Research Article

QoS-Based Path Switching Mechanism in Mobile SCTP

Li Han,1 Shimin Sun,1 Bokgyu Joo,2 and Sunyoung Han1

1 Department of Computer Science and Engineering, Konkuk University, Hwayang, Gwangjin, Seoul 143-701, Republic of Korea
2 Department of Computer and Information Communication, Hongik University, Sejong Campus, Sejong 337-701, Republic of Korea

Correspondence should be addressed to Sunyoung Han; syhan@cclab.konkuk.ac.kr

Received 26 June 2014; Accepted 18 August 2014; Published 15 September 2014

Academic Editor: Sung Won Kim

Copyright © 2014 Li Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In mobile SCTP, a mobile terminal has two or more network interfaces and vertical handover occurs when it moves from one network to another. The delay due to the handover process and the slow-start phase of SCTP’s congestion control after handover cause substantial performance degradation. If the mobile node goes back and forth frequently, excessive handovers occur and data transmission quality deteriorates. In order to provide the required level of QoS for on-going application, the frequency of handovers should be kept minimized. In this paper, we propose a transport layer handover mechanism using the mobile SCTP. We take the QoS requirements of application as the major criterion in deciding path switching. In our mechanism, the mobile node in overlapping area does not perform handover if the current network metrics satisfy the QoS requirements of on-going application. Both analytic evaluation and simulation results show that the proposed mechanism significantly improves the throughput by suppressing unnecessary handovers. Our research results can also be applied to distributed mobile sensor networks.

1. Introduction

As the radio technologies advance and the cost of chips decreases rapidly, it becomes common for a mobile device to use multiple wireless access technologies. Currently, WiFi, WiMAX, and 3G/LTE coexist on single mobile phone, and the device participates in more than one type of access networks at a time. For example, Apple's devices with version later than IOS 7 support simultaneous connections by 3G/LTE and WiFi. In many distributed wireless sensor networks (WSNs), especially in wireless multimedia sensor networks (WMSNs) which include various kinds of sensor nodes, some sensor nodes may support handover and be multi-homed. In heterogeneous wireless network environments, a mobile device or sensor node has more than one connection option in overlapped areas. If better service would be supplied by another network, the multihomed device changes its connection proactively, which is known as vertical handover. However, evaluating alternative connections and decision making for optimal path is a challenge.

Stream Control Transmission Protocol (SCTP) [1] has been developed and standardized as a new transport layer protocol. SCTP is a protocol that combines the features of TCP and UDP. SCTP makes it easier to manage connections over a wireless network, especially while mobile terminals are moving across different wireless networks.

Two major capabilities are newly designed into SCTP: the support of multihomed devices and the support for multiple streams in a single association. Multihomed devices allow a single SCTP association to run across multiple links, hence achieving link redundancy. With this capability, an SCTP association can perform fast failover from one path to another without significantly interrupting the data transmission. This feature provides a great profit to users by reducing the QoS degradation and session disruption during vertical handover.

In order to support transport layer mobility, DAR (Dynamic Address Reconfiguration) [2] is added to SCTP, and the new protocol is called mobile SCTP (mSCTP) [3]. The basic idea of mSCTP is to enable SCTP end-points to update its IP addresses and automatically inform the remote peer about the primary address change.

As in TCP, mSCTP still suffers from performance degradation in path switching process during vertical handover. The reasons for that are (a) entering into slow-start phase of new primary path after handover and (b) extra time needed to reorder packets. Many works have tried to improve
the mSCTP path switching scheme, but they still encounter problems due to unnecessary handovers.

In this paper, we provide an approach to optimize the handover process of mSCTP by exploring the efficient path switching mechanism. We first consider the QoS requirements of on-going application of a mobile node, and then we estimate network metrics to check if they meet the QoS requirements. The path switching decision is made only if the current network metrics do not meet the requirements while the alternative ones do. Our mechanism decreases the number of handover occurrences in many typical scenarios compared to the previous works.

The rest of this paper is organized as follows. In Section 2, related works on vertical handovers are presented. In Section 3, we describe the estimation method of network attributes and elaborated our path switching algorithms. In Section 4, simulation results and analytic evaluation are presented. Section 5 concludes our research works.

2. Related Works

Many mobility management techniques have been proposed to solve the handover related issues. Some researchers have focused on the mobility management at the application layer using SIP (Session Initiation Protocol) [4]. The majority of handover research works focus on the network protocols such as Mobile IPv4 (MIP), Mobile IPv6, Hierarchical Mobile IPv6, Fast Mobile IPv6, and Proxy Mobile IPv6. However, they introduce unavoidable delay and dependency on additional network components such as home agent, foreign agent, and proxy, to maintain connections during handovers.

Although network-layer scheme makes mobility transparent to upper layers, it increases the burden on the Internet infrastructure. Another promising approach is to do handover at the transport layer. A transport layer scheme is an end-to-end approach that keeps the Internet infrastructure unaware of mobility by having two end-points to take care of it.

Approaches based on the transport layer are gaining an increasing attention in recent years.

MPTCP (Multipath Transmission Control Protocol) [5] speeds up data transmission by introducing a new set of migration options for TCP to support mobility. However, it does not actually use MPTCP to send packets over WiFi and 3G/LTE at the same time. Rather, it sets up MPTCP and then continues to communicate over WiFi.

SCTP EFC (Efficient Flow Control) [6] is presented in order to minimize the change of throughput during path switching process by updating the new path with same throughput as the old one to skip the slow-start phase. In heterogeneous networks, however, access network features such as bandwidth and delay may be very different. If path switching is happening from high quality path to low quality path, packet loss and retransmission timeout occur. This causes the connection to go into a slow-start phase.

As mobile nodes are equipped with multiple interfaces, mobility in heterogeneous networks is also widely researched. A number of mechanisms have been proposed to support vertical handover based on mSCTP [7, 8]. They still suffer from the problems of unavoidable delays, packet loss, and connection interruption during handover process. The performance of MIP and mSCTP over IPv6 networks is evaluated in [9]. The results show that mSCTP performs better than MIP via its multihoming feature. Performance comparison of handover of MIP, SIP, and mSCTP in 3G/WLAN networks is elaborated in [10]. Simulation results show that mSCTP produces considerably lower delay in handover than SIP.

To provide a better throughput during the handover process, reference [11] proposes the data duplication technique on coexisting paths. This is useful for the primary path with high loss rate, but it results in creating redundancy in stable context.

Many studies use many criteria in making path switching decision to select optimum network path during vertical handover. They consider some of the network attributes as the criterion matrix: received signal strength (RSS), bandwidth, RTT, packet loss rate, velocity, and cost. [12] proposes SNIR (signal interference and noise ratio) based decision criteria. Improved decision algorithms based on fuzzy logic are present in [13]. Reference [14] introduces a network selection scheme based on the metrics of bandwidth, dropping probability, and cost.

Some proposed algorithms focus on the reduction of unnecessary handovers. Reference [15] restricts certain handover direction: if the handover from a WLAN to cellular system occurred before, the handover back to that WLAN is not allowed. The drawback of this scheme is that the prediction of user movement is complicated for mobile terminals. [16] proposes an algorithm based on the dwell-timer for eliminating ping-pong effect by reducing the shadow fading effect. But it is not feasible to use dwell-timer for individual users based on their needs.

In this paper, we provide the methods to dynamically detect some dominating attributes of associated links in real time. In order to estimate the critical network attributes, we utilized the intrinsic monitoring messages of SCTP rather than introducing extra messages. The handover decision is made only by the comparison between the QoS requirements of current application and the real time network status. This mechanism significantly reduces the possibility of unnecessary handovers and improves the user’s quality of experience (QoE).

3. QoS-Based Path Switching Mechanism

Our mechanism improves throughput by avoiding unnecessary handovers in heterogeneous network environments. QoS requirements of application are accounted into decision criterion to compare with each path's quality attributes. A detailed adaptive path switching algorithm is developed based on QoS requirements of application. We use the intrinsic monitoring messages (HEARTBEAT and HEARTBEAT-ACK) in order to detect network quality attributes in real time.
This section is arranged as follows. Firstly, the basic knowledge of handover in mSCTP is introduced in Section 3.1. Secondly, the application's QoS requirements are discussed in Section 3.2. The measurement method for RTT, bandwidth, and some other network attributes follows in Section 3.3. Finally, the path switching algorithm is presented in Section 3.4. In our preliminary work [17], we had proposed to compare RTT and bandwidth of associated links with QoS requirements of application.

3.1. Overview of SCTP, mSCTP, and Handover. SCTP is a reliable, message oriented transport layer protocol. It allows each end-point to provide the corresponding nodes during association startup with a list of IP addresses through which it can be reached. The SCTP binds multiple source and destination IP addresses for a single association between both end-points. These IP addresses are exchanged and verified during the association setup and they can be used interchangeably for communication.

An SCTP packet contains the common SCTP header and various control or data chunks. The control messages are used to initiate (INIT, INIT-ACK, COOKIE-ECHO, COOKIE-ACK), tear down (SHUTDOWN, SHUTDOWN-ACK, SHUTDOWN COMPLETE), and monitor (HEARTBEAT, HEARTBEAT-ACK) associations. The process of association establishment by the initiation messages and termination by the tear down messages was elaborated in [1]. Two monitoring messages are used to monitor the association periodically. We use them in Section 3.3 for network attributes measurement.

The DAR extension [2] allows SCTP to dynamically add IP addresses, delete IP addresses, and assign the path switching decision, to an active SCTP association. The process of path switching is achieved by two new chunk types: ASCONF and ASCONF-ACK with six new parameters: Add IP Address, Delete IP Address, Set Primary Address, Error Cause Indication, Success Indication, and Adaptation Layer Indication. ASCONF is sent in an authenticated way to avoid the risk of association hijacking.

New IP addresses are added to the association when a new link is detected on other interfaces. Only addresses belonging to this association can be set as the primary address; otherwise the Set Primary Address message is discarded. Before setting a new address as the primary address, the path verification procedure and path status detection are achieved by means of HEARTBEAT/ACK periodically.

Handover occurs when the criterion is met: suppose a mobile node (MN) is moving into a higher-quality network (e.g., WLAN) from ubiquitous network (e.g., UMTS). To maintain on-going sessions, the MN goes through a series of steps detailed in [4]. The scenario is illustrated in Figure 1, and the vertical handover process is elaborated in Figure 2. In this example, the corresponding node (CN) is only with one IP address, although it could also be multihomed. During the handover process, data transmission is sustained. However, due to the congestion control mechanism, the connection will start at a low transmission rate after handover during slow-start phase. This will cause an obvious degradation in throughput.

3.2. Application’s QoS Requirements. Application’s QoS Requirements in a network can be classified into many different ways. A practical approach defined in [18] provides service differentiation by defining different service classes.

QoS is a set of service level requirements to be met by a network in transmitting a dataflow. This contractual commitment is based on quality metrics, such as throughput, delay, jitter, packet loss, and availability. QoS in a networking context usually looks at four traffic characteristics:

(1) Bandwidth: the number of bits per second that can be expected to be transmitted.
(2) Delay: the elapse time for a bit of data to travel across the network from one node to another. The total network delay is composed of processing delay, queuing delay, transmission delay, and propagation delay.
(3) Jitter: the variation in latency as measured in the variability over time of the packet latency across a network.
(4) Packet loss: the failure of transmitted packets to arrive at their destination.

Since different applications have different service features, it is vital to specify the service requirements to realize quality assurance on computer network. In this paper, we consider four attributes listed above as the main criteria.

As SCTP being a transport layer protocol, it is easy to classify the services by port number and protocol. For example, HTTP usually uses destination port 80. In principle, applications can be simply identified by the server port and mapping this port to an application using the IANA list of registered ports. However, port-based application classification has limitations: not all applications are registered in IANA. Many application classification mechanisms have been proposed [19–24]. We follow the classification presented in [19] and choose the typical classes as in Table I.

(1) Requirements for conventional web browsing: representative of best effort applications. Delay less than 400 ms is expected for all best effort network traffic. Jitter is not applicable in HTTP because jitter has little impact on traditional static text and picture web browsing. Required bandwidth is less than 40 kbps for typical applications of web browsing. Since they use reliable transfer protocol, with packet retransmission, packet loss rate should be zero.

(2) Requirements for VoIP: real time and low bandwidth applications. VoIP requires end-to-end delay shorter than 150 ms to give the users an imperceptible difference between audio and real speech. A jitter rate less than 400 ms is necessary. Packet loss far less than 1% in the G.729 codec is recommended. Different voice coding technologies lead to different bandwidth requirements while 64 kbps is adequate for most coding standard.
(3) Requirements for video: real time and high bandwidth applications. They use UDP-liked transport layer protocol for data transmission. Real time video requires end-to-end delay shorter than 100 ms. Beyond 100 ms, although speech is still comprehensible, users may become irritated with the service. Streams with a higher compression ratio are more sensitive to packet loss and error. It is possible for errors to be recovered only in the case that the error rate does not exceed 0.1%. The required bandwidth depends on the video quality and compression format. 3Mbps is sufficient bandwidth for common quality video, such as 720p HDTV.

(4) Requirements for FTP service: typical high throughput applications. FTP requires a relatively high bandwidth and greater tolerance for delay and jitter. The expected packet loss rate for FTP service is zero. Based on the analysis above, delay and bandwidth are both important for best effort applications and have low relationship with packet loss and jitter. For real time applications, latency is the most sensitive attribute and bandwidth should meet the basic requirements. Packet loss is also important to achieve a good user experience. For the common data transmission applications like FTP, higher bandwidth is preferred while it is not sensitive to delay, packet loss, and jitter. The proposed path switching algorithm will take into account the requirements listed above.

3.3. RTT and Bandwidth Estimation Methods. By default, an SCTP end-point monitors the reachability of the association by sending a HEARTBEAT chunk periodically to the destination addresses. A HEARTBEAT chunk is sent once per retransmission timeout (RTO) of the destination plus the protocol parameter "HB.interval", with jittering of ±50%, and exponentially backed off the RTO if the previous HEARTBEAT is unanswered [1].
Table 1: QoS metrics of four typical applications.

| Class         | Delay (ms) | Bandwidth (bps) | Packet loss | Jitter (ms) | Category        |
|---------------|------------|-----------------|-------------|-------------|-----------------|
| Web browsing  | <400       | <40 K           | Zero        | N/A         | Best effort     |
| VoIP          | <150       | <64 K           | <1%         | <400        | Real time       |
| Video         | <100       | <3 M            | <0.01%      | <400        | Real time       |
| FTP           | Med        | High            | Zero        | N/A         | High throughput |

Based on this definition, HEARTBEAT/ACK pairs are launched from the mobile node periodically when an association is established. The information of transmitting these couple messages can be used to calculate path QoS parameters efficiently. This provides adequate information about associated paths when making path switching decision.

3.3.1. RTT Calculation. Normal RTT in SCTP, calculated from the HEARTBEAT/ACK pair, might be inaccurate if a sudden network congestion or packet loss occurs. To account for this, we use smoothed RTT along with a filtering method.

Assuming only two paths are associated, one is primary path and the other is alternative path. MN has measured RTT for n times, and then the RTT of a path at ith measurement round can be expressed as:

Primary path:
\[
\text{RTT}_{(P)}^i = \delta \cdot \text{RTT}_{(P)}^{(i-1)} + (1 - \delta) \cdot \text{RTT}^*_{(P)\text{sample}} \quad (1)
\]

Alternative path:
\[
\text{RTT}_{(A)}^i = \delta \cdot \text{RTT}_{(A)}^{(i-1)} + (1 - \delta) \cdot \text{RTT}^*_{(A)\text{sample}} \quad (2)
\]

RTT^*_{(P)\text{sample}} and RTT^*_{(A)\text{sample}} are RTTs of primary path and alternative path at ith measurement round, respectively. \( \text{RTT}^{(i-1)}_{(P)} \) and \( \text{RTT}^{(i-1)}_{(A)} \) are the previous (i-1)th round RTTs. \( \text{RTT}^*_{(P)\text{sample}} \) is the newest sampled RTT and \( \delta \) is the smoothing filtering weight. By applying the smooth filtering weight \( \delta \) to the equation, the influence of abnormal RTT is reduced. In order to perform more fine-grained steps and react to deal with sudden RTT fluctuations more conservatively, we define that those RTTs will not be included in the calculation process when a value of RTT is larger than double size of RTO. The calculation formula of RTO in SCTP is

\[
\text{RTO} = \text{RTT} + 4 \ast \text{RTTVAR} \quad (3)
\]

When the first RTT is measured, RTTVAR = 0.5 * RTT.

When a new RTT is measured, RTTVAR is calculated using the similar smoothing method as used in calculating RTT.

3.3.2. Available Bandwidth Calculation. Available bandwidth depends on the bottleneck of a link and cross traffic. Many different active probing techniques have been developed to measure available bandwidth. The packet pair technique [23, 24] is one of the most popular among them. The basic idea of packet pair paradigm is that the sender sends pairs of packets to a receiver. The pair packets are sent close enough in time to cause the packets to queue together at the bottleneck link. By measuring the changes in the packet spacing, the receiver can estimate the amount of cross traffic that enters between the two packets at the bottleneck link and then infers bandwidth proprieties of the network path.

The problems of the existing mechanisms are as follows.

1. Only receiver can get the estimated value of available bandwidth, whereas the sender needs the value for path switching decision.
2. They do not collect available bandwidth periodically.

We efficiently use HEARTBEAT/ACK message pairs that are defined in SCTP protocol.

1. HEARTBEAT message should be replied by the receiver immediately when received.
2. Time stamp is contained in HEARTBEAT/ACK pair information field.
3. HEARTBEAT messages are periodically sent by sender.

Before the path switching decision is made, MN has to measure the bandwidth of active links for n times. Packet pair bandwidth estimation methods send a pair of big size packets to receiver: packet length is set to Maximum Transmission Unit (MTU) and the packet pair is sent close enough. It causes the packets to queue together at the bottleneck link and separate at the outgoing edge. When the receiver received these two packets, the time interval \( \Delta t \) between two packets will be obtained, illustrated in Figure 3. The bottleneck bandwidth can be calculated using

\[
B_e = \frac{\text{Packet size}}{\Delta t}, \quad (4)
\]

where Packet size is the size of sent packet, usually set to MTU.

The limitation of existing methods is that only the receiver can get the time interval \( \Delta t \). The available bandwidth could only be calculated by receiver side. This problem can be solved by using specific message pairs (HEARTBEAT/ACK). HEARTBEAT messages are sent with time stamp field and padding with nonce to let the packet size reach MTU. After the sender receives both HEARTBEAT and HEARTBEAT-ACK, it can calculate \( \Delta t \) by calling the time stamp fields.
Based on the calculated bandwidth, to increase the accuracy of the bandwidth, we get a fine value by comprehending the historical records. The bandwidth of both paths at \( i \)th measurement round can be expressed as

\[
B_e^{(i)} = \gamma \cdot B_e^{(i-1)} + (1 - \gamma) \cdot B_e,
\]

where \( B_e^{(i)} \) is the available bandwidth, \( B_e^{(i-1)} \) is the last available bandwidth, \( B_e \) is the real measured \( i \)th bandwidth, and \( \gamma \) is the weight.

3.3.3. Other Parameters. Jitter is generated along with RTT measurement, marked as RTTVAR.

When the first RTT is measured, RTTVAR = 0.5 * RTT.

When a new RTT value is obtained, \( i \)th RTTVAR is calculated by the following equation:

\[
\text{RTTVAR}_{(i)} = (1 - \mu) \cdot \text{RTTVAR}_{(i-1)} + \mu \cdot |\text{RTT}_{(i)}|.
\]

Packet loss is also generated from periodical HEARTBEAT/ACK message pairs and message pairs used for bandwidth calculation.

3.4. Primary Path Switching Criteria. To facilitate the primary path switching mechanism, we compare the matrix of the application’s QoS requirements with the real time network QoS matrix. The network quality of both paths is the criterion for path switching algorithms. We classify the possible scenario into four cases when considering the application’s requirements and real time network conditions of two paths: primary path and alternative path.

Case 1 (only primary path satisfies the requirements). When MN detects a new network with lower quality, primary path switching will not be executed because the current link satisfies the application’s requirements.

Case 2 (only alternative path satisfies the requirements). When MN detects a new network with much better quality and the current link has some limitation to meet the QoS of on-going application, MN initiates path switching immediately.

Case 3 (both paths satisfy the requirements). When MN detects a new network with good quality, the current link is also good enough to deal with the required transmission rate. In this case, the current link does not need to be interrupted and path switching will not be executed.

Case 4 (neither path satisfies the requirements). In this case, a relative better path will be chosen as the primary path.

To clearly explain the algorithm, we categorize applications into two: known applications and unknown applications. Known applications are those that can be found in IANA registered ports. Known applications are also categorized into best effort, real time, and high throughput applications. Some notations are as follows.

- \( B_{(P)} \): Bandwidth of primary path
- \( B_{(A)} \): Bandwidth of alternative path
- \( B_{(R)} \): Bandwidth of application required
- \( \text{RTT}_{(P)} \): RTT of primary path
- \( \text{RTT}_{(A)} \): RTT of alternative path
- \( \text{RTT}_{(R)} \): RTT of application required
- \( \eta, \kappa \): Comparison factors.

If some applications use unregistered ports, algorithm is applied in Algorithm 1.

For best effort applications, such as web browsing and text chatting, which are not sensitive to network attributes, they usually need a little bandwidth and tolerate high latency. Those applications are usually TCP-based applications; retransmission of TCP packets can compensate the effect of packet loss. Thus, packet loss of the network is not the critical attribute. Therefore, we make the decision based on the distinction extent around attributes. Pseudo path switching decision algorithm for best effort applications is present in Algorithm 2.

For real time related applications such as video and audio, delay should be considered the main criterion and bandwidth should be adequate. Meanwhile, packet loss represented by \( L \) should be considered. Pseudo path switching decision algorithm is present in Algorithm 3. \( L_R \) is the required packet loss by applications.

For high throughput applications such as FTP, bandwidth is the main concern and RTT is more of a complement. Algorithm 4 shows the pseudo path switching algorithm for the case.

4. Discussion and Simulation

4.1. Discussion. Several mobile network technologies have been implemented in real world. We listed four major technologies in Table 2 and compared their data rate and latency. Mobile nodes (MNs) move around those networks, and they make handover decisions. As we analyzed in this paper, handover is not necessary in some cases when application’s QoS requirements are lower than the QoS of the current network. Table 3 summarizes switching decisions made by our algorithms. Handover is not necessary, in many cases,
because the current network quality is enough for on-going application. Previous mechanisms would do path switching in all cases because MN moves into better quality networks.

In real commercial mobile networks, price should be considered as an important factor. In this simulation, however, the price factor is not taken into account for simplicity. The reason is that each Telco has different pricing policy on different network types, and users’ behavior has to be considered in selecting the links: students, for example, prefer low price links even for high-data rate applications like video and ftp.

A mobile node could run several on-going applications at a time. In this research, however, we assumed that only one application is doing data transmission to simplify the simulation.

4.2. Environments Setup. We use the simulation tool Network Simulator 2 (NS2) [25] to build the network topology. For the SCTP module, we used the one designed by the Protocol Engineering Laboratory of University of Delaware [26]. We modified the source code of the SCTP module in order to make the module support mSCTP.

In many cases, the proposed mechanism would decide not to switch path even if the network quality of new path is better than the current path, which is marked as “No” entries in Table 3. Most of the existing switching mechanisms prefer to switch path in those cases. It is necessary to analyze the influence of throughput in path switch and in no path switch. Among them we choose 4 typical cases: (a) web browsing from 3G to WiFi, (b) VoIP from WiFi to LTE, (c) video from WiMAX to LTE, and (d) FTP from WiFi to WiMAX.

The network model used in this simulation is shown in Figure 4. Both MN and CN have two interfaces, each connecting with different type of networks. Nodes communicate with each other using mSCTP protocol in transport layer. The transmission path is decided by our path switching algorithm. The network parameters are based on the values in Table 2. Other simulation parameters are listed in Table 4.

In Table 4, two comparison factors bandwidth comparison factor ($\kappa$) and RTT comparison factor ($\eta$) should be selected based on the user's conception on QoS of network. We simply choose the value 1.5 in the simulation.
Algorithm 3: Real time application path switching algorithm.

```plaintext
Join SCTP session
Begin loop
  If (RTT_{(P)} < \eta RTT_R && B_{(P)} > \kappa B_R && L_{(P)} < L_R)
    Stay on current path
  Else if ((RTT_{(P)} < \eta RTT_R && B_{(P)} > \kappa B_R) && (RTT_{(A)} < \eta RTT_R && B_{(A)} > \kappa B_R && L_{(A)} < L_R))
    Sends “Primary-Switching”
  Else if (RTT_{(P)} > \eta RTT_R && B_{(P)} > B_R && L_{(P)} < L_R)
    Sends “Primary-Switching”
  Else
    Stay on the current path
End switch
End loop
Leave SCTP session
```

Algorithm 4: High throughput application path switching algorithm.

```plaintext
Join SCTP session
Begin loop
  If (B_{(P)} > \kappa B_R)
    Stay on current path
  Else if (B_{(P)} < B_R && B_{(A)} > \kappa B_R)
    Sends “Primary-Switching”
  Else if (B_{(A)} > \kappa B_R > B_{(P)} > B_R && RTT_{(P)} > \eta RTT_{(A)})
    Sends “Primary-Switching”
  Else
    Stay on the current path
End switch
End loop
Leave SCTP session
```

4.3. Simulation Results

4.3.1. Web Browsing from 3G to WiFi. 3G network quality satisfies the QoS requirements of web browsing. Thus, unnecessary handovers could be avoided, when a MN is moving from 3G to WiFi. We compare the throughput difference between the case with path switching and the case without. We generated a Constant Bit Rate (CBR) with a 40kbps bitrate to imitate a web browsing application in NS2. Figure 5(a) shows the throughput by path switching when connecting to WiFi and Figure 5(b) shows the throughput without path switching. By analyzing those figures, we can conclude that performance by our mechanism is much better than existing ones. The analytic evaluation of throughput distinction is presented in Section 4.5.

The throughput is significantly affected by the handover process. It is caused by the switching of transmission path and the starting of transmission from initialization phase. It becomes more crucial when a MN moves in and out of WLAN frequently. In this case, the speed of loading webpages is greatly influenced.

4.3.2. VoIP from WiFi to LTE. VoIP QoS requirements in Table 1 are much more sensitive to delay than web browsing. However, both delay and bandwidth of WiFi and LTE are sufficient for VoIP. We used CBR with a 64 kbps bitrate to imitate VoIP application in NS2. Figure 6(a) shows the throughput of path switching case and Figure 6(b) shows the throughput without path switching.

4.3.3. Video from WiMAX to LTE. Video is quite sensitive to bandwidth and to delay. However, both delay and bandwidth of WiMAX and LTE are sufficient for video applications. We
Figure 5: Throughput of web browsing in use in overlapped area of 3G and WiFi.

Figure 6: Throughput of VoIP in use in overlapped area of WiFi and LTE.

Figure 7: Throughput of video in use in overlapped area of WiMAX and LTE.
### Table 3: Path switching decision between different networks.

|                  | 3G → others | WiFi → WiMAX | WiFi → LTE | WiMAX → LTE |
|------------------|-------------|--------------|-----------|-------------|
| Web browsing     | No          | No           | No        | No          |
| VoIP             | Yes         | No           | No        | No          |
| Video            | Yes         | No           | Yes       | No          |
| FTP              | Yes         | No           | Yes       | Yes         |

#### Figure 8: Throughput of FTP in use in overlapped area of WiFi and WiMAX.

(a) Handover from WiFi to WiMAX

(b) Stay in WiFi network without handover

### Table 4: Other simulation parameters.

| Parameters                      | Value       |
|--------------------------------|-------------|
| Simulation duration            | 80 seconds  |
| Bandwidth comparison factor: $\kappa$ | 1.5         |
| RTT comparison factor: $\eta$   | 1.5         |

used CBR with a 3 Mbps bitrate to imitate video application in NS2. Figure 7(a) shows the throughput of taking path switch and Figure 7(b) shows the throughput without switching path.

#### 4.3.4. FTP from WiFi to WiMAX.

FTP prefers a higher bandwidth network always. According to the path switching algorithm in Algorithm 2, MN performs path switching in all scenarios except moving from WiFi to WiMAX. The difference of QoS between WiFi and WiMAX is very little. Although WiMAX can provide a little higher bandwidth than WiFi, the MN will not do handover according to the path switching criteria. From the simulation, we can conclude that Figure 8(b) shows the better throughput than Figure 8(a).

#### 4.4. Summary of Simulation Results.

From the simulation results of some scenarios where handover is not necessary, it became clear that handover causes a lot of throughput degradation during path switching period. In other words, if the current path capacity is adequate for bearing the ongoing application, unnecessary handover will just degrade the performance and will affect user’s QoE. Therefore, we insist that the handover decision should be made depending on the requirements of on-going application rather than the network quality of alternative paths. This paper provides adaptive decision making algorithms for path switching.

#### 4.5. Analytic Evaluation.

The simulation results indicate that efficient path switching algorithm provides a good deal of throughput in many scenarios. For analytic evaluation, we present a mathematical comparison of throughput distinction. We took the second scenario as an example: VoIP from WiFi to LTE. In Figure 9, we mark off four regions $(a,b,c,d)$ of Figure 6(a). When our mechanism is applied, the throughput is the total area of $(a,b,c,d)$; the throughput becomes regions $(b,c,d)$ when handover happens. Region $a$ is the extra throughput. Hereon, we will approximately calculate the percentage of the extra throughput (region $a$).

Considering the line of throughput as two strips of parabolas, one is the edge of region $b$ and the other is the edge of region $c$. We arrange the origin at the start point of handover, so the intersection point of $(b,c,d)$ is just at coordinate $(10,40)$. We assume that the parabolas can be represented approximately by standard equation of parabola:

$$y^2 = 2px.$$  \hspace{1cm} (7)

As the definition of parabola, $p$ is the distance between fixed point (the focus) and fixed line (the directrix). For region $b$, $p$...
5. Conclusion

In distributed sensor network in the future, mobile sensor nodes as well as regular mobile terminals will move around heterogeneous networks. In this paper, we presented an improved vertical handover mechanism using mSCTP. The contribution of our research work is to keep the frequency of handovers to the minimum.

The proposed algorithm first checks if the current network attributes satisfy the QoS requirements of on-going application. Then, the handover decision is made only if the current link does not satisfy the requirements and there is an alternative one that does. The simulation results with typical handover scenarios along with analytic evaluation show that our mechanism significantly improves the throughput by reducing unnecessary handovers.

Only a few QoS attributes were considered in this research. In reality, however, other factors such as price, network stability, and user habits of network usage also affect path switching decision. To be applied in real world, we should consider those factors in our algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by Research and Development Project of Korea National Information Society Agency (NIA) (SDN Multicasting on KOREN/APII/TEIN).

References

[1] R. Stewart, "Stream Control Transmission Protocol," IETF RFC 4960, September 2007.
[2] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, and M. Kozuka, "Stream control transmission protocol (SCTP) dynamic address reconfiguration," IETF RFC 5061, 2007.
[3] S. J. Koh and S. W. Kim, "mSCTP for vertical handover between heterogeneous networks," in Proceedings of the 3rd International Conference on Human.Society@Internet (HSI '05), pp. 28–36, July 2005.
[4] H. Schulzrinne and E. Wedlund, "Application-layer mobility using SIP," ACM Mobile Computing and Communications Review, vol. 4, no. 3, pp. 47–57, 2000.
[5] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF RFC 6824, 2013.
[6] K. J. Lee, S. S. Nam, and B. I. Mun, "SCTP efficient flow control during handover," in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06), pp. 69–73, April 2006.
[7] J. Shim, S. Song, and B. Rhee, "Enhancement of mSCTP mobility scheme for seamless vertical handover," in Proceedings of the 3rd International Conference on Ubiquitous and Future Networks (ICUFN '11), pp. 268–272, Dalian, China, June 2011.
[8] Y. Kim and S. Lee, “MSCTP-based handover scheme for vehicular networks,” *IEEE Communications Letters*, vol. 15, no. 8, pp. 828–830, 2011.

[9] P. K. Dong and J. K. Seok, “Analysis of handover latency for mobile IPv6 and mSCTP” in *Proceedings of the IEEE International Conference on Communications Workshops (ICC ’08)*, pp. 420–424, Beijing, China, May 2008.

[10] A. Mahmoud, A.-A. Al-Helali, M. Abu-Amara, T. Al-Kharobi, and T. Sheltami, “Comparative performance study for integrated 3G/WLAN networks using mobile IP, SIP, and m-SCTP protocols,” in *Proceedings of the IEEE 71st Vehicular Technology Conference (VTC ’10)*, May 2010.

[11] S.-H. Kwon, S.-J. Koh, T.-W. Um, and W. Ryu, “Mobile SCTP with bicasting for vertical handover,” in *Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT ’08)*, vol. 1, pp. 123–125, Busan, Republic of Korea, November 2008.

[12] Y. Kemeng, I. Gondal, Q. Bin, and L. S. Dooley, “Combined SINR based vertical handoff algorithm for next generation heterogeneous wireless networks,” in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM ’07)*, pp. 4483–4487, November 2007.

[13] A. Mehbodniya, F. Kaleem, K. K. Yen, and F. Adachi, “A fuzzy MADM ranking approach for vertical mobility in next generation hybrid networks,” in *Proceedings of the 4th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT ’12)*, pp. 262–267, October 2012.

[14] J. Chen, Z. Wei, Y. Wang, L. Sang, and D. Yang, “A service-adaptive multi-criteria vertical handoff algorithm in heterogeneous wireless networks,” in *Proceedings of the IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’12)*, pp. 899–904, Sydney, Australia, September 2012.

[15] Y. Kirsal, E. Ever, G. Mapp, and O. Gemikonakli, “Enhancing the modelling of vertical handover in integrated cellular/WLAN environments,” in *Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA ’13)*, pp. 924–930, Barcelona, Spain, March 2013.

[16] S. R. V. Reddy and S. D. Roy, “Vertical handoff decision algorithms in integrated heterogeneous networks,” *International Journal of Energy Information and Communications*, vol. 4, pp. 71–82, 2013.

[17] L. Han, S. Sun, N. Nonhlanhla, S. Cho, S. Bae, and S. Han, "Adaptive path switching for vertical handover using mSCTP," IEICE Technical Report, Internet Architecture IA2012, vol. 112, pp. 1–5, 2012.

[18] J. Babiarz, K. Chan, and F. Baker, “Configuration guidelines for DiffServ service classes,” IETF RFC 4594, 2006.

[19] Y. Chen, T. Farley, and N. Ye, “QoS requirements of network applications on the internet,” *Information Knowledge Systems Management*, vol. 4, pp. 55–76, 2004.

[20] E. Biersack, C. Callegari, and M. Mattijasevic, "Data traffic monitoring and analysis: from measurement, classification, and anomaly detection to quality of experience," in *Computer Communication Networks and Telecommunications*, Springer, 2013.

[21] C.-N. Lu, C.-Y. Huang, Y.-D. Lin, and Y.-C. Lai, “Session level flow classification by packet size distribution and session grouping,” *Computer Networks*, vol. 56, no. 1, pp. 260–272, 2012.

[22] T. T. T. Nguyen, G. Armitage, P. Branch, and S. Zander, “Timely and continuous machine-learning-based classification for interactive IP traffic,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1880–1894, 2012.

[23] M. AlSabah, K. Bauer, and I. Goldberg, “Enhancing Tor’s performance using real-time traffic classification,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS ’12)*, pp. 73–84, October 2012.

[24] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC ’03)*, pp. 39–44, October 2003.

[25] SAMAN, CONSER, and ACIRI, “The Network Simulator 2,” http://www.isi.edu/nsnam/ns/.

[26] "SCTP module for NS-2," Protocol Engineering Lab, University of Delaware, http://pel.cis.udel.edu.
Submit your manuscripts at
http://www.hindawi.com