Special Issue

(1) Research and Trials for Reliable VoIP Applications

Guest Editors:
F.-N. Pavlidou, A. Pitsillides, H. Schulzrinne and D. Sisalem

(2) Wireless Multimedia Sensor Networks

Guest Editors:
O.B. Akan, P. Frossard, Q. Zhang and N. Jayant
Contents

(1) Research and Trials for Reliable VoIP Applications

F.-N. Pavlidou, A. Pitsillides, H. Schulzrinne and D. Sisalem
Guest Editorial: Research and trials for reliable VoIP applications 2447

J. Liao, J. Wang and X. Zhu
A multi-path mechanism for reliable VoIP transmission over wireless networks 2450

M. Kalama, G. Acar, B. Evans and A. Isoard
VoIP over DVB-RCS satellite systems: Trial results and the impact of adaptive speech coding using cross-layer design 2461

S. Jelassi and H. Youssef
Adaptive playout scheduling algorithm tailored for real-time packet-based voice conversations over wireless ad-hoc networks 2473

H.-Y. Hsieh, Y.-E. Lin and H.-P. Lin
Enhancing VoIP service for ubiquitous communication in a campus WLAN with partial coverage 2489

A. Estepa and R. Estepa
Accurate resource estimation for homogeneous VoIP aggregated traffic 2505

D. Zisiadis, S. Kopsidas and L. Tassiulas
VIPSec defined 2518

(2) Wireless Multimedia Sensor Networks

O.B. Akan, P. Frossard, Q. Zhang and N. Jayant
Guest Editorial: Special issue on wireless multimedia sensor networks 2529

N. Saxena, A. Roy and J. Shin
Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks 2532

C. Na, Y. Yang and A. Mishra
An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks 2543

N. Tezcan and W. Wang
Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints 2558
A multi-path mechanism for reliable VoIP transmission over wireless networks

Jianxin Liao *, Jingyu Wang, Xiaomin Zhu
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, P.O. Box 296, Beijing 100876, PR China

ABSTRACT

With the advances in audio encoding standards and wireless access networks, voice over IP (VoIP) is becoming quite popular. However, real-time voice data over lossy networks (such as WLAN and UMTS), still poses several challenging problems because of the adverse effects caused by complex network dynamics. One approach to provide QoS for VoIP applications over the wireless networks is to use multiple paths to deliver VoIP data destined for a particular receiver. This paper introduced cmpSCTP, a transport layer solution for concurrent multi-path transfer that modifies the standard stream control transmission protocol (SCTP). The cmpSCTP aims at exploiting SCTP’s multi-homing capability by selecting several best paths among multiple available network interfaces to improve data transfer rate to the same multi-homed device. Through the use of path monitoring and packet allotment techniques, cmpSCTP tries to transmit an amount of packets corresponding to the path’s ability. At the same time, cmpSCTP updates the transmission strategy based on the real-time information of all of paths. Using cmpSCTP’s flexible path management capability, we may switch the flow between multiple paths automatically to realize seamless path handover. The theoretical analysis evaluated the model of cmpSCTP and formulated optimal traffic fragmentation of VoIP data. Extensive simulations under different scenarios using OPNET verified that cmpSCTP can effectively enhance VoIP transmission efficiency and highlighted the superiority of cmpSCTP against the other SCTP’s extension implementations under performance indexes such as throughput, handover latency, packet delay, and packet loss.

© 2008 Published by Elsevier B.V.

1. Introduction

Under the current mobile wireless systems infrastructure, quality of service (QoS) of voice over IP (VoIP) is still poor and inconsistent. The degradation in quality of VoIP applications is partly due to variations in delays as well as losses experienced by packets sent through mobile wireless systems. Although the VoIP application can tolerate some degree of missing information, significant losses degrade an application’s QoS. Time-varying path characteristics such as fluctuating bandwidth, delay or loss patterns may degrade the voice quality below an acceptable level. One approach to provide QoS for VoIP applications over the wireless networks is to use multiple paths to deliver VoIP data destined for a particular receiver, i.e., this data is fragmented into packets and the different packets take alternate routes to the receiver. One advantage of this approach is that the complexity of QoS provision can be pushed to the network edge and hence improve the salability and deployment characteristics while at the same time provide a certain level of QoS guarantees.

The common view among researchers of the next generation mobile communication is that it will be a heterogeneous network environment, offering seamless services such as VoIP across multiple wireless access technologies. In the future there will be more multimode devices which...
can access multiple radio access networks. Moreover in the future we will see greater overlap between the coverage provided by the differing access technologies. A host is multi-homed if it can be addressed by multiple IP addresses, as is the common case when the host has multiple network interfaces. Multi-homing is increasingly economically feasible and can be expected to be the rule rather than the exception in the near future. A Multi-homing host may be simultaneously connected through multiple access technologies, and even multiple end-to-end paths to increase resilience to path failure [1]. For instance, a mobile user could have simultaneous Internet connectivity via a wireless local area network using 802.11 and a wireless wide area network using UMTS.

Stream control transmission protocol (SCTP) [2] is the third transport layer protocol to be ratified by the Internet engineering task force (IETF). It was originally developed to transport signaling system 7 (SS7) across IP packet networks. SCTP provides reliable, connection oriented communication to endpoints that may have multiple IP addresses. Allowing a connection to span across multiple IP addresses is known as multi-homing, and it is just one of the features of SCTP which has researchers interested in using it as more than a signaling transport protocol. Yet, the multi-homing feature of SCTP can only exploit at most one of the available paths at any given time. That is, SCTP uses multiple interfaces only for redundancy: every host chooses a primary destination address, normally used for the transmission of the data units, “data chunks” in SCTP terminology, whereas the alternate addresses are considered as secondary, whose conditions are periodically monitored with the transmission of probe chunks called Heartbeat. The backup path is used only (i) to transmit lost data chunks, in order to increase the probability of successful retransmissions, (ii) to transmit new data chunks when, due to the excess of the number of (e.g., five) consecutive timeouts on the primary path, the default interface is declared as “inactive”. In the latter case, SCTP transmits new data chunks toward the backup interface and Heartbeat chunks toward the primary one. As soon as the Heartbeats reception on the primary interface is confirmed, its state is toggled to “active” and the transmission is resumed.

In this paper, we propose and design cmpSCTP, which is designed to use all the available paths for the association at the same time, instead of using only the primary path like SCTP. Therefore cmpSCTP provides full multi-homing support through the simultaneous utilization of all available paths. Furthermore, all available paths are classified as active path or monitor path, and different strategies are introduced for load sharing and redundant transmission according to their connection states. In addition, cmpSCTP monitors the states of the available paths and as their states change, i.e., new paths become active or existing paths break; it updates the transmission strategy, which can be used to switch the flow between multiple paths smoothly. As such, this work naturally leads to another fundamental issue of end-to-end support for seamless handover, specifically for VoIP. Furthermore, we performed the theoretical analysis on the model of multi-path flow control for the first time.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 proposes the concurrent multi-path SCTP, with original contribution presented in detail; and, Section 4 describes analytical models to evaluate the performance of cmpSCTP and compare with other extensions based on SCTP. In Section 5, simulation models are described, and numerical results are presented to investigate the performance. Finally, conclusions and possible future work are described in Section 6.

2. Related work

Research on extending SCTP to support concurrent multi-path transfer, e.g., simultaneously sending data over multiple available paths to increase the association bandwidth, is currently in progress [3–7].

The work in [3,4], by the original SCTP proposers, suggests to change the SCTP sender operation to compensate for the problems introduced by using a unique sequence number space for tracking packets sent over multiple paths. The sender maintains a set of per-destination virtual queues and spreads the packets across all available paths as soon as the congestion window allows it. Retransmissions are triggered only when several Selective Acknowledgments (SACKs) report missing chunks (SCTP protocol data units) from the same virtual queue.

Al et al. [5] propose load sharing–SCTP (LS–SCTP), a mechanism to aggregate the bandwidth of all the paths connecting the endpoints and dynamically add new paths as they become available. Path monitoring is used to distribute packets among all available paths ensuring that the association does not stall because of high loss rates or temporary path unavailability. The key idea is to introduce a per-association, per-path data unit sequence numbering that extends the per-association SCTP congestion control to a finer-grained, per-path congestion control.

Hsieh et al. [6] propose pTCP (parallel TCP) based on the transmission control protocol (TCP). pTCP has two components – striped connection manager (SM) and TCP-virtual (TCP-v). For each pTCP socket opened by an application, pTCP opens and maintains one TCP-v connection for every interface over which the connection is to be distributed on. The TCP-v’s are separate connections that are managed by the SM. This decoupling of functionality allows for intelligent scheduling of transmissions and retransmissions. Similarly, mTCP [7] also significantly modifies TCP to use multiple paths provided by a special routing layer that implements from a resilient overlay network (RON), and also employs mechanisms to handle reordering side-effects. mTCP also outlines a heuristic for the detection of disjoint paths.

However, none of the previous proposals fully addresses the case in which the paths comprised in the SCTP association exhibit widely-different bandwidths and round trip times (RTTs). In such scenario, the packets sent by the source reach the destination out of order, triggering a lot of retransmissions on the underlying TCP SACK on which SCTP is based. The most promising solutions proposed to mitigate packet reordering are based on: (1) estimating the available bandwidth and the RTT on each path,
(ii) using an appropriate packet scheduling algorithm to distribute packets across all paths, so that they reach the destination almost in order.

3. A concurrent multi-path SCTP

As a general remark, we found our modifications to be similar to those recommended by other multi-homing related papers [3–7]. In this section, we will refer to the features of cmpSCTP different with those techniques and present the key design elements of the cmpSCTP protocol.

3.1. cmpSCTP Design

Similar to LS–SCTP [5], cmpSCTP is also based on the idea of separating the association flow control from congestion control. In cmpSCTP the flow control is on an association basis; thus both the sender and receiver endpoints use their association buffer to hold the data chunks regardless of the path on which these data chunks were sent or received. On the other hand, congestion control is performed on a per-path basis; thus the sender has separate congestion control for each path. Especially, the congestion control mechanism on each path can follow the standard SCTP [2], TCP friendly rate control (TFRC) [8] or other congestion control algorithms, so as to insure fair integration with other traffic in the network.

To support the decoupling of functionalities, cmpSCTP uses several novel mechanisms including multi-buffer structure, multi-state management, two-level sequence number, and cooperative SACK strategy to realize effective bandwidth aggregation. Also cmpSCTP includes an overall retransmission technique that prevents the side-effects of simultaneous transmission of data on paths with different characteristics, including unnecessary fast retransmissions, which ensures fast delivery of lost data chunks to prevent stalling the association.

Through extending dynamic address reconfiguration [10], cmpSCTP keeps ongoing end-to-end paths alive and provides adaptive load sharing on multiple paths. In addition, cmpSCTP extends the SCTP path monitoring feature, through regular transmission of actual effective data chunks to update the list of unstable paths suitable for load sharing.

3.2. Function modules and interfaces

Specifically, the structure of cmpSCTP consists of two modules: One is the association management (AM) which carries out the flow control function for an association and the other is the single-path management (SPM) each of which performs single-path congestion control for the path for which it is responsible independently. Fig. 1 provides an overview of the cmpSCTP architecture and key data structures. cmpSCTP as a transport layer protocol interacts with the application and IP, and acts as a container loaded with one AM engine and several SPM engines. When an application opens a cmpSCTP socket, by default one SPM module corresponding to the current network interface in use is created. For each additional interface that becomes active during the lifetime of the association (e.g., when the mobile host moves into the coverage area of another network), the AM module creates one more SPM module. For each interface that becomes inactive, the AM closes the corresponding SPM module and removes it from the association. The number of SPM modules that co-exist in a cmpSCTP association thus depends on the number of active network interfaces used by the application.

The decoupling of functionalities allows network specific congestion control schemes to be used for individual SPM modules without affecting the functionalities performed by the AM module. The AM does not mandate the specific mechanisms used by each SPM module to perform congestion control. Any congestion control scheme proposed for various wireless environments and applications can potentially be plugged into the AM for enhancing the performance of cmpSCTP. Toward this end, the AM provides a well-defined interface with the SPM that does not depend on the details of the congestion control mechanism used. The interface functions between AM and SPM include opening and closing a SPM module, sending and receiving data chunks, and controlling the amount of data sent through each module.

After opening a cmpSCTP socket, the application writes data to the association send buffer using the write() interface. The AM uses the open() interface to initiate association setup in each SPM module that it created. A SPM module returns with the established() call once established. The AM module gives the packet to the SPM module using the send() interface. Upon receiving a cmpSCTP data chunk from the AM module, the SPM module appends the cmpSCTP chunk header to the data chunk, maintains the binding between the application data and the cmpSCTP data chunk in the bindings data structure, builds its own cmpSCTP data chunk based on its state variables such as the congestion window size and initial single-path sequence number, and sends the data chunk to the IP layer using ip-output(). The SPM module continues to issue ip-output() calls until there is no more space left in the congestion window, or there is no unbound data left in the buffer.
When the SPM module receives an incoming data packet via the cmpSCTP-recv() interface, it processes the cmpSCTP packet, updates its state variables, and generates selective ACKs. Each SPM module provides the AM with the data chunk using the receive() interface, and enqueues the cmpSCTP data chunk in the recv buffer using the association sequence number, as will be discussed in Section 3.5. Then the in-sequence data enqueued in recv buffer can be delivered to the application via the read() interface. When the SPM module receives an ACK control chunk from IP, it passes the AM module. Then the AM module processes the ACK and updates state variables of the concerned SPM module. The cmpSCTP selective ACK carries cumulative association level ACK information and receiver window size (rwnd) for each path that can be used to perform per-path congestion control as will be discussed in Sections 3.6 and 3.8. Summing all of the rwnds for each path AM can obtain the association receiver window (arwnd). The APM module calculates its congestion window (cwnd) as given by the congestion control algorithms, and informs the AM module of a new association congestion window (acwnd) value by update_acwnd() function. The SPM uses the shrink() interface to notify the AM of any change in the size of its module (e.g., change in the congestion window). Upon receiving the shrink() call, the AM performs dynamic reassignment as will be discussed in Section 3.10. If any path becomes revived, the AM uses the resume() interface to “de-freeze” the SPM module of the corresponding path, making it start asking for transmissions as before.

Finally, the AM uses the close() interface to tear down a SPM module. Once the SPM module returns with the closed() interface, the AM removes it from the socket. Note that cmpSCTP binds application data to a SPM module only when the concerned module asks for transmissions. Hence adding one more module to the aggregate association has the effect of draining the AM send buffer at a faster rate, and deleting a module implies a slower rate. The dynamic addition or deletion of SPM modules does not influence the functionalities that cmpSCTP performs including reliability.

3.3. Multi-buffer structure

The single buffer architecture of SCTP has been replaced by a multi-buffer structure, meaning that each connection now has its own send buffer, and the total association has a single, shared send buffer. When a new chunk is received from the upper layer, the traffic scheduler is invoked to determine the path it will be sent over, as will be described in Section 4, the chunk is then queued for the chosen path send buffer.

However, a single receiver buffer is presented in cmpSCTP, as in standard SCTP, though each connection in the association is assigned a virtual buffer from the unique receiver buffer for per-path congestion control. In fact, all the chunks from different paths are collected by a single association receiver buffer. At the receiver, the data chunks do not compete for an association buffer, as the sender controls the amount of data injected on all the paths, based on its view of the free space in the receiver’s association buffer (arwnd), as well as the total outstanding data on all the available paths. In order to compensate for the differences in the paths’ RTT, the minimum receiver association buffer should be based on the available bandwidth of all the paths and the maximum RTT, as shown in the following Eq. (1):

\[
\text{Recv Buffer}_{\text{Min}} = B_{\text{avail}} \times \text{RTT}_{\text{Max}}.
\]

where \(B_{\text{avail}}\) is the available bandwidth through all available paths, \(\text{RTT}_{\text{Max}}\) is the maximum RTT.

The multi-buffer structure guarantees connection independence as far as transmission is concerned, but it introduces the need for modifications to selective acknowledgement (SACK) management at the source as will be discussed in Section 3.8. As a matter of fact, SACKs are generated as in AM and transmitted over the path from which the last data chunk was received, but they also carry information about other ongoing connections of the same association, as is the case for delayed acknowledgments. Thus, when a SACK arrives at the source, the information is processed on each interface and the relative send buffer is refreshed.

Since the beginning of the association, each single connection proceeds without interference from other connections, handling only those packets in its send buffer. When a new data chunk needs to be transmitted, it is inserted in the send buffer of the connection indicated by scheduling algorithm (round-robin, bandwidth-aware, or other more optimal algorithm). From then on, it is the connection’s responsibility to ensure that it is delivered: the data chunk remains in the send buffer until acknowledged; it causes head-of-the-line blocking to its own connection, but not to the other connections. When a retransmission timeout occurs, a retransmission process is declared. The cmpSCTP will use an alternative path with lowest packet drop probability. In this case, lost chunks queued in the send buffer of the failed path are shifted into the send buffer of the path chosen for the retransmission.

3.4. Dynamic multi-state management

cmpSCTP is a multi-state transport protocol that creates and maintains one SCTP state for each network interface used by the application. As shown in Fig. 2, a SPM
(SCTP-singlepath) module is created for each active interface used in a cmpSCTP association, to manage the per-path SCTP state including the pair of IP addresses, SCTP ports, and congestion control parameters. While each SPM module (and the corresponding state) is addressed using the conventional connection 4-tuple, a cmpSCTP socket can be addressed through its association id of AM module. All SPM modules in a cmpSCTP association are pointed to the same association id. cmpSCTP dynamically adds or deletes states in an association depending on the connectivity (and the change in addresses) between the end hosts. Therefore, although each state can only be associated with one pair of network addresses specified when the state is created, cmpSCTP allows the addresses of the end points in an association to be changed dynamically. Moreover, as a multi-state transport protocol, cmpSCTP allows multiple SPM modules to co-exist in an association, and hence seamless handover of the transport layer states are possible.

3.5. Two-level sequence number

The standard SCTP uses the transmission sequence number (TSN) as sequence number in its congestion control algorithms. However TSN might simultaneously use only one path although it is used by entire association. Therefore we still continue to use the TSN as the sequence number that can be used for independent congestion control over each path.

The proposed cmpSCTP operates at two-levels, involved in connection level and association level. Every level has its own sequence number. At the connection level, we still use the TSN, which is not used for the entire association, but still represents the sequence of SCTP DATA chunk transmitted through per-path, used for reliability and congestion control on each path. TSN may split the first 4-bit as path ID (PID) denoted the transmission path, the remaining 12-bit part represents the per-path-sequential TSN. Data chunks sent to the same path are assigned the same path ID (PID) and per-path-sequential TSN. At the same time, data chunks sent to different paths carry different PIDs, since TSN is used only by SPM to keep track of the states of data chunks sent through each path and per-path congestion control is activated.

At association level, the association sequence number (ASN) of each data chunk, which is a per-association sequence number, is used to reassemble all received data chunks from different paths to an integrated file. In other words, we use ASN to reorder the received data chunks at the receiver association buffer, regardless the path from which they have been received.

Thus, we defined a new modified data chunk as in Fig. 3, by adding two new parameters to the standard SCTP data chunk [2]. The first parameter is a 4-bit Path Identifier (PID), which identifies the path used for the data chunk transmission. The second parameter is a 16-bit association sequence number (ASN), which is a monotonically increasing sequence number for the data chunks transmitted over the association. In addition, cmpSCTP continues to use the SSN for ordering the data chunks within the association streams.

3.6. Independent congestion control

In order to perform per-path congestion control, cmpSCTP uses a number of internal variables to control the rate at which data is injected into the network. These parameters for each path are as following; all are maintained in a send buffer and receiver buffer, respectively. In addition, these buffers keep track of the TSNs sent and received on each path.

- rwnd – this corresponds to a sender’s view of the receiver’s incoming buffer space. Summing all of the rwnds can obtain the available space in the receiver’s association buffer and is kept on an association basis.
- cwnd – this corresponds to the sender’s view of network conditions. The initial value of cwnd is less than or equal to twice the path maximum transmission unit (MTU), corresponding to the most aggressive value recently adopted for SCTP.
- Slow-start threshold (ssthresh) – the sender uses this to distinguish between slow-start and congestion avoidance phases.
- Partial bytes acked – the sender uses this to limit cwnd adjustment to only once per RTT during congestion avoidance phases.

The congestion control algorithms of slow-start, congestion avoidance, fast retransmission, fast recovery and timeout retransmission may used by the standard SCTP will all be applied to each path, on the other hand, other congestion control schemes, for example TFRC [8] and wireless TCP, etc., may also be used and enable congestion control for special applications like VoIP and various scenarios.

The per-path receiver still uses TSN to facilitate its operations. When a receiving DATA chunk advances the TSN for a path and current receiver window of that path is being fully utilized, the receiver window of that path is allowed to be increased by at most the lesser of (1) the total size of the previously outstanding DATA chunk(s) acknowledged at that path, and (2) the destination’s path MTU. Then the sender sends per-path receiver window reports.
from the receiver through modified cmp-SACK as described in Section 3.8, all per-path congestion window are refreshed.

3.7. Decoupling of functionalities

The design of cmpSCTP is the key to decoupling of functionalities. cmpSCTP decouples the transport layer functionalities associated with single-path characteristics from those pertaining to the whole association. The SPM opened for each path handles the single-path state, while the AM handles the whole association. For example, congestion control, the mechanism that estimates the available bandwidth along a path, is a single-path functionality and is handled by the SPM. On the other hand, the application interacts with the transport layer through the socket buffer, and hence the buffer management (including data resequencing) is handled by the AM.

The SPM is a slightly modified version of SCTP that handles only one path connection. Since cmpSCTP controls the association send and receive buffers, the SPM only needs to handle per-path send and receive virtual buffers and perform the regular congestion control as SCTP does. On the other hand, all incoming packets received and fragmented by cmpSCTP will be scheduled to the individual SPM module. As we show in Fig. 1, the process is possible since the SPM module is the only component that interacts with the network layer (IP). An advantage resulting from the decoupling of functionalities is that intact cmpSCTP packets generated by a SPM module with the same TSN can be bound to different paths. However, a retransmission is not at individual SPM modules, but needs the AM module with ASN.

3.8. Handling of cmp-SACK chunk

In order to acknowledge the received data chunks, cmpSCTP defines concurrent multi-path SACK (cmp-SACK) as Fig. 4. The cmp-SACK chunks, which can also be called ASN-based SACK, received by a cmpSCTP source endpoint actually reflect the reception of ASNs sent through all the paths used by a cmpSCTP association. The cmp-SACK chunk includes four different parameters than the standard SCTP SACK: Cumulative ASN ACK, Time Stamp, Path ID and advertised receiver window credit.

- The cumulative ASN Ack is a per-association cumulative acknowledgement instead of cumulative TSN ACK in the standard SCTP SACK, which records the highest sequential ASN received.
- Time stamp is used to order the cmp-SACKs received from the different paths.
- Every path ID and its corresponding advertised receiver window credit, reflects the current capacity of each receiver’s inbound virtual buffer.
- A sequence of gap Ack blocks records any out of sequence ASNs received.
- A sequence of duplicate ASNs records any ASNs for which duplicates have been received.

Data is not considered fully delivered until the cumulative ASN Ack point advances past its ASN. Thus the information in the Gap Ack Blocks corresponds to SCTP SACK blocks. Under normal conditions cmpSCTP uses a delayed acknowledgement scheme which sends one cmp-SACK for every mean RTT (RTTmean) within the association incoming packet which contains one or more new data chunks.

Considering the chunks of the same ASN that it is possible to receive from different paths on a certain redundant transmission strategy, the acknowledgement mechanism should be designed based on ASN to prevent unnecessary retransmissions unless data chunk losses occur on every transmission path. Fast retransmission is triggered by four consecutive duplicate SACKs in an association. Whenever a data chunk needs to be retransmitted, cmpSCTP will use an alternative path with the packet lowest drop probability. Since in our concurrent multi-path transfer, current cwnds of all paths are available, such a modified retransmission scheme is more efficient than that defined in RFC 2960: selected alternative path uses current cwnd to retransmit all lost packets. In such a strategy, there are N disjointed paths between sender and receiver to transmit simultaneously and retransmit lost packets via alternative path with lowest packet drop probability.

A difference compared with the standard SCTP SACKs run by each connection is that when the SACK is received, it is processed only for those paths whose carried chunks were acknowledged by the SACK itself; the cmp-SACKs return on the path from which the last data chunk was received, but it is possible that a cmp-SACK is reporting information about other ongoing connections of the same association. Thus, when a cmp-SACK arrives at the source, the information is processed at each interface and, consequently, all send buffers are refreshed. Since the information of the receiver window is required by the sender per-path prior to the congestion window increase, this change avoids that a connection parameter be out of date due to its RTT being too large.
3.9. Flexible multi-path management

The mobile SCTP (mSCTP) [9] is the SCTP with the ADDIP extension. It utilizes dynamic address reconfiguration [10] to manage the possible changes of IP addresses such as adding new addresses and deleting obsolete addresses while keeping ongoing end-to-end connections alive. However, the current SCTP specification designates only one path at each destination host as the primary path, and all new data is transmitted to only the primary path. If ever the primary path fails, new data transmission hands over to an alternate reachable destination path. Furthermore, the SCTP association experiences a reduction of traffic rate immediately after handover regardless of the available traffic state of the new primary path.

In order to mitigate these negative effects, we refer to this new extension as cmpSCTP. It aims to achieve higher throughput and seamless handover in an SCTP association by concurrently using all independent paths between a sender and receiver for data transfer.

By cmpSCTP, we mean that the mobile host is maintaining connections with more than one path. These paths are classified by the following two sets.

The *Active Set* includes the paths that form a cmpSCTP connection to the mobile host, which allows transferring data packets for the MH.

The *Monitored Set* is the list of candidate paths that the mobile hosts continuously measures, but whose bandwidths are not sufficient to be added to the *Active Set*.

The cmpSCTP handover support can be achieved by using modified address configuration change (ASCONF) and address configuration acknowledgement (ASCONF-ACK) control chunks of mSCTP, which may contain the four new modified and appended request parameters for the paths. These parameters signal:

- **0xC001-AddPath**: the path specified is to be added to the *Monitored Set*;
- **0xC002-DeletePath**: the path specified is to be removed from the *Monitored Set*;
- **0xC007-ActivePath**: the path specified is to be added to the *Active Set*;
- **0xC008-DeactivePath**: the path specified is to be removed from the *Active Set*.

In a handover situation, MH sends CT ASCONF chunks with these four different types of parameters. To add a new path, for example, the MH should send ASCONF chunk of the 0xC001 type, which should be acknowledged by an ASCONF–ACK chunk. Through using the concurrent multi-path feature, cmpSCTP could reduce latency and increase throughput during performing vertical handover. This means that packets are not lost during the handover and there is no interruption to service, making it suitable for handover of real-time VoIP traffic.

Commonly, the monitored path is assigned the probing data have been originally sent to the active path. Such data chunks can be carried by multiple paths to minimize the impact of delayed loss recovery. Specifically, the data carried by the first path that recently suffered from a blackout will be bound to another path, such that the association can continue receiving data while the concerned path probes for the duration of the blackout.

The multi-path management (MPM) algorithm determines the *Active Set* that is likely to yield the maximum throughput for upper application. The sender assigns a sending rate to each of the active paths. The rest of the candidate paths are kept in a *Monitored Set*, from which replacement paths will substitute for failed or degraded paths from the *Active Set*. These unstable paths are put into the *Monitored Set* to detect their available network condition. These paths are periodically checked against the so called “triggering conditions”. If a triggering condition is fulfilled, the MH decides if a path should be added to the *Active Set*. Then, the MH creates a report which is sent to the sender. The strategy continues as long as there is no need to switch to a different *Active Set*. A switch is needed if a path fails or the network path becomes congested. At that time, the topology is updated with new values. The algorithm is described in Fig. 5.

---

1. Assume MH is connected to any mobile network, then MH find out another mobile networks
2. MH enters concurrent multi-path transfer
   - MH keeps an association to several of paths simultaneously
   - AddPath (the new path);
   - MH connects k available paths including an *Monitored Set* (n candidate paths) and *Active Set* (m stable paths)
3. MH measures the path rate received from the *Monitored Set* and *Active Set* paths periodically.
4. For i = 1...n
   - If (Monitoring Set.Path.state==CON_AVOID & & Expectation of recently *Monitored Set*.Path, rate) > Rate threshold & & Variance of recently *Monitored Set*.Path, rate < Variance threshold)
     - Activepath(*Monitored Set*.Path);
5. For j = 1...m
   - If (Active Set.Path.state==CON_AVOID || Expectation of recently *Active Set*.Path, rate) < Rate threshold || Variance of recently *Active Set*.Path, rate > Variance threshold)
     - Deactivepath(*Active Set*.Path);
6. Update_Fragmentation_Strategy (*Active Set*);
7. If Receive(*Path*.state==LOSS) DeletePath(*Path*);
8. When MH only connected one mobile network, MH leaves concurrent multi-path transfer

---

Fig. 5. Multi-path management algorithm.
3.10. Effective bandwidth aggregation

While the multi-state design of cmpSCTP allows it to leverage seamless handover when the mobile host migrates from one access network to another, it also introduces problems of its own. Specifically, since cmpSCTP aims to provide the reliable and in-sequence semantics that SCTP provides, head-of-line blocking due to mismatches in the capacities of the individual SPM modules to deliver data can aggravate the performance. When multiple modules of different bandwidths or round trip times share the same send and resequencing buffers to collectively deliver one single stream of data, packets held up in the slower modules prevent packets arrived through the faster modules from being consumed by the application (waiting for the head-of-line "hole" to be filled), thus potentially causing buffer overflow and then association stalls after flow control kicks in [6]. In the target environment, the fact that multiple modules belong to heterogeneous wireless networks and exhibit very different characteristics in networks and exhibit very different characteristics in terms of bandwidth fluctuations and blackouts further aggravates the problem.

4. Analytical modeling

For the sake of illustration, consider two network hosts, host A and host B, where host A is sending data to host B. Assume there are K distinct paths from A to B, where the time needed to send a packet along path j is a linear function of flow size.

It is assumed that the path in Active Set has reached a steady state, that is, its transport protocol can be modeled with a Markov chain; for reference [11,12]. We consider the arrival traffic rate at a link queue (segments/s) as queue with Poisson arrivals with rate \( \lambda \). The service time of packet \( i \) is assumed to have an exponential distribution with rate \( \mu_i \), where \( \mu_i \) are independently and identically distributed (i.i.d.) random variables with an arbitrary distribution. The service times of packets are assumed to be independent. Once a packet has entered the queue, it does not leave until it completes service. The queue can be analyzed by considering a single cycle consisting of a busy period and an idle period.

As the utility for each path, we use the power for the path. Since the throughput is equivalent to the rate of the arrival flow (i.e., the arrival rate), from [13], the overall power \( P \) of the system and the power \( p_i \) of path \( l \) are given as

\[
P = \sum p_i, \quad p_i = \begin{cases} 
\frac{\lambda}{T(n)} & \text{if } \lambda \leq n\mu \\
0 & \text{otherwise}
\end{cases}
\]

and

\[
p_i = \begin{cases} 
\frac{\lambda}{T(n)} & \text{if } 0 \leq \lambda \leq n\mu - \sum_{j \neq l} \lambda_j \\
0 & \text{otherwise}
\end{cases}
\]

where \( T \) denotes the mean delay time in the system and \( l = 1, 2, \ldots, n \), respectively. Note that \( \lambda = \sum\lambda_i \), we have

\[
P = \sum p_i, \quad p_i \text{ is positive for } \lambda \leq n\mu, \text{ and zero for } \lambda > n\mu.
\]

We can formulate, for the system based on cmpSCTP described above, two typical optimal flow control schemes: the noncooperative scheme (I) and the overall cooperative scheme (II). The two schemes in different queuing system are presented as follows:

1. The noncooperative scheme: Each path strives to maximize unilaterally its own power. That is, the noncooperative scheme is to find \( \lambda^*_l \) for each \( l = 1, 2, \ldots, n \), that satisfies

\[
p_l' = \max_{\lambda \geq 0} p_l(\lambda^*_l, \lambda^*_2, \ldots, \lambda^*_n).
\]

2. The overall cooperative scheme: A single agent maximizes the overall power of the system, that is, it strives to find \( \lambda^* \) that satisfies

\[
p^* = p(\lambda^*) = \max_{\lambda \geq 0} p(\lambda).
\]

Based on the above definitions, we consider two queuing system models as shown in Fig. 6: \( S_0 \) - a system consisting of \( n \) separated \( M/M/1 \) queues (top) and \( S_n \) - an \( M/M/n \) queue (bottom). In the former, the flow control schemes are concerned with each separated \( M/M/1 \) queue. On the other hand, in the latter, the flow control schemes are concerned with the \( M/M/n \) system, where \( K \) is the number of active paths that are serviced at the same time over specific network. We recall that the latter is the result of grouping together the former.

For \( n \) separated \( M/M/1 \) queues, an arrival traffic rate that is mutually independent and forms a Poisson arrival process with rate \( \lambda_i, l = 1, 2, \ldots, n \). Note that \( \lambda = \sum\lambda_i \) and that

Fig. 6. Model of cmpSCTP queuing system.
the transmission time is independent of the player that sends the job.

Assume the average arrival rate of every queue is $\lambda' = \frac{\lambda}{n}$.

From $M/M/1$ queuing theory, System utilization is $\rho' = \frac{\lambda'}{\mu}$, and Probability of no packet in the system is $P_0 = 1 - \rho'$.

The average number of packets in the system is

$$L_q' = \frac{\lambda'^2}{\mu(\mu - \lambda')}.$$  \hspace{1cm} (6)

Further, by Little's law, the average delay in multi-queues system is

$$T' = \frac{1}{\mu - \lambda'}.$$ \hspace{1cm} (7)

For an $M/M/n$ queue, the system utilization is $\rho = \frac{\lambda}{m\mu}$, and Probability of no packet in the system is

$$P_0 = \frac{1}{\sum_{\lambda=0}^{\infty} \frac{(\lambda/\mu)^\lambda}{\lambda!(1-\rho)}}.$$ \hspace{1cm} (8)

The mean number of segments in the queue can be calculated as

$$L_q = \frac{(\lambda/\mu)\rho}{n(1-\rho)}P_0.$$ \hspace{1cm} (9)

By Little's law, we can model the average delay in system as

$$T = \frac{1}{\mu}L_q + \frac{1}{\mu}.$$ \hspace{1cm} (10)

According the queuing theory, the system power of the noncooperative scheme consisting of $n$ separated $M/M/1$ queues is not as good as the cooperative scheme of $M/M/n$ queue, that is $p' < p$. Thus from the view of the whole system flow model, cmpSCTP needs to use the following mechanisms to achieve the overall cooperative scheme avoiding the noncooperative scheme:

1. Delayed fragmentation: In cmpSCTP, data will be bound to a path only when the data is ready to transmit. cmpSCTP does not allow data to be queued up inside each path as soon as possible.
2. Dynamic reassignment: If a path reports losses or bandwidth fluctuations, cmpSCTP immediately unbinds the data that is lost or overflows, such that other paths ready for transmission can timely deliver the concerned data.
3. Cooperative retransmission: When a data segment can be lost through one path, cmpSCTP will assign another most reliable path to carry the retransmission data segment in order to minimize the impact of data loss.

5. Simulation and results

The proposed cmpSCTP protocol has been implemented in the network simulator OPNET [14], and tested with various network configurations. The purpose of the extensive simulations is two-fold: first to investigate the performance of the proposed cmpSCTP with various network parameters, and second to compare the cmpSCTP protocol with other multi-path transport protocol.

In our simulation, we created a network topology consisting of two hosts. To investigate the impact of various network parameters on the performance of the cmpSCTP, the multiple overlapping cells are also varied by using different simulation configurations including the number of overlapping cells, and available bandwidth in the cell the mobile host is entering. Available bandwidth in a cell is varied by changing the average of the Poisson distribution used to generate background traffic in all of cells.

5.1. Effect of path number

In this preliminary experiment, we tested the capability of the cmpSCTP protocol to provide load sharing over multiple paths. We created the network including multiple overlapping cells consisting with several different base stations.

In Fig. 7, referring to the four-path case, we can see that using cmpSCTP as the transport layer, the traffic load is shared among the available paths. Along with the number of paths, the per-path throughput lessens little due to the overhead and network congestion, and the association throughput reaches maximum value.

Moreover, Fig. 8 shows the delay of packet at the receiver depending on the number of paths between the two nodes. In all cases, the delay stays below 160 ms. This, together with measured average delays in the 40–160 ms range, lead to excellent performance for VoIP applications. The more the number of paths introduced, the less every packet contributes to delay. That is to say, we can achieve the more stable performance through the concurrent multi-path transfer of cmpSCTP.

5.2. Impact of bandwidth disparity

In this experiment we tested the robustness of the cmpSCTP protocol in dynamic conditions. We assumed that we have two paths, namely path 1 and path 2, in order
to examine the performance of cmpSCTP under the condition of paths with diverse capacities. The speed of mobile host movement is assumed to be 15 m/s (meters per second) and the RTT is assumed to be 60 ms. In case 1, the mobile host moves from a cell with a larger amount of available bandwidth (2 Mbps) to a cell with a smaller amount of available bandwidth (1 Mbps). In case 2, a mobile host moves between two homogeneous cells with the same amount of available bandwidth (2 Mbps). In case 3, a mobile host moves from a cell with a smaller amount of available bandwidth (2 Mbps) to a cell with a larger amount of available bandwidth (5 Mbps). In our performance study we used the association throughput as a performance metric, which is defined as the amount of data delivered to the receiver’s application layer per second. Simulation results are presented to evaluate the performance of the cmpSCTP.

Fig. 9 shows the association throughput during handover. The x-axis in Fig. 9 represents time, while the y-axis represents the effective association throughput excluding the duplicate packets. As can be seen from Fig. 9, despite the difference in the bandwidths of the paths, the association throughput achieved by cmpSCTP is close to the ideal throughput during handover. The high throughput achieved by cmpSCTP is due to its striping mechanism that is based on the rate of the bandwidth of the paths.

5.3. Sensitivity to the schedule strategy

In our simulation, we created a cmpSCTP association consisting two paths between the two multi-homed cmpSCTP hosts: cmpSCTP source and destination, for which available bandwidth is 1 Mbps and 2 Mbps, respectively. As the above analysis in Section 4, we simulate the LS–SCTP or pTCP as n independent M/M/1 queues, and the cmpSCTP approaches to an M/M/n queue model, which the n is 2. The simulation result represents the association throughput and the average delay of every packet or packet’s sojourn time.

As can be seen from Fig. 10, all three schemes show relatively small fluctuations in throughput. The proposed cmpSCTP shows higher throughput due to a series of efficient improvement mechanisms.

Fig. 11 shows the delay performance by using the concurrent multi-path transfer feature. We can see that the end-to-end delay during performing cmpSCTP is much lower than the other multi-path transport protocol, while in most situations it brings noticeable improvements in jitter, packet loss percentage and reordering delays at the receiver.
6. Conclusion and future work

In this paper, we proposed a new cmpSCTP protocol to better support reliable VoIP applications over heterogeneous wireless networks. cmpSCTP keeps two or more end-to-end paths concurrent, transferring new data from a source to a destination host, and distributes the data on the available paths based on an estimation of the available bandwidth of each path. We presented the design and details of the proposed approach, and evaluated its performance through simulation experiments. Our simulation results demonstrated that cmpSCTP can lead to satisfactory performance which is able to utilize the available bandwidth efficiently. We compared the performance of cmpSCTP with LS–SCTP and pTCP. Results show that cmpSCTP dramatically outperforms LS–SCTP and pTCP in terms of throughput and delay, especially in a heterogeneous wireless environment.

Further investigation is planned to address some of the issues associated with impact studies on the other path factors such as packet loss rate and cost, and the mechanisms to mitigate it. Also, the assumption of independent paths is being dropped; we then plan to enable the sender to dynamically decide between either shared and distinct congestion control across paths through incorporating an end-to-end bottleneck detection mechanism [15]. The analysis and evaluation of these issues is our future work.

Acknowledgement

This work was jointly supported by: (1) National Science Fund for Distinguished Young Scholars (No. 60525110); (2) National 973 Program (No. 2007CB307100, 2007CB307103); (3) Program for New Century Excellent Talents in University (No. NCET-04-0111); (4) Development Fund Project for Electronic and Information Industry (Mobile Service and Application System Based on 3G); (5) National Specific Project for Hi-tech Industrialization and Information Equipments (Mobile Intelligent Network Supporting Value-added Data Services). We express our gratitude to the Research Institute of China Mobile. Thanks also to Jing Wang, Tong Xu, Yan Zhang, Cong Liu and the anonymous reviewers for their helpful suggestions which helped to improve the paper.

References


Jianxin Liao was born in 1965, obtained his Ph.D. degree at University of Electronics Science and Technology of China in 1996. He is presently a professor of Beijing University of Posts and Telecommunications. He has published hundreds of papers in different journals and conferences. His research interests are mobile intelligent network, broadband intelligent network and 3G core networks.

Jingyu Wang was born in 1978, He is currently working toward the Ph.D. degree at the Department of Computer Science and Technology, Beijing University of Posts and Telecommunications. His research interests span broad aspects of performance evaluation for Internet and overlay network, traffic engineering, image/video coding, multimedia communication over wireless network.

Xiaomin Zhu was born in 1974, obtained his Ph.D. degree from Beijing University of Posts and Telecommunications in 2001. His major is Telecommunications and Information Systems. His research interests span the area of intelligent networks and next generation networks with a focus on Internet interworking and protocol conversion. He has published over 110 papers, among which there are more than 30 first-authored ones, in different journals and conferences. His personal home page is http://zhuxm.ik8.com.