

A Network-Controlled Architecture for SCTP Hard Handover

Khadija Daoud, Karine Guillouard, Philippe Herbelin

Orange Labs, Issy Les Moulineaux, France
{first name.last name}@orange-ftgroup.com

Noël Crespi

Institut Telecom, Telecom SudParis, Evry, France
Noel.crespi@it-sudparis.eu

Abstract—SCTP faces three performance problems in hard handover scenarios: high network handover delay, high transport handover delay and throughput under-utilization. Existing solutions assume the mobile extension of SCTP (m-SCTP) as a unique way to handle handovers and rely on terminal mechanisms. Nevertheless, they are not efficient as they lack the necessary information to perform on-time and fine SCTP configuration tuning during handovers. We propose in this paper an overall mobility management architecture (UFA) replacing m-SCTP. UFA uses network-controlled mechanisms and SIP protocol to reduce the network handover delay and drive the optimal SCTP configuration on both SCTP endpoints. We simulate and compare UFA and m-SCTP. Results are promising and show better performances for UFA, even when compared with enhanced m-SCTP solutions.

Keywords—component; m-SCTP, SIP, UFA (Ultra Flat Architecture).

I. INTRODUCTION

The proliferation of heterogeneous access technologies and the evolution of mobile terminals towards multi-interfaced devices have influenced most of research effort to consider soft handovers only. However, hard handovers should not be neglected for the following reasons: 1) depending on user locations multiple access technologies may or may not be available; 2) even in heterogeneous environments users may choose to activate only one access technology on their devices because of battery life, cost or network capacity criteria.

A large part of applications such as web, file transfer or streaming require reliable data transfer. These applications are based on connection oriented protocols like TCP or SCTP (Stream Control Transport Protocol) [1]. SCTP is currently gaining more and more interest given its advantages. Firstly, it encompasses basic functionalities of TCP and adds a number of mechanisms like multihoming and path failure detection. Secondly, its mobile extension (m-SCTP) [2] may be applied in any network architecture in a simple and scalable way. Indeed, m-SCTP uses end-to-end signaling messages which enable to get rid of any additional network nodes as required for other mobility protocols like MIP [3] and its variants. Despite these advantages, SCTP applications combined with m-SCTP still face performance problems in hard handover situations: high network handover delay [4], high transport handover delay [4] [5] and throughput under-utilization [6].

Based on a thorough analysis of these problems in section II and the state of the art in section III, we underline in section IV the limitations of current solutions based on terminal mechanisms and identify the need of an overall mobility management architecture with network-controlled mechanisms.

We then propose in section V a new architecture (UFA) and compare its performances to m-SCTP based solutions in the final section VI.

II. SCTP PROBLEMS DURING HARD HANDOVER

A. SCTP and m-SCTP overview

Multihoming feature brought by SCTP enables to establish an SCTP association between two endpoints taking into account a set of IP addresses and an SCTP port for each of these endpoints. Then, a path from one endpoint to a destination endpoint is characterized by one of the destination endpoint addresses. An endpoint chooses one path (called primary path) for data sending, the other paths (called secondary paths) being only used to retransmit data lost over the primary path. Like TCP, SCTP rely on specific data transmission and congestion control mechanisms to ensure reliable data delivery and efficiently use the available network resources. As paths may have different congestion states, SCTP sender separately maintains for each of them a set of congestion control parameters. These are the congestion window (cwnd), the slow-start threshold (ssthresh) and the Retransmission TimeOut (RTO). Cwnd limits the size of data a sender can send over a particular path without requiring any acknowledgement (SACK). To transmit data over a given path, SCTP first sets the congestion parameters to their default values (cwnd=2MTU¹, ssthresh=65536bytes, RTO=3s) and enters in slow start mode during which cwnd size increases exponentially. When cwnd reaches ssthresh, SCTP switches to congestion avoidance mode during which cwnd increases linearly. To control data delivery over a given path, the sender triggers a retransmission timer (T3-rtx) each time a packet is sent on that path. When T3-rtx reaches RTO (T3-rtx expiration) and data is still not acknowledged (non reception of SACK), the packet is considered as lost and SCTP falls back to slow start mode on that path with cwnd equal to 1 MTU, ssthresh divided by two and RTO doubled. The lost packet is retransmitted on a secondary path considering the congestion parameters of that path. On the other hand, if the sender receives 3 SACKs indicating that a given packet is missing (through GAP ACK Blocks field [1]) and the current T3-rtx has not expired yet, SCTP switches to fast retransmit mode by immediately retransmitting the missing packet on a secondary path without waiting for the expiration of the current T3-rtx.

To handle mobility, a new extension of SCTP called Mobile SCTP (**m-SCTP**) has been introduced in [2]. It makes possible the dynamic addition and deletion of IP paths to an

¹ MTU: Maximum Transmission Unit

established association through the usage of m-SCTP signaling messages (ASCONF). When a Mobile Node (MN) acquires a new IP address, it sends an ASCONF (ADD IP) to its Corresponding Node (CN) so that CN can consider the new address as a secondary path. Then, if MN wants that its new address is considered by CN as a primary path, it sends to CN an ASCONF (SET PRIMARY). After receiving an ASCONF (ADD IP), CN performs path verification towards the new address. Path verification consists in sending a HEARTBEAT message to the new address and waiting for the reception of HEARTBEAT ACK. On the MN side, MN is allowed to use the new address for data sending only after receiving the ACK response to ASCONF (ADD IP), confirming that CN has received the ASCONF (ADD IP).

B. SCTP problems

SCTP encounters a set of problems during hard handover when used with m-SCTP. We consider scenarios where MN is receiving data from CN and performing hard handover from one gateway (GW) to another one, both GWs belonging to different IP subnets. As illustrated in Figure 1, hard handover includes different time periods: (D_1) the time necessary for MN to detach from the Source GW (GW_S) and attach at Layer 2 to a Target GW (GW_T), (D_2) the time necessary to receive IP subnet information from GW_T through Router Advertisement (RA) [5], (D_3) the time necessary for MN to configure its interface with the new IP address and be able to use it (including DAD), and (D_5) the time necessary to exchange m-SCTP signaling messages (ASCONF (ADD-IP), ASCONF ACK, HEARTBEAT, HEARTBEAT ACK, ASCONF (SET-PRIMARY), ASCONF ACK). Network handover delay is the Layer 2 and Layer 3 disconnection delay and is equal to ($D_1+D_2+D_3$). Transport handover delay (SCTP_HO_Delay) is the delay perceived by SCTP layer on MN side and represents the time between the last packet received before HO and the first packet received after HO.

Typical values for network handover delay ($D_1+D_2+D_3$) vary between 1.5s and 3s [5]. These values directly impact the transport handover delay (see Figure 1). Indeed, during ($D_1+D_2+D_3$) period, CN continues to transmit data on MN old address which is no more reachable. Each time T3-rtx expires, CN retransmits the non acknowledged packet towards the same address (the only one known by SCTP) after setting ($cwnd=1MTU$, $ssthresh=ssthresh/2$, $RTO=RTO*2$) and arming T3-rtx with the new RTO value. When m-SCTP signaling is received by CN, CN cannot transmit data on the new MN IP address before T3-rtx expiration on the old MN address [1], [4], [5]. Thus, the larger the network handover delay ($D_1+D_2+D_3$) is, the higher the number of T3-rtx expiration is, the higher RTO is, and the higher the transport handover delay is.

On the other hand, if T3-rtx has never expired when m-SCTP signaling is received by CN (i.e $D_1+D_2+D_3 < 1s^2$), CN transmits immediately new packets towards the new MN address without waiting for T3-rtx expiration [1], [4]. The SACKs generated by MN after receiving these new packets indicate the missing packets lost during handover. This trigger

on CN fast retransmits to recover all of the missing packets. During the recovery period, $cwnd$ associated with the new path remains constant [1] [7]. Consequently, even for low network handover delay SCTP performances are decreased due to packet losses.

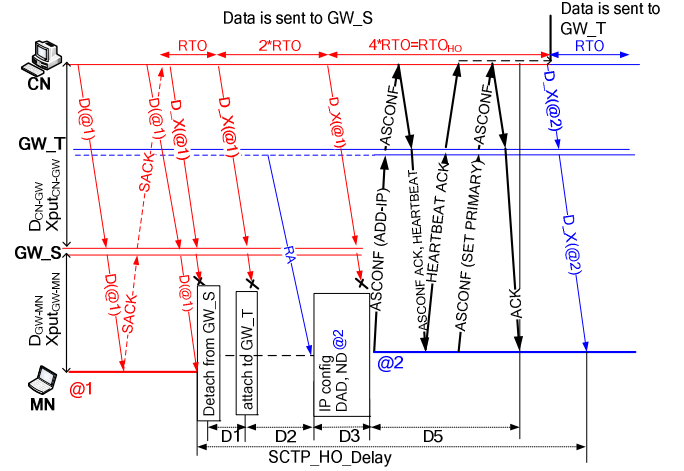


Figure 1. Handover delay components with m-SCTP

Another problem encountered by SCTP concerns throughput under-utilization on the new path after handover. Indeed as this path is initiated with default values for SCTP congestion control parameters ($cwnd=2MTU$, $ssthresh=65536$, $RTO=3s$), a period of time is necessary for the $cwnd$ to reach the optimal value enabling the maximum usage of the network resources available on the link. This value is equal to the product of the bottleneck link throughput ($Xput$) and the sender-receiver link round trip time (RTT), named BDP (Bandwidth Delay Product) [8].

III. RELATED WORK

Regarding the long transport handover delay problem [4] and [5], supposing a high network handover delay, propose a solution based on m-SCTP that triggers a particular SCTP configuration we call **m-SCTP+**. Upon the reception of ASCONF (SET-PRIMARY), CN immediately retransmits data without waiting for the current T3-rtx expiration. The disadvantage of such solution is that it requires cross-layering mechanisms within MN to detect hard handovers and send ASCONF (SET-PRIMARY) message in addition to ASCONF (ADD-IP). Moreover, as hard handover is not the only case where ASCONF (SET PRIMARY) may be sent, triggering m-SCTP+ configuration may be inappropriate. [4] additionally shows enhanced performances for SCTP in case of low network handover delays; however it does not indicate the mobility architecture enabling such low delays.

Link throughput under-utilization problem is also encountered by TCP. Work in [9] addresses this problem as well as the long transport handover delay issue. It proposes a TCP-HO solution where MN reports to CN handover events and the BDP of the new link. CN stops transmission during one RTT and then begins transmission with $cwnd$ equal to BDP. The disadvantage of this solution is that it does not specify how MN gets the BDP of the new link and does not take into account the delay necessary to get this information from the network. Reference [6] has the same disadvantage.

² minimal RTO value is 1s [1].

IV. REQUIREMENTS FOR A NEW SOLUTION

M-SCTP is only an end-to-end mobility signaling protocol: it does not provide tools to optimize mobility execution and relies on terminal mechanisms. The above analysis has proven that terminal mechanisms are not sufficient to deal with SCTP problems as they lack the necessary information to perform on-time and fine SCTP configuration tuning. Therefore an optimized mobility management architecture with network controlled-mechanisms driving SCTP configuration shall be defined. The first requirement of the target architecture is to reduce the network handover delay being the cause of the long transport handover delay. The second requirement is to support a proactive mechanism able to determine the throughput available on the target link without impacting the network handover delay. The third requirement is to provide SCTP with explicit triggers regarding handover events and adequate SCTP configuration containing new endpoint IP addresses and adapted congestion control parameters. This latter requirement could not be performed without using a dedicated signaling protocol that interacts with SCTP protocol. We decide to conceive the dedicated signaling protocol using SIP [10].

V. UFA: A NETWORK-CONTROLLED ARCHITECTURE SOLVING SCTP PROBLEMS

UFA (Ultra Flat Architecture), presented in [13] and [14] meets a part of the requirements discussed in the previous section: 1) it implements network-controlled cross-layer techniques driving terminals' configuration at all layers; 2) it provides a mobility procedure with proactive mechanisms and a reduced network handover delay; 3) it relies on SIP which is suitable to transport explicit triggers and information. The network control is enabled through the implementation in the UFA gateway (UFA_GW) of a SIP Back-To-Back User Agent (B2BUA) that modifies and generates SIP messages. UFA as described in [13] and [14] was introduced in order to solve scalability issues in mobile networks and support mobility procedures integrating QoS. However, it was defined only for SIP native applications based on RTP [15] (e.g. VoIP). In this paper, UFA is evolved to support non SIP native applications based on SCTP.

A. Support of SCTP applications by UFA

Each SCTP application is managed through a SIP session. The SIP session transports the SCTP application characteristics to the UFA_GW so that it can control the handover of this application. A SIPcrossSCTP (SxS) module is implemented within MN, CN and UFA_GW to maintain the binding between SIP sessions and SCTP applications and ensure the interaction between them.

UFA mobility procedure is based on two phases as shown in Figure 2: 1) a preparation phase (messages 1, 2, 3, 4) aiming at pre-determining the MN OSI layers configuration after its HO and the new CN SCTP layer configuration due to MN HO; 2) an execution phase (messages 5, 6, 7, 5A, 6A, 7A, 8) aiming at providing MN and CN with the predetermined configuration. Both phases are controlled by the source UFA_GW (UFA_GW_S) as detailed hereafter. When UFA_GW_S anticipates the need of HO for MN because of coverage loss, it sends to a set of candidate UFA_GWs a RESOURCE QUERY REQUEST (1) that includes application characteristics, user

profile, etc. Each of these candidates answers in RESOURCE QUERY RESPONSE (2) with *Allocated Throughput* ($Xput_{GW_T-MN}$) according to the received information and available resources. UFA_GW_S then selects a target UFA_GW (UFA_GW_T) and pursues HO preparation by sending CONTEXT TRANSFER (3) to UFA_GW_T. UFA_GW_T pre-determines an IP address for MN (*Add_IP_Addr*), checks its uniqueness, confirms the *Allocated throughput*, calculates the associated BDP_{GW_T-MN} and includes them with other UFA_GW_T related information in ACK message (4) towards UFA_GW_S. Based on the received message, UFA_GW_S builds two SIP re-INVITE messages (5, 5A) sent to CN and MN respectively. SIP Re-INVITE (5) message towards CN includes *UFA_Appli_Config* header and BDP_{GW_T-MN} . SIP Re-INVITE (5A) message towards MN includes the same *UFA_Appli_Config* header and *UFA_Terminal_Config* header:

- *UFA_Appli_Config* header is depicted in Table 1. It indicates the new SCTP association addresses. With this header m-SCTP signaling is no more needed: the reception of message 7A by MN directly validates *Add_IP_Addr* as a new source address; and the reception of message 7 by CN directly validates *Add_IP_Addr* as a new destination address to MN.

- *UFA_Terminal_Config* is depicted in Table 2. It contains the reconfiguration necessary for MN Layer 2 and Layer 3 to handover to UFA_GW_T.

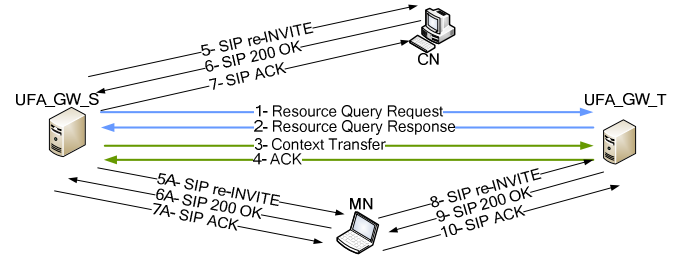


Figure 2. UFA handover procedure

Table 1. SIP header for application configuration (*UFA_Appli_Config*)

Add_IP_Addr	The new MN address.
Del_IP_Addr	The old MN address.

Table 2. SIP header for Layer 2/Layer 3 configuration (*UFA_Terminal_Config*)

UFA_GW_T_MAC_Addr	Used for Layer 2 HO
UFA_GW_T_ESSID	
UFA_GW_T_Channel	
UFA_GW_T_IP_Addr	Used for Layer 3 HO
UFA_GW_T_Netmask	
Add_IP_Addr	

UFA handover timing diagram is illustrated in Figure 3. When MN attaches to UFA_GW_T it sends a SIP Re-INVITE (8) message to UFA_GW_T. UFA_GW_T buffers data received from CN until reception of SIP Re-INVITE (8). We define D4 as the time necessary for SIP layer to detect IP address change and send SIP Re-INVITE (8). Compared to m-SCTP, UFA mobility mechanism enhances the network handover delay. Indeed the equivalent D2 delay does not exist and the equivalent D3 is very low as MN address determination and Duplicate Address Detection are performed proactively to

HO execution. In [14] (D1+D3+D4) was measured on a testbed to 150ms. This value is valid for SCTP as it is independent of the transport layer.

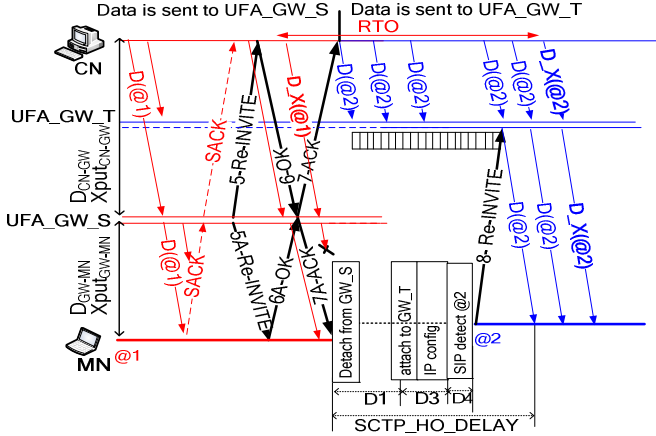


Figure 3. Handover delay components with UFA

B. Different UFA options driving SCTP configuration

We define three incremental UFA options depending on the configured SCTP parameters upon the reception UFA SIP messages for handover:

- **UFA**: is the minimal and basic SCTP configuration to support SCTP applications handover. It supposes the consideration of SIP headers presented in the previous section.
- **UFA+**: as handover delay in UFA is low, performance problems raised in section II.B appear. UFA+ resolves these problems by triggering on CN side after receiving message 7 immediate sending of lost packets before any new packet, preventing thus cwnd from remaining constant.
- **UFA++**: is based on UFA+ and solves in addition the link throughput under-utilization problem raised in section II.B. With UFA++, when BDP_{GW_T-MN} is received by CN within message 5, the SxS module calculates the SCTP congestion control parameter values related to Add_IP_Addr and informs the SCTP layer to immediately apply the calculated values. Cwnd is set to BDP_{GW_T-MN} which cancels the time necessary to attain this value, ssthresh is set to BDP_{GW_T-MN} and RTO is kept to 3s.

BDP_{GW_T-MN} is calculated by UFA_GW_T during handover preparation procedure using formulas (1) and (2) and considering the transmission delay of a 1500 bytes-length packet for the assessment of RTT_{GW_T-MN} . RTT_{CN-GW_T} can be determined based on measurements performed by UFA_GW_T as described in [16].

$$BDP_{GW_T-MN} = RTT_{CN-MN} * Xput_{GW_T-MN} \quad (1)$$

$$RTT_{CN-MN} = RTT_{CN-GW_T} + RTT_{GW_T-MN} \quad (2)$$

During throughput allocation, UFA_GW_T checks whether its free buffer size is compliant with the rule of thumb (buffer_size = BDP_{GW_T-MN}) [17] and allocates it accordingly.

C. Discussion

Giving the increasing interest for SIP [10] in many standardisation instances and research works, the choice of SIP as a dedicated signaling protocol to manage SCTP applications

seems particularly justified. In addition, SIP has been already proposed to interact with SCTP. In [11], MN uses the SIP registration messages necessary for reachability functions to also update in the CN the new address acquired by the MN. Therefore m-SCTP signaling is no more necessary and the overall signaling cost is reduced. In [12], SIP is used to manage simultaneous MN and CN movements while maintaining ongoing SCTP applications. However, these references differ from our work since they employ SIP to solve problems different from ours which consists in enhancing SCTP performances during hard handovers. Another important difference is that our work proposes the management of SCTP applications not only through the interaction with SIP but also with the support of a network-controlled architecture fulfilling all of the defined requirements.

Security issues raised in m-SCTP and treated through authenticated ASCONF messages and HEARTBEAT exchange for path verification [1] [2] are not encountered with UFA, since we assume an underlying trust relationship between the network and the endpoints.

VI. PERFORMANCE EVALUATION

In this section, UFA, m-SCTP as well as their enhanced configurations (UFA+, UFA++, m-SCTP+) are compared.

A. Simulation model and inputs

We construct a simulation model using Network Simulator 2.33 [18]. During a given simulation time (500 seconds), CN sends a file to MN. Hard handover of MN is simulated by periodically switching between two GWs. The switching periodicity determines the number of handovers (HO_nbr) occurring during data downloading. Links between CN and GW (respect. GW and MN) are characterized by a propagation delay D_{CN-GW} and a throughput $Xput_{CN-GW}$ (respect D_{GW-MN} , $Xput_{GW-MN}$). The receiver buffer size is 65536bytes. For m-SCTP and m-SCTP+, we consider 2s for the network handover delay ($D1+D2+D3$). Simulations are conducted using different network scenarios given in Table 1.

Table 1. Considered network scenarios (Sc) for simulation

		Sc1	Sc2
D_{CN-GW}	(ms)	10	100
$Xput_{CN-GW}$	(Mbps)	10	10
D_{GW-MN}	(ms)	2	2
$Xput_{GW-MN}$	(Mbps)	1	Variable (0.1 ... 3)
HO_nbr		Variable (1..13)	6

B. Performance results

To compare the different performances, we measure different pertinent key performance indicators such as the transport handover delay, the mean throughput and the size of downloaded data during the simulation time (500s). Due to limited space, we only show here results for the most global indicator, which is the size of downloaded data.

We first compare m-SCTP, m-SCTP+, UFA, UFA+ and UFA++ for the network scenario Sc1 (see Table 3) considering different number of handovers (HO_nbr). A HO_nbr value corresponds to a given MN velocity. For example, HO_nbr=6 corresponds to a pedestrian walking at 5 km/h in an area covered by 100m-diameter cells or a user in a car travelling at 51 km/h in 1km-diameter cells. **Figure 4** gives the additional

data volume m-SCTP+, UFA, UFA+, UFA++ enable to download compared to m-SCTP. For m-SCTP, the downloaded data volume is 62, 61, 60, 59, 57 Mbytes for HO_nbr equal to 1, 3, 6, 9, 13 respectively. We observe that all UFA options enable to download more data volume than m-SCTP. Moreover they are more efficient than m-SCTP+ considered in the state of the art as the best enhancement to m-SCTP performance with regards to the long transport handover delay: m-SCTP+ enables a gain ranging from 0.2% to 2% compared to m-SCTP; and UFA enables a gain ranging from 0.4% to 7.8% compared to m-SCTP. For this network scenario Sc1, UFA+ and UFA++ do not provide remarkable gains compared to UFA as both D_{CN-GW} and $BDP_{GW-T-MN}$ are low. These enable in UFA lost packets to be recovered rapidly (SACKs are returned rapidly) and cwnd to attain rapidly $BDP_{GW-T-MN}$.

We therefore compare the performances of UFA, UFA+ and UFA++ using the network scenario Sc2 (see Table 3) considering higher values for D_{CN-GW} and $BDP_{GW-T-MN}$ (higher value for $Xput_{GW-MN}$). We set the receiver buffer size to 200000bytes in order to take into account the high throughputs ($Xput_{GW-MN}$ is 2Mbps or 3Mbps). Figure 5 shows the additional data volume UFA+ and UFA++ enable to download compared to UFA. For UFA the downloaded data volume is 6, 28, 57, 116, 168 Mbytes for $Xput_{GW-MN}$ equal to 0.1, 0.5, 1, 2, 3 Mbps respectively. We observe that compared to UFA:

- UFA+ enables a gain varying from 2% to 7% for an $Xput_{GW-MN}$ varying from 1 Mbps to 3 Mbps, and
- UFA++ enables a gain varying from 4% to 9% for an $Xput_{GW-MN}$ varying from 1 Mbps to 3 Mbps.

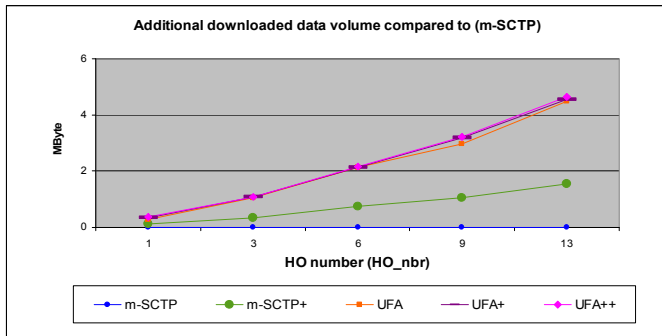


Figure 4. Performance comparison for network scenario Sc1

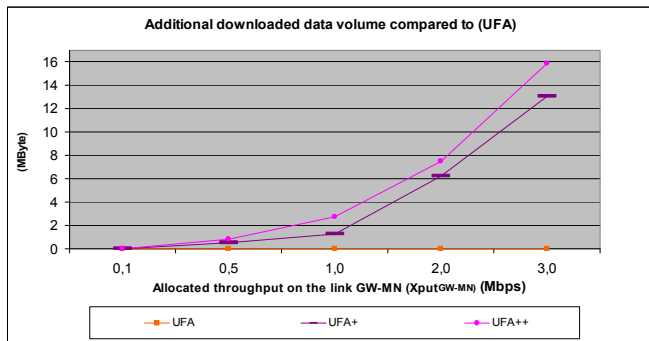


Figure 5. Performance comparison for network scenario Sc2

We conclude that UFA+ and UFA++ provide better performances than UFA. In general, although the additional downloaded data volume may appear relatively low, this one

shall not be neglected as it has been calculated during a short time period (simulation time=500s) and for a single terminal. The gain for an operator is important as it is proportional to the number of terminals and the duration of data downloading (higher than 500s).

CONCLUSION

In this paper we underline the weaknesses of m-SCTP and terminal-based approaches and propose an overall network-controlled architecture (UFA) for SCTP handover management. Besides the network-controlled and cross-layer mechanisms, UFA relies on SIP to drive optimally SCTP configuration on both endpoints. We conceived three incremental UFA options depending on the configured SCTP parameters after handover. The first option, shortly named UFA, is the basic one. It configures SCTP layer with the new MN addresses and reduces the handover delay. In UFA+ based on UFA, SCTP is enhanced by immediately sending first lost packets through the new link. UFA++ based on UFA+ additionally updates SCTP congestion control parameters according to the new link bandwidth. Performance results are promising:

- 1) UFA provides better performances than m-SCTP.
- 2) UFA is even better than enhanced m-SCTP solutions.
- 3) Both UFA+ and UFA++ improve UFA performance.

The gain of UFA and its options is important for an operator; it is proportional to the number of terminals and the duration of data downloading. Further comparative simulations are planned to provide the most appropriate UFA option.

REFERENCES

- [1] IETF RFC 4960; "Stream Control Transmission Protocol".
- [2] IETF RFC 5061; "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration".
- [3] IETF RFC 3775; "IP Mobility Support in IPv6".
- [4] M. Chang, M. Lee, H. Lee, "An enhancement of transport layer approach to mobility support", LNCS, volume 3391, 2005.
- [5] M. Honda et al, "SmSCTP: A Fast Transport Layer Handover Method Using Single Wireless Interface", ISCC 2007.
- [6] M. Afif et al, "Radio aware SCTP extension for Handover in EGPRS", PIMRC 2006.
- [7] Y.Qiao et al, "SCTP Performance Issue on Path Delay Differential", LNCS, 2007, Volume 4517/2007
- [8] T.V. Lakshman, U.Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss", IEEE/ACM transactions on networking, VOL. 5, NO.3, June 1997.
- [9] X.Wu, et al, "TCP Handoff: A practical enhancement for heterogeneous mobile environments", in Proc. of ICC 2007.
- [10] IETF RFC 3261, "SIP: Session Initiation Protocol".
- [11] Y.Chen, K.Chui, R.Hwang, "SmSCTP: SIP based MSCTP scheme for session mobility over WLAN/3G heterogeneous networks"
- [12] G. De Marco, S. Loreto, L.Barolli "Performance Analysis of IP Micro-mobility Protocols in Single and Simultaneous Movements", EUC Workshops 2005, LNCS 3823, pp. 443, 2005.
- [13] K. Daoud, P.Herbelin, N. Crespi, "UFA: An Ultra Flat Architecture for High Bitrate Services in Mobile Networks", PIMRC 2008.
- [14] K. Daoud et al, "Performance and implementation of UFA: A SIP-based Ultra Flat Architecture Mobile Network Architecture", PIMRC 2009.
- [15] IETF RFC 3550, "A Transport Protocol for Real-Time Applications".
- [16] R. Fracchia et al, "A wise extension of SCTP for wireless networks", ICC 2005.
- [17] G. Appenzeller et al, "Sizing router buffers", SIGCOMM 2004.
- [18] "Network Simulator NS2", <http://www.isi.edu/nsnam/ns/>.