PAPER *Special Section on New Generation Mobile and Sensor Networking and Future Networks*

Virtual Single Network Path by Integrating Multiple and Heterogeneous Challenged Networks

Akira NAGATA[†*a)], **Shinya YAMAMURA**[†**], ***and*** **Masato TSURU**[††], ***Members***

SUMMARY Motivated by the question of how to quickly transfer large files if multiple and heterogeneous networks are available but each has insufficient performance for a requested task, we propose a data transfer framework for integrating multiple and heterogeneous challenged access networks, in which long delays, heavy packet losses, and frequent disconnections are observed. An important feature of this framework is to transmit the control information separately from the transmission of data information, where they are flexibly transferred on different types of communication media (network paths) in different ways, and to provide a virtual single network path between the two nodes. We describe the design of the mechanisms of this framework such as the retransmission, the rate adjustment of each data flow, and the data-flow setup control. We validate a prototype implementation through two different experiments using terrestrial networks and a satellite communication system.

key words: *DTN, challenged network, reliable transfer, multipath, heterogeneous network, wireless network*

1. Introduction

There will be an increasing demand for the transfer of large data sets anytime, anyplace, and even in a mobile environment by using high speed wireless access networks (i.e., ubiquitous networking). Wireless broadband services have been developed to support high-speed data transfer. For example, cellular 3.5G and 802.11 wireless LAN hotspots are now widely deployed in many areas. It is expected that LTE or WiMAX will also become widely available in the near future. High-speed satellite communications (which we used in the experiment in this study) have been implemented. However, because of high costs, it is difficult to provide wireless broadband to all areas with sufficiently high performance. Some areas experience unstable wireless communications and uncertain availability. Disconnected communications and serious degradation of communication quality, for instance, can occur at blind spots or at the edges of coverage areas where hosts have little contact with the access point or base station. This is similar to the case of mobile hosts that are frequently in motion.

In those cases, the use of wireless broadband transmission can become "challenged," and end-to-end data transfer by conventional TCP can be inefficient. In the conventional Internet, which assumes TCP, both data and control information are either conveyed in the same packet or, if they are in different packets, by the same communication medium (network path). When a network path is transiently disconnected, not only data but also control information are blocked, which may result in significant degradation of reliable data transfer performance. Exchange of essential information before application data transmission, such as address resolution and authentication, may also fail due to unstable connections.

Note that a wireless channel that covers a larger area generally has a lower transmission rate, and thus such channels are not suitable for large data transfer due to the narrow bandwidth of the channel. However, they can be useful for the stable exchange of small-sized data such as the control information required for file transfer.

In our study, we construct a framework for quickly transferring large data sets in wireless challenged environments, based on the integration of unstable networks with a high data rate and stable networks with a low data rate into a virtual single network path. We have proposed an architecture that separates control information flow from data flow for end-to-end transport in a challenged environment [1]. The control information flow is not necessarily on the same communication medium as the data flow, but on a more sustainable medium covering a wider area, which could improve communication efficiency in challenged environments. Steady and therefore timely exchange of control information via separate links would make data transfer more efficient, even if the data flow was dynamically disrupted. For example, mobile hosts in rural areas or in vehicles moving at high speed that are using multiple and different types of wireless channels can benefit from this integration. We have validated our prototype implementation through two different field experiments. The first experiment [1] used a combination of a low-speed but widely-covered satellite communication link and terrestrial communication links (3G and Wi-Fi). The second experiment [2] used a pair of a high-speed satellite communication link and a terrestrial cellular link (3G).

In the present paper, we describe the design of the mechanisms including the retransmission, transmission rate adjustment of each data flow, and data flow setup control, and examine a prototype implementation through an exper-

Manuscript received September 30, 2010.
Manuscript revised February 4, 2011.

[†] The authors are with National Institute of Information and Communications Technology (NICT), Tokyo, 100-0004 Japan.

[††] The author is with Network Design Research Center, Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.

[*] Presently, with Network Application Engineering Laboratories Ltd.

[**] Presently, with Fujitsu Kyushu Network Technologies Limited.

a) E-mail: nagata@nalab.jp

DOI: 10.1587/transcom.E94.B.1546

iment using an emulated network. Then we present evaluation results of our prototype system through two field experiments using terrestrial networks and a satellite communication system, which were briefly reported in our previous papers. The experimental results demonstrate the effectiveness of our prototype implementation.

The rest of this paper is organized as follows. In Sect. 2, a brief overview of the proposed system for integrating multiple wireless access networks and related research are described. A prototype of the proposed system is described in Sect. 3. In Sect. 4, experiments involving the prototype system are presented. Finally, the key conclusions of the present study are presented in Sect. 5.

2. Proposed Approach and Related Research

We briefly review our concept of file transfers over multiple network paths (communication media) and the related work in literature. Suppose long delays, leaks (heavy packet losses) and frequent disconnection are observed in these paths; hence, this is a challenged environment.

We consider the situation that multiple and heterogeneous networks are simultaneously available. At least one is a stable network, even if its speed is too slow for a given file to be transferred within a reasonable time. Here, "*stable*" means continuously available for a longer time and in a wider area. Other networks have some challenged characteristics even if their speed is fast. Figure 1 shows a conceptual example of the proposed system. Assume that three types of communication media (NW_1, NW_2, and NW_3) are available to transfer large data sets (hundreds of Mbytes). NW_1 is a stable network with a low data rate (e.g. dozens of kbps). NW_2 and NW_3 have sufficiently high data rates but are not sufficiently stable; the probability of loss is high (e.g., more than 5–10%), and the networks frequently drop connections.

In conventional approaches, a communication medium is selected either manually by a user or dynamically by horizontal and/or vertical handover [3]. More than one media may be simultaneously used by multi-homing such as stream control transmission protocol (SCTP) [4] or concurrent multiple transfer [5], which is the extension of SCTP for overcoming its performance degradation problem due to re-ordering. However, since no network is efficient in the above-mentioned challenged environment, it takes a long time to complete the data transfer over such networks and communication may fail due to frequent timeouts, irrespective of the communication media selected.

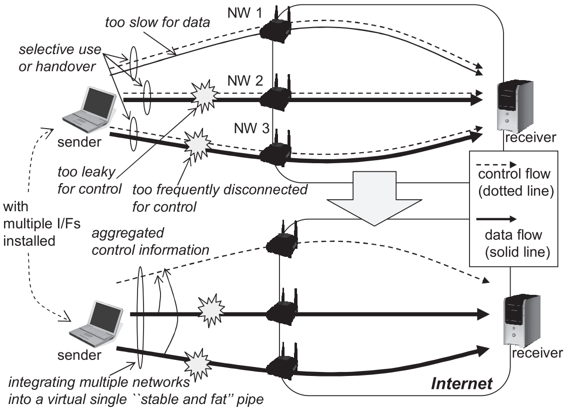

Fig. 1 Proposed concept.

Delay, Disruption, and/or Disconnection Tolerant Networking (DTN) is an emerging research area for addressing problems due to characteristics of challenged networks, in which the conventional TCP/IP-based model does not work well [6]. LTP-T [7] is the multi-hop extension of LTP [8] standardized in DTNRG [9]. It is a retransmission-based reliable transport protocol tolerant of end-to-end long delays and frequent disconnections, and can be an efficient transport solution in a challenged environment. It cannot, however, make the best use of multiple network resources in our assumed condition.

P. Kyasanur et al. [10] proposed the MAC protocol of separating the layer 2 control message (CTS or RTS) and layer 2 data (user data or ACK) into different frequency bands and using them simultaneously in order to improve the throughput of 802.11 wireless LAN. In a resource allocation protocol for intentional DTN (RAPID) [11], in order to get a report or exchange control information for routing from nodes other than the neighbor, the usage of an external network is referred to. N. Banerjee et al. [12] proposed simultaneous usage of different and multiple wireless communication media installed in the same system: Wi-Fi and Xtend (900 MHz radio). Xtend with a longer range than that of Wi-Fi detects a Wi-Fi client in a moving vehicle into the area of Wi-Fi hot-spot, and wakes the Wi-Fi access point in sleep mode before the client comes near.

The proposed system takes advantage of integration of multiple, heterogeneous and diverse networks, not simply for bandwidth aggregation but also particularly for providing stable control information exchange for end-to-end reliable transfer. In the proposed approach, file transfer is performed by multiple separated communication flows of control information for reliable and efficient data transmission (control flow), and those of the data transmission itself (data flow). Each flow is distinguishably transmitted and can be conveyed by different communication media, depending upon the characteristics of the flow and communication medium. A control flow conveys information for retransmission (ACK or NACK), rate adjustment, and so on. The size of control flow is relatively much smaller than that of the data, so the bandwidth required for network to convey it can be very small. Instead, it needs stable transfer. In the proposed system, control flow may not necessarily be through the same communication media as data flows; it may reach the receiver through more stable communication medium that also covers a wider area. For example in Fig. 1, steady and consequently timely exchange of control information via a stable network such as NW_1 enables data transfer in an unstable or intermittent network such as NW_2 or NW_3.

The proposed system supports both senders (as shown

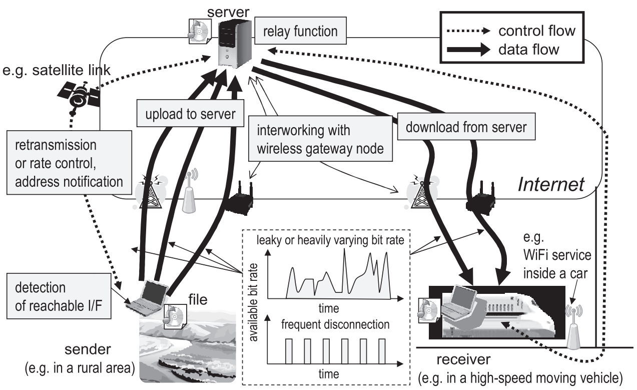

Fig. 2 An overview of our proposal.

in Fig. 1) and receivers in challenged networks. Using the control flow, the receiver notifies the sender host of the preferred interfaces that may receive data flows. When both senders and receivers are in a challenged network, a relay node in the Internet, as shown in Fig. 2, is required for the efficient transfer of large data sets.

One interesting question is which is a better approach — reactive or proactive to error recovery for achieving reliable transfer? A simple duplicated transfer is one of proactive approaches. Parity-added transfer by Forward Error/Erasure Coding (FEC) is a more sophisticated proactive approach [13], and can be used for efficiently and flexibly utilizing, in some cases, as great an amount of resources in a leaky network as possible [14]. In proactive approaches, however, it is not easy to prevent total performance (i.e., goodput or transfer time) from degrading by useless transfer due to excess redundancy or duplicate data. This is especially true when multiple users share the same challenged networks with limited resources. Therefore the proposed system prototype in Sect. 3 adopted a reactive approach using NACK-based retransmission that is expected to suppress excess data transfer by timely feedback, which makes resource (bandwidth) consumption minimal. More sophisticated combinations of proactive and reactive approaches, adaptive to the situation, should be studied in future work.

The concept of our reactive approach in retransmission method is basically similar to a classical and well-known method, selective repeat ARQ [15] as a way that a sender retransmits a segmented piece of data (we call it a sector) as lost one. The main difference is the mechanism of loss detection which triggers a feedback on retransmission from a receiver to a sender. Our proposed method does not always react and give a feedback every time a receipt of sector occurs in irregular sequence, in contrast to selective repeat ARQ. Since multiple data paths are simultaneously used in our proposal, each sector may experience a different delay depending on the data path through which it travels. The number of data paths can dynamically change during transfer, according to the condition change of each access network. In addition, we assume the challenged network in which packet loss often occurs. Therefore sectors often reach at a receiver out of order, which makes it difficult for a receiver to quickly judge whether a certain unreceived sector has been lost somewhere in the network, or has not yet reached because of a longer delay than expected.

3. Multi-Network Data-Transfer System

The design of the mechanisms in the prototype system based on the concept presented in Sect. 2 is described. The present study focuses on large-sized data transfer applications.

3.1 System Overview

To achieve efficient large-sized data transfer over challenged networks, two agents substitute direct end-to-end communication; one is a client that initiates data transfer, the other is a server that responds to a client request. The system supports both senders and receivers in challenged networks. A client agent works as a sender and a server as a receiver when uploading a file, and vice versa for download. It is assumed that the agent host in a challenged environment can simultaneously access multiple and different types of wireless networks to take advantage of the combination, and that at least one is a stable network even if it is at lower speed. A file is transferred between these two agents by exploiting multiple available wireless networks; therefore we call this multi-network data transfer.

The control flow is set up between agents over a TCP connection via a stable network. The file to be transmitted is then fragmented and concurrently sent out over UDP flows via multiple available networks according to the conditions of the data medium. Data transmission between agent hosts adapts UDP-based transmission, which attempts to send as much data as possible to each data link despite their disrupted or lossy condition. The application layer adjusts and controls UDP transmission based on the feedback information via the control path. Missing sector information enables retransmission control, and received data rate information enables rate adjustment control. Note that the stable network is not dedicated only for control flow, but can be also concurrently used for control and data flows.

Figure 3 shows the case in which the sender agent on the left side transfers a file to the receiver agent on the right side with an interface to access a stable network and multiple interfaces (e.g., three interfaces in the figure) to access an unstable one with a higher data rate. The procedure of multi-network data transfer between the client and the server agent in the prototype system is described as follows:

- Two agents attempt to synchronize the files stored in each local cache. A client agent sets up a control connection to a server agent when a file transfer request occurs. The agents periodically check the synchronization of the cached file with each other via the stable control path. If files are not found in the cache of the other agent, then the local side of the agent reports the details of those files.
- Files to be transferred are fragmented into sectors of

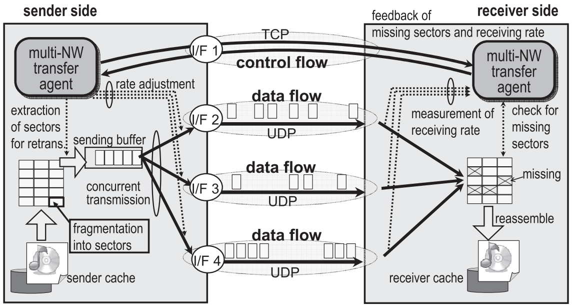

Fig. 3 Data transfer prototype system.

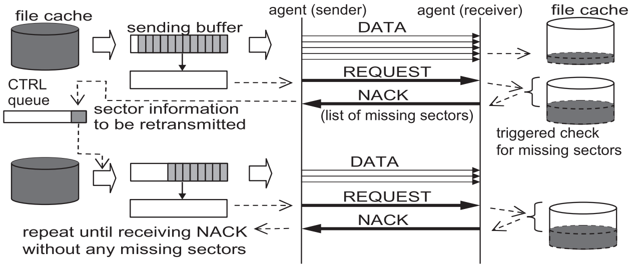
file cache
sending buffer
agent (sender)
agent (receiver)
file cache
DATA
REQUEST
CTRL queue
NACK
sector information to be retransmitted
(list of missing sectors)
triggered check for missing sectors
DATA
REQUEST
NACK
repeat until receiving NACK without any missing sectors

Fig. 4 Retransmission control (1).

fixed-size and synchronized by sector. Each sector is identified by a unique identifier within the file, e.g., a sequential number. The sender–agent notifies the number of sectors consisting of the file to be transferred to the agent on the receiver side.

- The receiver–side agent checks whether the specified file in the local cache has unreceived sectors. If there are any, the receiver agent returns to the sender agent the list of unreceived sector identifiers. Note that all sectors are regarded as unreceived at the beginning phase of the transfer.
- The sender agent extracts specified sectors from the file to be transferred, puts them into a send buffer, and transfers them to the receiver agent over all the available links, which has been previously configured as data links for usage. Multiple sectors can be contained in a UDP payload, as long as the total size of them does not exceed it. The size of sector is set to 256 bytes in our prototype system.
- The receiver agent stores the received sectors and reassembles all fragmented data into the original file. This agent sends explicit feedback information, namely, the missing sector information and the received data rate of each UDP flow, to the sender side via the control path. The sender agent receives the feedback information and, based on it, retransmits specified sectors and controls the sending rate.
- Every sector is retransmitted by a sender agent as long as a receiver requires retransmission on that sector in its feedback message. In other words, there is no upper limit on retransmission count.
- The file transfer finishes when the receiver agent receives all the sectors and reassembles the original file. After transfer between agents, the receiver agent uploads the reassembled file to the destination remote host, according to the upload information, by traditional means such as FTP.

Data transmission between agent nodes adapts UDP-based transmission, which attempts to send as much data as possible to each data link in spite of their disrupt condition. Application layer adjusts and controls UDP transmission based on the feedback information via the control channel.

3.2 Retransmission Control

The retransmission is based on explicit feedback, including the result of the check for unreceived sectors by the receiver agent. In order to trigger the unreceived sector check by the receiver agent, the following combination of two methods is used.

3.2.1 Basic Request-Response Style

A sender agent prompts the receiver agent to check whether the receiver agent has any unreceived sectors.

As shown in Fig. 4, after the sender agent sends out all sectors in the send buffer to the receiver agent, the sender agent generates and sends a control message (REQUEST) to the receiver agent. This message notifies the receiver agent of the completion of transfer from the sender agent's perspective and requests sector arrival information for retransmission. When the receiver agent receives a REQUEST message, it checks whether a specified file has any unreceived sectors. Sectors unreceived at that time are regarded as lost sectors, which must be retransmitted. To respond to a REQUEST message, the receiver agent generates a control message (NACK), which includes a list of identifiers of the unreceived sectors. A sender agent judges that sectors listed in the received NACK message must be lost and extracts these sectors into the send buffer for retransmission. A NACK message with no sector identifier implies that all sectors are successfully received at the receiver agent, which signals the end of a transfer. The sender agent repeats this procedure until receiving such an NACK message.

Control message exchange for retransmission occurs only when the send buffer becomes empty in the sender agent. There is low frequency of control message exchange. Therefore this is robust against accidental disconnection of the control path, because the control path is not always needed and can be setup only when necessary.

However, when the amount of remaining unreceived sectors becomes small, i.e. at the final stage of transfer, a sender agent may experience a long transmission idle due to waiting for a round-trip request-response procedure if the control path has a long delay. Additionally, if the network conditions of data paths are leaky, such a retransmission with idle time may occur several times. Therefore, in basic request-response style, a sender agent possibly suffers

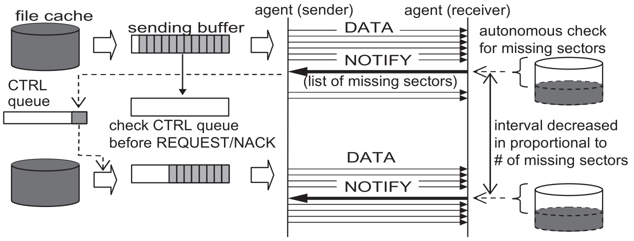

Fig. 5 Retransmission control (2).

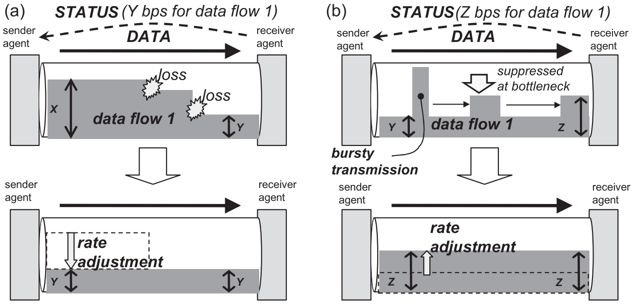

Fig. 6 Transmission rate control.

from this last-segments transmission idle.

3.2.2 Receiver-Initiated Notification

To improve the performance degradation of last–segments transmission idle caused by the long delay of the control path and the leaky data paths, a receiver-initiated notification method is also introduced. This method is a self-directive check of the unreceived sector in the receiver agent, independent of receiving a REQUEST message generated by the sender agent as illustrated in Fig. 5.

The receiver agent autonomously checks unreceived sectors of the received file during the transfer. The check occurs at certain intervals. A receiver agent generates a control message (NOTIFY) to notify the sender agent of the results of the autonomous check. A NOTIFY message includes the same content as the NACK message. A sender agent regards sectors listed in this message as lost sectors and extracts them into the send buffer for retransmission. Note that the sender agent reacts to the received NOTIFY message only when its send buffer becomes empty. If multiple and different NOTIFY messages are received before the send buffer becomes empty, the sender agent processes only the most recently received message and discards the out-of-date messages.

In this method, the interval of the autonomous check can be equivalent to a retransmission timeout. A shorter interval tends to trigger aggressive retransmission, but if too short it often causes excess retransmission for the initial period of transfer time. If it is too long, it may contribute little to the improving on the performance degradation. The interval of the autonomous check can be changed adaptively so as to decrease proportionally to the percentage of unreceived sectors. The final stage starts when the percentage of successfully received sectors (i.e. a file transfer progress) exceeds a certain threshold. The autonomous check works only in this stage, meaning that the aggressive retransmit is enabled only at the final stage of transfer. Currently we use a static configured value for this threshold. It is our future work to dynamically and automatically determine this threshold depending on the status of the transfer or networks.

3.3 Transmission-Rate Control

In challenged environments, it is necessary to fully and efficiently use valuable network resources that are not always available because of frequent disconnection or severe changes in bandwidth. It is efficient that data packets are sent out with as high a data rate as possible in order to achieve better transfer performance. At the same time, however, it is necessary to avoid an overloaded transmission rate which would cause excess retransmission and would lengthen the transfer time.

The proposed system controls the transmission rate of each of multiple data flows independently. A sender agent cooperates with a receiver to estimate the bottleneck available bandwidth of the end-to-end path through which each data flow travels. Figure 6 shows that a sender agent controls its sending rate on data flow 1, which is one of data flows used in multi-path transfer. A receiver measures the averaged receiving rate of each data flow and gives feedback about its rate information to a sender agent by a control message (STATUS) via control path. Basically, the sender agent adjusts the data flow's transmission rate according to the measured receiving rate seen in the STATUS message. As shown in Fig. 6(a), if a receiver detects that the receiving rate Y is lower than X measured before, a receiver sends a STATUS message indicating the newly measured rate Y. A sender changes its sending rate from X to Y.

The available bandwidth may change dynamically. To follow its change and transmit data flow at as high a data rate as possible, the sender agent additionally makes periodical bursty transmission with a higher data rate than that adjusted by the STATUS message. If the available bandwidth of the network path increases after a previous adjustment, this bursty transmission makes the receiver agent measure and give feedback of a higher rate than before. As shown in Fig. 6(b), when a receiver detects that the averaged receiving rate Z, led by burst transmission of a sender, is much higher than Y measured before, the receiver sends a STATUS message indicating the measured rate Z.

In each case, a STATUS message is generated only when receiving rate of a data flow changes at a certain level to avoid oscillation by too frequent changes. Note that this approach to estimating the end-to-end available bandwidth is not always accurate. However, since we use an averaged receiving rate of a sequence of the burst test packets and also check losses of those packets, our method can be usable for the purpose of triggering an increase of the sending rate. To

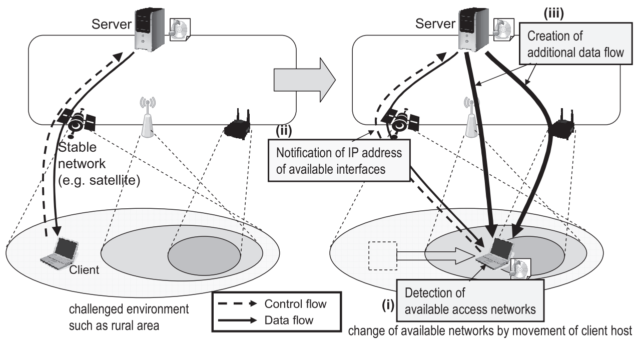
Server
Stable network (e.g. satellite)
Client
challenged environment such as rural area
Control flow
Data flow
Server
(iii)
Creation of additional data flow
(ii)
Notification of IP address of available interfaces
(i)
Detection of available access networks
change of available networks by movement of client host

Fig. 7 Topology control.

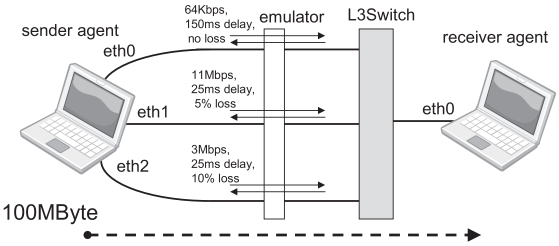
sender agent
eth0
eth1
eth2
64Kbps, 150ms delay, no loss
emulator
L3Switch
receiver agent
11Mbps, 25ms delay, 5% loss
eth0
3Mbps, 25ms delay, 10% loss
100MByte

Fig. 8 Configuration of emulated network.

Table 1 Characteristics of emulated network.

	path	bandwidth	delay	loss probability
NW_0	control	64 kbps	150 ms	0%
NW_1	data 1	11 Mbps	25 ms	random 5%
NW_2	data 2	3 Mbps	25 ms	random 10%

implement a more accurate estimation method is one of our future works.

3.4 Data-Flow Setup Control

When a receiver agent has multiple interfaces desired to receive data flow, a sender agent tries to transfer with multiple data flows to exploit multiple network accesses to a receiver. An issue arises that it is difficult for a sender agent to understand the status of data interfaces at a receiver agent: IP address or connectivity (up or down). It is especially true, for an example, when a client agent in the widely moving vehicle requests download and plays a receiver role. In this case, an IP address of data interface of a client agent may dynamically and often change.

As shown in Fig. 7, in our proposed system, a receiver agent generates a control message (TOPOLOGY) including a list of a set of device names (e.g., ppp0 or wlan0), IP address and connectivity information of a data interface. This message is generated not only at the start of transfer, but also whenever an IP address or its connectivity is changed. TOPOLOGY message is sent to a sender agent. Note it is assumed in this paper that which device to be used for receiving data flow has been previously configured.

The TOPOLOGY message enables a sender agent to immediately follow the change of interface status of a receiver agent. A sender agent reacts according to the contents of the received TOPOLOGY message. When the connectivity status of a device that has never been up becomes up, it sets up a new data flow targeted to the specified destination IP address. When a device that has been up becomes down, it removes the corresponding data flow. When an IP address of a device changes in TOPOLOGY message, it takes away corresponding data flow and sets up a new data flow targeted to the specified destination IP address.

4. Experiment

We performed two experiments using a combination of terrestrial communication links and a satellite communication link, in order to validate our prototype implementation including basic request-response, receiver-initiated retransmission control, and transmission-rate control. The first experiment was conducted in May 2008, and used a satellite communication link, ETS-VIII [16], which was regarded as a *low-speed but stable* network and, because of its mobility awareness, was better suited for control flow in that experiment. The second experiment was conducted in June 2010, and used a high-speed satellite communication link, WINDS (Wideband InterNetworking engineering test and Demonstration Satellite) [17], which was regarded as a *higher-speed but unstable* network that was better suited for data flow.

4.1 Experiment Using Emulated Network

Ahead of field experiments using a satellite link, we present a basic evaluation of the prototype system to transfer a 100M-byte file between two agents in an indoor experiment. As shown in Fig. 8, three different emulated networks (NW_0, NW_1, and NW_2) are available for a sender agent. The characteristics of each network are shown in Table 1. While the characteristics of NW_0 are low speed and long delay, but very stable (no loss), those of NW_1 and NW_2 are higher speed but very leaky. Therefore, NW_0 is used for the control path, and the others are used for the data paths.

4.1.1 Multiple Data-Path Transfer

Figure 9 shows the result of a 100M-byte file transfer using a separated control flow (NW_0) and two data flows (NW_1 and NW_2). The left y-axis shows the receiving rate of the data flow. The right y-axis shows the percentage of successfully received sectors as the file transfer progresses. The file-transfer progress reaches 100% at the completion of the transfer. Both of the methods shown in Sects. 3.2.2 and 3.2.1 are used for retransmission. The threshold for enabling the receiver-initiated notification (autonomous notification of a receiver agent) is set to 70. The average throughput of data flows 1 and 2 is 8.76 Mbps and 1.98 Mbps respectively. On the other hand, Figs. 10 and 11 show the results of the cases using only one data flow (NW_1 and NW_2, respectively) and a separated control flow. The average throughput of data

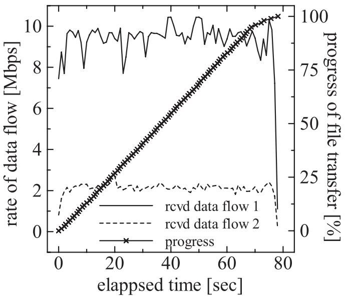

Fig. 9 Result of emulated network using two data flows (data path 1 and 2).

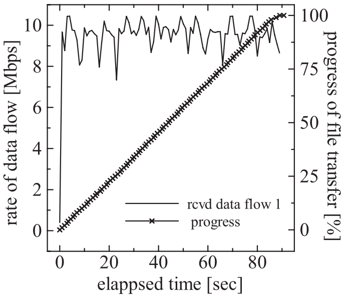

Fig. 10 Result of emulated network using one data flow (data path 1).

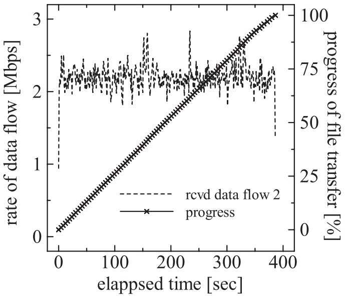

Fig. 11 Result of emulated network using one data flow (data path 2).

flow 1 in Fig. 10 is 9.08 Mbps. That of data flow 2 in Fig. 11 is 2.12 Mbps. This experiment demonstrates that our proposed system effectively utilizes the bandwidth of multiple data paths by aggregation.

4.1.2 Impact of Control Path Characteristic

This section shows the evaluation on how the difference of control path characteristics impacts on the transfer performance. The transfer time of a 100M-byte file is evaluated in the same configuration as that of Sect. 4.1.1, except for the delay and loss probability of NW_0.

First, we focus on the impact of control delay only and assume that no loss occurs in control path. As stated in Sect. 3.2.2, a longer control delay will lead to a large transmission idle on the sender side without a receiver-initiated notification. Tables 2(1) and (2) show a comparison of the transfer time with or without a receiver-initiated notification in a different control-path delay, 150 ms and 500 ms.

Table 2 An average transfer time using two data paths.

Control-path	(1) no loss 150 ms delay	(2) no loss 500 ms delay	(3) 10% loss 150 ms delay
Retrans. w/o NOTIFY	81.0 s (σ =6.2)	97.4 s (σ =9.3)	– –
Restans. w/ NOTIFY	79.6 s (σ =2.7)	80.4 s (σ =2.5)	123.2 s (σ=31.3)

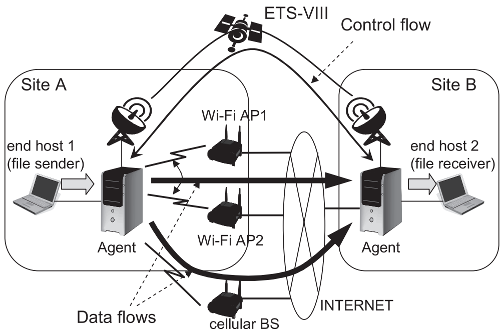

Fig. 12 Configuration of experiment using ETS-VIII.

σ shows the standard deviation. It shows approximately the same average transfer time in both the cases with NOTIFY (79.6 s) and without NOTIFY message (81 s), when the control-path delay is 150 ms. However, when the control-path delay is 500 ms, it takes 80.4 s in the case with NOTIFY but 97.4 s in the case without NOTIFY. This difference might come from the transmission idle time of the sender at the final stage of transfer. The introduction of NOTIFY suppresses its performance degradation.

Second, we focus on the impact of control-path stability, i.e. loss probability in control-path. Table 2(3) shows the transfer time when the loss probability of the control-path is equal to 10%. The average transfer time is 123.2 s, and the standard deviation is 31.3. As compared to (1), they are much larger and dispersed. When a control-path suffers from heavy packet losses, some STATUS or NOTIFY message are lost or significantly delayed. This causes a sender to fail to perform an appropriate rate control or retransmission, which results in a longer and more variable transmission time.

4.2 Experiment Using ETS-VIII

4.2.1 Configuration

We tested the prototype system in a file transfer experiment between two locations in Kitakyushu (Fukuoka) in Japan. As shown in Fig. 12, the control channel is setup through a satellite link of the engineering testing satellite ETS-VIII. The data links are an IEEE802.11g Wi-Fi and a cellular Internet service. The file upload source moves over several Wireless Access Points (APs) during data upload, which results in disconnection and disruption on the Wi-Fi link. In this experiment, an agent node at site A switches its associating AP between AP1 and AP2 every 30 seconds. While moving through an AP area, the transmission rate of its

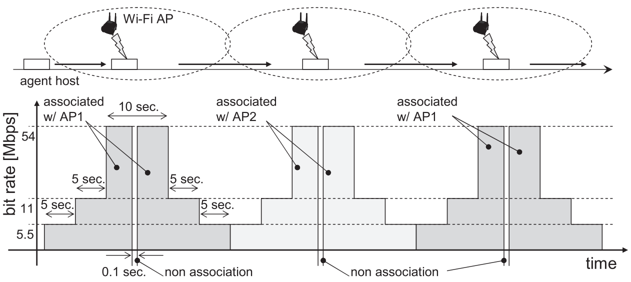

Fig. 13 Wi-Fi scenario in ETS-VIII experiment.

wireless link changes from 5.5 Mbps to 54 Mbps, and an instantaneous interruption also occurs. These conditions of the Wi-Fi link are manually operated to simulate the scenario as shown in Fig. 13. In this scenario, we simulated a mobile host moving across multiple Wi-Fi hotspot areas. The cellular link (384 kbps uplink) has a high latency, and its delay is not constant. In this section our main goal is to validate if our prototype implementation works well on an agent host installed multiple interfaces each of which has a different type of characteristics in the real environment.

4.2.2 Evaluation Result

Two experimental combinations of media and functions (data/control) shown in Table 3 are examined for performance comparison. Case 1 corresponds to the traditional case in which an application uses two parallel TCP connections at the same time to fully utilize the available media (communication paths), while Case 2 represents the proposed proxy data transfer system with two UDP data flows and extra control channel via satellite link. Figure 14 shows an example of the experimental results where the time required to transfer a 100-Mbyte file from an agent node at site A to one at site B is evaluated. The result of Case 1 shows the summation of the amount of data transferred by each of two independent TCP connections. It takes 237 seconds to transfer 100-MByte data. Contrastively, in the proposed scheme shown as Case 2, it takes 116 seconds only, around half of Case 1. Especially on a Wi-Fi link, while the rate of Case 2 increases according to a change of the Wi-Fi in its bit rate when a higher bit rate becomes available, the rate of Case 1 increases much less than the rate of Case 2. This is possibly because an instantaneous interruption during a transmission at a high bit rate or in case of AP switching makes the TCP connection window size of a sender small significantly, and this mechanism prevents a prompt increase of its sending rate after the disconnection.

4.3 Experiment Using WINDS

4.3.1 Configuration

We tested the prototype system in a file transfer experiment between two locations in Japan, namely, Koganei (Tokyo) and Kashima (Ibaraki), in June 2010.

Table 3 Test cases of experiment using ETS-VIII.

	Flows		
	via ETS-VIII	via Wi-Fi	via 3G
Case 1	–	Cubic TCP	Cubic TCP
Case 2	CTRL	DATA	DATA

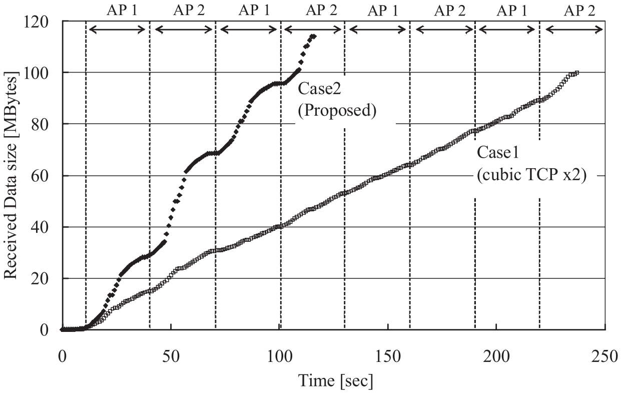

Fig. 14 Results for Cases 1 and 2 of experiment using ETS-VIII.

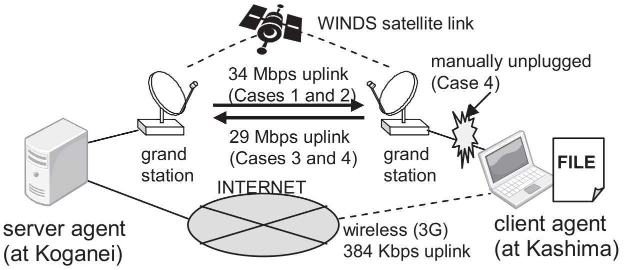
WINDS satellite link
34 Mbps uplink (Cases 1 and 2)
29 Mbps uplink (Cases 3 and 4)
manually unplugged (Case 4)
grand station
grand station
INTERNET
FILE
server agent (at Koganei)
wireless (3G) 384 Kbps uplink
client agent (at Kashima)

Fig. 15 Configuration of experiment using WINDS.

As shown in Fig. 15, a client agent located at Kashima has two available networks to a server located at Koganei: a satellite link and a terrestrial cellular link. A WINDS grand station antenna is set up at each location and is connected to a local agent by Ethernet cables. The agents have connectivity to each other through this satellite link, in which the actual link speed is 34 Mbps (from Koganei to Kashima) and 29 Mbps (from Kashima to Koganei), and its average round-trip-time is approximately 800 ms. The client also has a terrestrial commercial cellular 3G connection (384 kbps uplink), and the server agent has a stable wired Internet connection (1 Gbps). The average round-trip-time from the client via a 3G link to the server is approximately 400 ms.

4.3.2 Evaluation Result

We tested four cases, as shown in Table 4. In Cases 1 and 2, a client in Kashima downloads a 100-Mbyte file from a server in Koganei, i.e., the direction of file transfer is from Koganei to Kashima, and, in Cases 3 and 4, a client uploads the file to a server. Cases 1, 2, and 3 are under stable conditions, i.e., with no disruption, whereas, in Case 4, the connection through a satellite link experiences temporary disruption during transfer in order to allow file transfer under

Table 4 Test cases of experiment using WINDS.

	Direction of Transfer	Flows	
		via Satellite	via 3G
Case 1	Koganei → Kashima (download scenario)	CTRL, DATA	–
Case 2		DATA	CTRL
Case 3	Koganei ← Kashima (upload scenario)	DATA	CTRL
Case 4		DATA (w/ disruption)	CTRL, DATA

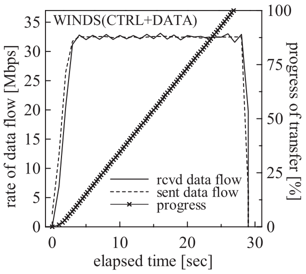

Fig. 16 Results for WINDS Case 1.

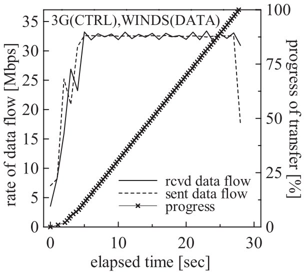

Fig. 17 Results for WINDS Case 2.

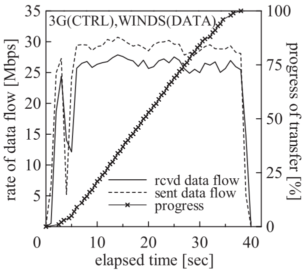

Fig. 18 Results for WINDS Case 3.

the challenged condition. For intentional disruption, an Ethernet cable between the grand station and the client agent was manually unplugged during transfer and was plugged in again after a short interval.

Figures 16 through 19 shows the results for Cases 1 through 4, respectively. The left y-axis shows the sending (dotted-line) and receiving (solid-line) rate of the data flow. The right y-axis shows the percentage of successfully received sectors as the file transfer progresses.

In Case 1, only a satellite link is used for transfer and both control and data flow travel through the satellite link. In Case 2, not only a satellite link but also a terrestrial 3G link are used. The control flow travels through the 3G link, and the data flow travels through the satellite link. In Case 1, it takes approximately 27 seconds to complete the file transfer, and, in Case 2, it takes approximately 28 seconds. The transfer time is almost the same in both cases. Thus these results reveal that the prototype works well in both cases and that there is no negative impact, even if the control flow travels through a different network.

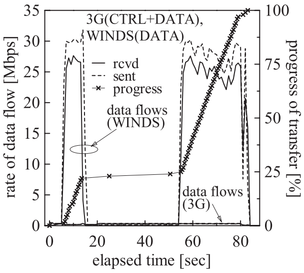

Fig. 19 Results for WINDS Case 4.

The condition of the satellite link is incidentally lossy in Cases 3 and 4. There is a non-negligible difference between the sending and receiving rates of data flow through the satellite link. The sending rate of the data flow at the client agent is approximately 29 Mbps, which is similar to the actual speed of the satellite link, whereas the receiving rate at the server agent is approximately 26 Mbps. This is considered to occur because the data flow experiences random losses by error packets in the satellite link, primarily due to bad weather around Kashima, which is the uplink side of the satellite link. There were torrential rains around Kashima at the time Cases 3 and 4 were tested, which may have caused unexpected heavy rain attenuation.

In Case 3, the configuration is the same as that for Case 2. It takes approximately 38 seconds to complete the transfer. The results for Case 3 cannot be simply compared to the results for Case 2 because of the difference in link speed. However, the results indicate that the sending rate of data flow is as high as possible, despite the lossy condition of the satellite link. This is owing to the steady exchange of control information via the separate 3G link.

In Case 4, the control flow and data flow travel through the 3G link, and an additional data flow travels through the satellite link. The Ethernet cable is unplugged approximately 15 s after the start of transfer, and the IP connectivity recovers in approximately 55 s. The data flow through the satellite link stops and restarts according to its connectivity status. This result indicates that as the stable 3G link maintains the control flow despite disconnection of the satellite link. The client (sender) agent can be notified of the disconnection and recovery of satellite link connectivity by feedback from the server (receiver) agent via control flow.

5. Conclusion

The present paper has described a prototype implementation of a data transfer system based on our previously proposed framework for large-sized data transfer by integrating multiple and heterogeneous challenged access networks. We conducted two field experiments. The first experiment used a combination of terrestrial communication links (3G and Wi-Fi) and a satellite communication link (ETS-VIII) which was regarded as a *low-speed but stable* network and better suited for control flow in that experiment. Our prototype implementation performs better than a simple aggregation of transfer by parallel TCP connections in this experiment. It clearly indicates the potential that the basic concept of steady extra control channel could improve transfer performance in the challenged environments.

The second experiment used a pair of a high-speed satellite communication link (WINDS) and a terrestrial cellular link (3G). The results of this experiment validated that a separated control flow via a *lower-speed but more stable* 3G network enables the transfer of large-sized data by exploiting a *high-speed* WINDS link even under lossy and/or disrupted conditions, i.e., *unstable* conditions.

Despite further advances in wireless technology, it will not be cost-effective to provide stable and high-speed Internet access via wireless networks in all areas and all situations. Therefore even in the future Internet, a combination of heterogeneous networks (i.e., *lower-speed but more stable* and *higher-speed but unstable* networks) is of practical importance for efficient large data transfers.

Acknowledgement

This study was supported in part by the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (B) (21300024).

References

[1] A. Nagata, S. Yamamura, M. Uchida, and M. Tsuru, "Proxy data transfer system with a stable control channel and dynamically changing data channels," 3rd ACM Workshop on Challenged Networks (CHANTS'08), pp.121–124, Sept. 2008.

[2] A. Nagata, S. Yamamura, and M. Tsuru, "Integrating multiple and heterogeneous challenged networks — Detailed design and experimental evaluation using satellite link," 3rd International Workshop on Information Network Design (WIND2010), pp.362–367, Nov. 2010.

[3] J. Roy, V. Vaidehi, and S. Srikanth, "Always best-connected QoS integration model for the WLAN, WiMAX heterogeneous network," Proc. 1st International Conference on Industrial and Information Systems, pp.361–366, Aug. 2006.

[4] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," IETF RFC2960, 2000.

[5] J.R. Iyengar, P.D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," IEEE/ACM Trans. Netw., vol.14, no.5, pp.951–964, 2006.

[6] K. Fall, "A delay-tolerant network architecture for challenged internets," ACM SIGCOMM'03, Aug. 2003.

[7] S. Farrell and V. Cahill, "Evaluating LTP-T: A DTN-friendly transport protocol," 3rd International Workshop on Satellite and Space Communications (IWSSC'07), Sept. 2007.

[8] S. Burleigh, M. Ramadas, and S. Farrell, "Licklider transmission protocol-motivation," IETF RFC 5325, Sept. 2008.

[9] "Delay tolerant networking research group," http://www.dtnrg.org

[10] P. Kyasanur, J. Padhye, and P. Bahl, "On the efficacy of separating control and data into different frequency bands," 2nd International Conference on Broadband Networks (BROADNETS), Oct. 2005.

[11] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," ACM SIGCOMM 2007, Aug. 2007.

[12] N. Banerjee, M.D. Corner, and B.N. Levine, "An energy-efficient architecture for DTN throwboxes," IEEE INFOCOM 2007, pp.776–784, May 2007.

[13] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," ACM SIGCOMM 2005, Oct. 2005.

[14] R. Mahajan, J. Padhye, R. Raghavendra, and B. Zill, "Eat all you can in an All-You-Can-Eat buffet: A case for aggressive resource usage," 7th ACM Workshop on Hot Topics in Networks (HotNets-VII), Oct. 2008.

[15] H. Burton and D. Sullivan, "Error and error control," Proc. IEEE, vol.60, no.11, pp.1293–1301, Nov. 1972.

[16] "Engineering Test Satellite VIII: KIKU no.8," http://www.jaxa.jp/projects/sat/ets8/index_e.html

[17] "Wideband InterNetworking engineering test and Demonstration Satellite," http://www.jaxa.jp/projects/sat/winds/index_e.html

Akira Nagata received the B.E. and M.E. degrees in Communications Engineering from Osaka University in 2000 and 2002, respectively. He worked at Fujitsu Laboratories Ltd. since 2002. In 2007, he moved to Network Application Engineering Laboratories Ltd. He was also a research member in NICT from April 2008 to March 2011. He has been engaged mainly in the research of network architecture and network system.

Shinya Yamamura received the B.E. degree in Electronic Engineering from Oita University in 1988. In 1988, he joined Fujitsu Kyushu Network Technologies Limited. He was also a research member in NICT from April 2008 to March 2011. He has been engaged mainly in the research of network architecture. He is a Ph.D. student with Kyushu Institute of Technology.

Masato Tsuru received B.E. and M.E. degrees from Kyoto University, Japan in 1983 and 1985, respectively, and then received his D.E. degree from Kyushu Institute of Technology, Japan in 2002. He worked at Oki Electric Industry Co., Ltd., Information Science Center, Nagasaki University, Japan Telecom Information Service Co. Ltd., and Telecommunications Advancement Organization of Japan. In 2003, he moved to the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology as an Associate Professor, and then has been a Professor in the same department since April 2006. His research interests include performance measurement, modeling, and management of computer communication networks. He is a member of the IPSJ, JSSST, ACM and IEEE.