EPI International Journal of Engineering
Vol. 1, No. 1, February 2018, pp. 60-68
DOI: 10.25042/epi-ije.022018.10

ISSN:2615-5109

Transmission Scheduling Method for Delivery of Large-Sized Data Object based on Delay Constrained Request over Scheduled Transportation Vehicles

Shigeru Kashihara*

Graduate School of Information Science, Nara Institute of Science and Technology. Email:shigeru@is.naist.jp

Abstract

The paper proposes two transmission scheduling methods to deliver large-sized data object based on delay constrained request over scheduled transportation vehicles. Scheduled transportation vehicles like buses and rails have the potential to provide data communication for delay-tolerant large data object based on delay-/disruption-tolerant network (DTN) concept as a part of the Internet. However, since the operation of vehicles follows the timetable, even if the vehicles carry users' data objects, they may not always deliver the objects to their destination within their requested delay. As one of the approaches, to satisfy the delivery with delay requests, our previous paper presented the design of data offloading to public transportation vehicles. In the model, if a data port accepts data objects from users in a first-come, first-served (FCFS) schedule, the system cannot guarantee to deliver data objects within requested delay. In this paper, as the first step under the design, we propose two transmission scheduling methods for uploading data at a data port. Simulation evaluations demonstrate that the proposed methods significantly improve the successful data delivery compared with FCFS.

Keywords: Data delivery; delay-tolerant data objects; DTN; large-sized data objects; scheduled transportation vehicles; transmission scheduling method

1. Introduction

With the widespread use of advanced functionality and high-performance smartphones, a smartphone has become the primary communication device for an end user. The user deploys diverse applications on the smartphone, producing and storing text, photos, and videos. In particular, the multimedia file size is sharply increasing as users demand higher resolution photos and videos ever. For instance, a photo of more than 5 MB is not uncommon, while a 10-minute video clip requires about 2 GB. Also, other applications for backup, business use, and so on, will create more large data. Current smartphones have relatively large storage, inviting users to create a more significant amount of data objects. At the same time, personal, business, and backup applications require that the delivery of these data objects be completed within defined time constraints to share, upload, and download these data objects in accordance to the circumstance.

Presently, a smartphone can send and receive relatively large data objects of a few photos through public Wi-Fi and 4G networks. However, high transmission of data files like many high-resolution photos, videos, and backup data is not feasible on public Wi-Fi or LTE, because of communication quality changes due to a density of users or the signal environment. Also, as the amount of data per month depends on the data plan, a user like a tourist wants to save money as much as possible by reducing the data communication. On the other, public Wi-Fi (e.g., coffee shops, malls, airports) have limited coverage and cannot support ubiquitous user mobility. Compared with Wi-Fi, 4G provides broader area access and may be the best technology to support extended user mobility. However, the communication speed of a 4G is still slow and is prone to congestion. For instance, Cisco reports that the average global mobile connection speed in 2016 was 6.8 Mbps [1]. That means that it would take about 39 minutes to complete the transmission of a 2 GB file in an ideal environment where communication quality does not change. Furthermore, a user who relies on a smartphone as a primary communication device may want to conserve its battery power for future use. A long session would also severely degrade the other cell users' throughput. As a result, the smartphones will consume more battery for the long communication. These limitations of the current (and future) wireless infrastructure motivate our research to find

*Corresponding author. Tel.: +81-743072-5213
8916-5 Takayama, Ikoma, Nara, Japan, 630-0192

effective alternatives for the delivery of large-sized data objects within a user's desired delay.

As illustrated in Fig. 1, to deliver a large data object within a user's desired delay, our previous paper considered a data delivery approach over a regularly scheduled vehicle [2]. Examples include buses and rails, among others. References [3, 4] reported that a similar approach that relies on the use of "data mule" carriers to extend the Internet connectivity in a rural area. We believe that the "data mule" concept can be leveraged to support delay-constrained transport of large-sized data objects via public transportation facilities.

To deliver large data objects, we envisage the system, depicted in Fig. 1, where a user drops a large data object off into a data port at a bus stop. Current trends indicate that smartphones will eventually equip with a proximity wireless transfer technology such as Transfer Jet [5] and WiGig [6]. It is capable of forwarding large data objects in a shorter time. We assume that a user would transfer their large object to the data port via one of the above technologies. Then, a bus picks the data up from the data port and carries the forwarded data to a bus terminal. At the bus terminal, the vehicle offloads the data to the Internet through a high-speed connection like an optical fiber cable.

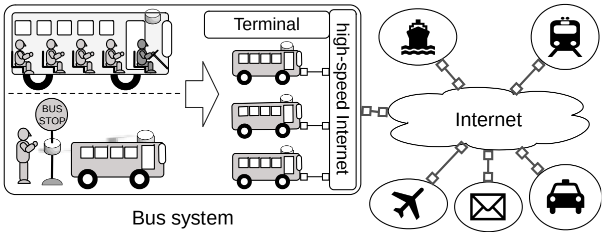


Figure 1. Data delivery over public transportation systems

To meet the user's specified deadline, we can estimate the time to deliver a large data object to its destination by utilizing the transportation service's timetable. However, since a vehicle's pick-up time follows the transportation system schedule, all data in the data port may not always be possible to be picked up within the stoppage time at a station. In this paper, we propose delay constrained large data object delivery using a scheduled public transportation system and introduce transmission scheduling algorithms for the data port. Note that as the first step for the study, this paper here focuses on uploading data flow because downloading flow requires a more complicated approach based on user's location information after solving the problems for uploading data. We first describe a communication approach for a bus system as an instance of the scheduled transportation system. We then show that the transportation system cannot deliver the required performance if the data port has no appropriate transmission-scheduling algorithm; that is if it operates in a first-come, first-served (FCFS) schedule. We propose two effective transmission scheduling methods and demonstrate through simulation experiments that the methods meet the desired constraints.

2. Communication approach with scheduled transportation vehicles

The section starts by describing how a scheduled transportation system can be exploited for delay constrained large data object delivery. Section 2.1 first describes the communication approach in the system, and Section 2.2 then analyzes the system performance considering data object size and user delay constraints as system parameters.

2.1. Communication approach and ideal transmission performance

Throughout this paper, we use a bus system as an example, but this concept is readily applicable to other transportation systems. Fig. 2 shows the communication flow in the system. We assume that a smartphone has an interface for proximity wireless transfer technology such as Transfer Jet. Likewise, a data port and bus employ Wi-Fi (802.11n) and Transfer Jet interfaces. Transfer Jet is used for communication between a smartphone and facilities (a data port or a bus), while communication between a data port and a bus employs 802.11n. We assume further that both the bus and the data port have huge but finite data storage capacity.

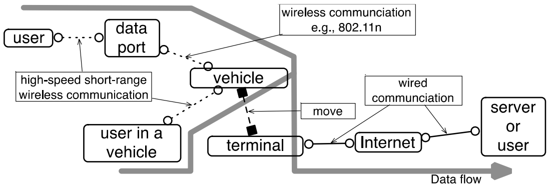


Figure 2. Communication flows over scheduled transportation vehicles

In our assumption, if an object size is relatively small or if the sending time via the current wireless network is tolerable for a user, a user may opt to send the data object through public Wi-Fi or 4G networks. Otherwise, a user tries to send the data object via the bus system. Besides, a user that boards a bus may upload the data object via Transfer Jet from his or her seat. As shown in Fig. 3, with the Transfer Jet's effective maximum throughput of 375 Mb/s [5], a user can send a large data object in a relatively short time. Also, since the connection distance of Transfer Jet is a few centimeters, the communication quality is expected to be very reliable against any multi-path fading and shadowing. On the other hand, if a user does not board a bus, a user drops a data object off at a nearby bus stop into a data port by using Transfer Jet. A bus picks up data from the data port via 802.11n. Once the data object is in the bus storage system, the bus carries the data to the bus terminal. After reaching the bus terminal, the bus then offloads all data objects to the Internet through a high-speed connection such as an optical fiber cable.

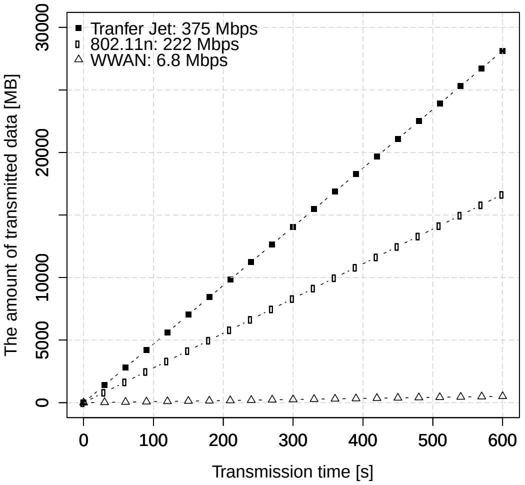


Figure 3. The amount of transmitted data over transmission time

Our primary emphasis is measuring the transmission performance for data objects in a data port at one bus stop since the overall transmission performance practically depends on data transfers from a data port to the bus. Although data objects uploaded by passengers from within the bus will eventually be delivered to a bus terminal, data objects that are picked up from the data port are restricted by the bus stoppage time at a bus station, according to a transit timetable. In other words, the transmission between a data port and a bus is the bottleneck in this system.

We first consider the ideal transmission performance. The effective transmission speed of Transfer Jet and 802.11n are here assumed to be 375 Mb/s [5] and 222 Mb/s [7], respectively. The amount of data from passengers per day is defined as alpha MB, while the amount of data from a data port per day is calculated as follows:

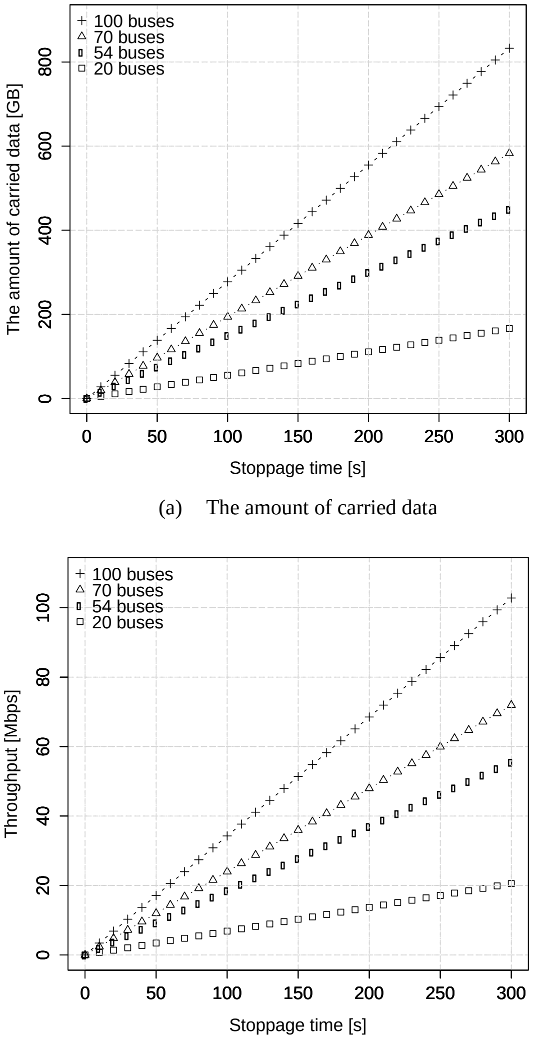
The amount of data from a data port at one bus stop in a day (MB) = (*Effective transmission throughput from a data port to a bus* (Mb/s) * *Stoppage time at a bus stop* (s) * *the number of buses in a day*) / 8.

The throughput during service operation is calculated as follows:

Throughput during service operation (Mb/s) = ((*The amount of data from a data port in a day* (MB) + *alpha* (MB)) * 8 / *Operation time* (s).

Thus, the transmission performance highly depends on stoppage time at a bus stop and the number of buses in a day.

Fig. 4 shows the ideal transmission performance for data objects in a data port. Fig. 4(a) shows the amount of carried data in a data port in a day, while Fig. 4(b) shows the throughput during service operation from 6:00 to 0:00. Each bus arrives at every 20 minutes, i.e., 54 buses per day and the stoppage time is 120 seconds. Thus, the ideal amount of carried data from the data port per day is approximately 179,820 MB/day with an effective throughput of 22.2 Mb/s. Since the results depend only on data objects in a data port at one bus stop, the system has higher performance if the data objects from passengers are included. Moreover, if a bus system has ten bus stops, the effective total throughput of the whole bus system is more than 222 Mb/s. Therefore, a scheduled transportation system has the potential for transporting large data objects.



(a) The amount of carried data

(b) Throughput

Figure 4. Ideal data transmission performance at one bus stop

2.2. Transmission performance under constraints of user's desired delay

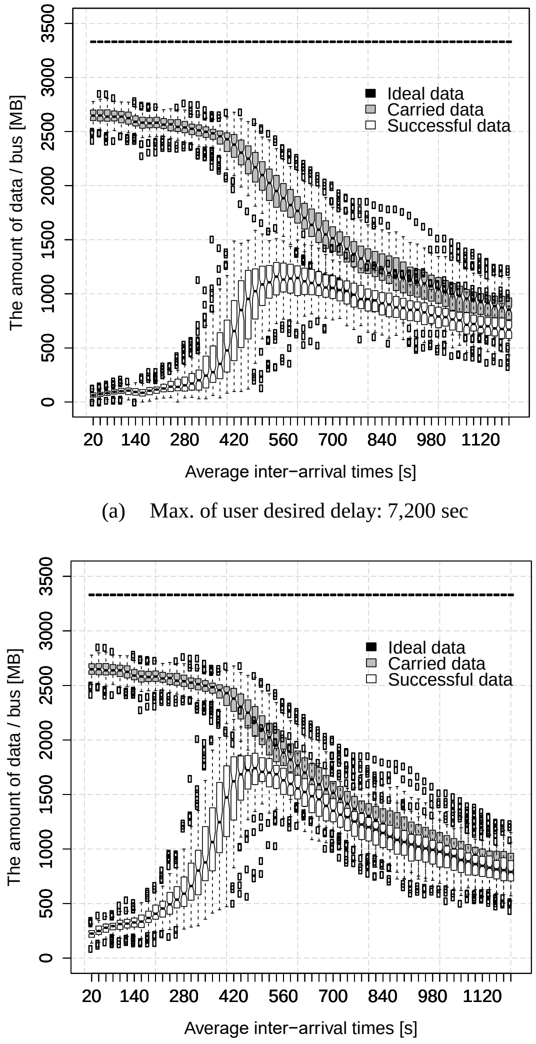
This section provides simulation results for the above system under the user's delay constraint. Note that we here focus only on the transmission performance for data objects placed at a data port.

Table 1 shows the simulation parameters under an assumption. We assume that user arrival at a bus stop follow a Poisson distribution, while data size and user desired delay for each data object are uniformly distributed. If a data object size is small, a user may choose to send the data via Wi-Fi or 4G. Otherwise, a user would be better off sending the large data object via the bus system. When a user arrives at a data port, a user is assumed to leave without sending the data object if another user is forwarding data at the data port. Otherwise, the user tries to forward the data object to the data port via Transfer Jet.

Table 1. Simulation Parameters

Parameters	Values
Average inter-arrival time (Poisson distribution)	From 20 to 1,200 sec
Data size (Uniform distribution)	Min: 1 MB, Max: 2,000 MB
User’s requested delay (Uniform distribution)	Min: 3,600 sec Max: 7,200 sec,14,400 sec
Throughput of Transfer Jet (From user to data port)	375 Mbps
Throughput of 802.11n (From data port to bus)	222 Mbps
Operation time	18 hours
Bus arrival interval	1,200 sec
Number of buses per day	54 buses
Stoppage time	120 sec

In our simulation experiments, the average inter-arrival time, the data size, and the user's desired delay are variable parameters. The average inter-arrival time varies between 20 and 1,200 seconds. The data size, uniformly distributed from 1 to 2,000 MB, represents the different sized data created by each user. The user's desired delay is assumed to be uniformly distributed between a minimum value (3,600 seconds) and a maximum value (7,200 or 14,400 seconds). The operation time for the bus system is from 6:00 a.m. to 12:00 a.m. with buses arriving every 20 minutes, for a total of 54 buses per day. The simulation analyzes only one bus stop since our focus is on the transmission performance of data objects uploaded from a data port. At a bus stop, the bus picks up data from the data port via 802.11n during stoppage time of two minutes. Without accounting for desired delays, the data port forwards data objects to the bus on the FCFS basis. Besides, data objects are not segmented. After leaving the bus stop, the bus transits to the bus terminal in about 50 minutes. Upon reaching the bus terminal, the bus assumes to offload all carried data objects to the Internet within 10 minutes.



(a) Max. of user desired delay: 7,200 sec

(b) Max of user desired delay: 14,400 sec

Figure 5. The amount of ideal data, carried data, and successful data per bus over average inter-arrival times under consideration of user’s desired delay

Figure 5 shows the amounts of ideal data, carried data, and successful data per bus under consideration of a user's desired delay. The ideal data object volume is defined as follows. The amounts of data objects picked-up within a bus stoppage time is its maximum allowance, and all the picked-up data objects are entirely delivered within the user's desired delays. Then, the amount of carried data and successful data are defined as the amount of data carried by bus and the amount of data delivered within user's desired delay for the carried data, respectively.

Figure 5(a) displays transmission performance with a maximum value of 7,200 sec for a user's desired delay, while Fig. 5(b) displays results for 14,400 sec. Under the above assumptions, the amount of ideal data per bus is 3,330 MB. Since a data object is kept intact and not divided, if forwarding the head-of-line object in the data port does not complete within the stoppage time, another attempt will be required with the next bus. In other words, even if the bus has a small data file, which can be picked up within the stoppage time, behind the head data in a data port, the bus cannot pick up other objects because of the inherent head-of-line blocking in FCFS. Furthermore, an object that is carried by bus under the FCFS discipline above is not guaranteed to be delivered within its desired delay. The rate of successful data per carried data may increase under a lower average inter-arrival time, yet, there are no assurances that all the data objects will be delivered within a user's desired delay. This is because a bus may carry data objects, which is going to be expired or is already expired, by FCFS under no consideration of a user's desired delay. From the graphs in Fig. 5, we also observe that the smaller maximum user's desired delay degrades the amount of successful data. Therefore, a data transmission scheduling algorithm in a data port is necessary to improve the transmission performance of a public transportation system.

3. Transmission scheduling method for data port

This section proposes two scheduling methods for the transmission from a data port to a bus. The objective is to improve the amount of data delivered within the user desired delay. We first discuss the design of a data port and then elaborate on each transmission scheduling method.

3.1. Design of data port

A data port estimates delivery time to show a user whether an object will be delivered within the user's desired delay. As illustrated in Fig. 6, a data port displays the estimated delivery time of the data object based on the object size from the smartphone and the current storage contents of the data port. As described in Section 2.1, the transmission between a data port and a bus is the bottleneck in this system. So, to improve the transmission performance, the data port should accept only data objects that will be delivered within the user delay threshold. Also, the user decides whether or not to send the data object from the estimated delivery time displayed. When a data-port cannot accept a data object, it suggests the current shortest delivery time for a user and the user decides whether or not send the data object.

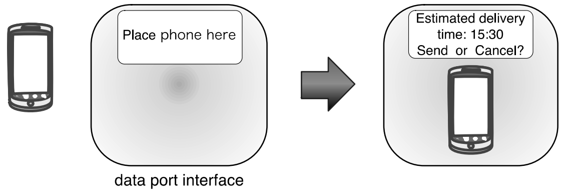
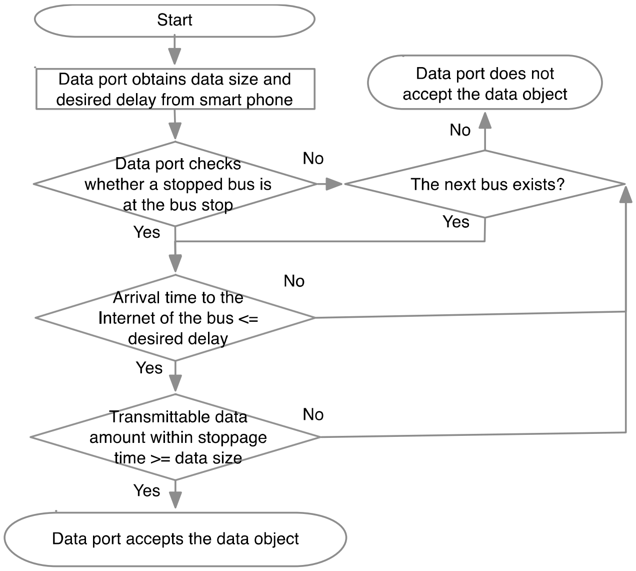


Figure 6. Example of interactive data port interface

To improve the transmission performance, a data port needs to schedule the transfer of data objects to buses according to the objects data sizes and users desired delay. So, the following sections introduce two scheduling methods.

3.2. Forward-packing transmission scheduling method (FPTS)

We first introduce a forward-packing transmission scheduling method (FPTS). Fig. 7 shows the flowchart of FPTS. When a smartphone is placed on the data port interface, the data port obtains the data size and the user's desired delay of the data object from the smartphone.


Start
Data port obtains data size and desired delay from smart phone
Data port does not accept the data object
No
Data port checks whether a stopped bus is at the bus stop
No
The next bus exists?
Yes
Yes
No
Arrival time to the Internet of the bus <= desired delay
Yes
No
Transmittable data amount within stoppage time >= data size
Yes
Data port accepts the data object

Bus #	transmittable data amount within stoppage time	arrival time to the Internet
1		
2		
3		
:		
the last bus		

Figure 7. Forward-packing transmission scheduling method (FPTS)

The data port then examines a bus that can deliver the data object within the delay. First, it checks the status of buses. If a bus is currently at the bus stop, the data port estimates the possibility of delivery within the delay and the transmission time to transfer the data object to a bus. If both are acceptable, the data port receives the data object from the smartphone.

If no bus is present or the data object cannot be delivered within the delay, the data port checks the scheduled time for the following buses in turn. In the process, the data port compares the delivery time to the Internet for the bus with the user's desired delay. If the delivery time to the Internet is larger than the user's desired delay, the data port checks the following buses iteratively, provided that the scheduled time does not exceed the user's desired delay. On the other hand, if the delivery time to the Internet is smaller than the user desired delay, the data port then checks whether the bus can pick up the data object within the stoppage time, i.e., transmittable data amount within stoppage time. If the bus has enough time, the data object is scheduled for that bus.

Otherwise, the data port checks iteratively whether the following buses will have enough time to deliver the data object within the user's desired delay. Therefore, FPTS packs the data object with a bus that can deliver it within the user's delay constraint, according to the bus arrival order.

3.3. Backward-packing transmission scheduling method (BPTS)

Similar to FPTS, the backward-packing transmission scheduling method (BPTS) also works according to data size and desired delay, but schedules data objects for a bus towards the upper bound of the user's desired delay. That is, BPTS examines a bus in turn from the last bus that can deliver the data object within the delay.

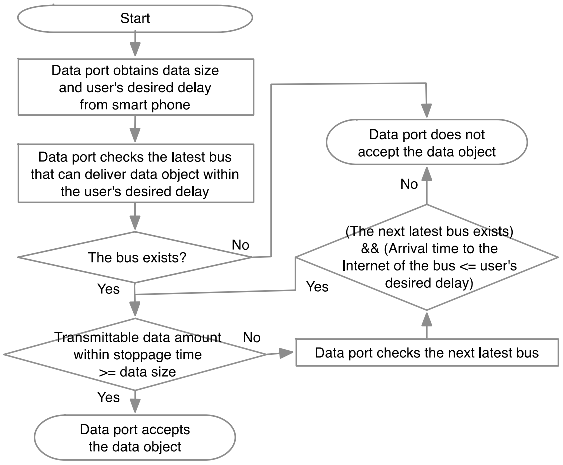

Start
Data port obtains data size and user's desired delay from smart phone
Data port checks the latest bus that can deliver data object within the user's desired delay
The bus exists?
No
Yes
Data port does not accept the data object
No
(The next latest bus exists) && (Arrival time to the Internet of the bus <= user's desired delay)
Yes
Transmittable data amount within stoppage time >= data size
No
Data port checks the next latest bus
Yes
Data port accepts the data object

Figure 8. Backward-packing transmission scheduling method (BPTS)

BPTS works as depicted in Fig. 8. After obtaining the data size and desired delay from a smartphone, a data port checks the last bus that can deliver the data object within the desired delay. If a bus does not exist, the data port does not accept the data object. If a bus exists, the data port then checks whether the bus has enough time to pick up the data object (i.e., the transmittable data amount within stoppage time). If the bus has enough time, the data port accepts the data object. Otherwise, the data port considers the previous bus iteratively and compares the delivery time to the Internet for the bus with the user's desired delay. If conditions are satisfied, the data port checks if there is enough time to pick up on the bus again. Otherwise, the data port does not accept the data object.

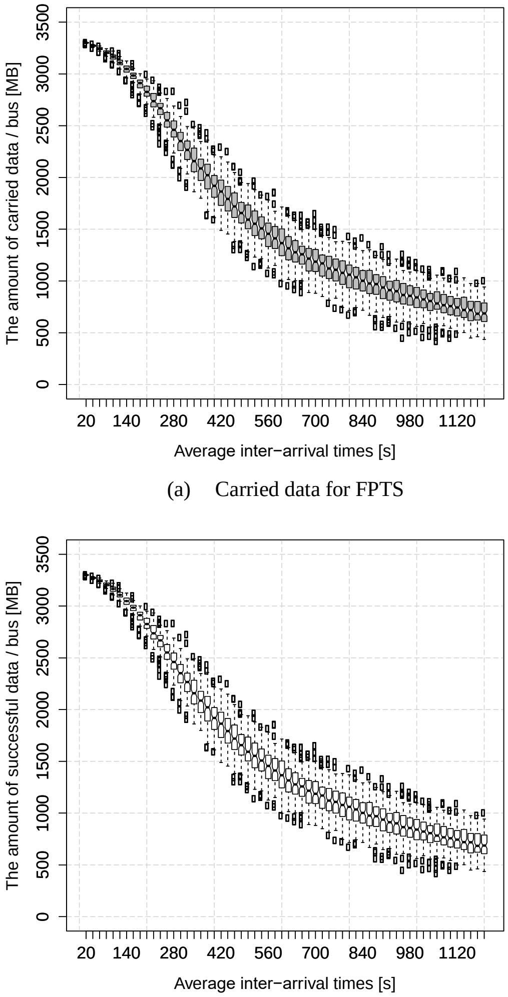
4. Performance evaluation

This section describes the performance evaluation and the discussion of the proposed method. Section 4.1 shows the transmission performance of the two scheduling methods compared with that of FCFS presented in Section 2. Section 4.2 then discusses remaining issues toward applying our methods to a real environment.

4.1. Simulation results

The section evaluates the transmission performance of FCFS, FPTS, and BPTS via simulation experiments. As previously mentioned in Section 2.2, our emphasis is on the performance of single data port, using the assumptions and parameters listed in Section 2.2. Note that the maximum user's desired delay is set to 7,200 sec.

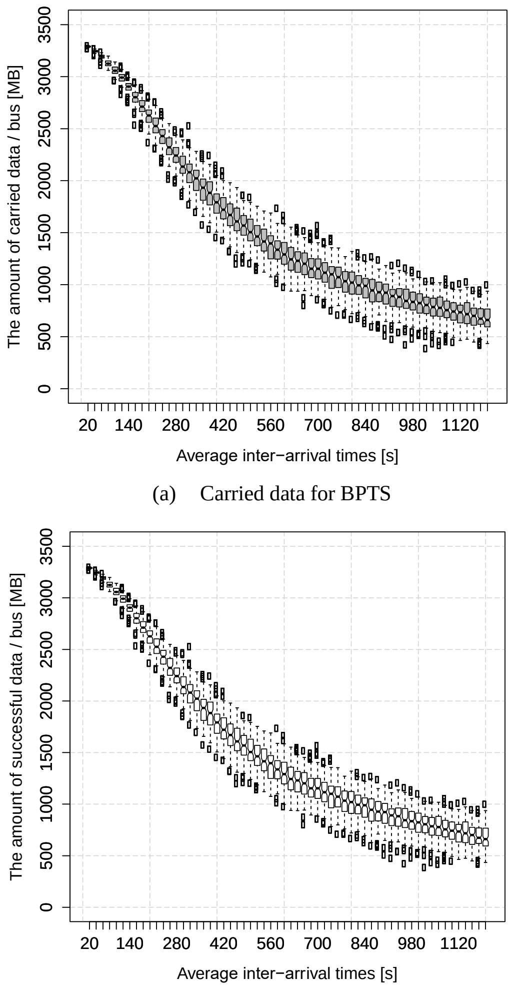
Figures 9 and 10 plot the amount of carried and successful data per bus for FPTS and BPTS, respectively. Figs. 9(a) and 10(a) show the amount of carried data by a bus, while Figs. 9 (b) and 10 (b) show the amount of successful data in the carried data. From the graphs, we observe that a bus completely delivers data objects within the user's desired delay in both FPTS and BPTS. However, in comparison with FCFS (see Fig. 5 (b)), the amount of carried data for FPTS and BPTS decreases because FPTS and BPTS selectively accept data objects according to data size and user's desired delay.



(a) Carried data for FPTS

(b) Successful data for FPTS

Figure 9. The amount of carried and successful data per bus for FPTS



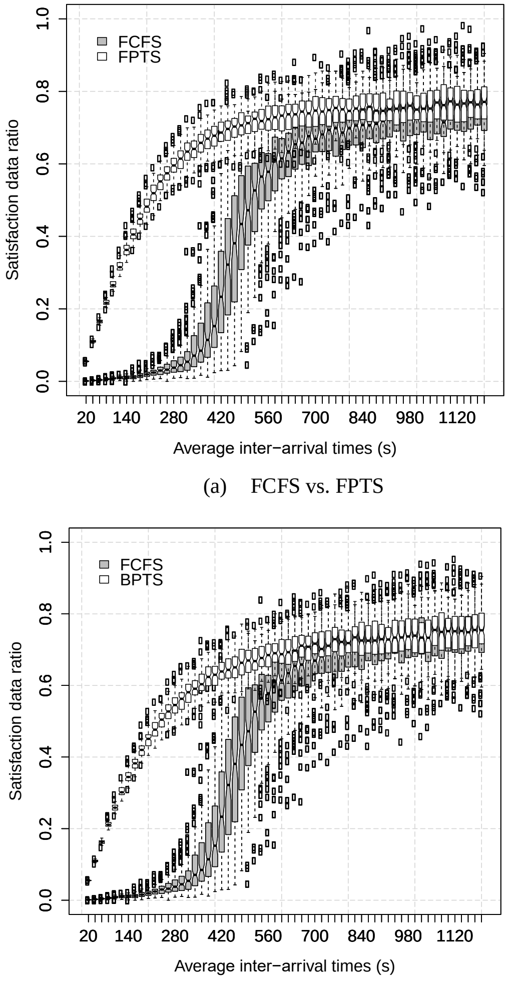
(a) Carried data for BPTS

(b) Successful data for BPTS

Figure 10. The amount of carried and successful data per bus for BPTS

Figure 11 shows the satisfaction data ratio for the three scheduling algorithms. The satisfaction data ratio is defined as the fraction of successful data objects for all large data objects from users who do not board a bus. All large data objects include two kinds of data objects: one is data objects stored in the data port, and the other is data objects not stored in the data port. Figure 11(a) plots the satisfaction data ratio of FCFS and FPTS, whereas Fig. 11(b) plots that of FCFS and BPTS. Compared with FCFS, both FPTS and BPTS sharply improve the satisfaction data ratio, especially in high average inter-arrival times. However, some of the bus data transport capacity may remain unused because a data object remains un-fragmented. On the other hand, in the case that the average inter-arrival times are long, all algorithms have nearly identical results, and none of them fills the transmission capacity because of the scarcity of data objects that transmit to a bus.

We also note that the satisfaction data ratio of FPTS performs slightly better than BPTS because BPTS schedules a data object from the last possible bus within a user's desired delay. Therefore, BPTS may result in capacity underutilization since earlier buses may have enough space. Besides, since BPTS operates based on the last bus service within the user's desired delay, there is a possibility that the satisfaction data ratio of BPTS may be damaged when an unprecedented bus delay occurs.



(a) FCFS vs. FPTS

(b) FCFS vs. BPTS

Figure 11. The satisfaction data ratio (the amount of successful data / the amount of all data)

4.2. Discussion

We also describe some remaining issues regarding delay constraints of large-sized data object delivery on a scheduled transportation system. First, the above simulation results show that our data port-to-bus transmission schedule sharply improves the performance of the scheduled bus system under no random bus delays beyond the bus system's ideal schedule. To analyze real-life transportation systems, we need to include realistic and random bus delays as the next step. Also, in regards to average inter-arrival time, data size and desired delay, more rigorous analyses are needed to identify these distributions and their parameters.

Second, to more improve the utilization of a bus transport capacity, a data object should be subdivided into smaller segments, with the added capability of tracking the divided data and reconstructing the original data object at a bus terminal or a user destination site. Moreover, different fragments may be placed on different buses, and; store and forwarding across buses could be considered, to meet different delay requirements.

Because of its simplicity, our approach differs from traditional Delay-/Disruption-Tolerant Networks (DTN) based on the bundle paradigm [8]. If data fragmentation (over multiple buses and routes) and bus switching were introduced, new communication protocols and policies would be required to guarantee end-to-end communication. In essence, as a difference from traditional DTN carries, to improve the transmission performance, our approach shifts to processing at the data port site close to users like edge computing.

5. Related Work

The concept of “data mule” has been proposed to extend the Internet connectivity in a rural area [4], [9]. At first, such a data mule, however, focused only on carrying data without considering desired delays. After that, researchers have attempted to apply the concept to public transportation systems in a city, to alleviate the congestion of mobile networks.

In [10], the authors explored how messages could be carried over aerial carriers and showed performance measures of delivery probability, delay, and hop counts. Their results provide a starting point for future research. Also, [3], [11–18] study data delivery over a bus network, similar to our research, but focuses on routing methods. As far as we know, there is not any known research on transmission scheduling method on a data port to meet user-specified delay constraints for large data object delivery over public transportations systems.

References [19, 20] show experimental results for data offloading via Wi-Fi. In reference [20], authors measured the throughput of 3G network and Wi-Fi network between Carnegie Mellon University and business district through walking and driving. The results showed that as for instantaneous throughput, Wi-Fi network is faster than the 3G network, while the average downloading throughput of the 3G network is faster than that of Wi-Fi network. On the other hand, in uploading throughput, Wi-Fi network is superior to the 3G network for instantaneous and average throughput because the uploading throughput of the 3G network is slow. In reference [19], authors measured the communication throughput of the 3G network and Wi-Fi network through driving in three cities, Amherst, Seattle, San Francisco. The measurement results showed that 3G network provides higher throughput in downloading and uploading than Wi-Fi network. From their results, we can see that users can obtain higher throughput when users use 3G network during movement, or when users use Wi-Fi network during staying.

To efficiently improve data offloading, references [19, 21] have studied data offloading for delay tolerant data. Reference [19] proposed a data offloading method that a smart mobile device sends/receives data within user’s tolerant delay by using Wi-Fi as much as possible. The method makes a threshold for delay tolerance, and considers data size and expected throughput in Wi-Fi area where a user will visit. If a user cannot offload all data within the desired delay through Wi-Fi networks, a user can use the 3G network to complete to send/receive data within the delay. In reference [21], authors measured Wi-Fi connectivity for about 100 iPhone users in Seoul, Korea. Based on the results, they then evaluated the efficiency of data offloading to Wi-Fi networks when user offload delay acceptable data to Wi-Fi networks, through simulation experiments. From the simulation results, in the case that 6 hours delay is acceptable, the efficiency of the data offloading rose up to 87.5 %. The authors, however, note that the results depend on the communicable area of Wi-Fi network and user’s mobility. Also, reference [22] proposed a data offloading method that a smart mobile device offloads data to another device by a Store-Carry-Forward routing, like DTN. As described above, data offloading that considers acceptable delay for users and applications is becoming a new trend.

Some studies [23, 24] proposed data offloading methods that consider power saving on smart mobile devices. In reference [24], authors investigated how the scanning of Wi-Fi APs impact to smart mobile device’s battery. They describe that the data offloading method that considers the scanning intervals is necessary. Reference [23] also showed that the scanning AP impacts to battery on a smart mobile device. From the results, we may save battery when the scanning interval is prolonged. However, since it may be late to find next Wi-Fi network, the connection period during movement may be reduced. Therefore, to efficiently offload data to Wi-Fi networks, AP scanning interval and battery consumption should be considered. Also, to efficiently make a connection to Wi-Fi networks, reference [25] proposed a connectivity forecast. The method forecasts Wi-Fi networks that a user will visit based on the information of user’s daily movement. Although various approaches have been studied for data offloading to Wi-Fi networks so far, it is essential to building a lot of Wi-Fi networks to improve the data offloading. Besides, as increasing APs, various issues such as deployment and radio interference are arising.

As described above, to utilize scheduled transportation vehicles as a part of the communication network in order to alleviate the congestion of the mobile network, many approaches have been proposed. Most of them focus on the routing method based on DTN. On the other hand, in this paper, to meet the users’ delay requirement for large data objects, we proposed the transmission scheduling algorithms that can deliver them as much as possible within delay requirements. As the next step, to realize the communication network over scheduled transportation vehicles, these existing work is expected to be combined organically.

6. Conclusion

This paper proposes a data delivery scheme that uses a public transportation system to deliver delay constrained large-sized data objects. We demonstrate that scheduled transportation vehicles have the potential to become an effective means for such data delivery service. We prove that, if the input data port has no transmission scheduling capability, in the case of high user demands, most of the data objects miss their deadlines. To efficiently exploit a public transportation system as a part of the delivery network, we proposed two transmission scheduling algorithms and have demonstrated via simulation that the fraction of objects delivered within a user's defined deadline significantly increases.

As mentioned in Section 4, our future work includes more rigorous analytic modeling and a system prototype implementation to study the feasibility of the public transportation-based delivery further. Although this paper focuses on measuring data uploads, download traffic may be of interest as well. For example, a user might request to deliver a data object to a data port at a location of interest. To accommodate download capabilities, we need to extend our investigation in this paper, to design and analyze the performance of a system supporting download traffic as well as upload traffic. A further opportunity for performance enhancement (with increasing complexity) is the introduction of message fragmentation in smartphones, a data port, and buses.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper." 2017.

[2] S. Kashihara, M. Y. Sanadidi, and M. Gerla, "Mobile, Personal Data Offloading to Public Transport Vehicles," in *The Sixth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2012)*, 2012, pp. 142–147.

[3] I. Komnios and E. Kalogeiton, "A DTN-based architecture for public transport networks," *Ann. Telecommun. - Ann. des télécommunications*, vol. 70, no. 11, pp. 523–542, 2015.

[4] M. Demmer and K. Fall, "DTLSR: Delay Tolerant Routing for Developing Regions," in *Proceedings of the 2007 Workshop on Networked Systems for Developing Regions*, 2007, p. 5:1--5:6.

[5] "Transfer Jet." [Online]. Available: https://www.transferjet.org/index.html.

[6] "WiGig." [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-wigig.

[7] Aerohive Networks Inc., "The Network Impact of 802.11n." Aerohive Networks Inc., 2010.

[8] L. Torgerson *et al.*, "Delay-Tolerant Networking Architecture," no. 4838. RFC Editor, 2007.

[9] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: rethinking connectivity in developing nations," *Computer (Long. Beach. Calif).*, vol. 37, no. 1, pp. 78–83, 2004.

[10] A. Keränen and J. Ott, "DTN over Aerial Carriers," in *Proceedings of the 4th ACM Workshop on Challenged Networks*, 2009, pp. 67–76.

[11] F. R. Segundo, E. Silveira e Silva, and J.-M. Farines, "A DTN routing strategy based on neural networks for urban bus transportation system," *J. Netw. Comput. Appl.*, vol. 64, pp. 216–228, Apr. 2016.

[12] T. Pögel, "Optimized DTN-Routing for Urban Public Transport Systems," in *17th GI/ITG Conference on Communication in Distributed Systems (KiVS 2011)*, 2011, vol. 17, pp. 227–232.

[13] S. Ahmed and S. S. Kanhere, "Cluster-based Forwarding in Delay Tolerant Public Transport Networks," in *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, 2007, pp. 625–634.

[14] S. Jiang, J. Ferreira, and M. C. González, "Clustering daily patterns of human activities in the city," *Data Min. Knowl. Discov.*, vol. 25, no. 3, pp. 478–510, 2012.

[15] S. Gaito, D. Maggiorini, C. Quadri, and G. P. Rossi, "On the Impact of a Road-Side Infrastructure for a DTN Deployed on a Public Transportation System," in *NETWORKING 2012*, 2012, pp. 265–276.

[16] H. Yao, D. Zeng, H. Huang, S. Guo, A. Barnawi, and I. Stojmenovic, "Opportunistic Offloading of Deadline-Constrained Bulk Cellular Traffic in Vehicular DTNs," *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3515–3527, 2015.

[17] S. Gaito, D. Maggiorini, G. P. Rossi, and A. Sala, "Bus switched networks: An ad hoc mobile platform enabling urban-wide communications," *Ad Hoc Networks*, vol. 10, no. 6, pp. 931–945, Aug. 2012.

[18] M. Doering, T. Pögel, and L. Wolf, "DTN routing in urban public transport systems," *5th ACM Work. Challenged Networks, CHANTS '10*, pp. 55–62, 2010.

[19] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting Mobile 3G Using WiFi," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, 2010, pp. 209–222.

[20] R. Gass and C. Diot, "An Experimental Performance Comparison of 3G and Wi-Fi BT - Passive and Active Measurement," 2010, pp. 71–80.

[21] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile Data Offloading: How Much Can WiFi Deliver?," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 536–550, 2013.

[22] B. Han, P. Hui, V. S. A. Kumar, M. V Marathe, G. Pei, and A. Srinivasan, "Cellular Traffic Offloading Through Opportunistic Communications: A Case Study," in *Proceedings of the 5th ACM Workshop on Challenged Networks*, 2010, pp. 31–38.

[23] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay Tradeoffs in Smartphone Applications," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, 2010, pp. 255–270.

[24] B. Han, P. Hui, and A. Srinivasan, "Mobile Data Offloading in Metropolitan Area Networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 14, no. 4, pp. 28–30, 2010.

[25] A. J. Nicholson and B. D. Noble, "BreadCrumbs: Forecasting Mobile Connectivity," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, 2008, pp. 46–57.