LoCO: Local Cooperative Data Offloading System Based on Location Information

Taku Yamazaki, Kazuma Asano, Satoshi Arai, Yusuke Shimomura, and Takumi Miyoshi

Shibaura Institute of Technology, Saitama, Japan

Abstract—The development of high speed mobile networks and the widespread use of smartphones have enabled users to easily obtain large data volumes via the Internet. This causes a heavy consumption of network resources, a burden on the available bandwidth. To solve such problems, a data offloading method with a wireless LAN access point has been used to distribute traffic from mobile 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 local cooperative data offloading system (LoCO) that reduces the overall traffic by sharing data, with direct communication between neighbors based on their location-related information. Moreover, the authors implemented the LoCO system on Android smartphones and clarified its performance in comparison with a traditional client/server system through experiments to download data in a real-world environment.

Keywords—cooperative offloading, distributed download, load balance, load reduction, location information, peer-to-peer.

1. Introduction

The development of high speed mobile networks and the widespread use of smartphones have enabled users to easily obtain a large volume of data 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 approximately 7 times from that of 2016. In particular, it is also predicted that, in 2021, video content traffic will account for approximately 78% of all mobile data traffic. The Ericsson mobility report [2] also predicts that the overall mobile data traffic will increase by approximately 8 times from that of 2017. In 2023, video content traffic will account for approximately 73% of all mobile data traffic. Thus, on the basis of the reports, to alleviate the effect of large volume data, such as a video content, the heavy use of mobile networks' resources must be dealt with.

For solving this problem, data offloading methods to distribute and reduce mobile traffic have been discussed [3]. As one of the major data offloading methods, a method with a wireless LAN access point [4]–[10], known as Hotspot [11], which changes the communication path and distributes data traffic from mobile networks to fixed networks via the wireless LAN access point, referred to as

vertical handover [12], has been widely used. As a different approach from the above, cooperative data offloading, downloading, and sharing methods in which terminals download data to cooperate with their neighbors [13]–[18] have been proposed.

Although the data offloading method with a wireless LAN access point can only distribute data traffic to move it from mobile networks to fixed networks, the overall data traffic is the same. Hence, the method is not able to reduce the data traffic generally. The cooperative offloading and downloading methods can reduce the data traffic. However, the methods require modifications to access point and terminal firmware. 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 determines the range of data for a partial download based on mobility prediction. However, its sharing efficiency may decrease under actual environments, since the method strongly depends on the accuracy of mobility prediction.

This paper proposes a local cooperative data offloading system (LoCO) which is able to download cooperatively and shares data among neighbors via direct communications based on location information, relying on peer-to-peer communications. In addition, LoCO is likely to be able to configure and select a cooperative download method. This paper also proposes cooperative download methods referred to as leader election-based cooperative download (LCD), distributed cooperative download (DCD) and enhanced DCD (eDCD). Moreover, this paper implements the LoCO system on smartphones that run it at the application-level, which means that it does not require any firmware-level modifications.

2. Related Work

Currently, a data offloading method with a wireless LAN access point is widely used to distribute data traffic from mobile networks to fixed networks [4], [8]–[10]. The deployment methods of wireless LAN access points have also been discussed in [5]–[7] for improving the data offloading effect. 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 may change the communication path from mobile to other networks. However, since this method only focuses on the data traffic distribution between mobile networks and fixed networks, it is not able to reduce data traffic. It is only applied to specified areas, because it only enables an area where there is a wireless LAN access point. Hence, the distribution effect of traffic strongly depends on the location.

The cooperative download method, where users download partial data and disseminate it to each other by passing it opportunistically based on a mobility prediction, has been proposed in [17], [18]. Here, each terminal sends a control message to a server. The message includes its own current position, destination, departure time and information about own partial data. Upon receiving the control message, the server generates a reply message which includes the ID of the terminal which is capable of encountering the sender of the control message, the probability of encountering the sender of the control message, the time of encountering and information about own data. Then, the server sends the message to each terminal. Based on the information received, each terminal predicts positions of encountering the other terminals which 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 prediction of user mobility is difficult under actual environments. Hence, the sharing efficiency of the method may decrease since the method strongly depends on the accuracy of mobility prediction.

3. Local Cooperative Data Offloading System Based on Location Information

This paper proposes a local cooperative data offloading (LoCO) system that reduces data traffic to cooperatively download and share data among local terminals called neighbors, based on their location information.
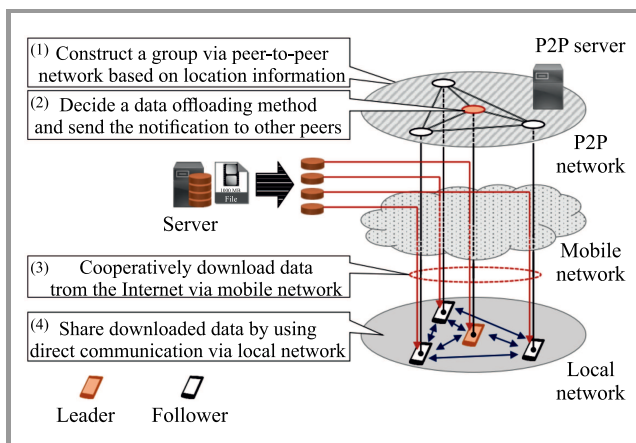

Fig. 1. Overview of the LoCO system.

Figure 1 shows the structure of the LoCO system. First, the LoCO constructs a local group among neighbors that request the same data via a peer-to-peer network on mobile networks. Then, group members exchange control messages with each other, which include the download method and the data sharing type, to decide the offloading algorithm based on the messages. This paper also proposes several cooperative download methods and data sharing types. The LoCO system supports uniform resource locator (URL) etc. to specify the data, since it is relied upon to perform the range request function [19] which is a kind of conditional request [20] within HTTP/1.1 [21], [22] to enhance its flexibility and versatility. The function is also usable in the future because it is compatible with HTTP/2.0 [23]. After finishing downloading data, they share the data among other group members via a local communication medium, such as wireless LAN, Bluetooth, etc. After that, the terminals leave the group and the procedure ends.

3.1. Local Group Structure

This section defines the structure of a local group in the LoCO system, which consists of a single leader that manages the group and single or multiple followers. It is constructed to download the same data cooperatively. The local group is arranged in a circle centered around the leader, has a pre-determined radius and requires that the leader communicate with others via direct communication. All terminals exchange control messages except for data traffic via a 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 [24], [25] specialized in sharing location information among peers, has already been proposed. The peer-to-peer architecture focuses on the construction of a local group based on location information of peers and supports the establishment of a connection among peers using session traversal utilities for NATs (STUN) [26] and traversal using relays around NAT (TURN) [27].

3.2. Local Group Construction

This section introduces the procedure of local group construction. Table 1 shows an example of peer information stored in the P2P server.

Table 1
Example of peer information in P2P server

Peer ID	Position	Content	Terminal ID	Requesting
P_1	(X_1, Y_1)	Data1	M_1	True
P_2	(X_2, Y_2)	Data2	M_2	True
P_3	(X_3, Y_3)	Data1	M_3	False
P_4	(X_4, Y_4)	Data3	M_4	True

First, when a terminal wants to obtain some data, it it searches its neighbors that also request the same data via the peer-to-peer network based on location informa-

tion. Then, if there is no terminal that requested the same data, terminal i promotes a leader and registers own peer ID P_i, own location (X_i, Y_i), request data name or type, terminal ID M_i, and a Boolean value that denotes either requesting or not in the peer-to-peer control server. Peer ID P_i, which is generated at the beginning of the process of searching for neighbors on the peer-to-peer network, is used for designating the peer on the peer-to-peer network. The requested data name denotes the uniform resource identifier (URI) [28] of the requested data and so on. The terminal ID M_i, is generated when the communication among neighbors starts via the local network, and is used for designating the direct communication terminal.

If a group already exists, the terminal connects to its leader and establishes a peer-to-peer connection and then it becomes a group follower. The leader and the followers periodically update the registered information in the peer-to-peer control server in order to eliminate obsolete information.

Therefore, the LoCO system searches terminals that request the same data and enables cooperation between the group members to use the local group via a peer-to-peer network.

3.3. Local Cooperative Data Offloading Method

The LoCO system cooperatively obtains and shares data among terminals in a group. The procedure of the local cooperative offloading method among neighbors is described below.

1. Initializing process of cooperative data offloading. First, after starting the group construction of a group for content j, 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 terminals within the local group N and assigns a group member ID k ($0 \leq k \leq N-1$) to each group member. Note that the group member ID k of 0 is assigned to the leader, and a group member ID k of 1 to $N-1$.

2. Cooperative download process via mobile network. After the initialization process, the leader ($k = 0$) sends control messages with containing the group member ID k of the receiver and the total number of members in the group N to all followers ($k = 1, 2, \ldots, N-1$). The control message also includes the method of cooperative data download and the type of data sharing which will be used for sharing content j in the group. Upon receiving the control message, each follower k 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 notification of completing data download to the leader. When the leader receives the notification from all followers, the leader initiates the establishment of a connection via direct communication with the followers, for sharing downloaded data in the local network.
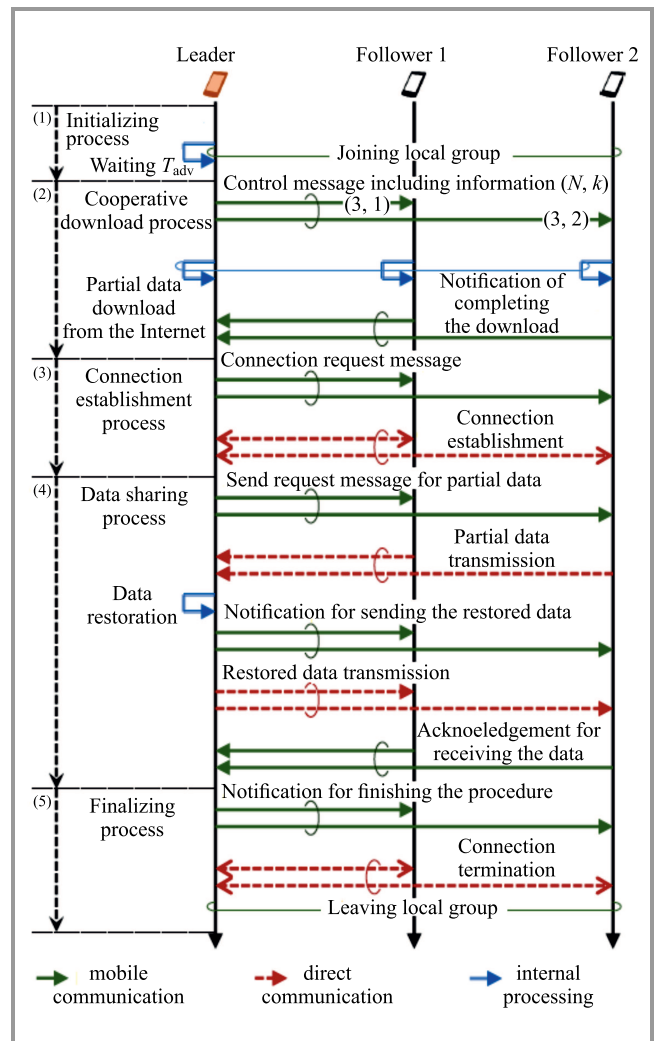

Fig. 2. Operation sequence of the LoCO system.

3. Connection establishment process via direct communication. After completing the above process, the leader sends a connection request message to establish a local connection with all followers in the local network. Upon 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. The LoCO system does not depend on a specific medium. Currently, it applies mboxWi-Fi Direct [29]–[32] or Bluetooth [33] as the local direct communication medium. Therefore, the LoCO system also needs to decide the medium for local communication during the above process.

4. Data sharing process in local network. After 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 based on information received in advance. When a follower obtains the complete data, it sends an acknowledgement to the leader, indicating completion of data reception.

5. Finalizing process of cooperative data offloading. When the leader receives acknowledgements from all fol-

lowers, the leader sends a notification informing them of the completion of cooperative data offloading. Upon receiving the notification, each follower terminates the local connection and leaves the local group.

3.4. Local Cooperative Download Method

This section proposes local cooperative download methods of the LoCO system which are (1) leader election-based cooperative download, (2) distributed cooperative download (DCD) and (3) enhanced DCD (eDCD) to be applied for downloading data from the Internet.

1. Leader election-based cooperative download. Leader election-based cooperative download (LCD) is a method where the leader downloads all data and, then, sends the downloaded data to all followers via a local network, using direct communication. Figure 3 shows an example behavior of LCD. Note that it is not necessary for the leader to be the same as the leader of the group, although the leader is defined, for convenience purposes, as a terminal responsible for the download of overall data.
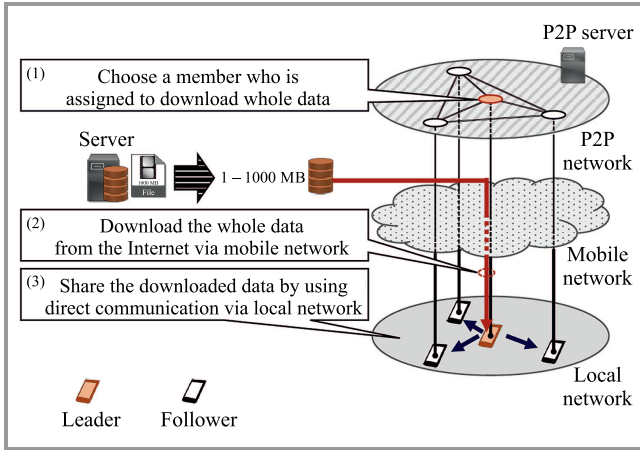

Fig. 3. Leader election-based cooperative download (LCD).

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

LCD has a simple procedure to share the data, since the leader only has to send the data to each follower. However, an unfairness in network resource consumption may occur, due to the characteristics of the leader election process. Therefore, this method is appropriate for a situation in which some content will be exchanged while changing the leader. In addition, another situation is that the leader is unrestricted or it has sufficient resources in terms of network resource consumption, such as fixed devices (e.g. digital signage, vending machine and so on) or devices connected to unlimited wireless LAN, etc.

2. Distributed cooperative download. Distributed cooperative download (DCD) is a method where all group

members download partial data and, then, transfer it to others. Figure 4 shows an example behavior of DCD.
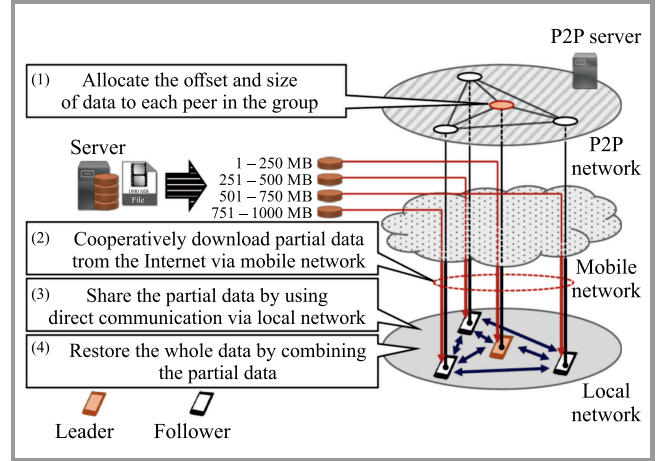

Fig. 4. Distributed cooperative download (DCD).

First, the terminals which request the same data construct a local group based on the same construction process as LCD. Then, the leader assigns the range and size of the data to followers. Here, the size and range $[s_{j,k}^{(\text{head})}, s_{j,k}^{(\text{tail})}]$ are calculated based on data size s_j of a content j, the number of terminals in group N and their group member ID k from:

$$s_{j,k}^{(\text{head})} = \left\lfloor \frac{s_j \times k}{N} \right\rfloor + 1 , \qquad (1)$$

$$s_{j,k}^{(\text{tail})} = \left\lfloor \frac{s_j \times (k+1)}{N} \right\rfloor . \qquad (2)$$

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

LCD achieves fairness in terms of network resource consumption in both mobile and local networks. However, due to its characteristics, DCD requires each terminal to connect with all others, like in the full mesh topology, because they need to send partial data between each other. Hence, construction of a local group may become complex, because DCD needs to establish and keep many connections. The switching and the keeping of many connections may cause higher delays.

3. Enhanced DCD. To ease the requirement, this paper also proposes an enhanced variant of DCD (eDCD) which operates differently from the normal DCD in order to reduce the complexity of the local network. eDCD mainly focuses on traffic reduction via the Internet without constructing a full mesh topology. Figure 5 shows an example behavior of eDCD.

In this variant, first, after group construction, each terminal download partial data by using the same method as the normal DCD. After downloading the partial data of all the members, all followers send their partial data to the
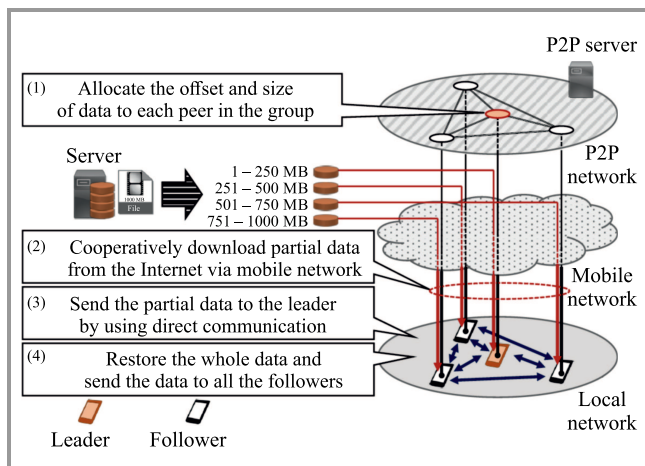
Fig. 5. Enhanced distributed cooperative download (eDCD).

leader. After that, the leader restores the overall data by combining them. Finally, the leader sends the whole data to each follower. Therefore, in eDCD, the leader collects all partial data from all followers. As a result, each follower only establishes a connection with the leader based on a tree topology which includes the leader as the root. Namely, the eDCD reduces the number of connections without compromising the advantage, since each follower does not need to establish a connection with other followers.

3.5. Data Sharing in Local Network

This section proposes 3 ways of data sharing with the use of the LoCO system, to execute a local cooperative download. Here, we define that the sender of direct communication is called master, and the 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 that has already finished the exchange of partial data. The 3 data sharing methods proposed are described below.

1. Data sharing type A. The master establishes a connection with all slaves through direct communication. After the connection is established, the master exchanges data with all slaves simultaneously. After that, they abolish the connection.

2. Data sharing type B. The master establishes a connection with all slaves through direct communication. After the connection is established, the master exchanges data with all slaves, on a one-by-one basis. After that, they abolish the connection.

3. Data sharing type C. The master establishes a connection with a slave through direct communication. After the connection is established, the master exchanges slave. After that, they abolish the connection. The master repeats the above for the unconnected terminals until they disappear.

4. Performance Evaluation

In this section, we evaluate the LoCO system based on real world experiments, by measuring the overall performance as well as comparison between cooperative download methods and data sharing types. In the experiments, we implemented the LoCO system as an application on Android OS [34] on 5 smartphones.

4.1. Overall Performance Evaluation Setup

This experiment evaluated the overall performance of the LoCO system to clarify its effectiveness. This experiment used NTT East FLET'S Hikari Next [35] as a backbone network and NTT Plala [36] as an Internet service provider, offering the downlink and uplink speed of 100 Mbps. In addition, a wireless LAN access point with IEEE 802.11g is used instead of the mobile networks to simplify the experiment. Wi-Fi Direct [29]–[32] was selected as the local direct communication method. The smartphones were fixed on a desk in a row, at 5 cm intervals. We chose the eDCD cooperative download method and the data sharing type B.

First, we boot the application on a single smartphone and the smartphone waits the requesting time T_{adv} which is set to 60 s. After that, we boot the application on the other smartphones, one-by-one, at 5 s intervals. Next, each terminal cooperatively downloads 10 MB of data via the LoCO system. Note that the data request and download via the Internet was performed based on the range request [20], which is a kind of a conditional request [19] in HTTP/1.1 [21], [22]. We also focused the evaluation on the effect on mobile and fixed networks, and thus we excluded the direct communication traffic from the result describing the total amount of traffic. We compared the LoCO system with the traditional client/server (C/S) model.

4.2. Results of Overall Performance Evaluation

Figure 6 shows total sent and received data of the traditional C/S model and the proposed system. The C/S model increases the total amount of data received in proportion to the number of terminals, since all of them need to download all data independently. In contrast to the C/S model, the total amount of data received in the case of the LoCO system is not changed, even if the number of terminals is increased. This is because the members of the LoCO group may cooperate with others, and thus data are divided into partial data based on the number of group members. Hence, they do not need to download all data. However, since LoCO requires cooperation among group members, the total amount of traffic sent is increased as the number of terminals increases, in comparison with the C/S model. Figure 7 shows the amount of sent and received data and control messages of each role on the LoCO system. Note that the result of the C/S model indicates the result of a single terminal. Each LoCO terminal reduces the received traffic in comparison to the terminal of the C/S model. As
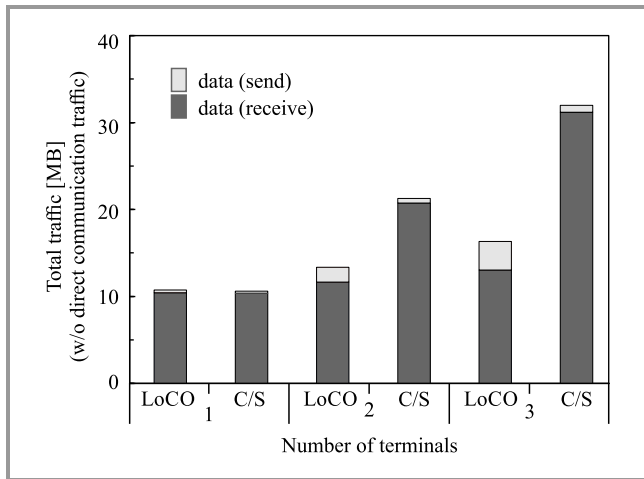
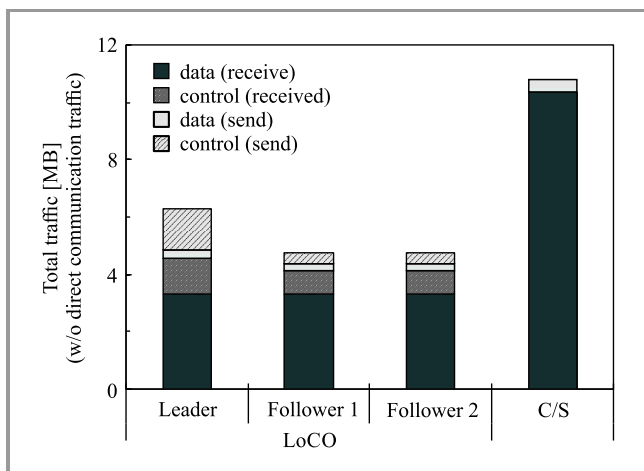***Fig. 6.*** Total traffic with varying number of terminals.

Fig. 7. Total traffic of each role in the LoCO system.

mentioned above, LoCO 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. In contrast with the amount of received data, the amount of sent data of increases in LoCO in comparison to the C/S model. In particular, it can be seen that the leader has the largest amount of sent data among all group members. This is because the leader sends control messages to all followers and needs to update group information on the peer-to-peer server. In addition, LoCO imposes control traffic on each member since it requires exchange control messages via the P2P network.

4.3. Setup for Comparing Data Sharing Types and Cooperative Download Methods

The experiment evaluated the time required to complete data sharing between all members, to compare the differences between data sharing types and cooperative download methods as well. Here, we chose Bluetooth as the direct communication method [33], based on Nearby Connections API 2.0 [37]. This experiment analyzes traffic flows and

enables to trace them using Wireshark [38] as a packet capturing tool.

3 to 5 terminals were used to send/receive 5 MB of data to each other, and the time required to complete the data sharing phase between all members, from the beginning to the end, was measured. Note that this experiment used throughput in a free flow scenario, as well as throughput in a congested scenario, which is the throughput pre-measured in the mobile network to eliminate the effect of deploying a fraction of the fixed network. A medium-rate congestion level was assumed as well. The throughput was measured seven times at Tokyo Big Sight, where a large-scale event was held, from 9:22 to 9:40 on November 31st 2017. Consequently, the throughput of the mobile network was determined to offer the uplink speed of 7.48 Mbps and the downlink speed of 0.8 Mbps in the congested scenario, to compare with the uplink speed and downlink speed of 11 Mbps for the normal scenario. Note that, in comparison with the cooperative download methods, data sharing type C was used, since it offers better performance than others, based on the experimental results. This experiment varies the size of data from 7.5 MB to 45 MB, with a varying number of terminals.

4.4. Results of Comparisons between Data Sharing Types and Cooperative Download Methods

Figure 8 shows the time required to complete the data sharing phase for 3 types of methods used. The results show that the difference in the time required to complete data sharing is small between type B and type C, though type A needs a longer time to complete the procedure in comparison to the remaining types. In particular, with 5 terminals, the increase observed in the case of data sharing type A is larger than in other scenarios. We investigated the reason by tracing data with the use of a packet capturing tool. Consequently, Bluetooth communication established via nearby connections is relied upon to send data using time-division multiple access (TDMA) scheduling from the master to the slave, when the terminals send data simultaneously. Therefore, if the number of terminals is increased, the delay
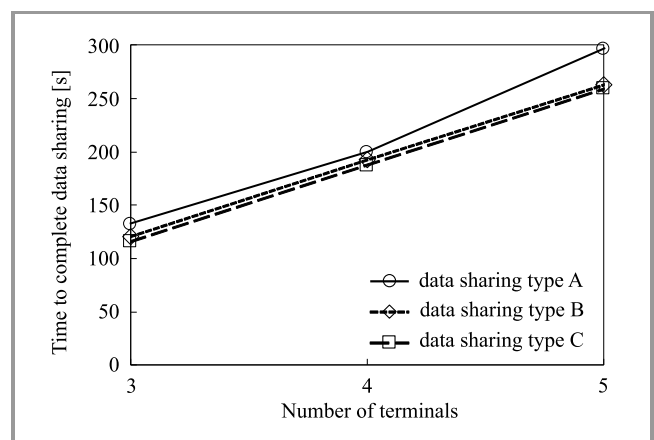

Fig. 8. Comparison between the proposed data sharing types of the LoCO system.

resulting from switching and waiting is much longer, since the switching of the slaves is more intense. In contrast with data sharing type A, the difference between type B and type C is small. Hence, they can avoid the above problem by sending data one-by-one, using their own scheduling algorithm even if some connections are established by the terminals.

Figures 9 and 10 show the time required to complete data sharing in LCD and DCD, in the normal scenario, without congestion, and in a scenario with congestion.
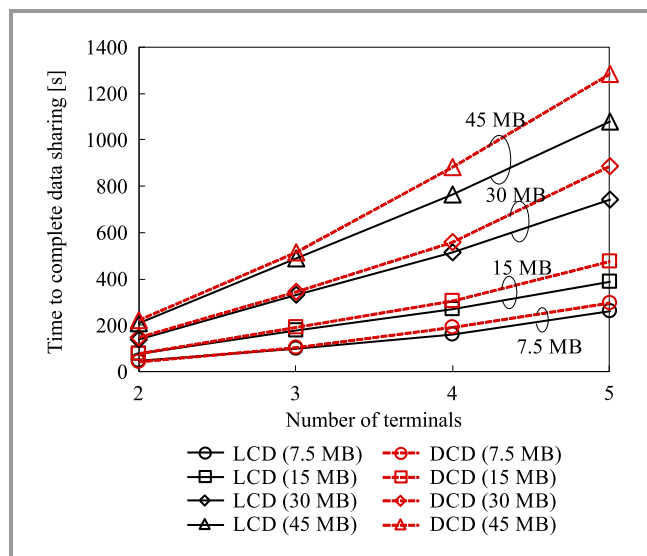

Fig. 9. Comparison between local cooperative download methods of the LoCO system in a normal scenario without congestion (uplink speed and downlink speed are set to 11 Mbps).

Figure 9 shows the time required to complete data sharing in LCD and DCD in a normal environment, where there is no congestion. The result shows that LCD has a shorter time required to complete data sharing than DCD, at least in most cases. In particular, when the data size becomes larger, the difference between LCD and DCD also becomes larger. This is because the scenario involved is a normal environment in which uplink and downlink speeds are set to 11 Mbps. In this case, the time for downloading data from the Internet is decreased in comparison to the congested scenario. As a result, the effect of the distributed download of DCD becomes relatively less evident. Therefore, LCD has the shorter time required to complete data sharing than DCD, since LCD has a simpler data sharing procedure, as it only sends all data to all followers in the local network. In addition, the result also shows that the increase in the time required to complete data sharing is larger in DCD than in LCD when the number of terminals increases. As mentioned above, this is because LCD only sends all data from the master to all slaves, in contrast to DCD. The count of connection switching tasks is $N-1$ when the number of group members is N, and thus the count of connection switching tasks increases according to the order of $\mathcal{O}(n)$. On the other hand, in DCD, all group members need to switch the connection to each other. In other words, the

count of connection switching tasks is $_NC_2$. Hence, the difference between LCD and DCD becomes larger when the number of terminals increases, because the delays caused by switching and waiting are imposed since DCD increases the count of connection switching tasks according to the order of $\mathcal{O}(n^2)$.
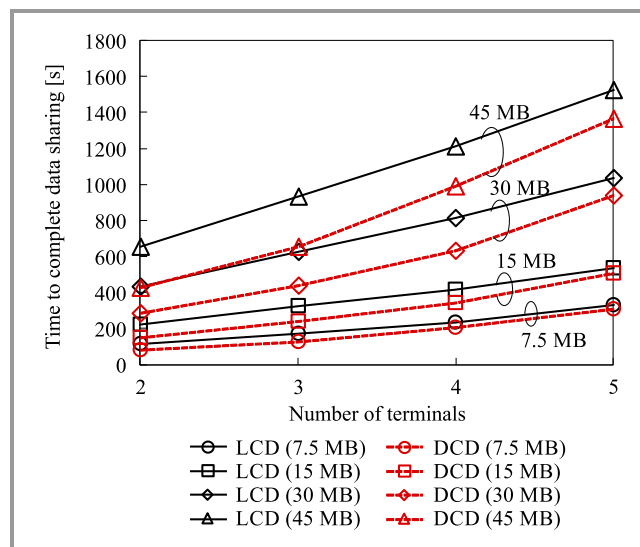

Fig. 10. Comparison between local cooperative download methods of the LoCO system in a congested scenario (uplink speed is set to 7.48 Mbps and downlink speed is set to 0.8 Mbps).

Figure 10 shows the time required to complete data sharing in LCD and DCD, in a congested scenario. The results show that DCD has a shorter time to complete data sharing than LCD, in contrast to the result of Fig. 9. This is because the size of data downloaded by each terminal from the Internet is decreased in DCD, since it downloads partial data, whereas the leader downloads all data in LCD. In DCD, each terminal may download partial data in parallel with other terminals, and thus the necessary download time is decreased in comparison to LCD. Especially, in this scenario, the effect of parallel downloads becomes much larger than in the previous case, since the backbone network is slow due to network congestion. In addition, when the size of data shared becomes larger, the time required to complete data sharing is shorter in DCD than in LCD. However, regardless of data size, although the time in LCD linearly increases as the number of terminals increases, DCD shows a higher increase ratio than LCD. Consequently, the difference between the time required to complete data sharing, observed between LCD and DCD, becomes small when the number of terminals is increased due to the characteristics of DCD mentioned above.

5. Conclusion

This paper proposed a local cooperative data offloading (LoCO) system based on location information, which realizes cooperative download and data sharing among neighbors by using peer-to-peer communication and local

communication. In addition, this paper also proposed two major types of cooperative download methods and three types of data sharing sequences. Furthermore, we implemented the LoCO system on Android smartphones as an application without firmware-level modifications. In order to enable the system to be used on smartphones, experiments were conducted in real world environments to evaluate the overall performance of LoCO and to establish the differences between the cooperative download methods and individual data sharing types. It has been shown that LoCO reduces data traffic by using local cooperation between neighbors when they download the same data.

In the future, in order to precisely assess the scalability of LoCO, its performance should be evaluated in a large-scale experiment, because performance of the cooperative data offloading process is affected by the number of group terminals.

The cooperative download methods remain primitive for the time being, as this paper only confirms their fundamental characteristics in real world environments. However, real world environments, the LoCO system should take into consideration various conditions, such as communication quality degradation caused by radio interference, terminal mobility and so on. Therefore, cooperative download methods taking the above conditions into consideration should be studied as future work.

Acknowledgements

References

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 201–2021" [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[2] "The Ericsson Mobility Report" [Online]. Available: https://www.ericsson.com/en/mobility-report

[3] F. Rebecchi et al., "Data offloading techniques in cellular networks: A survey," IEEE Commun. Surveys & Tutor., vol. 17, no. 2, pp. 580–603, 2014 (doi: 10.1109/COMST.2014.2369742).

[4] "docomo Wi-Fi" [Online]. Available: https://www.nttdocomo.co.jp/service/wifi/docomo_wifi [in Japanese]

[5] X. Kang, Y.-K. Chia, and S. Sun, "Mobile data offloading through a third-party WiFi access point: An operator's perspective", in Proc. IEEE Globecom Worksh. GC Wkshps 2013, Atlanta, GA, USA, 2013, pp. 696–701 (doi: 10.1109/GLOCOMW.2013.6825069).

[6] E. Bulut and B. K. Szymanski, "WiFi access point deployment for efficient mobile data offloading", in Proc. 1st ACM Int. Worksh. on Pract. Issues and Appl. in Next Gener Wirel. Netw. PINGEN 2012, Istanbul, Turkey, 2012, pp. 45–50 (doi: 10.1145/2348714.2348723).

[7] E. Bulut and B. K. Szymanski, "WiFi access point deployment for efficient mobile data offloading", ACM SIGMOBILE Mob. Comput. Commun. Rev., vol. 17, no. 1, pp. 71–78, 2013 (doi: 10.1145/2502935.2502948).

[8] 3GPP, "Architecture enhancements for non-3GPP accesses", 3GPP TS 23.402, March 2018 [Online]. Available: https://www.3gpp.org/DynaReport/23402.htm

[9] 3GPP, "Access to the 3GPP evolved packet core (EPC) via non-3GPP access networks", 3GPP TS 24.302, March 2018 [Online]. Available: https://www.3gpp.org/dynareport/24302.htm

[10] 3GPP, "Access network discovery and selection function (ANDSF) management object (MO)", 3GPP TS 24.312, June 2018 [Online]. Available: https://www.3gpp.org/dynareport/24312.htm

[11] "Hotspot 2.0 (release 2) technical specification", Wi-Fi Alliance, Dec. 2016.

[12] J. Márquez-Barja, C. T. Calafate, J.-C. Cano, and P. Manzoni, "Review: An overview of vertical handover techniques: Algorithms, protocols and tools", Comp. Commun., vol. 34, no. 8, pp. 985–997, 2011 (doi: 10.1016/j.comcom.2010.11.010).

[13] U. Lee et al., "P2P content distribution to mobile Bluetooth users", IEEE Trans. on Veh. Technol., vol. 59, no. 1, pp. 356–367, 2010 (doi: 10.1109/TVT.2009.2030893).

[14] 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. Mobile Comput., vol. 5, no. 6, pp. 633–645, 2005 (doi: 10.1002/wcm.332).

[15] S. Sharafeddine, K. Jahed, N. Abbas, E. Yaacoub, and Z. Dawy, "Exploiting multiple wireless interfaces in smartphones for traffic offloading", in Proc. 1st Int. Black Sea Conf. on Commun. and Netw. BlackSeaCom 2013, Batumi, Georgia, 2013, pp. 142–146 (doi: 10.1109/BlackSeaCom.2013.6623398).

[16] A. Le et al., "MicroCast: Cooperative video streaming using cellular and local connections", IEEE/ACM Trans. on Netw., vol. 24, no. 5, pp. 2983–2999, 2016 (doi: 10.1109/TNET.2015.2501349).

[17] H. Hanano, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito, "A cooperative download method for low cost video ads dissemination through WiFi-cellular hybrid network", IPSJ J., vol. 51, no. 2, pp. 440–452, 2010 [Online]. Available: https://core.ac.uk/download/pdf/75905394.pdf [in Japanese]

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

[19] R. Fielding, Y. Lafon, and J. Reschke, Eds., "Hypertext transfer protocol (HTTP/1.1): Range requests", IETF RFC7233, June 2014 (doi: 10.17487/RFC7233).

[20] R. Fielding and J. Reschke, Eds., "Hypertext transfer protocol (HTTP/1.1): Conditional requests", IETF RFC7232, June 2014 (doi: 10.17487/RFC7232).

[21] R. Fielding and J. Reschke, Eds., "Hypertext transfer protocol (HTTP/1.1): Message syntax and routing", IETF RFC7230, June 2014 (doi: 10.17487/RFC7230).

[22] R. Fielding and J. Reschke, Eds., "Hypertext transfer protocol (HTTP/1.1): Semantics and content", IETF RFC7231, June 2014 (doi: 10.17487/RFC7231).

[23] M. Belshe, R. Peon, and M. Thomson, Eds., "Hypertext transfer protocol version 2 (HTTP/2)", IETF RFC7540, May 2015 (doi: 10.17487/RFC7540).

[24] Y. Shimomura and T. Miyoshi, "Location-based peer-to-peer communication system", IEICE Tech. Rep., vol. 117, no. 114, pp. 77–80 2017 [in Japanese] [Online]. Available: https://www.ieice.org/ken/paper/201707078bvr/eng

[25] T. Miyoshi, Y. Shimomura, and O. Fourmaux, "G-LocOn: A P2P-based communication framework for geo-location oriented networks", in IEICE Inform. and Commun. Technol. Forum ICTF 2018, Graz, Austria, 2018.

[26] J. Rosenberg, R. Mahy, P Matthews, and D. Wing, "Session traversal utilities for NAT (STUN)", IETF RFC5389, Oct. 2008 (doi: 10.17487/RFC5389).

[27] 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 (doi: 10.17487/RFC5766).

[28] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform resource identifier (URI): Generic syntax", IETF RFC3986, Jan. 2005 (doi: 10.17487/RFC3986).

[29] "Wi-Fi Direct" [Online]. Available: http://www.wi-fi.org/discover-wi-fi/wi-fi-direct

[30] "Wi-Fi peer-to-peer (P2P) technical specification", Wi-Fi Alliance, July 2016.

[31] "Wi-Fi peer-to-peer Services (P2Ps) technical specification", Wi-Fi Alliance, July 2015.

[32] "UPnP file transfer service technical specification", Wi-Fi Alliance, June 2015.

[33] Bluetooth [Online]. Available: https://www.bluetooth.com

[34] Android [Online]. Available: https://www.android.com

[35] NTT East FLET'S Hikari [Online]. Available: https://flets.com/english

[36] NTT Plala Inc. [Online]. Available: http://www.nttplala.com/english

[37] Nearby Connections API [Online]. Available: https://developers.google.com/nearby/connections/overview

[38] Wireshark [Online]. Available: https://www.wireshark.org

E-mail: bp14004@shibaura-it.ac.jp
College of Systems Engineering and Science
Shibaura Institute of Technology
Saitama, Japan

Satoshi Arai received his B.E. degree in Electronic Information Systems from the Shibaura Institute of Technology, Tokyo, Japan, in 2018. He is presently an engineer at Nippan Computer Technology Inc., Tokyo, Japan. His research interest is in mobile data offloading.

E-mail: bp14006@shibaura-it.ac.jp
College of Systems Engineering and Science
Shibaura Institute of Technology
Saitama, Japan

Taku Yamazaki – for biography, see this issue, p. 12.

Yusuke Shimomura – for biography, see this issue, p. 66.

Takumi Miyoshi – for biography, see this issue, p. 22.

Kazuma Asano received his B.E. degree in Electronic Information Systems from the Shibaura Institute of Technology, Tokyo, Japan, in 2018. He is presently a master's course student at the Graduate School of Systems and Information Engineering, University of Tsukuba, Ibaraki, Japan. His research interests include mobile data offloading and computer vision.