End-to-End Seamless Handover using Multi-path Transmission Algorithm

Shigeru Kashihara, Katsuyoshi Iida, Hiroyuki Koga, Youki Kadobayashi and Suguru Yamaguchi
Graduate School of Information Science,
Nara Institute of Science and Technology, Ikoma-shi, Japan.
Email: {shige-ka, katsu, koga, youki-k, suguru}@is.aist-nara.ac.jp

Abstract— The current Internet is being constructed from various kinds of wired and wireless access networks, and mobile hosts can connect to the Internet at any time and anywhere. In such an environment, mobile hosts need new technologies without closing connection or degrading the goodput during roaming these different networks. To overcome this, we propose an end-to-end seamless handover mechanism. In our approach, a mobile host needs to coordinate with its corresponding host to move across different wireless access networks without connection severance and degradation of communication quality on an end-to-end basis. In this paper, we propose a multi-path transmission algorithm for end-to-end seamless handover. The main purpose of this algorithm is to improve the goodput during handover by sending the same packets along multiple paths, as reducing unnecessary consumption of network resources. We evaluate our algorithm through simulations and show that mobile hosts gain a better goodput.

Index Terms— Multi-path transmission algorithm, End-to-end seamless handover, SCTP.

I. Introduction

The current Internet is being constructed from various kinds of wired and wireless access networks. To connect with these networks, mobile hosts have many network interfaces. As a result, mobile hosts can connect to the Internet at any time and anywhere. In such an environment, it is an urgent problem to enable mobile hosts to move across these different networks without connection severance and degradation of communication quality. Moreover, since real-time communications, such as video streaming and Voice over IP (VoIP), are expected to increase more than non real-time communications, we believe that to keep the quality of a real-time communication during handover is essential and must be solved. For example, a Wireless Local Area Network (WLAN) is relatively inexpensive, wideband and has stable connectivity, but its coverage is very limited. Although a cellular network provides a wide coverage area, it is relatively expensive, narrowband and has unstable connectivity. Therefore, since each wireless access network has different features, mobile hosts need to move across these networks while keeping the communication quality.

To accomplish this, we propose an end-to-end seamless handover mechanism in which a mobile host can roam across different wireless access networks. In our approach, a mobile host needs to coordinate with its corresponding host to move across different wireless access networks on an end-to-end basis.

In this paper, we employ the Stream Control Transmission Protocol (SCTP) [1], [2] as one of end-to-end protocols. This is because we think SCTP is appropriate research platform of end-to-end protocols, since it has many functions that have arisen from decades of research in transport protocols. As one of such functions, SCTP has multihoming function to achieve reliable transport service for VoIP signaling. The multihoming function simultaneously handles multiple network interfaces for sustaining the reliable transfer of data between two hosts. We make use of this multihoming function for an end-to-end seamless handover. However, the current SCTP uses only one of the network interfaces at a time.

Our main contribution in this paper is to propose a multi-path transmission algorithm. For a seamless handover with minimum consumption of network resources, our algorithm has two modes: a single-path transmission mode and a multi-path transmission mode. Single-path transmission is a normal mode. If the transmission path currently used becomes unstable, the mode is changed to the multi-path transmission mode to avoid quality degradation. Moreover, after the condition of the transmission path returns to the stable state, the mode also return to the single-path transmission mode to reduce unnecessary consumption of network resources.

This paper is organized as follows: Section II describes related work on our end-to-end seamless handover. We give an overview of SCTP and propose a multi-path transmission algorithm in Section III, and evaluate the effectiveness of our algorithm through simulations in Section IV. Section V concludes the paper with a description about future work.

II. Related Work

Handover technologies in the Internet are being investigated by a number of researchers. Most of this research are classified into network assisting mobility support (NAMS). A list of

NAMS is the following: Mobile IP [3], [4], Hawaii [5], [6], Cellular IP [7], and Hierarchical Mobile IP [8]. In NAMS, each mobile host communicates with a network device to move across different wireless access networks. Thus, the special network devices need to maintain movement states of mobile hosts, causing high operating and equipment cost. Because of the above reasons, some other researchers have proposed a notion of end-to-end assisting mobility support (EAMS).

In EAMS, each mobile host coordinates with its corresponding host to move across different wireless access networks. Thus, EAMS does not need special network devices to enable mobile hosts to roam across different wireless access networks. Instead, EAMS needs a special end-to-end protocol. Examples of EAMS are TCP migration [9], and MMTP (Multimedia Multiplexing Transport Protocol) [10]. TCP migration, which has been proposed by Snoeren et al., is a modification of normal TCP, allowing changes of IP address of mobile hosts. When a mobile host changes its own IP address, it will send a Migrate SYN packet to notify the change of IP address to its corresponding host. On the other hand, MMTP, which has been proposed by Magalhaes et al., is a novel transport protocol for transferring real-time streaming data on mobile hosts. Due to the nature of wireless links, wireless access networks provide relatively small bandwidth.

Therefore, MMTP is designed to aggregate the available bandwidth from multiple channels to create a virtual channel with more bandwidth. However, all of existing EAMS does not support seamless handover.

One of the definition of seamless handover is appeared in [11]: "Ensuring a seamless (or transparent) migration of an element from one domain to another". The major difficulty of this is how to hide from applications any differences between the service during the migration interval and the normal service. Note that difficulties in seamless handover is also described in a context of the third generation wireless system in [12].

III. Multi-Path Transmission Algorithm

First, we define an end-to-end seamless handover. Since the conventional terminology of "handover" is assumed as network assisting mobility support (NAMS), the terminology combination of "end-to-end" and "seamless handover" gives us a feeling of wrongness. Our definition is this: seamless migration of a mobile host from one network to another in end-to-end assisting mobility support (EAMS) systems. Since existing techniques of seamless handover are only for NAMS, we need to implement a new technique.

As a key technique, we propose a multi-path transmission algorithm. In our algorithm, a mobile host can switch between multi-path transmission mode and single-path transmission mode to achieve seamless handover with minimum consumption of network bandwidth.

In this section, we describe our multi-path transmission algorithm. First, we explain our motivation for choosing SCTP as a base protocol of end-to-end seamless handover in Section III-A. We then describe modifications to SCTP in Section III-B. Finally, we give a description about our multi-path transmission algorithm in Section III-C.

A. Motivation for Choosing SCTP

This section describes why we chose SCTP as our base protocol. The Stream Control Transmission Protocol (SCTP) is a novel transport protocol designed for both reliable and unreliable data transmissions [1], [2]. One major advantage of SCTP is that it has a multihoming function. Since this function can handle multiple network interfaces, it will be useful in switching between different wireless network interfaces when one of the wireless links is disconnected. We think the multihoming function will be a basic function to realize seamless handover.

Moreover, there is an enhancement of SCTP, called ADD-IP [13]. ADD-IP enables mobile hosts to dynamically add and/or delete IP addresses without disruption of the connection. This function should be co-utilized along with our algorithm. Since our main focus in this paper is the algorithm, we are not concerned with ADD-IP.

Note that we explain our terminology related to SCTP. *Path* is a combination of source and destination addresses. Due to the multihoming function of SCTP, one transport connection may contain multiple paths in general. Among theses paths, *primary path* is an actively chosen path for data transmission. The other paths are called *backup paths*.

B. Modifications to SCTP

Since the SCTP in RFC2960 provides only reliable data transmission, as does TCP, some modifications for supporting real-time communications are needed by way of implementing unreliable transmission mode. Moreover, to implement seamless handover, we modify mechanisms of failure detection and recovery. Here is a list of modifications to SCTP.

1) Disable the congestion control mechanism and the retransmission mechanism
2) Periodically send HEARTBEAT packets to the primary path
3) Change how to increase and reset an error counter
4) Change the sending interval of HEARTBEAT packets
5) Implement the multi-path transmission algorithm

The first, second and third modifications are for supporting real-time communications. Since the SCTP in RFC2960 is optimized for non-real-time communication, it has the congestion control mechanism and the retransmission mechanism. However, these two mechanisms are obstacles for real-time communications, so that disabling these two mechanisms is the first modification. After we have made the first modification, we also need the second, third and fourth modifications.

In RFC2960, there are two ways of failure detection: a data/acknowledgment packet, and a HEARTBEAT/HEARTBEAT Acknowledgment (HEARTBEAT-ACK) packet. Since we omitted the data packet retransmission mechanism in the first modification, our failure detection mechanism only relies on the HEARTBEAT mechanism. We next explain the HEARTBEAT mechanism.

Hosts periodically send HEARTBEAT packets to each other to test reachability. After the corresponding host receives the HEARTBEAT packet, it sends back a HEARTBEAT-ACK packet to the sender host. Since HEARTBEAT packets are only sent to the idle path in RFC2960, the primary path is not tested by the HEARTBEAT mechanism. Therefore, our modification is that hosts send HEARTBEAT packets to all paths. The second modification is for reachability testing of the primary path.

Moreover, hosts maintain parameters called error counters that are a metric of reachability for each path. In RFC2960, the error counter is increased by one when data sent through the path is timed out. On the other hand, the error counter is reset to zero when a host receives an acknowledgment packet. However, this condition cannot appropriately give changes of reachability for each path in wireless access network environment. Then, we change how to increase and reset an error counter. We describe our condition as follows.

- When a host sends a HEARTBEAT packet along the path, the error counter for the path is increased by one.
- When a host receives a HEARTBEAT-ACK packet along the path, the error counter for the path is reset to zero.

This is third modification.

For the fourth modification, we then describe our HEARTBEAT interval (H) calculation, which is given by

$$H = HB.Interval \times (1 + \delta) \quad (1)$$

where $HB.Interval$ is a constant, and δ is a random value, uniformly distributed between -0.5 and 0.5, to give a fluctuation of loads of computers and networks.

The fifth modification is our key proposal. We describe it in the following section.

C. Multi-path Transmission Algorithm

In this section, we describe our multi-path transmission algorithm for end-to-end seamless handover. In wireless access networks, there are many packet losses which degrade the quality of real-time communication, so that more redundant packet transmission is one possible implementation of seamless handover. However, redundant packet transmission consumes network resources; thus we need to reduce unnecessary packet transmissions as much as possible. In order to achieve these two contradictory goals, we provide two modes: a single-path transmission mode and a multi-path transmission mode. The mode will change according to packet loss occurrences of the path for seamless handover, with only a small consumption of network resources.

When the quality of the primary path is degraded significantly, the host switches its transmission mode to the multi-path transmission mode. In the multi-path transmission mode, the sender host sends the same packet to two paths simultaneously. One of these is the primary path, and the other is one of the backup paths. After either of the paths becomes stable, the sender chooses the stable path as the new primary path and switches back to single-path transmission.

Figs. 1 and 2 illustrate pseudo codes which represent a part of our algorithm in sending HEARTBEAT packets and receiving HEARTBEAT-ACK packets, respectively. The algorithm in sending HEARTBEAT packets is executed for each path at every interval of H that is calculated by Eq. (1). The timing of execution is different for each path. The algorithm in receiving HEARTBEAT-ACK packets is executed when a host receives a HEARTBEAT-ACK packet.

```
[Primary path]
  send(HB);
  ErrorCount++;
  if (ErrorCount > MT){
       mode = MultiPath;
  } else if (ErrorCount > PMR) {
       state = inactive;
  }

[Backup paths]
  send(HB);
  ErrorCount++;
  if(ErrorCount > PMR){
       state = inactive;
  }
```

Fig. 1. Pseudo code in sending HEARTBEAT packets

```
[All paths]
  receive(HBack);
  if(mode == SinglePath){
       ErrorCount = 0;
  }else if(mode == MultiPath){
       ErrorCount = 0;
       if(seqnum == HBack_seqnum - 1){
            StabilityCount++;
            if(StabilityCount > ST){
                 mode == SinglePath;
            }
       }else {
            StabilityCount = 0;
       }
  }
  seqnum = HBack_seqnum;
```

Fig. 2. Pseudo code in receiving HEARTBEAT-ACK packets

Our algorithm mainly uses an error counter as an indicator to switch to the multi-path transmission mode. As we described above, an error counter is a metric of packet loss occurrences for each path. There is a threshold for the error counter called $Path.Max.Retrans$ (PMR). PMR is a threshold to indicate that the primary path should be changed due to some network trouble. In addition to this threshold, we provide $Multi\text{-}path.Threshold$ (MT). MT is a threshold for the error counter to switch from the single-path transmission mode to the multi-path transmission mode. This process is shown in Fig. 1. Note that MT must be much smaller than PMR.

To switch back from to the single-path transmission mode, we provide a stability counter and $Stability.Threshold$ (ST) as a new counter and a new threshold respectively. The stability counter is used only in the multi-path transmission mode. The initial value of the stability counter is zero. The stability counter is increased by one when two consecutive HEARTBEAT packets are acknowledged. To confirm this, we provide slight modifications about HEARTBEAT and HEARTBEAT-ACK packets to incorporate sequence number. When the stability counter exceeds ST, which means the path becomes stable, then the host switches from the multi-path transmission mode to the single-path mode, shutting down the operation of the stability counter. The stability counter is maintained in both paths currently in use, choosing the better path when the mobile host switches to the single-path mode. The reset condition of the stability counter is when two consecutive HEARTBEAT-ACK packets cannot be received. This process is illustrated in Fig. 2.

IV. Simulation Experiments

In this section, we do some simulation experiments to get some basic evaluations. The base simulator used is the Network Simulator version 2 (NS-2) [14] with SCTP module [15]. We implement our algorithm as a modification of the base simulator. In Section IV-A, we give a description of our simulation model, including mobile scenario and network topology. We then show the simulation results of our algorithm in Section IV-B.

A. Simulation Setting

Since our objective is seamless handover, we consider the following scenario as shown in Fig. 3. Our scenario is that a mobile host has two different wireless network interfaces including IMT-2000 as a cellular service and IEEE 802.11b as a WLAN hotspot service. First, mobile host A is only within the IMT-2000 service area. After the simulation starts, it moves towards the WLAN Hotspot service area. Time lines of all the simulations follow Fig. 3. The simulation start time is 0s. Between 15s and 45s, mobile host A is within the overlapping area. Simulation ends at 60s.

We then show our network topology in Fig. 4. Mobile host A communicates with corresponding fixed host B. Both hosts have two network interfaces, IF1 and IF2, respectively. IF1 of mobile host A and IF1 of fixed host B form Path1. Similarly, IF2 of mobile host A and IF2 of fixed host B form Path2. Both paths are contained by one SCTP connection. The network capacity of IF1 of mobile host A is 384kb/s due to the assumption of IMT-2000. Similarly, that of IF2 is 11Mb/s. On the other hand, since both network interfaces of fixed host B are assumed to be Ethernet, the network capacity is 10Mb/s. The moving speed of mobile host A is walking velocity. When it goes into the overlapping area of two wireless access networks, the packet loss rate of IF1 increases linearly whereas that of IF2 decreases linearly as a function of time, as illustrated in Fig. 3. Our assumed of application is real-time video streaming at a rate of 300kb/s. The streaming is uni-directional from mobile host A to fixed host B.

B. Results

In this section, we give some simulation results including parameter selection and performance comparisons. First, we select parameters for SCTP with the Multi-path Transmission Algorithm (MTA). These parameters are selected according to requirements. We then compare performance between SCTP with MTA and SCTP without MTA.

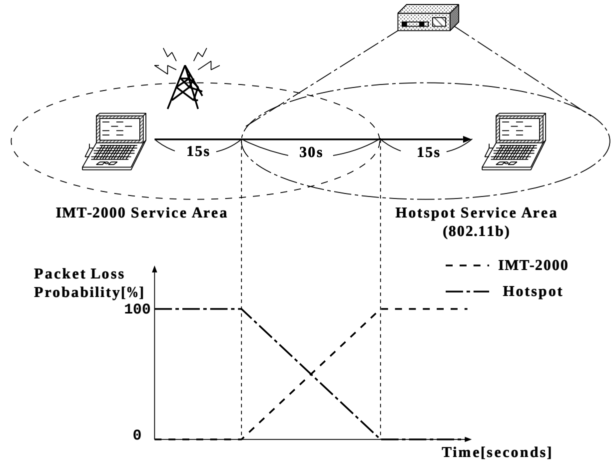


Fig. 3. Mobile scenario

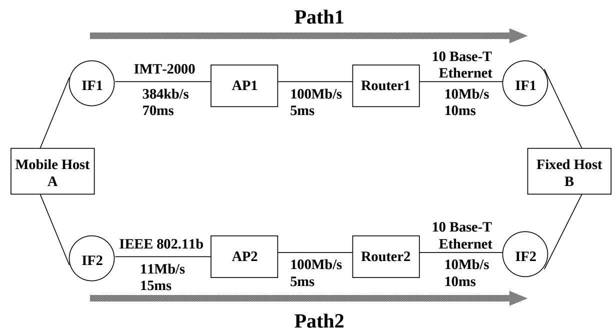


Fig. 4. Network topology

1) Parameters tuning of SCTP with MTA: As listed in Table I, SCTP with MTA has three parameters, $HB.Interval$, MT and ST. Moreover, it has three important performance metrics: goodput, total multi-path transmission period and communication overhead caused by data packets and/or HEARTBEAT packets. In this section, we investigate the relationship among these parameters and performance metrics, and discuss parameter selection.

We first would like to determine $HB.Interval$, to simplify the rest of the discussions. The communication overhead caused by HEARTBEAT packets can be calculated through a simple equation, which is given by

$$\text{Overhead} = \frac{\text{HEARTBEAT packet size} \times 8}{HB.Interval}, \qquad (2)$$

where the HEARTBEAT packet size is a constant 60 Bytes. Therefore, this metric is only affected by the $HB.Interval$. If we choose 0.1s as the $HB.Interval$, the communication overhead becomes 4.8kb/s, which seems not to be an acceptable value.

Fig. 5 shows goodput as a function of ST. The MT value is set to 3. Goodput is the total number of bytes received by the receiver when a mobile host is in the overlapping area. The purpose of this figure is to find a good value for the $HB.Interval$. We can see from the figure that a small value for the $HB.Interval$ gives us high goodput. For example, if we select 0.3s as $HB.Interval$ value, this results in a

TABLE I

SIMULATION PARAMETERS FOR SCTP WITH MTA

$HB.Interval$	0.1, 0.3, 0.5s
$Multi\text{-}path.Threshold$	1, 2, 3
$Stablility.Threshold$	1–10

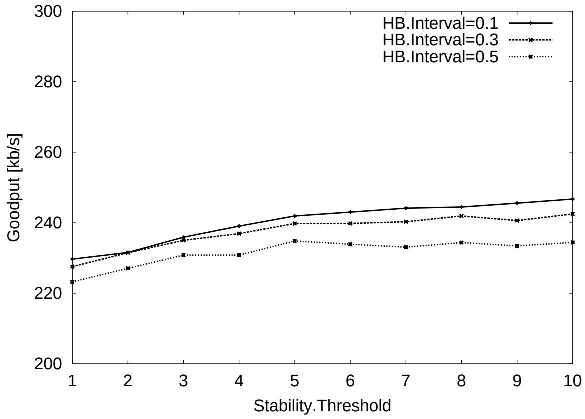


Fig. 5. Goodput versus $Stability.Threshold$ and $HB.Interval$

communication overhead of 1.6kb/s. This would be acceptable in the IMT-2000 environment at a rate of 384kb/s. Note that the simulation results listed in this section are average values from 100 experiments for each set of parameter values.

We then select values for MT and ST. First, we investigate the relationship between these parameters and performance metrics. Figs. 6 and 7 show goodput and total multi-path transmission period as a function of ST in setting the $HB.Interval$ to 0.3s. From Figs. 6 and 7, goodput is high when MT is set to 1, but the communication overhead is also high due to a long multi-path transmission period. Moreover, as ST increases, goodput becomes high, but the communication overhead also becomes high. We explain the reason for this relationship. If MT is small, the mobile host switches the transmission mode to multi-path as soon as the primary path becomes unstable. This makes goodput higher, but the communication overhead also increases. On the other hand, if ST is small, the mobile host switches the transmission mode to single-path as soon as one of the paths becomes stable even for a moment. This makes the communication overhead low, but also makes goodput low. Also, the number of mode changes increases in Fig. 8. That is, MT and ST should be determined in consideration of a tradeoff between goodput and the communication overhead. In this paper, we also focus on reducing the communication overhead. To decrease the communication overhead, we select 3 as MT.

We then try to set ST to reduce the communication overhead. For example, if we keep the total multi-path transmission period within 15 seconds, the range of ST from 1 to 9 satisfies the condition in Fig. 7. In another example, if we keep the total multi-path transmission period within 10 seconds, the range of ST from 1 to 4 satisfies the condition. To be high goodput, we select 4 under a condition of the total multi-path transmission period within 10s

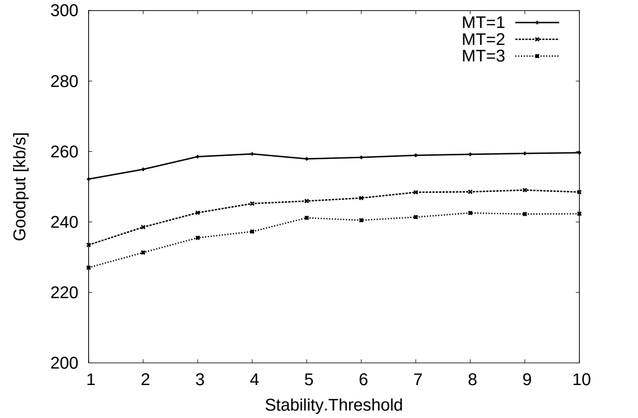


Fig. 6. Goodput versus $Stability.Threshold$ and $Multi\text{-}path.Threshold$

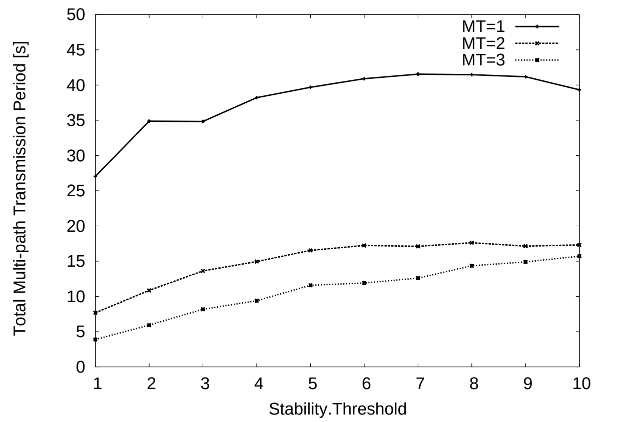


Fig. 7. The total multi-path transmission period versus $Stability.Threshold$ and $Multi\text{-}path.Threshold$

2) Comparisons: We compare goodput between SCTP with MTA and without MTA. Table II shows the parameter values for SCTP with MTA and without MTA. The parameter value for SCTP with MTA follows the results of parameter selection in the previous section. On the other hand, in SCTP without MTA, we similarly choose 0.3s as $HB.Interval$. Then, we choose 3 as PMR, which is a threshold to switch to a backup path, because MT in out algorithm achieves the function of PMR.

Figs. 9 and 10 illustrate throughput and goodput performance of SCTP with MTA and without MTA, respectively. Fig. 9(a) shows there is a time period when both paths are sending packets, while Fig. 10(a) does not have such a period. As a result, goodput performance of SCTP with MTA is better than that of SCTP without MTA. The average goodput of SCTP with MTA is 236.9kb/s, and that of SCTP without MTA is 218.5kb/s. The difference between these two values is about 18kb/s. One thing we would like to emphasize is that Fig. 10(b) has a drastically decreased goodput period less than 100kb/s. On the other hand, SCTP with MTA shows that goodput is kept high (i.e., more than 150kb/s) in almost all periods.

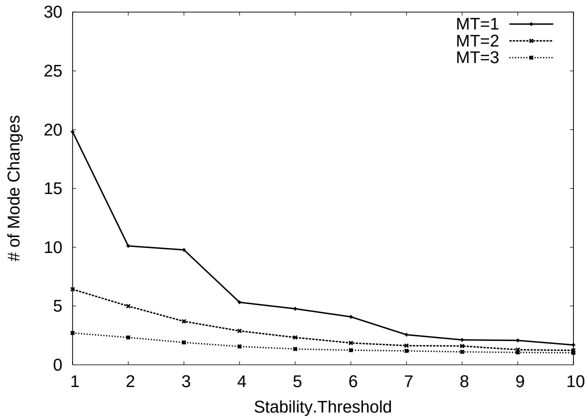


Fig. 8. The number of mode changes versus $Stability.Threshold$ and $Multi\text{-}path.Threshold$

TABLE II

SELECTED PARAMETER VALUES FOR SCTP WITH AND WITHOUT MTA

SCTP with MTA

$HB.Interval$	0.3
$Multi\text{-}path.Threshold$	3
$Stability.Threshold$	4

SCTP without MTA

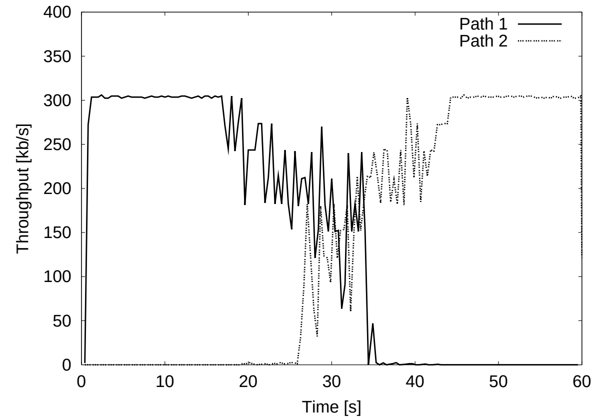
$HB.Interval$	0.3
$Path.Max.Retrans$	3

V. CONCLUSION

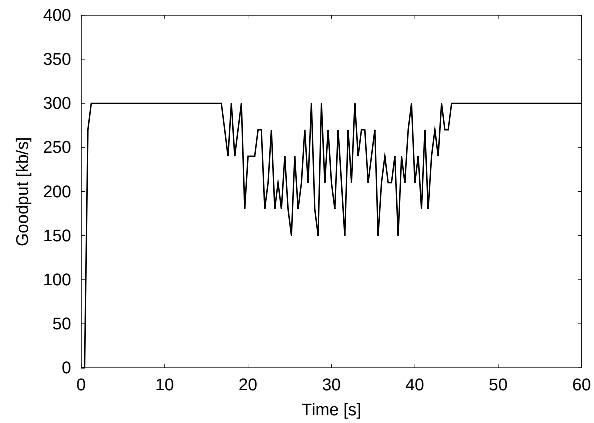
In this paper, we have proposed a multi-path transmission algorithm (MTA) for end-to-end seamless handover across heterogeneous wireless access networks. The aim of the MTA is to improve goodput during handover by an end-to-end assisting handover. To realize our algorithm, we have modified SCTP which can handle multiple network interfaces. In the MTA, there are two modes: single-path and multi-path transmission modes. If a mobile host detects that network condition of the primary path becomes unstable, the mode is switched to multi-path transmission mode to avoid quality degradation. In the multi-path transmission mode, the mobile host sends the same packets to the multiple paths simultaneously. Moreover, after the condition of one of the transmission paths becomes stable, the mode returns to single-path mode to reduce unnecessary consumption of network resources. Through simulations, we have tuned parameters of our algorithm, and derived a set of recommenced values. Finally, we have shown our multi-path transmission algorithm gives us high goodput with limited unnecessary consumption of network resources.

ACKNOWLEDGEMENT

The work of K. Iida was supported in part by grants from the Support Center for Advanced Telecommunications Technology Research (SCAT), Japan. The work of Y. Kadobayashi was supported in part by the Telecommunication Advancement Organization of Japan.



(a) Throughput



(b) Goodput

Fig. 9. Throughput and Goodput performance of SCTP with MTA

REFERENCES

[1] R. Stewart, et al., "Stream Control Transmission Protocol," *IETF RFC2960*, October 2000.

[2] R. Stewart and Q. Xie, *Stream Control Transmission Protocol (SCTP) A Reference Guide*, Addison-Wesley, 2001.

[3] C. Perkins, Ed., "IP mobility Support for IPv4," *IETF RFC 3344*, August 2002.

[4] D. Johnson, et al., "Mobility Support in IPv6," *IETF Internet-Draft*, draft-ietf-mobileip-ipv6-21.txt, February 2003.

[5] R. Ramjee, et al., "HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-area Wireless Networks," *Proc. IEEE ICNP'99*, October 1999.

[6] A. T. Cambell and J. Gomez, "IP Micro-Mobility Protocols," *ACM, Mobile Comp. and Commun. Rev.*, Vol. 4, No. 4, October 2001.

[7] A. Valkó, "Cellular IP: A New Approach to Internet Host Mobility," *ACM Comp. Commun. Rev.*, Vol. 29, No. 1, January 1999.

[8] E. Gustafsson, et al., "Mobile IPv4 Regional Registration," *IETF Internet-Draft*, draft-ietf-mobileip-reg-tunnel-07.txt, October 2002.

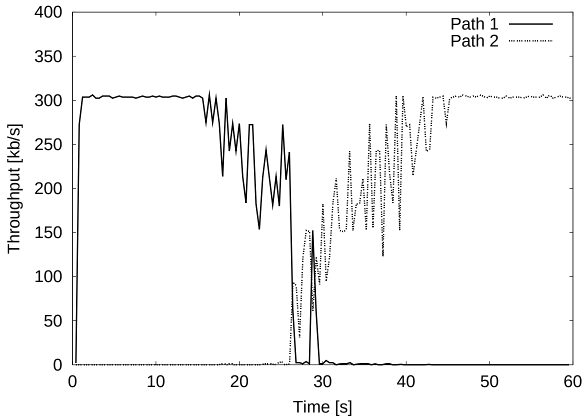
[9] A. C. Snoeren and H. Balakrishinan, "An end-to-end approach to host mobility, *Proc. 6th ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, August 2000.

[10] L. Magalhaes and R. Kravets, "MMTP: Multimedia Multiplexing Transport Protocol," *ACM first Workshop of Data Communications in Latin America and the Caribbean (SIGCOMM-LA 2001)*, April 2001.

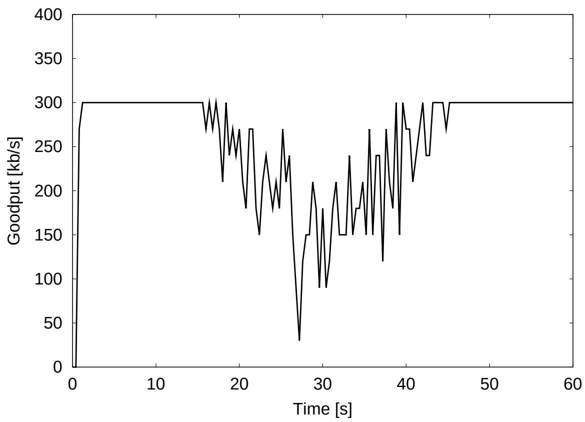
[11] M. Endler and V. Nagamuta, "General Approaches for Implementing Seamless Hnadover," *Proc. of ACM Principles of Mobile Computing (POMC 2002)*, Toulouse, pp. 17–24, October 2002.

[12] L. Taylor, et al., "The Challenges of Seamless Handover in Future Mobile Multimedia Networks," *IEEE Personal Communications*, 1999.

[13] R. Stewart, et al., "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration," *IETF Internet-Draft*, draft-ietf-tsvwg-addip-sctp-07.txt, February 2003.



(a) Throughput



(b) Goodput

Fig. 10. Throughput and Goodput performance of SCTP without MTA

[14] The Network Simulator – ns-2 –; `http://www.isi.edu/nsnam/ns/`.
[15] Protocol Engineering Lab; `http://www.pel.cis.undel.edu/`.