HMTP: Multi-path Transport Protocol for Multi-homing Wireless Erasure Networks

Younghak Hwang, Abhijit Saha, Hanbit Choi,
Hyuk Lim, and Brownson O. Obele

Abstract

This paper proposes a fountain-code-based multi-path transport protocol, called heterogeneous multi-path transport protocol (HMTP), and evaluates its performance. HMTP improves the throughput performance and path utilization of multi-homing wireless erasure networks. It solves the receive buffer blocking problem and eliminates the need for retransmissions and in-order packet delivery, both of which severely degrade the performance of existing multi-path transport protocols in multi-homing networks. The encoding and decoding algorithms of HMTP are based on efficient and computationally inexpensive coding algorithms such as Luby transform (LT) codes. To minimize the fountain encoding/decoding overhead, complexity, and computational cost while ensuring the desired decoding performance, the appropriate values of the parameters of the LT codes are determined. We implement and extensively evaluate the performance of HMTP using both NS-2 simulator and an 802.11 wireless local area network (WLAN) laboratory test-bed. Our simulations and experimental results show that HMTP significantly outperforms the concurrent multi-path transfer – stream control transmission protocol (CMT-SCTP), especially in multi-homing 802.11 WLANs.

Index Terms

Multi-homing, receiver blocking, fountain codes, wireless erasure channel.

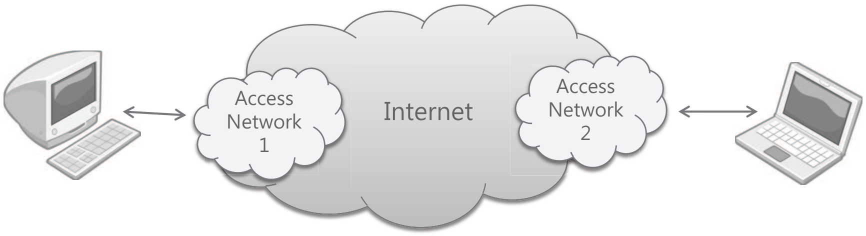
I. INTRODUCTION

Owing to recent advancements in hardware design and technology, Internet hosts such as PCs, notebooks, tablet PCs, and smart phones commonly have two or more heterogeneous network

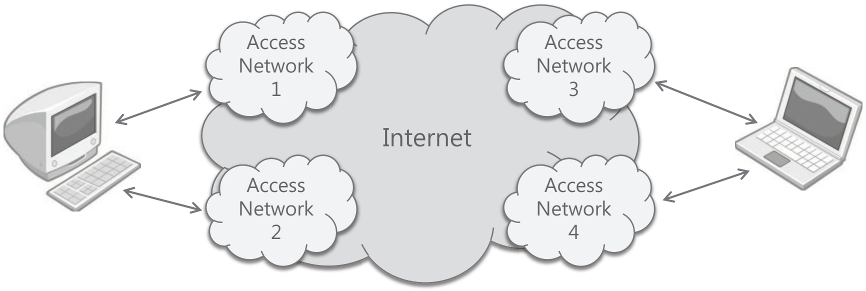
---

A preliminary version of this paper was presented in ACM CoNEXT Student Workshop 2010 [1].

The authors are with the School of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, Republic of Korea. E-mail: hlim@gist.ac.kr



(a) Single-homed network



(b) Multi-homed network

Fig. 1. Schematic diagram comparing a single-homed network and a multi-homed network.

interfaces such as Ethernet, Wi-Fi, WiMAX, 3G, Bluetooth, and GPRS. Moreover, if a host has only a single network interface, multiple additional interfaces can be easily and inexpensively added. Nonetheless, most of aforementioned hosts can only use one of their network interfaces at a time to connect to Internet services, as shown in Fig. 1(a). In fact, the performance of such hosts can be improved significantly if they can simultaneously establish network connections with two or more network interfaces, as shown in Fig. 1(b). This type of network is called a multi-homing network. Multi-homing networks can also provide spatial diversity and are particularly resilient to network path failures. Furthermore, an efficiently designed simultaneous multi-path transmission protocol for multi-homed hosts can significantly increase the network performance. Consequently, multi-homing and multi-path transports are attracting considerable interest from the research community [1]–[3].

It is widely believed that the number of wireless/mobile Internet hosts will exceed the number of wired hosts in the near future. Wireless multi-homing features multi-path diversity. In order to gainfully exploit multi-path diversity, a new multi-path transport protocol that can reliably and efficiently distribute data packets through multiple paths is strongly desired. Currently, however, most existing multi-path transport protocols are based on transmission control protocol (TCP) [2] and stream control transmission protocol (SCTP) [4], both of which rely on automatic repeat request (ARQ) schemes for reliable data transfer. Furthermore, ARQ schemes mandate retransmissions and in-order delivery of packets.

Under heterogeneous multi-path environments, the performance of ARQ-based transport protocols may be degraded because of the *“receive buffer blocking problem”* [5]. For example, a sequence of packets transmitted over multiple paths from a sender may arrive out-of-order at the receiver because of the heterogeneous characteristics of these paths. In such situations, the delivery of the received packets to the upper layers is delayed until the packets have been received in-order. Moreover, if the receive buffer becomes filled with out-of-order packets, newly arriving packets will be dropped, thereby further delaying the in-order delivery of received packets to the upper layer. The receive buffer blocking problem occurs more frequently when the size of the available receive buffer is smaller. It is worth noting that the number of data packets that the sender can simultaneously transmit is limited by the minimum value among `cwnd` and `rwnd`. The receive buffer blocking problem may severely degrade the network performance, especially when the packet loss characteristics and end-to-end delay differ among the multiple paths. The receive buffer blocking problem with asymmetric multiple paths was investigated in detail by Dreibholz *et al.* [6]. They showed that both the sender and receiver queue blocking problems existed in asymmetric paths, causing the multi-path transfer protocols to have poor performance.

In this paper, we consider a special class of wireless networks in which each host exchanges data packets with other hosts simultaneously through two or more network interfaces connected to different heterogeneous networks. The packet transmissions among hosts follow a simple communication channel model, called the ”*packet erasure channel,*” in which sequential packets are either received or lost. Digital fountain codes [7]–[10] are types of forward error correction (FEC) codes for erasure channels. For example, in vehicular ad hoc networks, a rateless coding approach was proposed for performing fast and efficient data dissemination through cooperating vehicles [11]. A fountain encoder continuously generates new encoded packets by the modulo-2 addition of randomly selected packets among k original packets. Irrespective of which encoded packets are collected and the order in which they are collected, if a sufficient number of encoded packets are collected, the fountain decoder can recover the original packets from any subset of k’ encoded packets with a probability of $(1-\delta)$, where k’ $= k(1+\varepsilon)$, and k’ is slightly greater

TABLE I

SUMMARY OF NOTATIONS

Notation	Description
S	Multihomed HMTP sender
D	Multihomed HMTP receiver
E_n	Encoded packet $n : n \in \{1, 2, ..., N\}$
I	Total number of network interfaces on the node
J	Number of network interfaces simultaneously in use by the node
k	Number of original packets at the sender
k’	Number of received encoded packets at the receiver
$\mathbb{S}$	Number of degree-1 encoded packets
L	Amount of load (application data) at S
R_j	Transmit rate on interface j
R_{eff}	Effective transmit rate of the node
δ	Decoding failure probability
ω	Decoding inefficiency

than k. The failure probability, δ, of decoding is bounded by $\delta \leq 2^{-\varepsilon k}$ and depends on the degree distribution used by the sender to encode the packets. Here, $(1 + \varepsilon)$ is a small positive real number called the *decoding inefficiency* [7]–[9]. Table I shows a summary of the notations used in this paper.

In this paper, we propose a heterogeneous multi-path transport protocol (HMTP), which is an adaptive fountain-code-based multi-path transport protocol that efficiently improves the throughput performance and path utilization of multi-homing networks. HMTP solves the receive buffer blocking problem and eliminates the need for retransmissions and in-order packet delivery. The NS-2 simulator and an 802.11 WLAN laboratory test-bed were used to implement and extensively evaluate the HMTP performance. The simulation and experimental results show that the overhead and computational cost of HMTP are reasonable and that HMTP significantly outperforms the existing transport protocol, i.e., the concurrent multi-path transfer – stream control transmission protocol (CMT-SCTP) [12], especially in multi-homed 802.11 WLANs.

The remainder of this paper is organized as follows. Section II discusses related works. Section III presents the HMTP protocol and its operational details. Section IV presents the results of extensive NS-2 simulations for various multi-homing heterogeneous networks. Section V presents the results of the network-emulation-based test-bed experiments, while Section VI summarizes and concludes this paper.

II. RELATED WORK

Recently, multi-path transmission has attracted considerable interest. However, many issues such as building a new transport protocol for multi-path transmission still exist. For example, the standard TCP does not support multi-homing. Therefore, new multi-path transport protocols have been introduced by modifying TCP. In [2], Huitema proposed the first multi-homed TCP. Subsequently, many other researchers proposed new multi-path transport mechanisms by modifying the standard TCP. Hsieh *et al.* proposed parallel TCP (pTCP) [13], which allows multiple connections to achieve the bandwidth aggregation offered by multiple paths irrespective of the individual characteristics of the paths. Therefore, pTCP could not achieve multi-path load balancing. Dong *et al.* introduced concurrent TCP (CTCP) [14] by extending the standard TCP; this protocol was implemented in the FreeBSD kernel. Multi-path load balancing was achieved in the transport layer, and CTCP was backward compatible with the standard TCP [15]. Rojviboonchai *et al.* proposed multi-path TCP (M/TCP) [16] based on the current performance of the Internet. The M/TCP, which is an alternative to the standard TCP, was designed to improve the reliability and performance of the Internet to support multi-path transports. Zhang *et al.* proposed a mobile TCP (mTCP) [17], i.e., an end-to-end transport layer protocol that can efficiently aggregate the bandwidth available on several paths in parallel. By striping the packets of the flow across multiple paths, mTCP can not only realize higher end-to-end throughput but also be more robust under path failure.

Some studies have focused on the multi-path transport scheme implemented in SCTP [18], which is a new transport layer protocol that has been proposed to overcome the problems inherent to TCP and user datagram protocol (UDP). SCTP performs functions similar to TCP, such as congestion control and reliable data transmission, and also performs some additional functions such as multi-streaming, multi-homing, and four-way handshakes. The multi-homing scheme enables the establishment of several network-layer connections between two end hosts via a single SCTP association. Some researchers believe that the current SCTP packet format already contains sufficient information for the data source to distinguish between all used paths. Al *et al.* proposed a load sharing SCTP (LS-SCTP) [19], which uses an independent sequence number for each path. Liao *et al.* proposed a concurrent multi-path SCTP (cmpSCTP) [20], which uses several new mechanisms including a multi-buffer structure, multi-state management, two-level sequence numbers, and a cooperative selective acknowledgement (SACK) strategy to achieve effective bandwidth aggregation. Ye *et al.* introduced the independent per-path congestion control SCTP (IPCC-SCTP) [21], which achieves per-path congestion control without any modifications to the packet format of SCTP by implementing an implicit path sequence number.

Multi-path data delivery has been actively discussed by the Internet engineering task force (IETF), in the form of the multi-path TCP (MPTCP) [22]–[24] extension to TCP and the CMT [12] extension to SCTP (CMT-SCTP). MPTCP can simultaneously deliver TCP packets over multiple paths and aggregate the available bandwidth of the paths. As the result, MPTCP is able to achieve a higher aggregate throughput performance by efficiently exploiting the available bandwidth of multiple paths. However, it was noted that the goodput of MPTCP was lower than the aggregate throughput due to out-of-order received packets. Out-of-order packets at the receiver may also cause a large variation in the end-to-end delay for multiple paths. CMT-SCTP was proposed to overcome the problems caused by using a unique sequence-number space for data transfers occurring concurrently over multiple paths. In [25], Iyengar *et al.* proposed five retransmission policies for CMT-SCTP in order to take into account the various multi-path characteristics for the retransmission over multiple paths. Further research on the effect of receiver buffer size and bottleneck queues on the end-to-end paths has been performed [5], [12]. The problems caused by the out-of-order received packets and the retransmission over multiple paths can be efficiently resolved by introducing digital fountain codes into the multipath transport protocols.

Recently, fountain codes-based multipath protocols have been proposed. A preliminary version of the proposed HMTP was presented in [1]. In comparison with the preliminary version, this paper includes a substantive extension as follows:

- Recent related work has been concisely summarized.
- A comprehensive description of HMTP with an illustrative flowchart and detailed explanation is provided.
- The selection of key parameters for HMTP is investigated.
- HMTP has been implemented in a Linux kernel and evaluated in heterogeneous network topologies by using a network emulator.

In [1], Hwang *et al.* revealed that the problems caused by the out-of-order received packets and retransmission over multiple paths can be efficiently resolved by introducing digital fountain codes into the multi-path transport protocols. Later, another fountain code-based multipath transmission control protocol (FMTCP) was proposed by Cui *et al.* [26]. FMTCP uses a fountain code to encode the transmission data, and uses the advantage of the random coding scheme to avoid the performance degradation due to retransmissions in the multi-path transport protocols such as MPTCP [22]. While FMTCP uses a similar approach to HMTP, FMTCP focuses more on the data allocation policy, which determines an appropriate path (subflow) among multiple paths to transmit a new packet. This policy affects the transmission efficiency and overall decoding performance. Under FMTCP, each sender uses the estimated path quality and knowledge of the number of received symbols as the policy input. Specifically, FMTCP estimates the expected delivery time (EDT) and expected arrival time (EAT). The EDT is used to measure the overall quality of the paths, while the EAT is used to compare the transmission time of all subflows as well as to determine the subflow allocation. Theoretical analyses and simulation results have shown that FMTCP outperforms MPTCP.

Conversely, HMTP depends on the underlying transport protocol for determining how much traffic will be transmitted over each path rather than using an active allocation scheme because the receiver can successfully decode the sent chunk if only a sufficient number of packets are received. Interestingly, Cui *el al.* [26] claimed that HMTP was a stop-and-wait protocol, and FMTCP introduced a prediction mechanism for more efficient data delivery. However, HMTP does not adopt a stop-and-wait protocol at the transport layer. The transport protocol for encoded packets is a pipelined reliable data transfer protocol such as TCP or SCTP. At the application layer, HMTP uses an acknowledgement scheme for each chunk. However, it can concurrently start to encode and transmit packets for the next available chunk without waiting for the acknowledgement of the current chunk in order to fully exploit the path capacity in the network.

III. HETEROGENEOUS MULTI-PATH TRANSPORT PROTOCOL (HMTP)

The proposed HMTP protocol consists of a fountain layer and transport layer. The transport layer is wrapped with the fountain layer. As illustrated in Fig. 2, a multi-homed sender accepts k original data packets from a data stream of the application layer and then encodes them into independent fountain-encoded packets. These packets are then concurrently transmitted over multiple

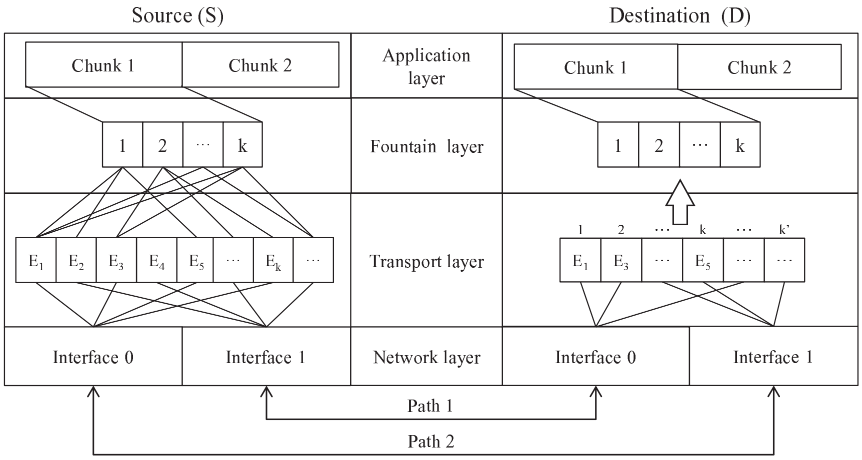


Fig. 2. Proposed multipath transport protocol.

heterogeneous paths. Because the packets are transmitted in-order over the heterogeneous paths, some packets may arrive at the receiver out-of-order and others may be missing because of the heterogeneous path characteristics, i.e., the loss rate and end-to-end delay. For the existing multi-path protocols, these out-of-order packets cannot be delivered to the application layer until the lost packets are retransmitted and the in-order set of packets is received. This results in increased delays and the receive buffer blocking problem. In HMTP, however, received fountain-encoded packets are delivered immediately to the upper layer irrespective of their arrival order and decoded into original packets. This approach eliminates the need for retransmissions and in-order delivery of packets because the fountain-encoded packets need not arrive in order and the loss of some such packets is tolerated. Therefore, the proposed HMTP effectively resolves the issue of managing out-of-order packets and eliminates the receive buffer blocking problem.

More specifically, irrespective of which encoded packets are received and the order in which they are received, an HMTP receiver can recover the k original packets when k’ encoded packets are received, with a probability $(1-\delta)$, where k’ $= k(1+\varepsilon)$ and k’ is slightly greater than k. The decoding failure probability, δ, is bounded by $\delta \leq 2^{-\varepsilon k}$ and depends on the degree distribution used by the sender to encode the packets. In addition, it is worth noting that the coding rate of HMTP is highly dependent on the number of packets k used for the fountain code. If k is reduced, then ϵ should be increased to maintain a certain level of δ, thereby resulting in higher coding rate. Therefore, the HMTP algorithm achieves better coding efficiency when a large data chunk is available for transmission. This is because if the size of data chunk increases, k increases. This implies that HMTP is more appropriate for achieving a reliable and efficient performance for data network applications with a sufficiently large amount of data, such as file downloading and progressive video streaming over heterogeneous multi-path networks.

A. Basic Operational Procedure of HMTP

Figure 3 shows the HMTP flowchart. The gray and white boxes indicate the operations of the sender and receiver, respectively. The basic HMTP operational procedure is as follows:

(1) At the multi-homed sender, a continuous stream of new data from the application layer is divided into chunks of size ($k \times L$), where k is the number of packets in the chunk and L is the length or size of each packet. If k is too large, a long delay may occur before the encoding process starts. Therefore, if k packets are not available from the application layer within a certain time interval, then the HMTP has to be implemented to forcedly start the encoding process.

(2) The original packets in the chunk are encoded into independent fountain-encoded packets by the modulo-2 addition of packets randomly chosen from among the k original packets that make up that chunk.

(3) The resulting fountain-encoded packets are assigned a fountain header that includes the chunk ID and the seed for the pseudo-random number generator. This seed will be used by the decoder at the receiver end to generate the same random sequence. The fountain-encoded packets are then delivered to the transport layer for heterogeneous multi-path transfer to the receiver.

(4) In the transport layer, the fountain-encoded packets are assigned sequence numbers (SNs) or transmission sequence numbers (TSNs) if the underlying transport protocol is TCP or SCTP, respectively. Thereafter, they are enqueued in the transmission buffers and then continuously transmitted over multiple paths to the receiver.

(5) After transmitting k' fountain-encoded packets, the sender waits for feedback concerning the number of correctly received fountain-encoded packets. At the same time, if the next chunk is available for transmission, the sender starts the abovementioned procedure for the next chunk.

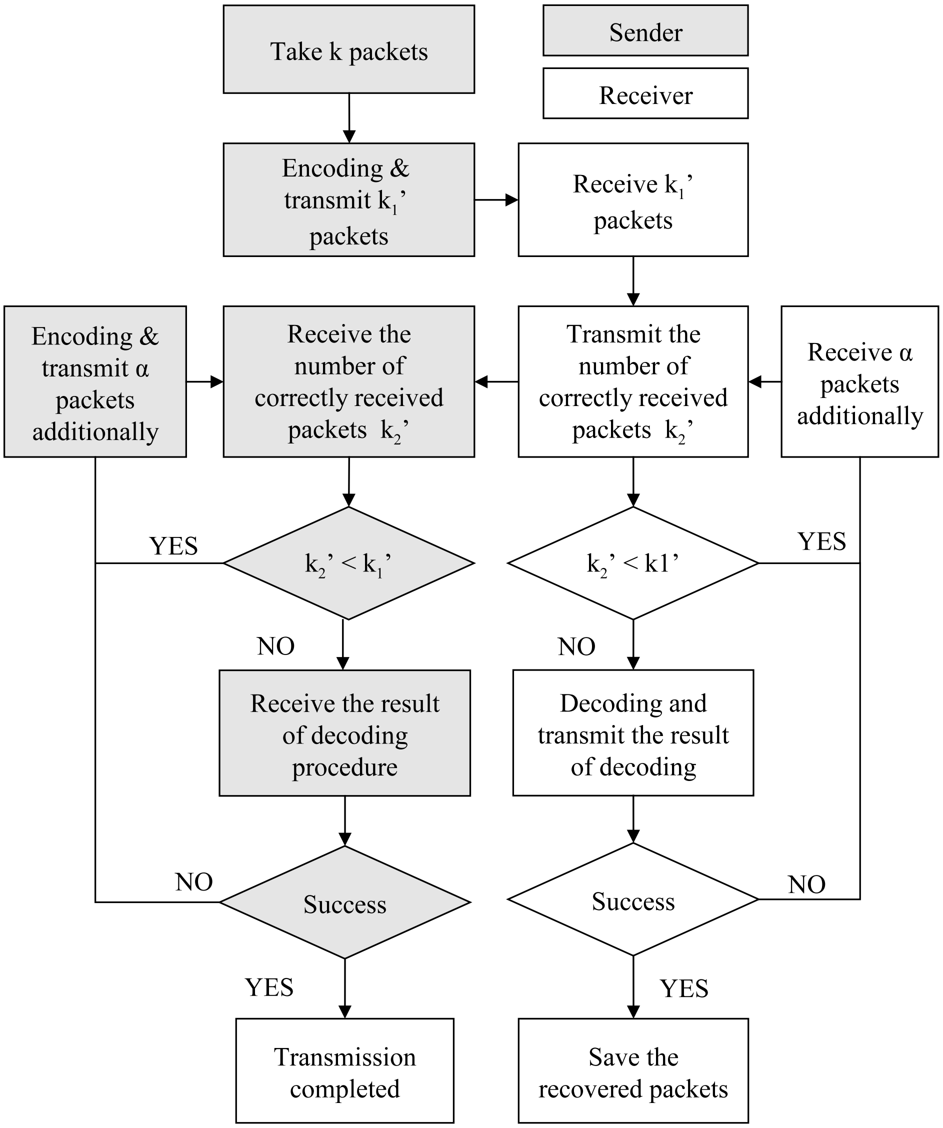


Fig. 3. Flowchart of the proposed multi-path transport protocol.

(6) If the sender receives feedback that the number of correctly received packets is smaller than k', it additionally transmits α packets. Otherwise, it continues the procedure for the next chunk.

(7) At the receiver, the received fountain-encoded packets are delivered to the fountain layer irrespective of their arrival order. Then, the receiver transmits the feedback concerning the number of correctly received fountain-encoded packets until k' fountain-encoded packets are correctly received.

(8) If k' fountain-encoded packets are correctly received, the decoding procedure begins. Then, the result of the decoding procedure is transmitted to the sender. Although $k(1+\varepsilon)$ packets are correctly received, the decoding process can fail with a probability of δ.

(9) If the receiver successfully recovers the k original packets, they are delivered to the application layer. Otherwise, the receiver waits for the additional fountain-encoded packets for the current chunk.

The proposed HMTP can be easily implemented by disabling the retransmission for the in-order delivery, which is required for the ARQ schemes of TCP and SCTP. In order to disable the retransmission, we modified the ARQ scheme of the existing transport layer as follows: at the transport layer of the sender, the fountain-encoded packets delivered from the fountain layer are assigned SNs or TSNs before they are transmitted over multiple paths. The newly generated fountain-encoded packets are assigned new SNs or TSNs in sequence. However, when 3 duplicate acknowledgements (ACKs) arrive, i.e., packet loss occurs, the SN or TSN of the lost packet is assigned to the newly generated fountain-encoded packet and not to a specific lost fountain-encoded packet because the proposed algorithm need not retransmit the specific lost packet. In most retransmission algorithms, a sender keeps as yet unacknowledged packets to respond to the retransmission request. However, in HMTP, the sender need not retain the fountain-encoded packets for retransmission.

At the receiver, the received fountain-encoded packets are delivered to the fountain layer irrespective of their arrival order. In other protocols such as TCP, when packets are received out-of-order because of the lost packets, the receiver sends a duplicate ACK to inform the sender of the packet loss and waits for the retransmission of the specific lost packet. In this case, the received packets cannot be delivered to the upper layer. While the receiver waits for the retransmission of the lost packet, the receiver buffer is filled with the newly received packets. If the receiver buffer is filled with out-of-order packets, the newly arrived packets are dropped. Therefore, the overall throughput decreases. However, in HMTP, the receiver need not wait for the retransmission of a specific lost fountain-encoded packet. It simply delivers all received fountain-encoded packets to the fountain layer irrespective of their arrival order. Therefore, the receiver buffer is always available, and the receive buffer blocking problem does not occur.

As briefly discussed at the end of Section II, HMTP does not work in a stop-and-wait manner. The flowchart in Fig. 3 shows the procedures of the transmission for a single chunk. After transmitting k' packets, it has to wait for the feedback from the receiver, similar to a stop-and-wait manner. However, this is a waste of the network path utilization. Therefore, HMTP immediately starts a new transmission for the next chunk if it is available from the upper layer, as explained in procedure **(5)**. Using this pipelined approach, HMTP does not have idle time wasted in waiting for the feedback from the receiver.

B. Parameter Selection for HMTP

The multi-homed sender selects the appropriate parameters such as k, c, and δ for fully utilizing the multiple paths, minimizing the encoding/decoding overhead, and maintaining the encoding/decoding performance at a predetermined level higher than its maximum effective transmit rate. Then, the sender continuously generates fountain-encoded packets by the modulo-2 addition of a number of randomly selected packets from k original packets and transmits them to the receiver. The number of random packets is determined by the selected probability (degree) distribution. This also influences the encoding/decoding process. Therefore, selecting a suitable probability is very important.

For an encoder of Luby transform (LT) codes, each fountain-encoded packet is generated by the modulo-2 addition of d randomly selected original packets. The degree, d, is randomly selected from a degree distribution such as an idle or robust soliton distribution. For an LT code, the idle soliton distribution $\rho(d)$ is defined as follows [7], [8].

- $\rho(1) = 1/k$
- For all $\quad d = 2, \ldots, k, \quad \rho(d) = 1/d(d-1)$

(1)

Here, the expected degree of an encoding packet is $\sum_{d=1}^{k} d/d(d-1) = H(k)$, where $H(k) \approx \ln(k)$ is the harmonic sum up to the number of original packets k. However, in practice, the idle soliton distribution works poorly because the expected number of encoded packets encoded using only 1 original packet, i.e., degree-1 packet, is too small. Therefore, the robust soliton distribution was developed to improve the decoding performance of LT codes. In this distribution, a new parameter $\mathbb{S}(\delta, k, c)$ is defined as

$$\mathbb{S}(\delta, k, c) = c \ln\left(\frac{k}{\delta}\right) \times \sqrt{k}, \tag{2}$$

where δ is the desired decoding failure probability at the receiver, and c is a suitable positive constant that has been found to significantly affect the code performance. By definition, $\mathbb{S}(\delta, k, c)$

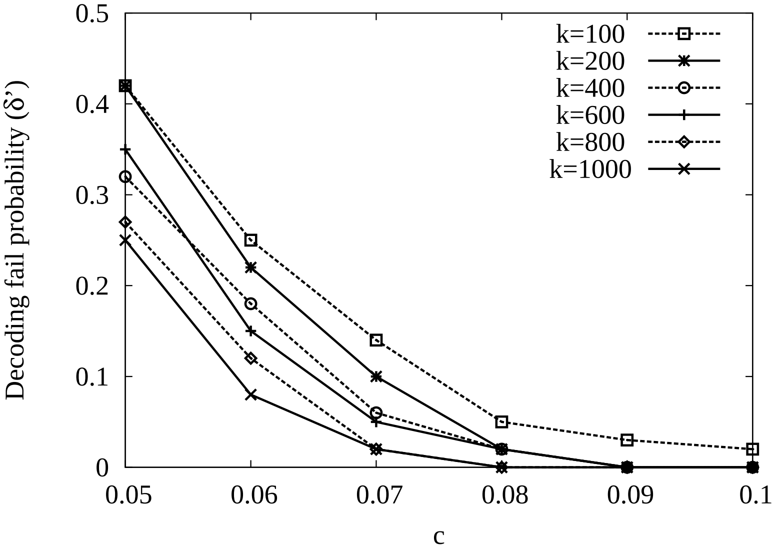


Fig. 4. Real decoding failure probability (δ') as a function of c when δ = 0.03.

is the number of encoded packets that were encoded using only 1 original packet (degree-1 packet). The positive function $\tau(d)$ is defined as

$$\tau(d) = \begin{cases} \mathbb{S}/dk & \text{for} \quad d = 1, \ldots, (k/\mathbb{S}) - 1 \\ \mathbb{S}\ln(\mathbb{S}/\delta)/k & \text{for} \quad d = k/\mathbb{S} \\ 0 & \text{for} \quad d = k/\mathbb{S} + 1, \ldots, k \end{cases} \tag{3}$$

The idle soliton distribution $\rho(d)$ is added to $\tau(d)$ and normalized to obtain the robust soliton distribution, $\mu(d)$:

- $\beta = \sum_{d=1}^{k} \rho(d) + \tau(d)$
- For all $\quad d = 1, \ldots, k, \quad \mu(d) = (\rho(d) + \tau(d))/\beta$

(4)

The robust soliton distribution ensures that the number of degree-1 packets is $\mathbb{S}$. Therefore, it can improve the decoding performance. However, the performance of the robust soliton distribution may be poor when the value of c is too small. The number of degree-1 packets plays a very prominent role in the success of the decoding procedure [8]. When k and δ are fixed, the value of $\mathbb{S}$ becomes a function of c. Fig. 4 shows the real decoding failure probability (δ') with respect to the value of c obtained from our implementation of the HMTP encoding/decoding algorithms. In the implementation, we used the message passing algorithm in our decoder. As shown in Fig. 4, when the value of c is very small, the real decoding failure probability obtained from the implementation is higher than the desired decoding failure probability. This is because for small values of c, the number of degree-1 packets is less than what is required for the success of the decoding procedure at the desired decoding failure probability. It has been shown that if there are no degree-1 packets, the decoding procedure fails with a probability of 1 [8]. However, when the value of c is 0.07, it is observed that $\delta' \simeq \delta$ for k = 800 and 1000. Consequently, it is important to properly set the value of c such that the desired decoding failure probability can be achieved. Fig. 4 also shows that the appropriate value of c that makes $\delta' \leq \delta$ changes according to the value of k. The value of k is determined by the size of the original data chunk. This implies that if the size of the original data chunk changes, we have to change the value of c according to k. Through extensive simulations, we have obtained the appropriate values for these parameters and tabulated them for an adaptive selection.

IV. PERFORMANCE EVALUATION

In order to demonstrate the effectiveness of HMTP, we have conducted extensive simulations using the NS-2 simulator [27] and compared its performance against that of CMT-SCTP. For the CMT-SCTP implementation, we used RTX-CWND as the retransmission policy because it was reported to be the best retransmission policy among the five policies recommended for CMT-SCTP [5]. In the RTX-CWND, lost packets are retransmitted to the destination for which the sender has the largest CWND. If there is a tie, it is broken randomly. For HMTP, the parameters δ and k for the fountain layer are set as 0.03 and 1000, respectively. Under this configuration, once the receiver receives 1137 fountain-encoded packets, it can reconstruct 1000 original packets. In the simulations, we have evaluated the performances of HMTP and CMT-SCTP in three different multi-homing environments.

A. Performance of HMTP in simple heterogeneous multi-homing networks

First, we have conducted a set of simulations for a simple heterogeneous multi-homing network, as shown in Fig. 5 to show how HMTP solves the receive buffer blocking problem. The bandwidth was 1 Mb/s for both paths. We defined heterogeneous path environments with different end-to-end delays and loss rates. While the end-to-end delay and loss rate on path 1

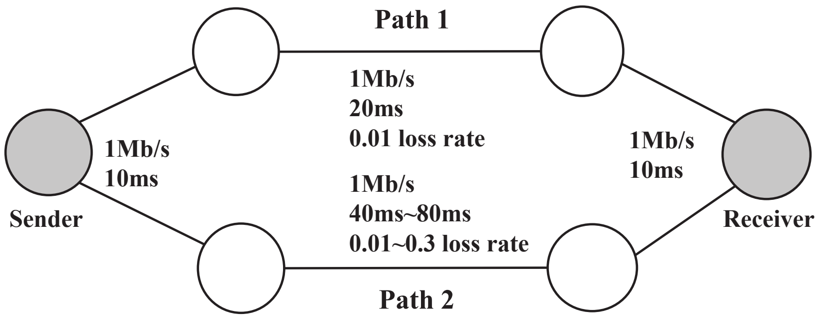
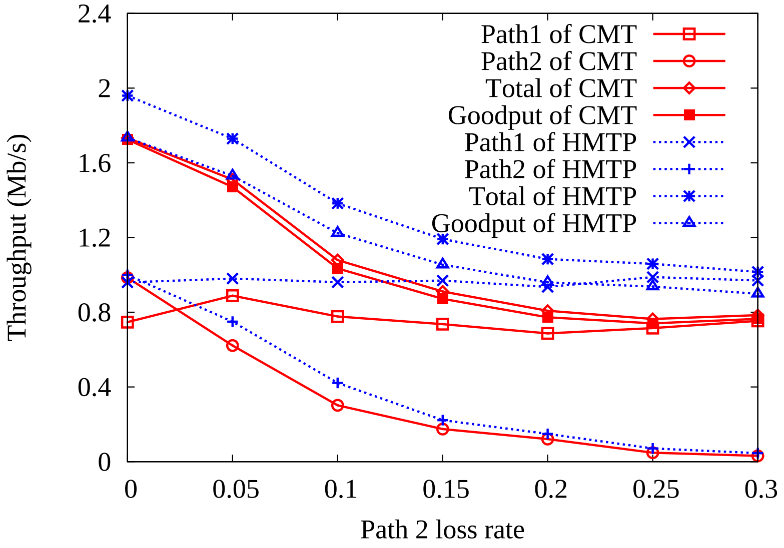


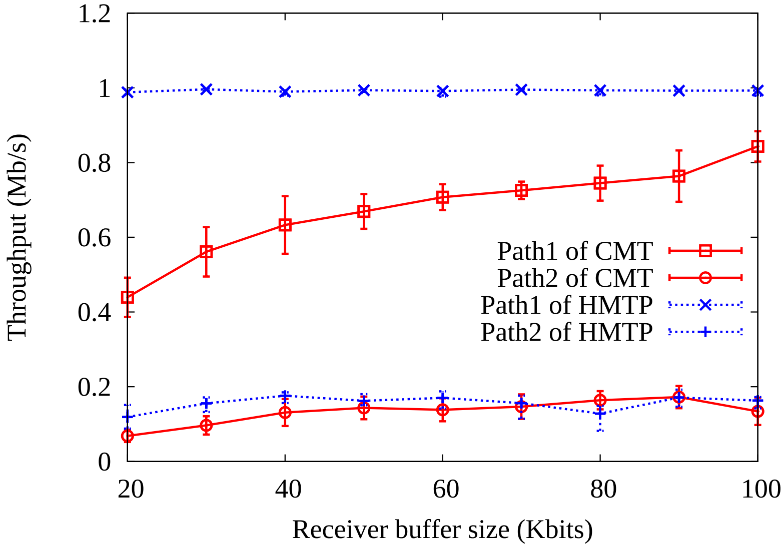
Fig. 5. Simple heterogeneous network topology.

were kept constant, those on path 2 varied from 40 to 80 ms and 0.01 to 0.3, respectively. We used the file transfer protocol (FTP) application module in the NS-2 simulator to generate data traffic. Each simulation ran for 100 seconds. The maximum amount of data traffic generated during each simulation was 200 Mb because the sender was connected to 2 links with a bandwidth of 1 Mb/s.

We compared the throughput performances of CMT-SCTP and HMTP when the loss rate of path 1 was fixed at 0.01 and that of path 2 varied from 0.01 to 0.3. Fig. 6(a) shows the throughput and goodput of CMT-SCTP and HMTP, when the end-to-end delay for both paths is 40 ms. Each point in Fig. 6(a) is the average value for 10 simulations. The goodput is the throughput measured at the application layer and does not count the packet transmissions used for lost packet retransmissions. For CMT-SCTP, the throughput of path 1 decreases as the loss rate of path 2 increases, as shown in Fig. 6(a). When the loss rate of path 2 is high, although the loss rate of path 1 is fixed, the throughput of path 1 decreases because of the receive buffer blocking caused by many lost packets on path 2. In contrast, for HMTP, although the loss rate of path 2 increases, the throughput of path 1 is maintained nearly constant at 1 Mb/s, as shown in Fig. 6(a). This implies that the throughput of path 1 for HMTP is not affected by the loss rate of path 2. As a result, HMTP realizes multiple completely independent paths because in the proposed protocol, the receiver need not wait for the retransmission of the loss packets and delivered the received packets to the fountain layer irrespective of their arrival order. In order to compare the actual HMTP performance, we have also compared the goodput of both protocols in the application layer, as shown in Fig. 6(a). The HMTP performance is better than that of CMT-SCTP when the loss rates of both paths are different. The discrepancy of goodput between



(a) Throughput of CMT-SCTP and HMTP



(b) Throughput with respect to receiver buffer size

Fig. 6. Simulation results for the simple topology with a receive buffer = 64 kb, loss rate = 0.01 for path 1, and end-to-end delays = 40 ms for both paths.

HMTP and CMT-SCTP increases as the loss rate of path 2 increases because of more frequent receive buffer blockings.

In order to further investigate the receive buffer blocking problem over multiple paths, we measured the throughput performance with respect to the receive buffer size when the loss rates of paths 1 and 2 were 0.01 and 0.2, respectively. As shown in Fig. 6(b), the throughput of path 1 was much higher than that of path 2 because the loss rate of path 1 was much lower than that of path 2. In Fig. 6(b), the vertical line indicates the 95 % confidence interval of each experiment. The variance of simulation results for CMT was larger than that for HMTP because the performance of CMT was more susceptible to the link delay changes in the topology. For CMT-SCTP, the receive buffer blocking problem was more severe for a small size buffer and became mitigated as the receive buffer size increases. We argue that the heterogeneous characteristics of multiple paths have a negative influence on the performance of multi-path transfer. In other words, the receive buffer blocking problem becomes worse when a path with a high loss rate or long end-to-end delay is used for multi-path transfer. Conversely, HMTP consistently achieves a high throughput performance regardless of the buffer size.

B. Performance of HMTP in complex heterogeneous multi-homing networks

In order to evaluate the performance of HMTP in complex and heterogeneous multi-homing networks, we generated a complex topology using GT-ITM in NS-2 [27]. The topology is shown in Fig. 7. There were 28 nodes and 56 links. The bandwidth and delay of last-hop links connected to the sender and receiver were set as 1 Mb/s and 10 ms, respectively. The other links had a bandwidth of 10 Mb/s and delay of 10 ms. In the topology, nodes 3 and 0 were the sender and receiver, respectively. We defined paths 1 and 2 as the paths passing through nodes 1 and 2, respectively. The total loss rate on path 1 was maintained at 0.01, whereas that on path 2 varied from 0.01 to 0.3. Here, we used the FTP application module in the NS-2 simulator to generate data traffic, and each simulation ran for 100 seconds.

Fig. 8(a) shows the throughput and goodput of CMT-SCTP and HMTP in the complex and heterogeneous multi-homing networks. As in the case of the simple topology, in HMTP, the throughput of path 1 is maintained at 0.92 Mb/s while the loss rate of path 2 increases because there is no receive buffer blocking. Fig. 8(b) shows the delay measured at the application layer of CMT-SCTP and HMTP. Note that the delay measurement includes the encoding and decoding

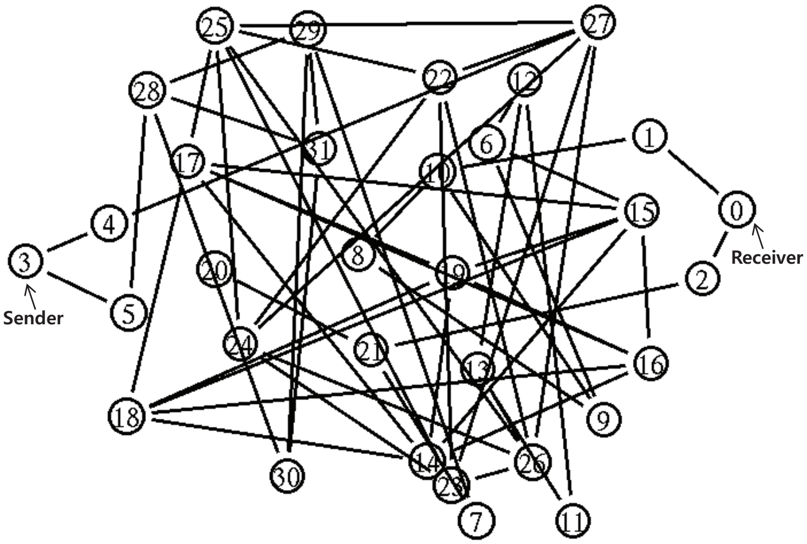
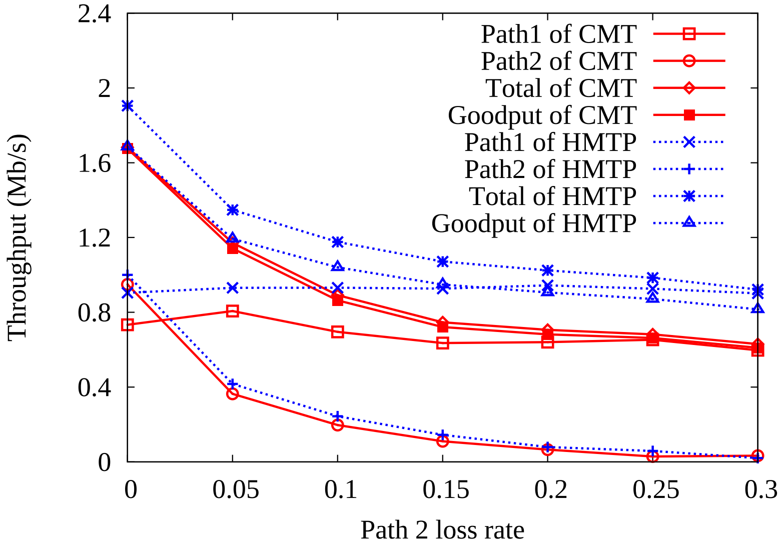


Fig. 7. Complex heterogeneous network topology.

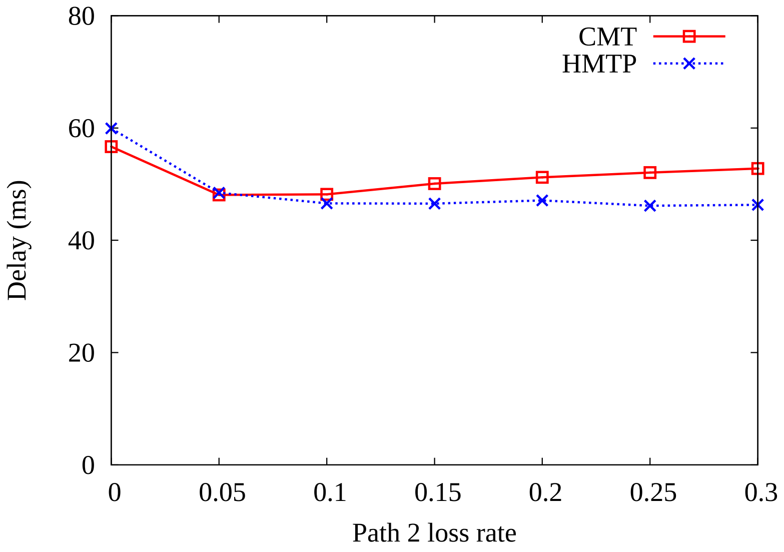
computing time, application layer waiting time, and network transmission delay. When the loss rate of path 2 is zero, HMTP incurs a longer delay because of the delay for encoding/decoding packets and the transmission of $(k' - k)$ more packets. This implies that an overhead for the encoding/decoding packets in HMTP exists, but it is negligible (less than 10% in comparison with CMT-SCTP when the loss rate of path 2 is zero). On the contrary, as shown in Fig. 8(b), the delay for CMT-SCTP increases as the loss rate of the path 2 increases, while the delay for HMTP is maintained constant.

C. Performance of HMTP in multihomed wired and wireless interconnected networks

Fig. 9 shows a multi-homed wired and wireless interconnected network consisting of a multi-homed mobile host, many single-homed mobile hosts, and two APs (BS1 and BS2). The multi-homed sender has two wireless network interfaces and it is independently associated with the 2 APs using its multiple interfaces. Three single-homed mobile hosts are associated with AP-2 (BS2). These single-homed hosts use TCP as the transport protocol. When the multi-homed sender associates with 2 APs, if the environment of each AP and the condition of each wireless link are heterogeneous, the transmit rate, which is assigned by each AP, would be different because the AP decides the transmit rate of each client by considering various factors such as the signal to interference plus noise ratio (SINR) between the AP and each host as well as the



(a) Throughput of CMT-SCTP and HMTP



(b) Delay of CMT-SCTP and HMTP

Fig. 8. Simulation results for the complex topology with receive buffer = 64 kb and loss rate = 0.01 for path 1.

number of clients associated with the AP. The loss rate and end-to-end delay of each path would also be different because the SINR between the multi-homed host and each AP as well as the number of clients associated with each AP are different. In order to prevent collisions between the 2 links established by each AP, we assume that each link uses an orthogonal channel. Therefore,

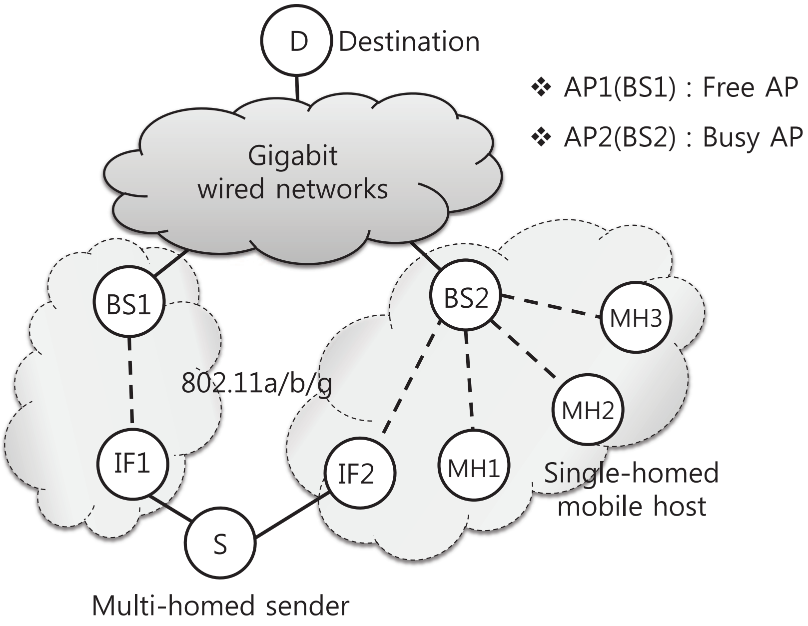


Fig. 9. Wired and wireless interconnected network topology.

the multi-homed sender has two independent and heterogeneous links for data transmission. The destination is connected through gigabit wired networks. In this multi-homing multi-rate wireless network, there exist heterogeneous wireless links and wired links; therefore, it is more difficult to reliably and efficiently distribute data packets through multiple paths.

We have evaluated the HMTP performance of this heterogeneous multi-homing multi-rate wireless network scenario. The connections with AP-1 and AP-2 are defined as paths 1 and 2, respectively. The bandwidth of the wireless links was set as 5 Mb/s. While AP-1 had no other clients, AP-2 had several clients. The number of single-homed clients of AP-2 varied from 0 to 3. Therefore, the transmit rates of both paths may differ according to the number of clients associated with each AP owing to the interference from the other associated clients. As the number of single-homed clients of the AP-2 increased to 3, the transmit rates of path 2 changed. For this case, many packets would be out-of-order and the receive buffer blocking problem would occur often. Fig. 10 shows the throughput and goodput performance of CMT-SCTP and HMTP. As shown in Fig. 10, in HMTP, the throughput of path 1 maintains its available bandwidth (4 Mb/s) because receive buffer blockings do not occur. In contrast, in CMT-SCTP, many out-of-order packets caused by lost packets transmitted through path 2 lead to frequent receive buffer

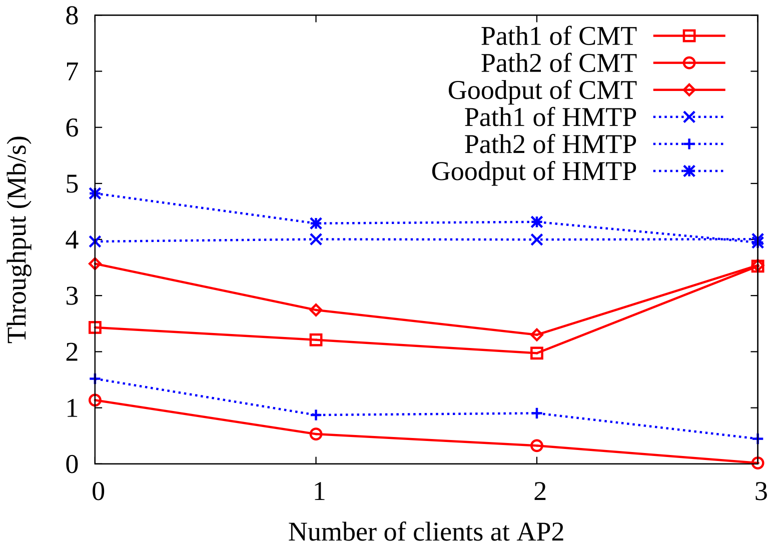


Fig. 10. Simulation results for the wired and wireless interconnected networks.

blockings. If the transmit rate of path 2 is very low because this AP is associated with many clients, most of data packets are transmitted through path 1. Therefore, for this case, the receive buffer blocking problem becomes more severe in CMT-SCTP. However, HMTP exhibits the better throughput and goodput performances for this multi-homing wireless networking scenario.

V. PERFORMANCE OF HMTP IN REAL TEST-BED EXPERIMENTS

In order to evaluate the performance of HMTP, we used a real test-bed based on a Linux machine. The experimental topology is similar to the multi-homed topology shown in Fig. 1(b). Two types of multi-homed laptops were used for constructing the test-bed: a laptop that had two Ethernet interfaces and a laptop that had one Ethernet interface and one IEEE 802.11 Wi-Fi interface. The two scenarios were configured using the network emulator *EXata* developed by Scalable Network Technologies, Inc. [28]. Note that the Internet shown in Fig. 1(b) was emulated using EXata, while the laptops transferred the real IP traffic through the network interfaces connected to the emulated networks.

The first scenario involved the data exchange between two multi-homed hosts on different wired access networks. The hosts in Fig. 1(b) were the laptops with two Ethernet interfaces. The second scenario involved the data exchange between two hosts that were simultaneously connected to wired and wireless access networks, i.e., one with wired link and the other with wireless link. The second scenario took into account the high loss rate and long end-to-end delay of the wireless links on the IEEE 802.11 WLAN.

HMTP was implemented in C++ on a Linux platform. All experiments have been conducted under Linux kernel-2.6.31.14. Because the standard Linux Ubuntu 9 package does not support SCTP, we upgraded the Linux kernel and installed the SCTP module lksctp-tools-1.0.11 in order to use SCTP as a transport layer. Note that the Linux kernel stream control transmission protocol (lksctp) project has implemented SCTP in the Linux kernel [29]. Specifically, HTMP was implemented as an application in Linux based on the SCTP module. It basically used SCTP as the transport protocol and modified it by enabling the concurrent transmissions on the multiple paths and disabling the packet retransmissions for HMTP. Then, the HMTP application called the standard socket functions to use the functions of the modified SCTP module in the Linux kernel.

A. Experimental results

We have evaluated the performance of HMTP in scenario 1, in which all the multi-homed hosts have two Ethernet interfaces and are connected to the Internet using two Ethernet interfaces. In this set of experiments, we set k, c, and δ for the fountain codes as 1000, 0.057, and 0.03, respectively.

In order to compare the performance, we measured the average time required to transmit a fixed number of packets using HMTP and CMT. Then, we calculated the throughput performance of each protocol in terms of the average number of packets transmitted per second (packets/s). Fig. 11 shows the results of these experiments. Notice that the performance of HMTP is better than that of CMT. However, the performance gain is small because all paths between the two multi-homed hosts are homogenous wired Ethernet links. For this set of experiments, the packet loss rates for both paths were smaller than 0.001, and the end-to-end delays for both paths were smaller than 1 ms. This implies that the receive buffer blocking rarely occurs because the heterogeneity of both paths is very small. Note that both laptops are connected to the same wired switch.

Then, we conducted another set of experiments in more heterogeneous environments. We evaluated the performance of HMTP in scenario 2, in which all multi-homed hosts had two

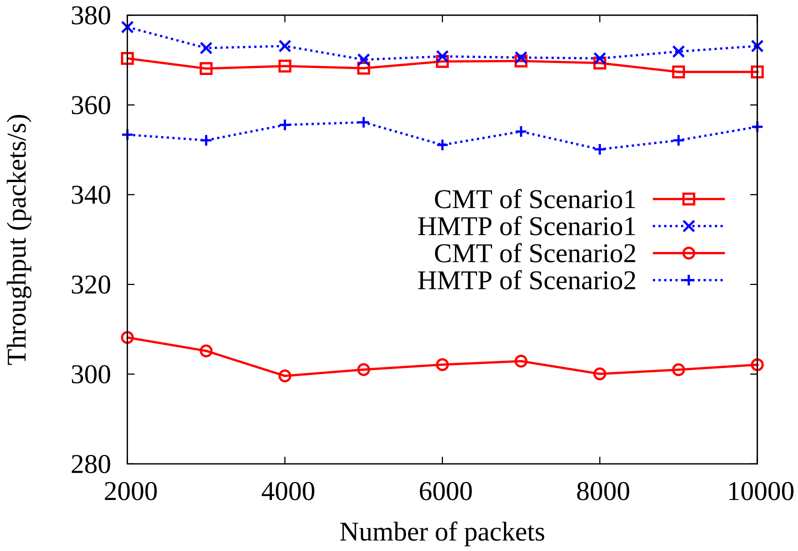


Fig. 11. Throughput results in the experimental test-bed.

different network interfaces, Ethernet interface and 802.11 Wi-Fi interface, and were connected to the Internet through them. The paths between the two multi-homed hosts were different in terms of the throughput and delay performance in this set of experiments: one was the wired Ethernet link and the other was the IEEE 802.11 wireless link. While the loss rate and end-to-end delay of the wired Ethernet link were similar to those in scenario 1, the corresponding values for the IEEE 802.11 wireless link were 0.1 and 5.7 ms, respectively. This implies that the receive buffer blocking occurs more frequently in scenario 2 than in scenario 1 because the heterogeneity of both paths is large in scenario 2. Fig. 11 shows the results from this set of experiments. As shown in Fig. 11, the performance gain of HMTP in scenario 2 is higher than that in scenario 1.

B. Summary

HMTP was implemented on a real test-bed using laptops and a network emulator. Its performance was evaluated using two scenarios. In order to evaluate the performance of HMTP, we measured the average time required to transmit a fixed number of packets over multiple paths using HMTP and CMT. We then compared the performance of the two protocols. The experiments indicated that HMTP outperforms CMT in heterogeneous multi-homing networks and mitigates the receive buffer blocking problem.

VI. CONCLUSION

The heterogeneous path characteristics of multi-homed networks present many challenges for reliable and efficient concurrent multi-path data transmission. The main objective of this work is to improve the overall network performance by solving the receive buffer blocking problem in heterogeneous multi-homing environments. Toward this end, we have proposed a new multi-path transport protocol, HMTP, that uses fountain codes. In HMTP, before a multi-homed sender transmits data packets, it encodes them into fountain-encoded packets. After a sufficient number of encoded packets have been successfully received at the multi-homed receiver, the original packets can be recovered irrespective of the order in which the packets were received across the multiple paths. The proposed scheme can reliably and efficiently transmit packets through heterogeneous paths in multi-homing networks. The results of extensive simulations and experimental test-beds show that HMTP achieves improved throughput and high path utilization.

REFERENCES

[1] Hwang YH, Obele BO, Lim H. Multipath transport protocol for heterogeneous multi-homing networks. In *Proceedings of ACM CoNEXT 2010 Student Workshop*, Philadelphia, USA, 2010.

[2] Huitema C. Multi-homed TCP. draft-huitema-multi-homed-01, Internet Engineering Task Force (IETF), 1995.

[3] Manilici V, Wundsam A, Feldmann A, Vidales P. Potential benefit of flow-based routing in multihomed environments. *European Transactions on Telecommunications* 2009; **20**(7): 650–659.

[4] Stewart R, et al. Stream Control Transmission Protocol. RFC2960, Internet Engineering Task Force (IETF), October 2000.

[5] Iyengar JR, Amer PD, Stewart R. Receive buffer blocking in concurrent multipath transfer. In *Proceedings of IEEE GLOBECOM*, Vol. 1, 122–126, St. Louis, Missouri, USA, 2005.

[6] Dreibholz T, Becke M, Rathgeb EP, Tuxen M. On the use of concurrent multipath transfer over asymmetric paths. In *Proceedings of IEEE GLOBECOM*, 1–6, December 2010.

[7] MacKay DJC. Fountain codes. *IEE Proceedings Communications* 2005; **152**(6): 1062–1068.

[8] Luby M. LT Codes. In *Proceedings of 43rd Annual IEEE Symposium on Foundations of Computer Science*, 271–280, Vancouver, BC, Canada, November 2002.

[9] Kumar D, Chahed T, Altman E. Analysis of a Fountain Codes Based Transport in an 802.11 WLAN Cell. In *Proceedings of 21st International Teletraffic Congress (ITC 2009)*, 1–8, Paris, France, September 2009.

[10] Gasiba T, Xu W, Stockhammer T. Enhanced system design for download and streaming services using Raptor codes. *European Transactions on Telecommunications* 2009; **20**(2): 159–173.

[11] Cataldi P, Tomatis A, Grilli G, Gerla M. CORP: Cooperative rateless code protocol for vehicular content dissemination. In *Proceedings of Med-Hoc-Net*, 1–7, June 2009.

[12] Iyengar JR, Amer P, Stewart R. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. *IEEE/ACM Transactions on Networking* 2006; **14**(5): 951–964.

[13] Hsieh HY, Sivakumar R. pTCP: An end-to-end transport layer protocol for striped connections. In *Proceedings of IEEE ICNP 2002*, 24–33, Paris, France, November 2002.

[14] Dong Y, Wang D, Pissinou N, Wang J. Multi-path load balancing in transport layer. In *Next Generation Internet Networks, 3rd EuroNGI Conference on*, 135–142, 2007.

[15] Jones EPC, Karsten M, Ward PAS. Multipath load balancing in multi-hop wireless networks. In *Proceedings of IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob 2005)*, Vol. 2, 158–166, 2005.

[16] Rojviboonchai K, Aida H. An evaluation of multi-path transmission control protocol (M/TCP) with robust acknowledgement schemes. *IEICE Transactions on Communications* 2004; **E87-B**(9): 2699–2707.

[17] Zhang M, Lai J, Krishnamurthy A, Peterson L, Wang R. A transport layer approach for improving end-to-end performance and robustness using redundant paths. In *Proceedings of the USENIX 2004 Annual Technical Conference*, 8–18, 2004.

[18] Stewart R, Metz C. SCTP: New transport protocol for TCP/IP. *IEEE Internet Computing* 2001; **5**(6): 64–69.

[19] Abd El Al A, Saadawi T, Lee M. LS-SCTP: A bandwidth aggregation technique for stream control transmission protocol. *Computer Communications* 2004; **27**(10): 1012–1024.

[20] Liao J, Wang J, Zhu X. cmpSCTP: An extension of SCTP to support concurrent multi-path transfer. In *Proceedings of IEEE ICC*, 5762–5766, Beijing, China, May 2008.

[21] Ye G, Saadawi TN, Lee M. IPCC-SCTP: An enhancement to the standard SCTP to support multi-homing efficiently. In *Proceedings of IEEE IPCCC*, 523–530, Pheonix, Arizona, April 2004.

[22] Ford A, Raiciu C, Handley M, Barre S, Iyengar J. Architectural guidelines for multipath TCP development. RFC 6182, Internet Engineering Task Force (IETF), March 2011.

[23] Ford A, Raiciu C, Handley M, Bonaventure O. TCP extensions for multipath operation with multiple addresses. RFC 6824, Internet Engineering Task Force (IETF), January 2013.

[24] Barre S, Paasch C, Bonaventure O. Multipath TCP: From theory to practice. *Lecture Notes in Computer Science on Networking* 2011; Vol. 6640, 444–457.

[25] Iyengar JR, Amer P, Stewart R. Retransmission Policies For Concurrent multipath transfer using SCTP multihoming. In *Proceedings of 12th IEEE International Conference on Networks (ICON 2004)*, vol. 2, 713–719, November 2004.

[26] Cui Y, Wang X, Wang H, Pan G, Wang Y. FMTCP: A fountain code-based multipath transmission control protocol. In *Proceedings of IEEE 32nd International Conference on Distributed Computing Systems (ICDCS)*, 366–375, June 2012.

[27] NS-2. *Univ. California, Berkeley, LBL, USC/ISI, and Xerox Parc*. http://www.isi.edu/nsnam/ns.

[28] EXata. *Scalable Network Technologies (SNT)*. http://www.scalablenetworks.com.

[29] Yarroll LM, Knutson K. Linux kernel SCTP: The third transport. http://lksctp.sourceforge.net, 2001.