Cooperative Data Offloading System with Neighbours Based on Location Information

Taku Yamazaki†, Kazuma Asano†, Satoshi Arai†, Yusuke Shimomura††, and Takumi Miyoshi†,††
† College of Systems Engineering and Science, Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama, 337–8570 Japan
†† Graduate School of Engineering and Science, Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama, 337–8570 Japan
Email: taku@shibaura-it.ac.jp, bp14004@shibaura-it.ac.jp, bp14006@shibaura-it.ac.jp,
mf17037@shibaura-it.ac.jp, miyoshi@shibaura-it.ac.jp

***Abstract*—The development of high speed mobile networks such as LTE (long term evolution) and the widespread use of smartphones have enabled users to easily obtain large volumes of content via the Internet. However, this causes a heavy consumption of network resources, a burden on the available bandwidth. To solve this kind of problems, a data offloading method with wireless LAN access point has been practically used to distribute traffic from mobile networks to fixed networks. However, the method using wireless LAN access points can only change the communication paths but cannot reduce the overall traffic. This paper proposes a cooperative data offloading system that reduces the overall traffic by sharing data with direct communication between neighbours based on their location information. Moreover, we implement the proposed system on Android smartphones and clarify its performance in comparison with a traditional client/server system through experiments to download data in an actual environment.**

***Index Terms*—cooperative offloading, distributed download, peer-to-peer, location, load balance, load reduction.**

I. Introduction

The development of high speed mobile networks such as LTE (long term evolution) and the widespread use of smartphones have enabled users to easily obtain a large volume of content anywhere via the Internet regardless of the user's location. However, Cisco's forecast [1] predicts that the overall traffic from mobile terminals in 2021 will increase by 7 times from that of 2016. In particular, it is also predicted that, in 2021, video contents traffic will account for approximately 78% of all mobile data traffic. Thus, to alleviate the effect of large volume data such as a video content, the heavy consumptions of network resource in mobile networks must be dealt with.

For solving this problem, data offloading methods to distribute and reduce mobile traffic have been discussed [2]. As one of the major data offloading method, a method with wireless LAN access point [3]–[5], which changes a communication path and distributes data traffic from mobile networks to fixed networks via the wireless LAN access point, has been widely used. As a different approach from the above, cooperative data offloading and downloading methods that terminals download data to cooperate with neighbours [6]–[10] have been proposed.

Although a data offloading method with wireless LAN access point can only distribute data traffic to move it from mobile networks to fixed networks, the overall data traffic does not change. Hence, the method is not able to reduce the data traffic. The cooperative offloading and downloading methods can reduce the data traffic. However, the methods require modifying both firmware of an access point and terminals. In addition, the discovery process of cooperation users has not been researched and discussed in many cases. The cooperative download method with terminals passed by opportunistically decides a range of data for partialy downloading based on a mobility prediction. However, its sharing efficiency may decrease under actual environments since the method strongly depends on an accuracy of the mobility prediction.

This paper proposes a neighbour cooperative data offloading system which is able to download cooperatively and share data among neighbours via direct communication based on location information relying on peer-to-peer communication. In addition, the proposed system is likely to be able to configure and select a cooperative download method. Therefore, this paper also proposes cooperative download methods called leader-elected cooperative download method and distributed cooperative download method. Moreover, we implement the proposed system on smartphones that runs it as an application-level and thus it does not require firmware-level modification.

II. Related Work

Currently, a data offloading method with wireless LAN access point has widely been used to distribute data traffic from mobile networks to fixed networks [3]. In addition, the deployment methods of wireless LAN access points have also been discussed [4], [5] for improving the effect of data offloading. In the access point-based method, users who connect to the Internet via mobile networks change their own connection to wireless LAN such as public wireless LAN services. Hence, the method can change the communication path from the mobile networks to other networks. However, since this method only focuses on distribution of data traffic between mobile networks and fixed networks, it is not able to reduce data traffic. In addition, this method is only applied to specified areas because it only enables an area where there is

wireless LAN access point. Hence, the distribution effect of traffic strongly depends on the place.

A cooperative download method [9], [10] where users download partial data and disseminate it to each other by passing it opportunistically based on a mobility prediction has been proposed. In this method, first, each terminal sends a control message to server. The message includes its own current position, destination, departure time, and information of own partial data. When receiving the control message, the server generates a reply message which includes the ID of terminal who has a possibility of encountering the sender of the control message, probability of encountering the sender of the control message, time of encountering, and information of having data and then the server sends the message to each terminal. Based on the information, each terminal predicts positions of encountering the other terminals who cooperate with each other and share partial data. Then, each terminal prioritizes to share the partial data which is difficult to obtain based on a mobility prediction. Although the method realizes cooperative data download based on a mobility prediction, the mobility prediction of users is difficult under actual environments. Hence, the sharing efficiency of the method may decrease since the method strongly depends on the accuracy of the mobility prediction.

III. Cooperative Data Offloading System with Neighbours based on Location Information

This paper proposes neighbour cooperative data offloading method that terminals, who are placed close position called neighbour based on location information relying on peer-to-peer communication, download data cooperatively, and then they share the data through direct communication among neighbours.

A. Overview

Figure 1 shows the structure of the proposed system. First, the proposed system constructs a local group among neighbours who request the same data via peer-to-peer network on mobile networks. Then, group members exchange control messages, which include a procedure of downloading and sharing

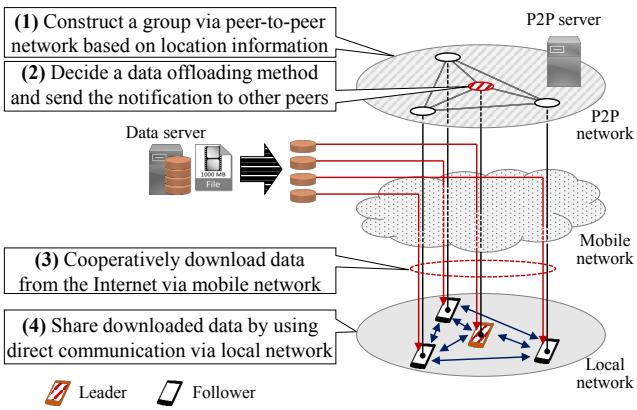

Fig. 1. Overview of the proposed cooperative offloading system.

data, to decide the procedure based on the messages. This paper also proposes several cooperative download methods and data sharing procedures. III-E describes the cooperative download methods. III-F describes the sequences of data sharing. Next, group members download data from the Internet based on the information decided in the previous step. Here, the proposed system supports URI (uniform resource locator) and so on to specify the data since it is realized to use a range request function [11] which is a kind of conditional request [12] within HTTP/1.1 [13], [14] to enhance its flexibility and versatility. In addition, the function is also usable in the future because it is compatible with HTTP/2.0 [15]. When they have finished downloading data, they share the data among other group members via local communication such as wireless LAN and Bluetooth. After that, the terminals leave the group and finish the procedure.

As shown by the above procedure, the proposed system can reduce data traffic by downloading cooperatively and sharing the data among neighbours.

B. Structure of Local Group

A local group which is consists of a single leader who manages the group and single or multiple followers constructed to request to download same data cooperatively. The local group is arranged in a circle centered at the leader with a pre-determined radius and requires communication with the leader via direct communication. Here, we take notice that all terminals exchange control messages except for data traffic via peer-to-peer network on mobile networks. This paper does not focus mainly on the peer-to-peer system since an existing peer-to-peer architecture [16], [17] specialized to share location information among peers has already been proposed. The peer-to-peer architecture focuses on construction of a local group based on location information of peers and support to establish a connection among peers using STUN (session traversal utilities for NATs) [18] and TURN (traversal using relays around NAT) [19].

C. Local Group Construction

First, a terminal requests data searches of neighbours who also request the same data via the peer-to-peer network based on location information.

Then, if there is no terminal requested same data, the terminal promotes a leader and registers own peer ID, own location, request data name or type, terminal ID, and a Boolean value that denotes either requesting or not into a peer-to-peer control server. Peer ID, which is generated when beginning to search neighbours on peer-to-peer network, is used for designating the peer on peer-to-peer network. The requested data name denotes as URI (uniform resource locator) of the requested data and so on. The terminal ID, is generated when starting to communicate among neighbours via local network, is used for designating the terminal on direct communication.

If there already exists a group, the terminal connects to its leader and establishes peer-to-peer connection and then it becomes a follower of the group. Leader and followers

periodically update the registered information in the peer-to-peer control server for eliminating obsolete information.

As the above procedure shows, the proposed system searches terminals who request the same data and determines the cooperation among the group members to use a local group via peer-to-peer network.

D. Neighbour Cooperative Data Offloading Method

The proposed system cooperatively obtains and shares data among terminals in a group. The procedure of neighbour cooperative offloading method is described below.

(1) Process of initializing cooperative data offloading

First, after starting the group construction described in III-C, the leader waits a certain time T_{adv}. After that, the leader stops to request data and fixes group members for downloading cooperatively. Then, the leader obtains the number of the terminals within the local group N and assigns a group member ID i $(0 \leq i \leq N)$ to each group member respectively. Note that the leader is assigned group member ID i as 0 and followers are assigned a group member ID as 1 to $N-1$.

(2) Cooperative download process via mobile network

After the initializing process, the leader sends a control message of the group member ID of the receiver to all the followers. The control message also includes a sequence of cooperative data offloading. The detailed information of the cooperative data offloading method and the sequences of data sharing are described in III-E and III-F respectively. On receiving the control message, each follower i decides the offloading procedure and starts to download data based on the method and the sequence in accordance with the information of the received message. After they finish downloading data, each follower sends a completion notification of receiving data to the leader. When the leader receives the notification from all the followers, the leader initiates the establishment of a connection via direct communication with the followers for sharing downloaded data in the local network.

(3) Connection establishment process via direct communication

After completing the above process, the leader sends a request message to establish a local connection with all the followers in the local network. On receiving the request message, each follower establishes the local connection using information of the leader obtained from the peer-to-peer control server in advance. III-F describes the detailed sequence and the timing of the establishment of the connection.

(4) Data sharing process in local network

After the establishment of the connection in the local network, the group initiates the sharing of the downloaded data based on the predefined sequence which is decided from the information received in advance. III-F describes the detailed sequence also. When a follower obtains the complete data, it sends a notification that indicates completion of data to the leader.

(5) Process of finalizing cooperative data offloading

When the leader receives notifications from all the followers, the leader sends a reply message to them to notify them of the end of the cooperative data offloading. On receiving the message, each follower disconnects the local connection and leaves the group.

E. Cooperative Download Method

This section proposes cooperative download methods which are (1) leader-elected download method (LCD) and (2) distributed cooperative download method (DCD) to be applied as a data download from the Internet.

(1) Leader-elected cooperative download method (LCD)

Leader-elected cooperative download method (LCD) is a method where the leader downloads overall data and, then, sends the downloaded data to all the followers via local network, using direct communication. Figure 2 shows example behavior of LCD. Note that it is not necessary that the leader must be the same as the leader of the group though the leader is defined as a responsible terminal to download overall data for convenience.

First, the neighbours request the same data construct as a local group based on the local group construction process. Then, the group elects a responsible terminal as a leader and the leader downloads overall data from the Internet. The leader sends and shares the overall data to all the followers via local network.

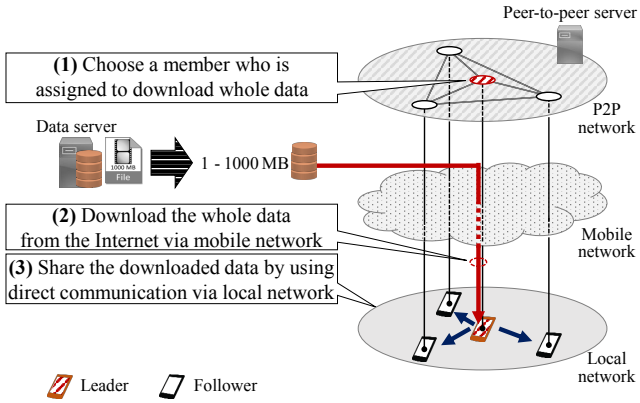

Fig. 2. Leader-elected cooperative download method (LCD).

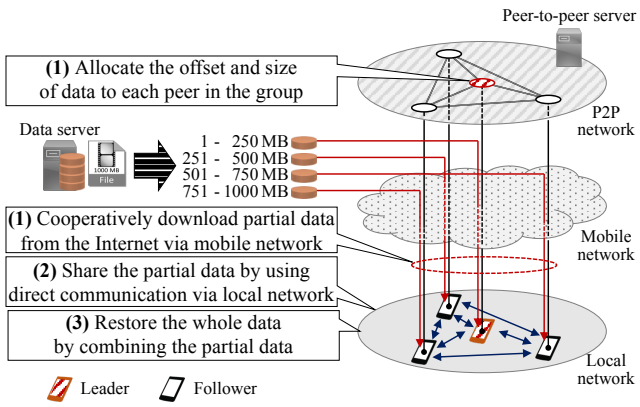

Fig. 3. Distributed cooperative download method (DCD).

(2) Distributed cooperative download method (DCD)

Distributed cooperative download method (DCD) is a method where all group members download partial data and, then, share the partial data with each other. Figure 3 shows example behavior of DCD.

First, the neighbours request the same data construct as a local group based on the local group construction process as well as LCD. Then, the leader assigns the range and size of the data to all the followers. Here, the size and range are calculated based on data size s_j, the number of terminals in the group N and their group member ID i from Eq. (1) and Eq. (2).

$$s_{i,j}^{(\mathrm{head})} = \lfloor \frac{s_j \times i}{N} \rfloor + 1 \quad (1)$$

$$s_{i,j}^{(\mathrm{tail})} = \lfloor \frac{s_j \times (i+1)}{N} \rfloor \quad (2)$$

As above calculation shows, the leader and followers download partial data. After downloading partial data, all the followers send a notification to the leader and then all the terminals share the partial data via local network. Last, they restore the whole data by combining the received partial data.

In addition, we also propose an enhanced variant of DCD which has different operations than the normal DCD to reduce connection switching in local networks. In this variant, after downloading the partial data of all the members, all the followers send their partial data to the leader. After receiving all the partial data, the leader restores the whole data by combining them. Last, the leader sends the whole data to each follower.

F. Data Sharing in Local Network

This section proposes 3-types ways of data sharing to execute LCD or DCD. Here, we define that a sender of direct communication is called master and a receiver of direct communication is called slave. Note that LCD executes the below sequence only once since it is only necessary to connect the master with the slave. In contrast with LCD, DCD requires execution of the below sequence more than once while changing masters. In DCD, a master does not reconnect with a slave who has already finished the exchange of partial data. The proposed 3-type ways of data sharing are described below.

(1) Data Sharing Type A: A master establishes a connection with all the slaves through direct communication. After the connection establishment, the master exchanges data with all the slaves simultaneously. After that, they abolish the connection.

(2) Data Sharing Type B: A master establishes a connection with all the slaves through direct communication. After the connection establishment, the master exchanges data with all the slaves one-by-one. After that, they abolish the connection.

(3) Data Sharing Type C: A master establishes a connection with a the slave through direct communication. After the connection establishment, the master exchanges data with the slave. After that, they abolish the connection. The master repeats the above for the unconnected terminals until they disappear.

IV. Experimental Evaluation

In this section, we evaluated the proposed system through the experiments by (1) the measurement of the overall performance as well as (2) the comparison among the cooperative download methods and the data sharing types. In the experiments, we implemented the proposed system as an application to Android OS on 5 smartphones (ASUS Zenfone2 Laser, Android 5.0.2).

A. Experiment 1: Setups of Overall Performance Evaluation

This experiment evaluated the overall performance of the proposed system to clarify the effectiveness of the proposed system. This experiment used NTT EAST FLET'S HIKARI NEXT [21], [22], which has the downlink speed is 100 Mbps and the uplink speed is 100 Mbps respectively, as the backbone network and used wireless LAN access point with IEEE 802.11g instead of mobile networks to simplify the experiment. Wi-Fi Direct [20] as a local direct communication method was used. The smartphones were fixed on a desk in a row at 5 cm intervals. In this experiment, we chose a cooperative download method as an enhanced variant of DCD and data sharing type B.

In the procedure of this experiments, first we boot the application on single smartphone and the smartphone waits the requesting time T_{adv} which is set to 60 seconds. After that, we boot the application on the other smartphones one-by-one at 5 seconds interval. After booting the application of all of the smartphones, each terminal cooperatively downloads 10 MByte data via the proposed offloading system. Note that the data request and download via the Internet was implemented on the basis of a range request [12] which is a kind of conditional request [11] in HTTP/1.1 [13], [14]. We also focused the evaluation as the effect for mobile networks and fixed networks, and thus we except the traffic of direct communication from the result of the total amount of traffic. We compared the proposed system with the traditional client/server (C/S) model.

B. Experiment 1: Results of Overall Performance Evaluation

Figure 4 shows total sent/received traffic. The traditional C/S model increases the total amount of the received data in proportion to the number of terminals since all the terminals need to download whole data independently. In contrast with the C/S model, the total amount of received data of the proposed system is not changed data even if the number of terminals is increased. This is because the group members of the proposed system can cooperate with other members in the group, and thus data are divided to partial data based on the number of group members. Hence, they do not need to download whole data. However, since the proposed system requires cooperation among the group members, the total sent traffic of the proposed system is increased as increasing of the number of terminals in comparison with the C/S model.

Figure 5 shows an amount of sent/received data and control messages of each role on the proposed system. Note that the result of the C/S model indicates the result of a single terminal. Each terminal of the proposed system reduces the received traffic in comparison to the terminal of the C/S model. As mentioned above, the proposed system can cooperatively download data among the group members, and thus the amount of received data of each terminal decreases since they only download partial data from the Internet. In contrast with the amount of received data, the amount of the sent data of the proposed system increases in comparison with that of the C/S model. In particular, it can be seen that the leader is largest amount of sent data of group members. This is because a leader sends control messages to all followers and needs to update the group information on the peer-to-peer server. In addition, the proposed system imposes control traffic on each member since the proposed system requires exchange control messages via P2P network on a mobile network.

C. Experiment 2: Setups of Comparing among Data Sharing Types and Cooperative Download Methods

The experiment evaluated (1) the time to complete the data sharing of all the members to compare the differences among the data sharing types as described in III-F and also (2) that of cooperative offloading methods as described in III-E. In this experiment, we chose a direct communication method as Bluetooth on the basis of Nearby Connections API 2.0 [23]. In addition, this experiment analyzes traffic flows to trace them using Wireshark [24].

This experiment used 3 to 5 terminals that send/receive 5 MByte data to each other and measured the time to complete the data sharing of all the members from beginning to end. Note that this experiment used pre-measured throughput in mobile network to eliminate the effect of the performance fraction of fixed network and the assumed congestion such as a middle of events. The throughput was measured seven times at TOKYO BIG SIGHT, where the large-scale event was held, from 9:22 to 9:40 on November 31st in 2017. Consequently, the throughput of the mobile network was set with a downlink speed of 0.8 Mbps and an uplink speed of 7.48 Mbps. Note that, in the comparison of the cooperative download methods, the data sharing type C was used, since it has a better performance than others based on the experimental results. This experiment varies the size of data from 7.5 Mbyte to 45 Mbyte, with varying number of terminals.

D. Experiment 2: Results of Comparing among Data Sharing Types and Cooperative Download Methods

Figure 6 shows that the difference of the time to complete data sharing between data sharing type B and the data sharing type C is small though the data sharing type A needs a longer time to complete the control than the others do. In particular, when there are 5 terminals, an increasing ratio of the data sharing type A is larger than that of the others. We investigate the reason to trace data. Consequently, the Bluetooth communication via nearby connections sends data using TDMA (time-division multiple access)-like scheduling from a master to the slave when the terminals simultaneously send data. Therefore, if the number of terminals is increased, the delay derived from switching and waiting is much imposed since the switching of the slaves is increased. In contrast with the data sharing type A, the difference between the data sharing type B and the data sharing type C is small. Hence, they can avoid the above problem by sending data one-by-one

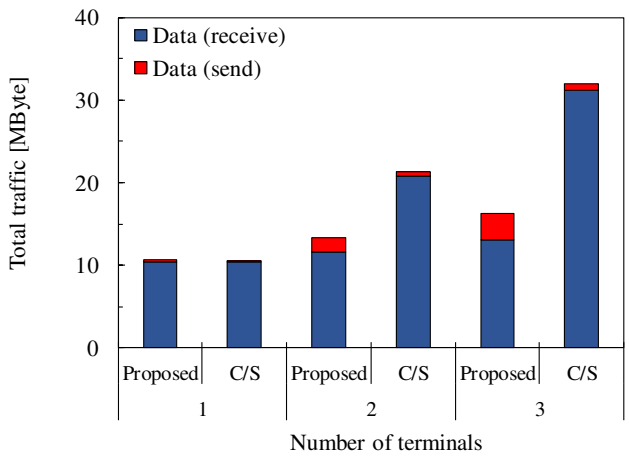

Fig. 4. Experiment 1: Total traffic.

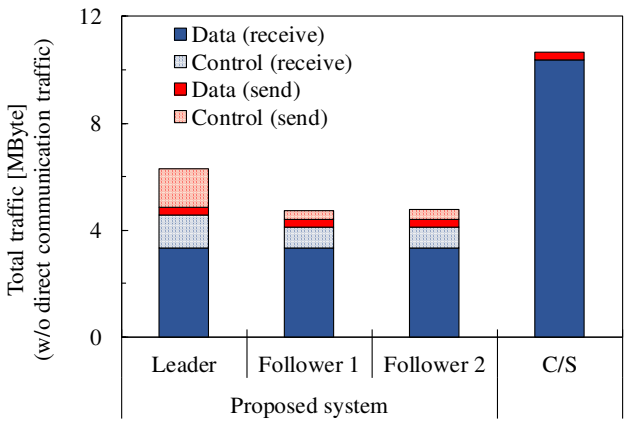

Fig. 5. Experiment 1: Total traffic of each role in the proposed system.

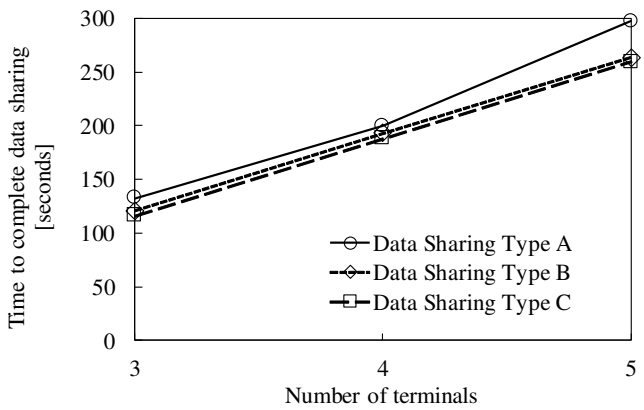

Fig. 6. Experiment 2: Comparison among the proposed data sharing types.

using own scheduling algorithm even if terminals establish some connections.

Figure 7 shows that the time to complete the data sharing of LCD increases in comparison to that of DCD as increasing of the sharing data size when the number of terminals is 2. The downloading data size of each terminal is decreased in DCD because it downloads partial data whereas a leader downloads overall data in LCD. Thus, each terminal in DCD can download partial data in parallel with the others and, thus, the necessary download time is decreased in comparison to LCD. In addition, when the size of sharing data becomes larger, DCD has shorter time to complete the data sharing than LCD. However, although the time in LCD linearly increases as the number of terminals increases, regardless of data size, DCD has a larger increasing ratio than LCD. Consequently, the difference of the time to data sharing between LCD and DCD becomes small when the number of terminals is increased. This is because LCD only sends whole data from a master to the all slaves. Hence, the count of connection switching is $N-1$ when the number of group members is N and thus the count of connection switching increases according to the order $\mathcal{O}(n)$. However, in DCD, all the group members are required to switch a connection to each other, and thus the count of connection switching is ${}_NC_2$. Hence, the difference decreases because the delay of switching and waiting is imposed when DCD increases the count of connection switching according to the order $\mathcal{O}(n^2)$.

V. Conclusion

This paper proposed a neighbour cooperative data offloading system based on location information and evaluated by experiments in actual environment. In addition, this paper also proposed two types of cooperative download methods and three types of sequences of data sharing. The experiments evaluated the overall performance of the proposed system and the difference among the cooperative offloading methods and the data sharing types. In the future, the performance of the proposed system should be evaluated through a large-scale experiment. In addition, the download method which considers redundancy should also be studied.

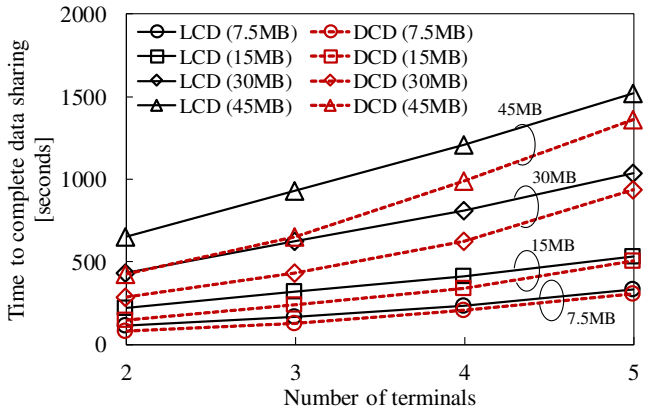

Fig. 7. Experiment 2: Comparison between the proposed offloading methods.

Acknowledgment

This work was supported by TAF Research Grant and JSPS KAKENHI Grant Number 17K12680 and 17K06441.

References

[1] Cisco Visual Networking Index, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[2] F. Rebecchi, M.D. Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," IEEE Commun. Surveys & Tutorials., vol.17, no.2, pp.580–603, Nov. 2014.

[3] docomo Wi-Fi (in Japanese), https://www.nttdocomo.co.jp/service/wifi/docomo_wifi

[4] X. Kang, Y.-K. Chia, and S. Sun, "Mobile data offloading through a third-party WiFi access point: An operator's perspective," 2013 IEEE Globecom Workshops, pp.696–701, Atlanta, USA, Dec. 2013.

[5] E. Bulut and B.K. Szymanski, "WiFi access point deployment for efficient mobile data offloading," 1st ACM Int. Workshop on Practical Issues and Applications in Next Gen. Wireless Netw. (PINGEN 2012), pp.45–50, Istanbul, Turkey, Aug. 2012.

[6] S.-S. Kang and M.W. Mutka, "A mobile peer-to-peer approach for multimedia content sharing using 3G/WLAN dual mode channels," J. Wireless Commun. & Mob. Comput., vol.5, no.6, pp.633–645, Sept. 2005.

[7] S. Sharafeddine, K. Jahed, N. Abbas, E. Yaacoub, and Z. Dawy, "Exploiting multiple wireless interfaces in smartphones for traffic offloading," 2013 1st Int. Black Sea Conf. Commun. and Netw. (BlackSeaCom 2013), pp.142–146, Batumi, Georgia, July 2013.

[8] A. Le, L. Keller, H. Seferoglu, B. Cici, C. Fragouli, and A. Markopoulou, "MicroCast: Cooperative video streaming using cellular and local connections," IEEE/ACM Trans. Netw., vol.24, no.5, pp.2983–2999, Oct. 2016.

[9] H. Hanano, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito, "A cooperative download method for low cost video ads dissermination through WiFi-celllar hybrid network," IPSJ Journal, vol.51, no.2, pp.440–452, Feb. 2010. (in Japanese)

[10] Y. Takamatsu, W. Sun, K. Yasumoto, Y. Yamauchi, and M. Ito, "Energy-aware cooperative download method for mobile phones utilizing street pass communication," IPSJ Journal, vol.53, no.2, pp.783–794, Feb. 2012. (in Japanese)

[11] R. Fielding (Ed.), Y.Lafon (Ed.), and J. Reschke (Ed.), "Hypertext Transfer protocol (HTTP/1.1): Range requests," IETF RFC7233, June 2014.

[12] R. Fielding (Ed.) and J. Reschke (Ed.), "Hypertext transfer protocol (HTTP/1.1): Conditional requests," IETF RFC7232, June 2014.

[13] R. Fielding (Ed.) and J. Reschke (Ed.), "Hypertext transfer protocol (HTTP/1.1): Message syntax and routing," IETF RFC7230, June 2014.

[14] R. Fielding (Ed.) and J. Reschke (Ed.), "Hypertext transfer protocol (HTTP/1.1): Semantics and content," IETF RFC7231, June 2014.

[15] M. Belshe, R. Peon, and M.Thomson (Ed.), " Hypertext transfer protocol version 2 (HTTP/2)," IETF RFC7540, May 2015.

[16] Y. Shimomura and T. Miyoshi, "Location-based peer-to-peer communication system," IEICE Tech. Rep., vol.117, no.114, pp.77–80, ICM2017–18, July 2017. (in Japanese)

[17] T. Miyoshi, Y. Shimomura, and O. Fourmaux, "G-LocOn: A P2P-based communication framework for geo-location oriented networks," IEICE Information and Commun. Tech. Forum (ICTF 2018), Graz, Austria, July 2018.

[18] J. Rosenberg, R. Mahy, P Matthews, and D. Wing, "Session traversal utilities for NAT (STUN)," IETF RFC5389, Oct. 2008.

[19] R. Mahy, P. Matthews, and J. Rosenberg, "Traversal using relays around NAT (TURN): Relay extensions to session traversal utilities for NAT (STUN)," IETF RFC5766, April 2010.

[20] Wi-Fi Direct, http://www.wi-fi.org/discover-wi-fi/wi-fi-direct

[21] NTT EAST FLET'S HIKARI, https://flets.com/english/

[22] NTT Plala Inc., http://www.nttplala.com/english/

[23] Nearby Connections API, https://developers.google.com/nearby/connections/overview

[24] Wireshark, https://www.wireshark.org