend to include significant delay so that the actual link quality is not reflected accurately. In contrast, since our method detects the congestion through a short-period measurement of frame retransmission counts, nodes can switch their forwarding links much more sensitively to complement dynamic metrics.

3.5 Switching Links to Forward Packets

In our method, when a node detects congestion on its primary link (as described in Section 3.4), the node switches the link to forward packets; as a new forwarding link, the node uses the link with less congestion between the primary and the secondary links. After changing the forwarding link, the node keeps using the link for a certain period of time T_c to prevent frequent changes in forwarding links. After time T_c has passed, the node returns to the primary link and again starts detecting for any congestion on it.

Our basic strategy on selecting the forwarding link in face of congestion is to use the less-congested link between primary and secondary links. To do this, we maintain three node states for each neighbor node, i.e., *primary*, *secondary*, and *return* states. The specific steps for node behavior includes the following states transitions, which we illustrate in **Fig. 4**.

Step 1: Initially, a node is in the *primary state*. In this state, the node transmits packets using the primary link l_p, and watches the average retransmission count r_{l_p} on that link. If r_{l_p} exceeds the given threshold T_r, the node transits to the *secondary state*, and the node stores the value r_{l_p} for a certain period of time.

Step 2: In the *secondary state*, a node transmits packets using the secondary link l_s, and watches the average retransmission count r_{l_s} of that link. If r_{l_s} exceeds the stored re-transmission count of the primary link, i.e., $r_{l_p} < r_{l_s}$, the node transits to the *return state*.

Step 3: In the *return state*, a node transmits packets using the primary link, and the node does not measure the retransmission count any more. In other words, the node will not transit to other states within the time period of T_c.

Step 4: No matter what state the node is in, when the time period T_c has passed after the node transits to the *secondary state*, the node returns back to the *primary state* to reset the forwarding link selection. When the next-hop node is changed by the routing protocol in face of topology changes, the node also returns to the *primary state* to reset the link selection behavior.

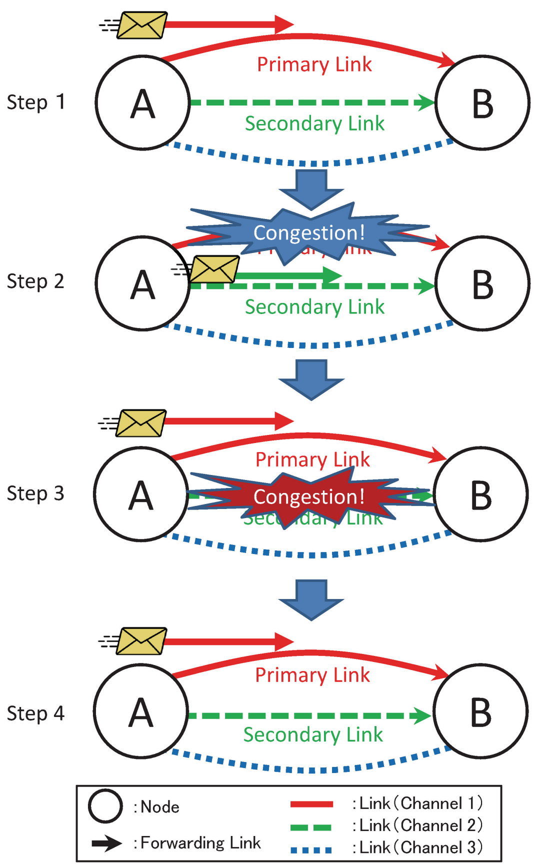

Fig. 4 The operational steps of the proposed method.

Steps 1 and 2 intend to select the less congested link to forward packets when a node detects congestion on its primary link. Note that, however, the quick repetition of link switching would bring confusion to traffic. Thus, we added Steps 3 and 4 to reduce unnecessary frequent switching. Consequently, node changes to the forwarding link can be as frequent as the interval T_c.

4. Evaluation

4.1 Simulation Scenarios

We evaluated the proposed method using the Qualnet 5.0 simulator. We employed the routing protocol OLSR version 2, which is implemented as the OLSRv2-Niigata [16] module included in the simulator. We implemented the proposed method on Qualnet 5.0. We also modified the module of OLSRv2-Niigata so as to implement WCETT, which is a dynamic routing metric designed for multi-channel WMNs. In **Table 1**, we show the simulation environment.

First of all, we explain common simulation settings. Each scenario runs for 6 minutes. We generated several CBR (Constant Bit Rate) flows with a uniform total rate. (e.g., if the number of flows is 4 and the total transmission rate of flows is 6.0 Mbps, then the rate per flow is 1.5 Mbps.) Every flow starts transmission at 1 minute after the scenario starts, and they stop at 5 minutes.

In this study, we evaluate the performance of the proposed method with different traffic patterns over WMNs. As traffic patterns for evaluation, we considered two scenarios in which severe interference is expected among flows, because the proposed method intends to improve network performance by reducing interference among flows. As the practical scenarios that include interference among flows, we tried two traffic patterns that corresponds to the stand-alone use and the Internet-connected use of the mesh networks, as shown as follows:

Scenario 1: Many Random flows among all nodes in the network.

Scenario 2: Many flows from all nodes to a gateway node.

Note that random flows in Scenario 1 and uplink flows to the gateway placed at the center of the network in Scenario 2 would incur considerable inter-flow interference that degrade the network performance. To see the effect of the proposed method to avoid this kind of interference is the point that we wish to evaluate.

In Scenario 1, we suppose the 5×5 grid topology shown in **Fig. 5**. We generate 25 CBR flows whose source and destination nodes are randomly selected. We varied the total transmission

Table 1 Simulation environment.

Item	Value
Topology	5×5 (Scenario 1) or 7×7 (Scenario 2) Grid with 300 m interval
MAC protocol	IEEE 802.11a (6 Mbps)
Transmission power	20 db
# of channels (k)	2, 3, 4, and 5
Pathloss Model	Two Ray
Routing module	OLSRv2-Niigata [16]
Deployed dynamic metric	WCETT
# of past packets to hold retransmission count (n)	3

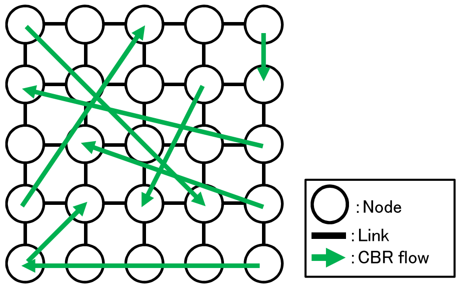

Fig. 5 Network topology in Scenario 1: In this topology, we generate 25 flows whose source and destination nodes are selected randomly.

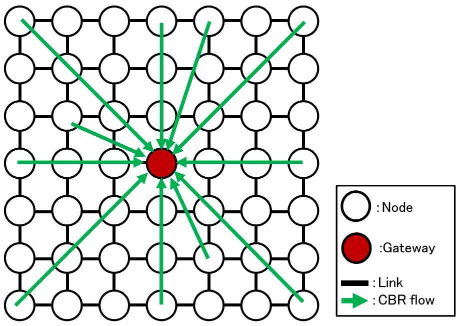

Fig. 6 Network topology in Scenario 2: In this topology, the central node behaves as a gateway, and 48 CBR flows are generated from all nodes (except the gateway) to the gateway.

rate between 3 Mbps and 10.5 Mbps with an interval of 1.5 Mbps. We performed 10 repetitions of simulations with different random seeds, in which flow patterns are also randomised, and obtained averages as the result.

In Scenario 2, we suppose the 7 × 7 grid topology shown in **Fig. 6**. The gateway that enables communications with the outside nodes is placed at the center of the topology. We generated 48 CBR flows in which each node except the gateway is the source node and the gateway is the destination node. We varied the total transmission rate between 3 Mbps and 9 Mbps with the interval of 1.5 Mbps. In Scenario 2, we performed 5 repetitions of simulations with different random seeds and compared the average values.

The comparison criteria in both Scenario 1 and 2 are (i) total throughput, (ii) the coefficient of variation C_v, (iii) the average delay, and (iv) the average jitter. The coefficient of variation C_v is used to examine the stability of throughput from the viewpoint of users. Formally, C_v is computed as

$$C_v = \frac{\sqrt{\sigma^2}}{\bar{x}}, \qquad (1)$$

where σ is the standard deviation of the throughput values measured every second, and $\bar{x}$ is the average over them. (i) The total throughput is the sum of all throughput values of flows. (ii) The coefficient of variation C_v represents the variation of throughputs each second. Thus, a small C_v indicates that communications are stable from the viewpoint of users. (iii) The delay is the time that each packet takes to reach its destination. (iv) The average jitter is measured as the average of the difference in delays between all succeeding pairs of packets in a flow that reach the destination. These four values are compared between the conventional method WCETT and the proposed method that works with WCETT.

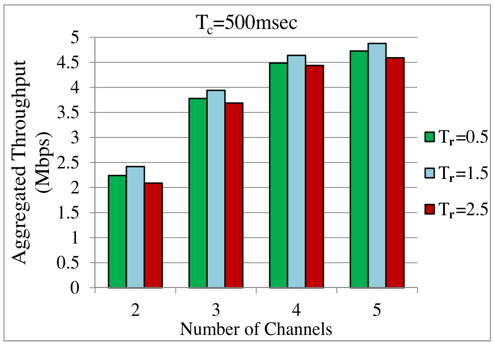

Fig. 7 Comparison of throughput under variation of parameter T_r.

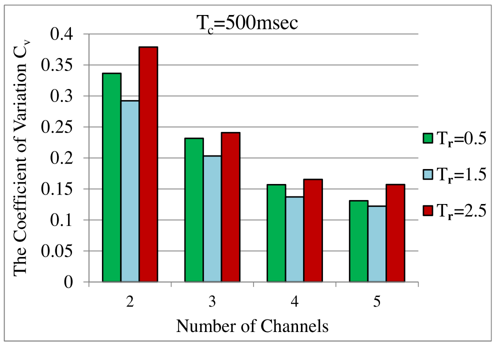

Fig. 8 Comparison of C_v under variation of parameter T_r.

4.2 Impact of Parameters T_c and T_r

We firstly evaluated network performance by varying the two threshold values T_r and T_c to explore the best values for T_r and T_c. Specifically, we measured the total throughput and C_v under the same situation as Scenario 2 with a total transmission ratio of 6 Mbps.

Figures 7 and **8** show the performance in throughput and C_v for several numbers of channels where T_c is fixed at 500 ms and T_r is varied at 0.5, 1.5, and 2.5. These results show that the performance is the best in both throughput and C_v when T_r is fixed at 1.5, implying that too small or too large values for T_r is not good. This is because, when T_r is too small, congestion detection becomes too sensitive and frequent switching of links degrades performance. Large values of T_r make the system insensitive to congestion, and performance cannot improve. An adequate value of T_r of around 1.5 leads to better performance.

Figures 9 and **10** show the performance when T_r is fixed at 1.5 and T_c is varied between 125 ms and 2,000 ms. Figure 9 shows that, for every number of channels, throughput is higher than when T_c is 500 ms or 1,000 ms. Figure 10 shows that, when the number of channels is more than 3, C_v is lower than others when T_c is 250 ms or 500 ms. That is, when T_c is too high, switching behavior is so slow that it cannot keep up with the transition of traffic; whereas when T_c is too low, switching occurs so frequently so that the overhead degrades network performance. These results show that an adequate value of T_c of around 500 ms provides good performance. Exceptionally, when the number of channels is 2, performance is the lowest when T_c is 125 ms. This is because networks are so saturated in case of 2 channels that the traffic pattern changes more frequently. In such cases, performance improves when switching occurres more frequently.

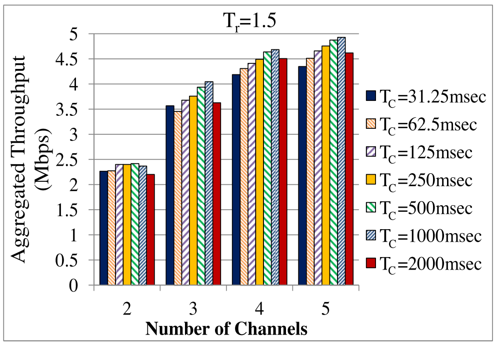

Fig. 9 Comparison of throughput under variation of parameter T_c.

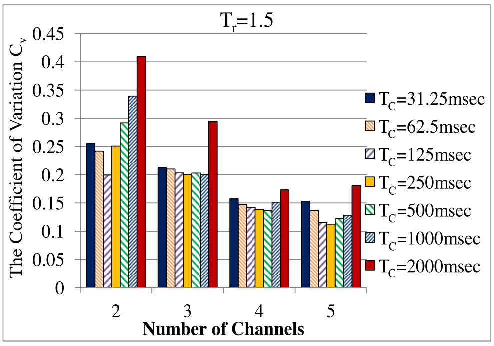

Fig. 10 Comparison of C_v under variation of parameter T_c.

From above, we conclude that the best value for T_r is 1.5 and for T_c is between 250 ms and 1,000 ms. Consequently, in the rest of our simulations, we fixed T_r and T_c at 1.5 and 500 ms, respectively.

4.3 Results in Scenario 1

Figures 11 and **12** show the results of Scenario 1 where we compare the aggregated throughput and C_v while varying transmission rate. The proposed method improved both throughput and communication stability as represented by C_v in most cases, where the maximum improvement is 22% in throughput and 33% in C_v. On the other hand, when the transmission rate gets larger than 9 Mbps, the difference in performance tends to shrink particularly in cases when there are less than or equal to 3 channels. In particular, the aggregated throughput gets lower although the transmission ratio gets higher. Note that, around this point, networks are so saturated, the traffic amount is relatively larger than the capacity. We conclude that the proposed method works effectively unless networks are saturated.

Figures 13 and **14** show the average delay and the average jitter while varying transmission rate. These results show that both delay and jitter are improved by the proposed method; the delay improved 60% and the jitter improved 30% at the maximum. When the number of available channels was 3 or less, the performance in jitter was degraded by the proposed method. This would be because forwarding links switch frequently, which makes two succeeding packets travel along different paths with higher probability than the conventional method.

In summary, from the results of Scenario 1, in which we assumed that all flows are generated within networks, we showed that the proposed method improves communication performance

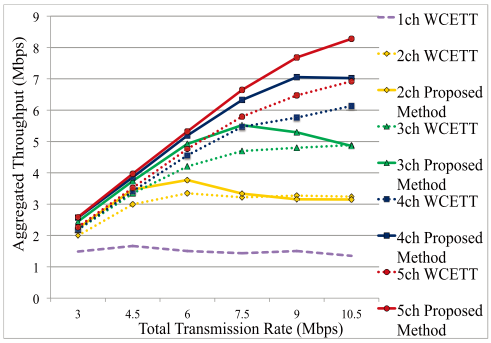

Fig. 11 Comparison of aggregated throughput in Scenario 1.

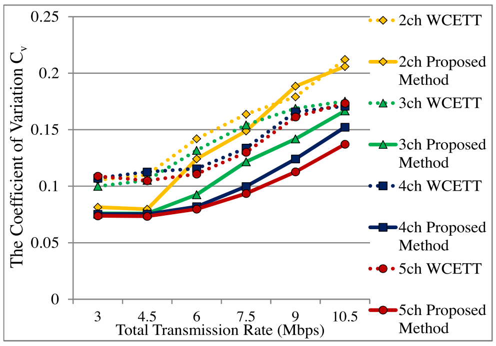

Fig. 12 Comparison of C_v in Scenario 1.

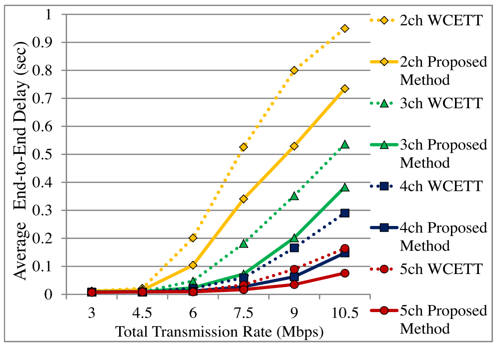

Fig. 13 Comparison of average end-to-end delay in Scenario 1.

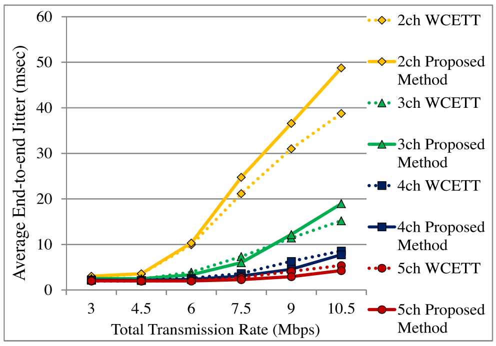

Fig. 14 Comparison of average end-to-end jitter in Scenario 1.

unless networks are saturated. Particularly, the average delay improves in every case whereas the average jitter is greatly affected by network saturation.

4.4 Results in Scenario 2

Figures 15 and **16** show the performance in throughput and C_v when transmission rate is varied. Scenario 2 also shows that

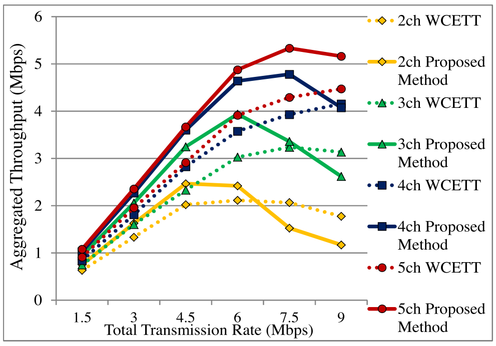

Fig. 15 Comparison of aggregated throughput in Scenario 2.

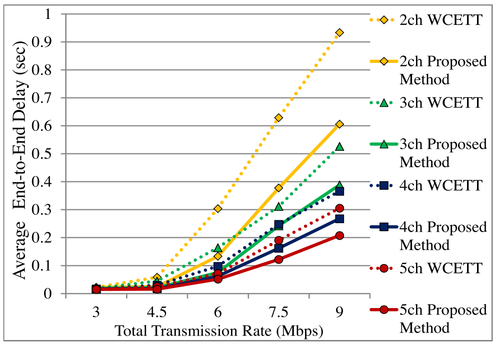

Fig. 17 Comparison of average end-to-end delay in Scenario 2.

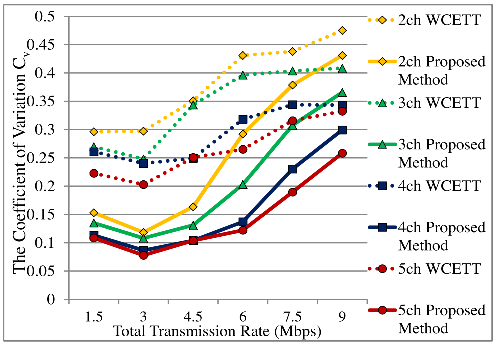

Fig. 16 Comparison of C_v in Scenario 2.

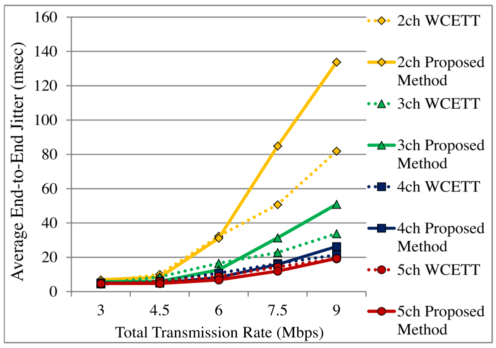

Fig. 18 Comparison of average end-to-end jitter in Scenario 2.

the proposed method improved performance in terms of both aggregated throughput and stability. Throughput improved by 40% and C_v improved by 64% at most. Note that, in Scenario 2, saturation is clearly seen when transmission rate is around 6 Mbps in the cases where the number of available channels is 2 or 3, and around 7.5 Mbps for the case of 4 channels. Similar to Scenario 1, we found a tendency for the performance of the proposed method to shrink when networks are saturated.

Figures 17 and **18** show the performance in terms of average delay and average jitter. Similar to Scenario 1, the proposed method improved the average delay in every case, whereas the average jitter improved slightly. The performance of the proposed method got worse than the conventional method when networks were saturated.

In summary, from the results of Scenario 2, in which we assumed communication via one gateway node, we also saw a similar tendency to Scenario 1. The proposed method improves communication performance unless networks are saturated.

4.5 Discussion

From the evaluation results, we found that the proposed method improves communication performance unless networks are saturated. We discuss this issue in more detail. We saw that the average delay and stability C_v are improved significantly by the proposed method in all cases. However, note that, although delay and C_v are improved, jitter can be degraded by the proposed mechanism, especially in the cases of less than 3 available channels. This may seem unexpected, but this is not a contradiction.

From a microscopic view, since forwarding links are frequently switched, and the jitter is measured as the average difference in delay between two succeeding packets, the proposed mechanism increases the average jitter, which may indicate unstable communications. However, since this switching mechanism follows more up-to-date quality of links, it enables us to choose better paths, and so packet delay in total is reduced. Note that the most critical cause of communication instability is the queuing delay due to congestion. Thus, by reducing the effect from this kind of delay, the proposed mechanism improved the communication stability C_v as measured from the users' viewpoint.

With the proposed method, we succeeded in improving the performance of dynamic metrics by complementing the delay drawback described in Section 2. Although dynamic metrics improve network performance, we pointed out that the metric values represent the link qualities in the past, which allows room to improve the network performance. Our approach for addressing this point is to switch the forwarding link as selected by dynamic metrics to the second-best one, if the primary forwarding link is actually judged not to be the best one through our instant measurement method. In other words, between the two best links that are selected by dynamic metrics, we choose the best one through our instant measurement. By achieving better performance with our method, we have shown that the delay drawback actually degrades the network performance, and that the link with the smallest metric may not always be the "best-quality" link. This is why the proposed method works effectively with dynamic metrics. We have evaluated the performance of our method with the best-known multi-channel metric WCETT. Although we did not try other metrics, our method would also work with them because other metrics proposed so far also include the same delay drawback.

On the other hand, we point out to readers that may wonder why each node in dynamic metric schemes uses a single channel to send packets rather than multiple channels simultaneously. This is simply because the effect of hidden terminals is significant. If every node uses all available channels simultaneously, the result would be that the interference would increase considerably and consequently the performance would significantly degrade. As evidence for this, see Fig. 11 again, where you see the result from the single channel case. The maximum throughput in the single channel case is about 1.6 Mbps at a transmission rate of 4.5 Mbps. In comparison, it is 3.8 Mbps with the two-channels case and 5.3 Mbps with the three-channels case, which are more than twice and three times the performance of the single channel case, respectively. This means that, even if the single-channel approach uses the radio resources of two or three channels simultaneously, the performance is less than the multi-channel approach with dynamic metrics and the proposed method.

5. Conclusion

In this paper, we pointed out that the performance of dynamic metrics is not optimal due to the delay between the real link quality and the metric value, and proposed a method that complements dynamic metrics. Our method prepares a secondary link (i.e., channel) in addition to the primary link (channel) that indicates the next-hop link under the shortest-path based routing, and switches the forwarding link between them according to a quick local measurement of link quality. We evaluated the effectiveness of the proposed method using a network simulator, and clarified that the proposed method improves communication performance in both throughput and stability in combination with a representative dynamic metric designed for multi-channel WMNs.

The central concern that we focused on in this study is the significant delay over link metrics that leads to a considerable degradation of communication performance. Because this property is not avoidable, dynamic metrics only follow relatively slow transitions of link quality. Thus, to catch up with the rapid transition of link quality, we proposed combining a local decision with the dynamic metric. Because our local decision method incorporates a quick and rough measurement of link quality, the method complements dynamic metrics since it aims for rapid transition in link quality.

As future work, we plan to introduce the local decision method proposed in this paper into general wireless mesh networks. In this paper, we supposed the WMNs in which k independent channels are all available for every pair of neighbors. Thus local decision means the selection of the forwarding channel so that the next-hop node does not change. In contrast, in general WMNs, because different channels may not be available to reach the next-hop node, local decisions would include the selection of paths to reach nodes within a short distance. Designing such a local decision scheme to complement routing metrics for general WMNs would be one of the interesting challenges for the future.

References

[1] Nelson, R. and Kleinrock, L.: Spatial TDMA: A Collision Free Multi-hop Channel Access Protocol, *IEEE Trans. Communications*, Vol.33, No.9, pp.934–944 (1985).

[2] Gore, A.D. and Karandikar, A.: Link Scheduling Algorithms for Wireless Mesh Networks, *IEEE Communication Surveys & Tutorials*, Vol.13, No.2 (2011).

[3] Ni, J., Tan, B. and Srikant, R.: Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks, *Proc. INFOCOM2010*, pp.1–5 (2010).

[4] Huang, P.K. and Lin, X.: Improving the Delay Performance of CSMA Algorithms: A Virtual Multi-Channel Approach, *Proc. INFOCOM'13*, pp.2598–2606 (2013).

[5] Akyildiz, I.F. and Wang, X.: *Wireless Mesh Networks*, John Wiley & Sons, Ltd, Publication (2009).

[6] Clausen, T. and Jacquet, P.: Optimized Link State Routing Protocol (OLSR), IETF RFC3626 (2003).

[7] De. Couto, D., Aguayo, D., Bicket, J. and Morris, R.: A High-Throughput Path Metric for Multi-Hop Wireless Sensor Networks, *Proc. 9th Annual International Conference on Mobile Computing and Networking* (*MOBICOM'03*), pp.134–146 (2003).

[8] Draves, R., Padhye, J. and Zill, B.: Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, *Proc. 10th Annual International Conference on Mobile Computing and Networking* (*MOBICOM'04*), pp.114–128 (2004).

[9] Raniwala, A. and Chiueh, T.: Architecture and Algorithms for an IEEE 802.11-based Multi-channel Wireless Mesh Network, *24th Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM'05*), pp.2223–2234 (2005).

[10] Alicherry, M., Bhatic, R. and Li, L.: Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks, *Selected Areas in Communications, IEEE Journal*, Vol.24, pp.1960–1971 (2006).

[11] Kodialam, M. and Nandagopal, T.: Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks, *Proc. 11th Annual International Conference on Mobile Computing and Networking* (*MOBICOM'05*), pp.73–87 (2005).

[12] Yang, Y., Wang, J. and Kravets, R.: Interference-aware Load Balancing for Multihop Wireless Networks, Technical Report UIUCDCS-R-2005-2526, Department of Computer Science, University of Illinois (2005).

[13] Jin, S. and Mase, K.: A Hidden-Exposed Terminal Interference Aware Routing Metric for Multi-Radio and Multi-Rate Wireless Mesh Networks, *IEICE Trans. Communications*, Vol.92-B, No.3, pp.709–716 (2009).

[14] Waharte, S., Ishibashi, B., Boutaba, R. and Meddour, D.E.: Design and Performance Evaluation of IAR: Interference Aware Routing Metric for Wireless Mesh Networks, *Mobile Networking Applications*, Vol.14, No.5, pp.649–660 (2009).

[15] Gupta, P. and Kumar, P.R.: The Capacity of Wireless Networks, *IEEE Trans. Information Theory*, Vol.46, No.2, pp.388–404 (2000).

[16] OLSRv2-Niigata, available from ⟨http://www2.net.ie.niigata-u.ac.jp/nOLSRv2/olsrv2/Welcome.html⟩ (accessed 2013-6-21).

Hiromichi Kanaoka received his B.E. and M.E. degrees from Wakayama University in 2012 and 2014, respectively. He is currently working with Duplo Seiko Corporation.

Takuya Yoshihiro received his B.E., M.I. and Ph.D. degrees from Kyoto University in 1998, 2000 and 2003, respectively. He was an assistant professor at Wakayama University from 2003 to 2009. He has been an associate professor in Wakayama University from 2009. He is currently interested in graph theory, distributed algorithms, computer networks, wireless networks, medical applications, bioinformatics, etc. He is a member of IEEE, IEICE, and IPSJ.