Measurement of the Performance of IEEE 802.11s

Naoshi Sakamoto and Katsuya Atsuta
Tokyo Denki University, Tokyo, Japan
Email: sakamoto@c.dendai.ac.jp; 18kmc03@ms.dendai.ac.jp

Abstract**—**IEEE 802.11s, the specification for mesh networking, has been available since 2012. Initial products to support this standard did not perform well. However, Google WiFi, which is based on IEEE 802.11s, was released recently. Nevertheless, it has often been noted that multihop mesh networks cause a drop in the throughput. In this study, we measure the performance of mesh networks that consist of the mesh points of our own making and those of Google WiFi. Finally, we find that the performance of multihop mesh networks is worse than the performance of single-hop networks. Then, we consider the problems with the performance of multihop mesh network based on the Mathis's TCP traffic model.

Index Terms**—**IEEE 802.11s, mesh network, TCP, throughput

I. Introduction

Wireless LAN, particularly WiFi (IEEE 802.11), is spreading widely. The infrastructure mode is mainly used as the transportation mode of IEEE 802.11, which consists of an access point and many directly connected terminals. Additionally, the ad hoc mode is also used by game machines and other systems, which enables two stations to connect directly.

In 2012, IEEE 802.11s, the specification for mesh networks, was included in IEEE 802.11. This introduced the concept of Mesh Points (MP). In a mesh network, multiple MPs are in radio contact with one another, organize one another, calculate the routing, and form a relay network. While in the infrastructure mode, every former access point requires a wired connection to the Internet; in the mesh network, it is not necessary for every MP to have a wired connection to the Internet. The mesh network enables the communication area to be enlarged by only installing MPs. Once the mesh networks are put to practical use, they could be very useful because they would work as the ports of a hub.

IEEE 802.11s began development in 2003. The open80211s group was established and developed the driver for Linux, which has been installed into the official Linux kernel since version 2.6.26. Thus, the IEEE 802.11s mesh network has often been considered and studied for a long time. It has been pointed out that increasing the number of hops causes a drastic decrease in throughput. Therefore, even though some products are being sold, we cannot say that they have spread widely.

Manuscript received May 26, 2019; revised November 10, 2019.
Corresponding author email: sakamoto@c.dendai.ac.jp.
doi:10.12720/jcm.14.12.1218-1223

In this study, in order to measure the properties of mesh networks, we produce our own MPs by using Raspberry Pi. Moreover, we also use Google WiFi [1], which has been sold in Japan since 2018. Then, we compare the results of the measurements with those of the theoretical model, and discuss the resulting throughput.

This paper is organized as follows: Section II describes the background information about IEEE 802.11s, TCP, and others. Section III presents the related works. In Section IV, we perform the experiments and discuss the results of our measurements. Finally, Section V concludes the paper.

II. Preliminaries

A. IEEE 802.11s

IEEE 802.11s is the specification for mesh networks. It was included in IEEE 802.11, the wireless LAN specification, in 2012 [2].

IEEE 802.11 specifies wireless local area networks, which mainly use the 2.4-GHz and 5-GHz bands. Its physical layer (PHY) is defined by 11b, 11g, 11n, and 11ac. It optionally uses packet request to send/clear to send (RTS/CTS) to control flows. Originally, it specified only the infrastructure and ad hoc network modes. In infrastructure mode, the network consists of a single access point and the terminals that connect to the access point. This is the most commonly used mode. On the other hand, in the ad hoc network mode, two terminals connect to each other directly. This mode is used by game machines and others.

Moreover, in the mesh network specified by IEEE 802.11s, multiple MPs stay in radio contact with one another by using 802.11 PHY, calculate the routing dynamically, then relay communication. In the present situation, up to 32 MPs are allowed. On the other hand, because each MP is allowed to be an access point, an MP can also act as a relay station of the access point.

IEEE 802.11s adopts the hybrid wireless mesh protocol as its routing protocol. This protocol consists of two modes: the on-demand path-selection mode and the proactive tree-building mode. In the on-demand path-selection mode, whenever an MP requires the information of the path to the destination, in order to calculate the routing, it floods the path request (PREQ) packets. On the other hand, in the proactive tree-building mode (optional), the MP designated as the root forms a spanning tree by using either the proactive PREQ or the proactive root announcement. Then, after the MP finishes calculating the routing, packets are relayed via the spanning tree on the mesh network. Thus, because the routing protocol is basically initiated for a data packet, the mechanism of relaying data packets in the mesh network is not as simple as a bucket brigade.

B. Transport Control Protocol

In order to evaluate the network traffic, the file transport speed of the transport control protocol (TCP) is usually used instead of the number of transported bits per unit time. This replacement is meaningful in practice. In order to measure this, the software called iperf is usually used.

On the internet, in order to transfer a file, we usually use TCP. TCP is an end-to-end protocol that transports files on a network and allows packet loss of a certain ratio.

When TCP sends a file, it divides the file into segments. TCP has many operations in that it divides a file into segments, rebuilds a file from segments, detects error packets, retransmits them, and controls the sending speed to avoid congestion. There are many algorithms to control congestion. Recently, TCP CUBIC is used by various OSes to control congestion according to packet loss.

Mathis et al. provided a TCP throughput model [3]. By letting the maximum segment size be MSS, round-trip time be RTT, and packet-loss probability be p, the model provides that the bandwidth is restrained by (1).

$$BW < \left(\frac{MSS}{RTT}\right)\frac{1}{\sqrt{p}} \tag{1}$$

Antunes confirms that Mathis's TCP throughput model can also be adopted to TCP CUBIC [4].

C. Traffic Property of 802.11

Funabiki et al. measured the throughput of the infrastructure mode of IEEE 802.11g and 11n for the distance by using iperf. They then measured the throughput of the mesh net [5]. It was shown that the TCP throughput in infrastructure mode steadily declines with increasing distance.

III. Related Works

The following related works study the traffic properties of IEEE 802.11s. They measured the properties under the condition that the number of hops is a variable parameter while each distance between the MPs is fixed. Furthermore, they assume the network topologies as a line and a square grid.

Lv et al. used ns-3 to simulate mesh networks where 16 MPs are placed as a 4 × 4 grid at intervals of 100 m [6]. They do not use TCP. However, they demonstrated that when the packet flow increases, the throughput decreases but the air time and packet-loss ratio increase.

González *et al*. produced four MPs of their own making using Raspberry Pi and USB wireless adapters. Then, they measure the response time of the ping, TCP throughput, and UDP loss ratio by using iperf. Finally, they compare the measurement results with the simulation by ns-3 [7].

Lin et al. prepared nine real machines. Then, they measured the TCP throughput under the conditions of 11b, 11g, and 11n, and with and without RTS/CTS in the following situations: [8]

1. Five MPs in a line at intervals of 50 cm,
2. MPs in a 3 × 3 grid at intervals of 50 m, and
3. MPs in a 25-m-wide 3 × 3 grid and the building-level height.

Rethfeldt *et al*. connected MPs by placing a register instead of antennas in between to drop the radio strength, then formed MPs in a line via wire [9]. They measured the TCP throughput and the UDP throughput for four MPs in a line.

Robitzsch *et al*. measured the TCP throughput and the UDP throughput of the network where six MPs are formed in a line for 802.11g, 802.11n at 2.4 GHz, and 802.11n at 5 GHz [10].

Hiertz et al. wrote an article introducing various technologies of IEEE 802.11s [11] and an open80211s project [12]. Then, they evaluated the driver developed by the open80211s project by measuring the discovery time and throughput from one hop to eleven hops.

Pandey et al. provided a design and implementation of the expansion function [13]. They discuss and then test and evaluate the mesh network consisting of five MPs. Moreover, they focus on the effect of flooding PREQ. They examine the relationship between the number of PREQs per second and the throughput, and then observe the results at 18 Mbps for 0 PREQs and 8 Mbps for 20 PREQs.

For each result, the TCP throughput decreases greatly when the number of hops is greater than two.

IV. Performance Tests

In this section, we form mesh networks and evaluate the communication property. Then, we discuss the measurement results. For the sake of analytical convenience, we would like to reduce the complexity of the mesh networks. Therefore, we focus on the mesh networks with a line topology. We measure the delay in the ping, packet-loss ratio of the ping, and TCP throughput by using iperf.

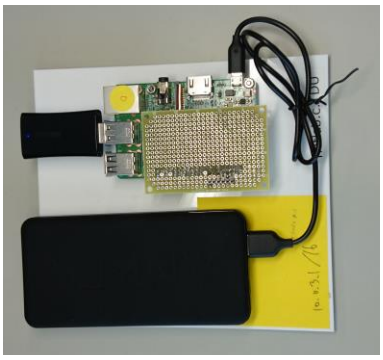

Fig. 1. Mesh point of our own design.

We use our own MPs and those of Google WiFi. An MP of our own making consists of a Raspberry Pi3 and a Buffalo WLI-UC-G-301N with Raspbian OS installed (Fig. 1). The driver is rt2x00, and the congestion control algorithm of TCP is CUBIC. Note that the transmission property of our own MP corresponds to IEEE 802.11g.

We measure 50000 pings, then measure the average delay time and loss ratio.

A. Experiment

Experiment 1: First, we evaluate the properties of a one-hop network with respect to distance. That is, we prepare two MPs, then measure the delay time, success ratio, and throughput with varying distance by using iperf.

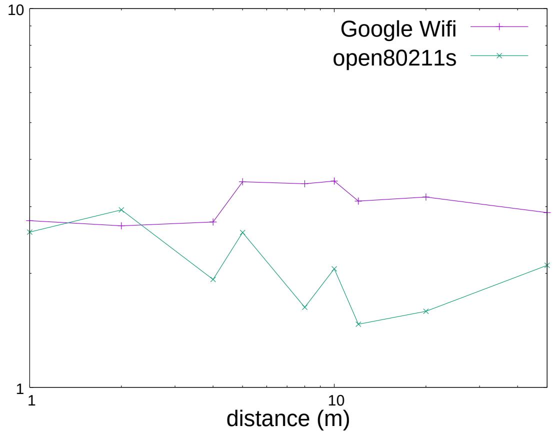

Fig. 2. Delay time with respect to distance.

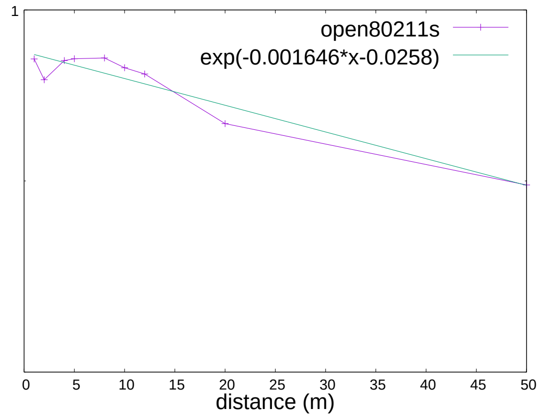

Fig. 3. Success ratio with respect to distance.

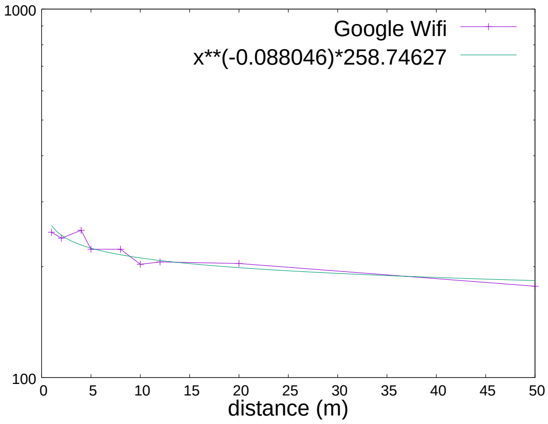

Fig. 4. Throughput of Google WiFi with respect to distance.

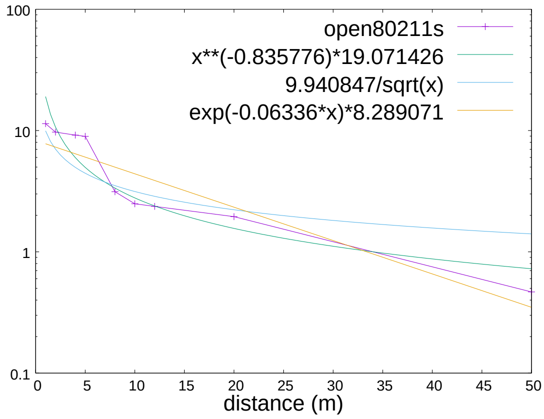

Fig. 5. Throughput of open80211s with respect to distance.

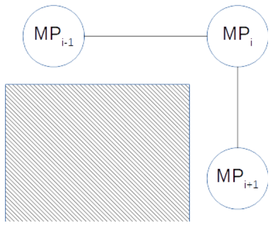

Fig. 6. MPs around an obstruction.

We show the results in Figs. 2, 3, 4, and 5. Moreover, we draw the approximate curves in Figs. 3, 4, and 5. Note that we omit the graph of the success ratio of Google WiFi because this system causes no packet loss. According to Fig. 2, we find that the delay time is almost independent of the distance. On the other hand, according to Fig 3, the loss ratio of open80211s increases with distance. We separate the chart of throughput into Fig. 4 and Fig. 5 because their performance results are greatly different, where Google WiFi corresponds to IEEE 802.11ac and the MPs of our own making correspond to IEEE 802.11g. Nevertheless, both are affected by distance. However, the effect of distance on the performance of Google WiFi seems to be less.

Experiment 2: The MPs in this experiment allow dynamic routing.

Moreover, we arrange the position of each MP around obstructions in order to restrict communications to only neighbors (Fig. 6). Then, we can form the mesh network in a line topology. To achieve this design, we place MPs from the third floor to the fourteenth floor of a ferroconcrete building so that the structures of the building are used as obstructions (Fig. 7). Thus, the distance between each neighboring MP is 30 m horizontally or 4 m vertically. Note that, because this building has four well-holes and the communication area spreads wider when a MP is placed above a well-hole, we avoid placing MPs above well-holes.

According to the measurement results, we draw charts in Figs. 8, 9, and 10. Moreover, we draw the approximate curves in Fig. 10.

According to Fig. 8, we find that whereas Google WiFi loses few packets, open80211s loses packets exponentially. On the other hand, according to Fig. 9, we find that the delay time of both increases linearly. That is, any relay MPs must delay for a fixed amount time.

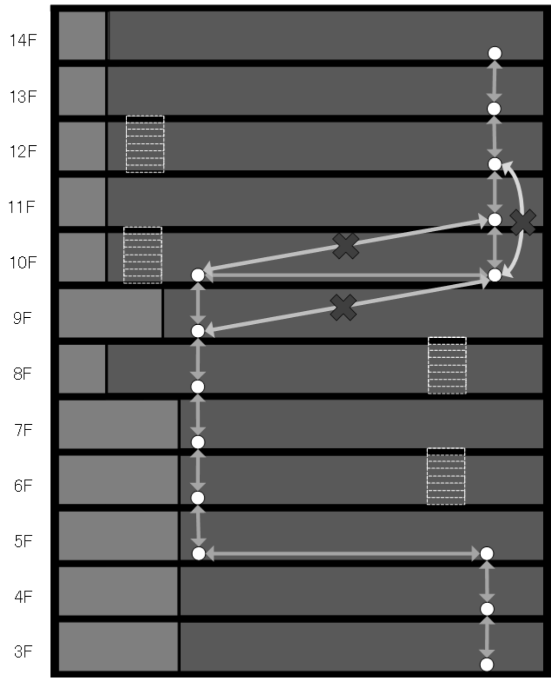

Fig. 7. Layout of MPs.

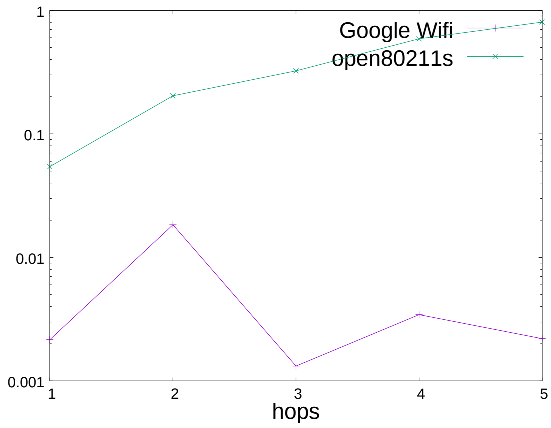

Fig. 8. Loss ratio with respect to hops.

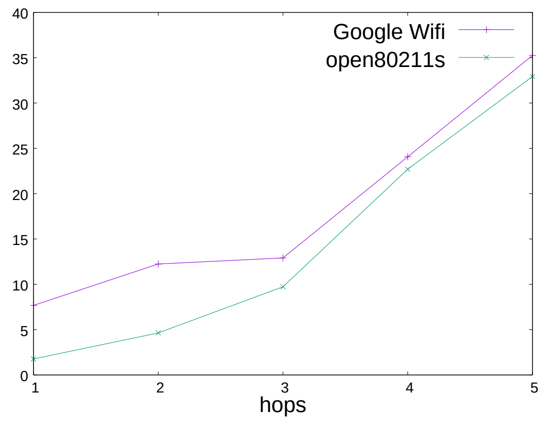

Fig. 9. Delay time for hops.

On the other hand, according to Fig. 10, the throughput of both decreases with increasing numbers of hops. We will discuss this in Section IV-B.

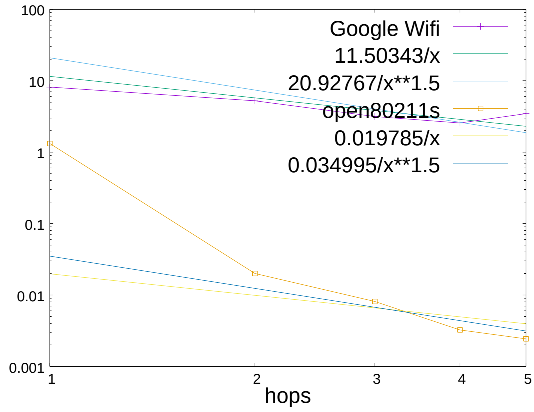

Fig. 10. Throughput with respect to hops.

Experiment 3: The MPs also allow dynamic routing in this case. Moreover, we arrange each position of the MPs around obstructions in order to restrict communications to neighbors only. However, unlike Experiment 2, in order to make each distance between neighbors be as short as possible, we place MPs along steel stairs to form a topology like the letter L or Z. We can place MPs at intervals of 4 m.

Similar to Experiment 2, we measure the delay time and the throughput.

We make charts that compare the results with the result of the one-hop network in Experiment 1.

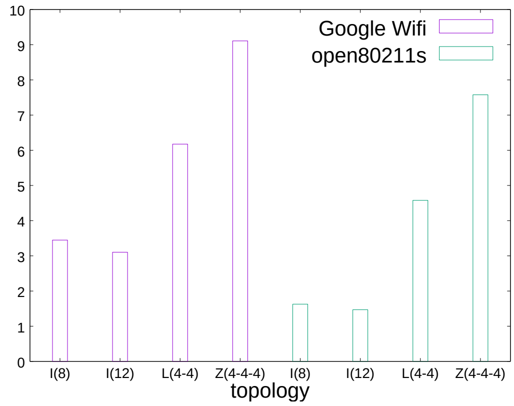

Fig. 11. Delay time with respect to hops in short range.

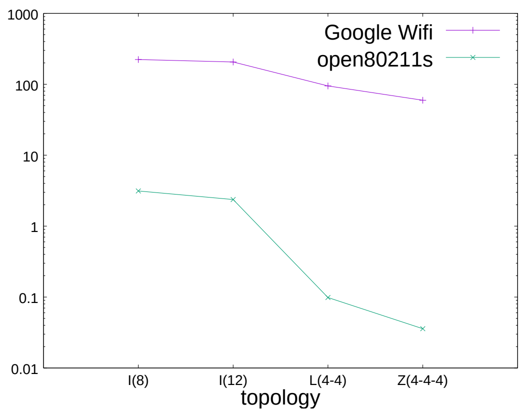

Fig. 12. TCP throughput for each topology.

According to Fig. 11, when the total length of each network is the same, the delay time is almost proportional to the number of hops. On the other hand, according to Fig. 12, the throughput drops drastically with increasing numbers of hops.

B. Discussion

1) Properties of the one-hop network: It has already been reported that the IEEE 802.11 network loses throughput with increasing distance between the access point and terminal [5]. On the other hand, Mathis's model describes the throughput as a function of MSS, RTT, and p.

Fig. 2 shows that the delay time is always independent of the distance between MPs. On the other hand, because the loss ratio of Google WiFi is almost equal to zero and the throughput decreases little, the property must mainly depend on the performance of their hardware, where Google WiFi is IEEE802.11ac and MPs of our own making are IEEE 802.11g. Suppose that the success ratio of distance x follows an exponential function such as $s_1 s^{x-1}$ where s_1 is the success ratio of distance 1. Then, by letting p, the loss ratio per unit distance, be $1-s$, we can approximate the loss ratio $p(x)$ as follows.

$$\begin{aligned} p(x) &= 1 - s_1 s^{x-1} \\ &= 1 - \frac{s_1}{s}(1-p)^x \\ &\sim \left(1 - \frac{s_1}{s}\right) + \frac{s_1}{s} px \end{aligned} \tag{2}$$

Moreover, for $q(x) = 1/\sqrt{p(x)}$, for all sufficiently large x, $q(x)$ can be proportional to $x^{-0.5}$. This can be observed roughly in Fig. 5. On the other hand, for Google WiFi, because the loss ratio is almost equal to zero and the delay time does not depend on the distance, we cannot estimate the relationship between the throughput and the distance based on Mathis's model.

2) Properties of the multihop network: On the simple model of a multihop network with a line topology, we can approximate that the delay time is additive and the success ratio is multiplicative for every hop. Let D be the delay time of one hop and P be the loss ratio of one hop. Suppose that the loss ratio of the total system is ρ (almost zero) and the success ratio for every relay is $(1-P)^a$, where the number of communications required for each relay is a. Then, we can approximate the delay time $d(n)$ and the loss ratio $p(n)$ for n hops as follows.

$$d(n) = D + k(n-1) \tag{3}$$

$$p(n) = 1 - (1-\rho)(1-P)^{an} \tag{4}$$

By applying Mathis's model, we can approximate the throughput $t(n)$ as

$$\begin{aligned} t(n) &= \frac{MSS}{d(n)\sqrt{p(n)}} \\ &= \frac{MSS}{(D + k(n-1))\sqrt{1-(1-\rho)(1-P)^{an}}} \\ &\sim \begin{cases} \dfrac{MSS}{k\sqrt{\rho}} n^{-1} & \text{if } a = 0, \\ \dfrac{MSS}{k\sqrt{(1-\rho)aP}} n^{-3/2} & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

Actually, according to the approximate curves in Fig. 10, the throughput of Google WiFi can be approximately proportional to n^{-1}, and the throughput of open80211s can be approximately proportional to $n^{-3/2}$.

Therefore, we can say that, for TCP algorithms dominated by Mathis's model, the throughput must drop in a multihop network. This conclusion can also be obtained from Figs. 11 and 12, which show this property at short range.

Moreover, because the throughput of open80211s is worse, its protocol itself might have a negative effect on the throughput. We conjecture that because its default routing protocol is reactive, the routing packets that are generated with each arrival of data packets cause a decrease in the throughput. Moreover, for ICMP packets in the ping command, because the packets leave and return, the packets might collide with each other in passing.

The drop in the throughput of multihop mesh networks is caused by the property of TCPs dominated by Mathis's model. On the other hand, because the communication property of open80211s is greatly different from that of Google WiFi, the protocol of open80211s should be improved.

V. Conclusions

In this study, in order to evaluate IEEE 802.11s mesh networks, we actually measured the properties of mesh points created by Raspberry Pi with the open80211s driver and those created by Google WiFi. Then, we found that although Google WiFi shows great improvement over open80211s, the throughput of both still drops. We conjecture that this is not caused by problems with Google WiFi, but is caused by the properties of TCP CUBIC.

In the future, we will measure mesh networks with more complex topology and discuss improvements to the protocols. Moreover, we will consider TCP BBR [14] provided by Google.

REFERENCES

[1] S. Noronha. (Nov. 2016). Google WiFi making a 'mesh' of your Wi-Fi. [Online]. Available: https://www.blog.google/products/google-wifi/making-mesh-your-wi-fi/

[2] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 2016.

[3] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82, Jul. 1997.

[4] J. Antunes. (Jul. 2013). Validating a very simple model for TCP throughput. [Online]. Available: https://blog.thousandeyes.com/a-very-simple-model-for-tcp-throughput/

[5] N. Funabiki, S. Fujita, T. Nakanishi, and K. Watanabe, "A modified active access-point selection algorithm considering link speed change in IEEE 802.11n for wireless mesh networks," in *Proc. Seventh International Conference on Complex, Intelligent, and Software Intensive Systems*, Jul. 2013, pp. 714–719.

[6] P. Lv, X. Wang, M. Xu, and Y. Chen, "Network-leading association scheme in IEEE 802.11 wireless mesh networks," in *IEEE ICC*, 2011.

[7] V. González, A. L. Santos, C. Pinart, and F. Milagro, "Experimental demonstration of the viability of IEEE 802.11b based inter-vehicle communications," in *Proc. 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, 2008, pp. 1:1–1:7.

[8] Y. D. Lin, S. L. Chang, J. H. Yeh, and S. Y. Cheng, "Indoor deployment of IEEE 802.11s mesh networks: Lessons and guidelines," *Ad Hoc Netw.*, vol. 9, no. 8, pp. 1404–1413, Nov. 2011.

[9] M. Rethfeldt, B. Beichler, H. Raddatz, F. Uster, P. Danielis, C. Haubelt, and D. Timmermann, "Mini-mesh: Practical assessment of a miniaturized IEEE 802.11n/s mesh testbed," in *WCNC*. IEEE, 2018, pp. 1–6.

[10] S. Robitzsch, C. Niephaus, J. Fitzpatrick, and M. Kretschmer, "Performance evaluation of wireless mesh networks using IEEE 802.11s and IEEE 802.11n," in *Proc. Fifth International Conference on Wireless and Mobile Communications*, 2009, pp. 272–278.

[11] G. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "IEEE 802.11s: The WLAN mesh standard," *IEEE Wireless Communications*, pp. 104–111, Feb. 2010.

[12] open80211s. [Online]. Available: https://www.open80211s.org/

[13] P. Pandey, S. Satish, J. Kuri, and H. Dagale, "Design & implementation of IEEE 802.11s mesh nodes with enhanced features," in *Proc. IEEE 6th International Conference on Mobile Adhoc and Sensor Systems(MASS)*, Oct. 2009, pp. 639–644.

[14] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Queue*, vol. 14, no. 5, pp. 50:20–50:53, Oct. 2016.

Naoshi Sakamoto was born in Japan in 1964. He graduated from the University of Electro-Communications in 1987, and received his M.S. and Dr.S. degrees from Tokyo Institute of Technology in 1989 and 2001, respectively.

From 1992 to 1997, he was a research assistant in the Computer Center of Hitotsubashi University. From 1997 to 2001, he was a research assistant at Tokyo Institute of Technology. From 2001 to 2014, he was an associate professor at Tokyo Denki University, and as of 2014, he has been a professor. He received the 1999 IEICE Excellent Paper Award. His research interests are distributed algorithms, randomized algorithms, and complexity theory.

Katsuya Atsuta was born in Japan in 1996. He graduated from Tokyo Denki University, and belongs to the graduate school of Tokyo Denki University.

He studies mesh networks.