De-Triangulation Optimal solutions for mobility scenarios with asymmetric links

Pedro Vale Estrela[1,2], Teresa Maria Vazão[1,2], Mário Serafim Nunes[1,2]

[1]Instituto Superior Técnico-TagusPark / Universidade Técnica de Lisboa, Av. Prof Cavaco Silva, Oeiras, Portugal
[2]INESC-ID, Rua Alves Redol, 9, 1000-029, Lisboa, Portugal

{pedro.estrela, teresa.vazao, mario.nunes}@tagus.ist.utl.pt

Abstract. In the field of mobility support, several mobility protocols resort to the use of triangulation mechanisms as a means of supporting fast handovers or basic connectivity. In order to reduce the maximum end-to-end delay of the packets, such triangulations can be later removed to enable direct routing of the data packets. However, using a simple update of this routing entry can cause the reception of out-of-order packets at the mobile node receiver, as the direct packets can arrive earlier than the triangulated packets.
This paper proposes a generic optimal de-triangulation mechanism that neither causes out-of-order packets nor increases the packet delay, for any combination of asymmetric links delays. After an analytic framework analysis contribution, the efficiency of the proposed algorithm is evaluated using simulation studies, in which the packet losses, packet delay, handover latency and control/data load metrics are measured.

1 Introduction

In the field of mobility support, several mobility protocols resort to the use of triangulation mechanisms as a means of supporting fast handovers or basic connectivity. By using these mechanisms, the routing of data packets destinated to mobile nodes is aided by a certain number of intermediate nodes, leading to longer paths and higher end-to-end delays.

Several mobility protocols with micro-mobility capabilities, which *Fast Handovers for Mobile IP* (FMIP) [1], *BRAIN Candidate for Mobility Management Protocol* (BCMP) [2] or *Micro-mobility support with Efficient Handoff and Route Optimization Mechanism* (MEHROOM) [3] are examples of, can provide a faster handover operation by forwarding the in-flight packets received at the previous Mobile Node (MN) location to the new location, using a simple triangulation scheme. In the same vein, the MIP *Route Optimization* (RO) option can be used to forward data between the involved Foreign Agents in an handover [4]. Also, macro-mobility protocols like MIPv4 [5] can permanently use the triangulation via the Home Agent (HA) as a means of achieving transparency to fixed Correspondent Nodes (CN), or of performing temporary data forwarding before the route optimization operation in MIPv6 [6]

In all cases, in order to reduce the maximum end-to-end delay, such triangulations can be later removed by a ***de-triangulation operation***, in which the sender node is updated with the most current MN location, enabling it to send the packets directly to the MN. In all the previously mentioned protocols, the triangulation is simply removed by updating the sender node with the newest MN location, using a simple update message. However, this simple update can cause the reception of out-of-order packets at the MN. As long as the direct path will typically impose lower latency than the triangulated path, the subsequent directly sent data packets can actually be received *earlier* than the last in-flight packets sent via the triangulated path.

Depending on the type of receiver, these out-of-order packets due to such de-triangulation operations can have a major impact on it, as additional actions may be needed to recover from such phenomena [7]. These out-of-order packets can either be considered as lost packets by UDP receivers which expect ordered arrival [9], or can trigger the retransmit mechanism and needlessly reduce the contention window of the majority of the TCP senders, as no packets have actually been dropped [10].

This problem was addressed by previous research work. The *Celular IP semi-soft handover* [11] solves it by delaying the packets sent through the direct path at a specific node, known as crossover node, for a constant amount of time, in order to allow an earlier reception of the triangulated packets. In spite of avoiding out-of-order packets, this mechanisms forces a constant delay, which is independent of actual de-triangulation phenomena, and results in excessive majorated delays to ensure an ordered reception. Alternatively, the *seamless MIP* [12] marks the packets and reorders them at the destination. This is an inefficient and of limited applicability process, as it requires packet marking in the IP header of all data packets.

This paper proposes a generic optimal de-triangulation mechanism that neither causes out-of-order packets *nor* increases the packet delay. The proposed mechanism achieves its goals by delaying the packets, which will be sent via the direct path, for the exact minimum period of time that causes them to arrive at the destination immediately after the last in-flight packets sent via the triangulated path. Simpler versions of the mechanism that either remain optimal, but assume the presence of symmetric links, or that are non-optimal but continue to guarantee packet ordering, are also presented.

To support this contribution, the paper presents an illustrative example followed by an analytical framework in section 2, which is used to study the problem in section 3. The proposed solutions are described and analyzed in section 4 using the analytical framework, and are additionally validated by NS2 simulation studies in section 5. The paper ends, in section 6, with some conclusions and future work.

2 Analytical framework and problem definition

2.1 Problem Statement

To illustrate the problem, let us consider the triangulation situation depicted in **Fig. 1**a. The figure comprises the crossover node, Node A, which is the node where the triangulated and the direct paths diverge; the old destination node, Node B, which is the one that was previously used to transfer information with the MN and will be used as a triangulated node; and the new destination node, Node C, which is the one that should be used to access the MN after it roams from Node B to Node C. For the sake of simplicity, the MN is omitted in the figure.

At first, when Node A receives data to the MN, it uses Node B the reach the MN, although it is now accessible through Node C; then, Node C will send an update message to Node A, so that data may be routed directly to it, removing the triangulation effect. Even though such operations do not typically incur in the drop of in-transit data packets, it can result in their reordering, as the direct path will typically have lower latency than the triangulated one (otherwise, the routing would benefit from being triangulated in the first place, and no de-triangulation mechanism should be performed). The exact latency difference will be the result of the actual link delays and the amount of data packets queued in the routers, among other factors.

This problem is illustrated in **Fig. 1**b, where packets sent through the direct path (Data packet 2, A→C) can be received earlier than packets sent through the triangulated path (Data packet 1, A→B→C).

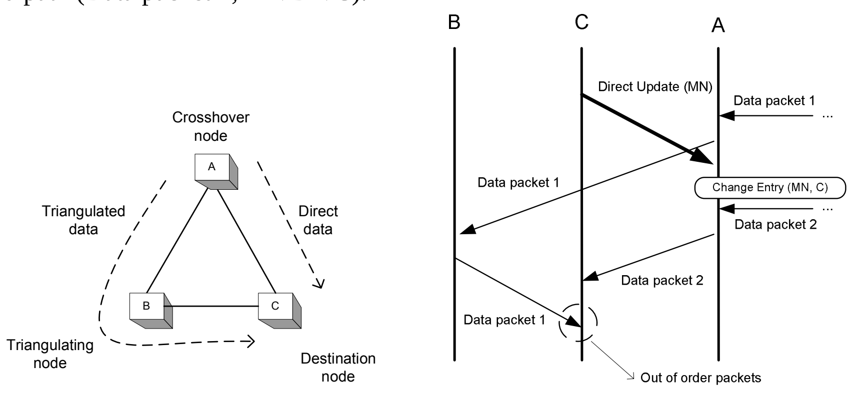

Fig. 1. de-triangulation problem illustration. a) overall problem b) messages exchanged

2.1 De-triangulation mechanism description

In this section, we introduce a new framework for analysing the de-triangulation problem.

Let us considered the situation represented in **Fig. 2**a where the data packet transmission and reception instants are depicted and the time instants represent:

- $t_{0:}$ the time when the de-triangulation process will start at the destination node (Node C);
- $t_{x} > t_{0:}$ the time when the ***last*** data packet is sent by the crossover node (Node A), through the ***triangulated*** path;
- $t_{x'} > t_x$ the time when this packet is received by the destination node;
- $t_y > t_{x:}$ the time when the ***first*** data packet is sent by the crossover node (Node A) through the ***direct*** path;
- $t_{y'} > t_y$ the time when this packet is received by the destination node.

Considering also the mean delays that occur between the involved nodes for the transmission of packets, depicted in **Fig. 2**b, which represent, separately, the downstream ($d_{(A,B)}$, $d_{(B,C)}$ and $d_{(A,C)}$) and the upstream ($d_{(B,A)}$, $d_{(C,B)}$ and $d_{(C,A)}$) paths. Each mentioned delay will contain the time needed to support all the operations involved to transmit a packet and to receive it, of which the propagation and queuing delays will be the main drivers.

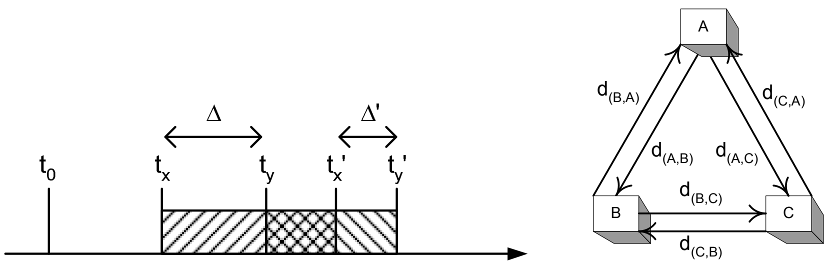

Fig. 2. a) Time instants of data packet transmission and reception b) asymmetric link delays

Using this notation, the following relations occur:

$$t_{x'} = t_x + d_{(A,B)} + d_{(B,C)} \quad \text{(Eq. 1)}$$

$$t_{y'} = t_y + d_{(A,C)} \quad \text{(Eq. 2)}$$

Considering now, the time Δ since the transmission of the last triangulated packet until the transmission of the first direct packet at the crossover node; and the time Δ' since the arrival of the first direct packet until the arrival of the last triangulated packet a the destination node, which are given by:

$$\Delta = t_y - t_x \quad \text{(Eq. 3)}$$

$$\Delta' = t_{y'} - t_{x'} \quad \text{(Eq. 4)}$$

A positive value of Δ results in buffering being applied to the packets at the crossover node, as incoming packets need to wait before they are forwarded. The value is directly proportional to the maximum buffering capacity required. A value of zero results in no buffering at all. Finally, a negative value of Δ is not used in the context of de-triangulation.

A negative value of Δ' results in the occurrence of out-of-order packets, but no additional delay, as triangulated packets are received later than the direct ones. On other hand, a positive value results in ordered delivery, but with an additional delay being experienced by the involved packets which is equal to Δ'. Finally, a value of zero will result in an optimal de-triangulation mechanism – no out-of-order packets and no delay increase.

Thus, the objective of an ordered de-triangulation mechanism is defined as the operation where the packets will be delayed at the crossover node for a certain period of time Δ, so that Δ' will be positive; additionally, the objective of an optimal ordered de-triangulation mechanism is to ensure that Δ' will be zero.

3 Analysing the direct de-triangulation problem

To illustrate the generic solution, let us consider the regular de-triangulation situation depicted in Fig.3.

Firstly, the destination Node C sends an Update Message directly to the crossover node (step 1). Then, the crossover node will update its own Routing Table, being able to start using the direct path (step 2). From now on, incoming packets will be sent directly to Node C, while already in-transit packets continue to use Node B (step 3). Every in-flight packet received by Node B is sent to Node C (triangulated path) (step 4), being received later than the direct paths, unless an ordered de-triangulation mechanism is used.

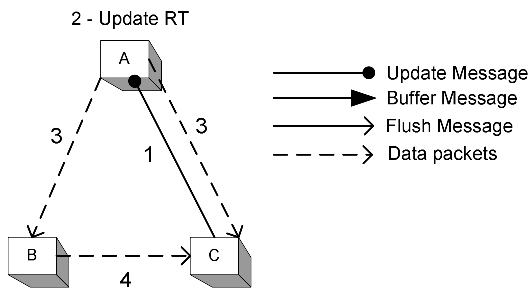

Fig. 3. Direct de-triangulation algorithm

Using the same notation as before:

$$t_x = t_0 + d_{(C,A)} \quad \text{(Eq. 5)}$$

$$t_y = t_x \quad \text{(Eq. 6)}$$

$$\Delta' = (t_y + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 7)}$$

$$\Delta' = d_{(A,C)} - d_{(A,B)} - d_{(B,C)} \quad \text{(Eq. 8)}$$

In order to avoid out-of-order packets, Δ' must be positive or equal to zero:

$\Delta' >= 0$ (Eq. 9)

$d_{(A,C)} >= d_{(A,B)}+d_{(B,C)}$ (Eq. 10)

As this last relation is not verified in the typical situations, as it would negate the major benefit of removing the de-triangulation, this results in out-of-order packets phenomena illustrated in section 2.

4 Proposed solutions

4.1 Conservative Algorithm

The first presented algorithm, named **conservative algorithm** (Fig. 4a), will ensure that no out-of-order packets will occur, by buffering the received packets at the crossover node, while waiting for the reception of the last packet via the triangulated path. Only when this happens, the buffered packets will be released by the use of a new message called ***Flush***.

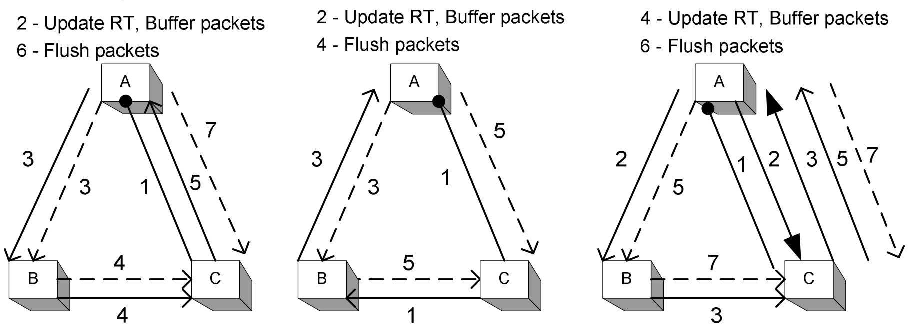

Fig. 4. Proposed solutions: a) conservative b) symmetric c) asymmetric

Firstly, the destination Node C sends an Update Message directly to the crossover node, as in the previous case (step 1), but now it starts buffering all incoming packets to a certain MN (step 2). Then, node A will send a Flush message that will pass through the triangulated and the destination node before returning to itself (step 3, 4 and 5). This forces the control packet to pass through the same paths as the in-transit data packets in the triangulated path (also steps 3 and 4, dashed lines). When node A receives the Flush Message, it transmits the buffered packets via the direct path, and stops the buffering of additional packets (steps 6 and 7).

Using the same notation as above, the last packet forwarded using the triangulated path will guarantedly be received at node C *earlier* than the first packet received via the direct path. However, this operation increases the previous handover latency for an additional amount of time, as shown by the following analysis:

$$t_x = t_0 + d_{(C,A)} \quad \text{(Eq. 11)}$$

$$t_y = t_x + d_{(A,B)} + d_{(B,C)} + d_{(C,A)} \quad \text{(Eq. 12)}$$

$$\Delta' = (t_y + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 13)}$$

$$\Delta' = (t_x + d_{(A,B)} + d_{(B,C)} + d_{(C,A)} + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 14)}$$

$$\Delta' = d_{(C,A)} + d_{(A,C)} \quad \text{(Eq. 15)}$$

Thus, out-of-order packets are avoided ($\Delta' > 0$), but the data packets are always unnecessarily delayed for ($d_{(C,A)} + d_{(A,C)}$) time units.

4.2 Optimal algorithm for Symmetric links

If the link delays can be assumed to be symmetric, then the previous algorithm can be refined to not incur in any extra delay, besides the imposed by triangulation itself. This has the advantages of solving the latency increase of the previous mechanism, and of reducing the buffering requirements of the **symmetric algorithm** (Fig. 4b).

Again, the destination node sends an Update Message directly to the crossover node (step 1) to start the buffering of all data packets (step 2). However, at the same the, the destination node also sends the Flush Message in parallel to the crossover node, which now passes through the triangulated node in the opposite direction of the data packets (step 1 and 3). As before, when the crossover node receives this message, it stops buffering additional packets (step 4) and transmits the buffered packets via the direct path (step 5).

Using the same notation as before:

$$t_x = t_0 + d_{(C,A)} \quad \text{(Eq. 16)}$$

$$t_y = t_0 + d_{(C,B)} + d_{(B,A)} \quad \text{(Eq. 17)}$$

$$\Delta' = (t_y + d_{(A,C)}) - (t_x + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 18)}$$

$$\Delta' = (t_0 + d_{(C,B)} + d_{(B,A)} + d_{(A,C)}) - (t_0 + d_{(C,A)} + d_{(A,B)} + d_{(B,C)}) \quad \text{(Eq. 19)}$$

$$\Delta' = (d_{(C,B)} + d_{(B,A)} + d_{(A,C)}) - (d_{(B,C)} + d_{(A,B)} + d_{(C,A)}) \quad \text{(Eq. 20)}$$

As symmetric links are assumed, $d_{(C,B)}=d_{(B,C)}$; $d_{(B,A)}=d_{(A,B)}$; and $d_{(A,C)}=d_{(C,A)}$; thus:

$$\Delta' = 0 \quad \text{(Eq. 21)}$$

Thus, the algorithm is optimal: out-of-order packets are avoided, and no extra delay is incurred in the data packets.

4.3 Optimal algorithm for Asymmetric links

If the link delays are asymmetric, e.g. due to different propagation delays or queuing, then the conservative algorithm can again result in out-of-order packets, in the situations where Δ' is negative. Such situation might happen when the control messages (Update and Flush messages) travel faster than the data packets and the delay imposed to data packets at the crossover node is smaller than needed. Although this could be solved by the addition of an extra small delay after the reception of the Flush Message, this would not result in an optimal solution. In contrast, the final **asymmetric algorithm** presented in this section is able to maintain optimality through slightly higher control and data loads (Fig. 4c).

Firstly, the destination node sends an Update Message directly to the crossover node (step 1). When it receives the message, it sends a Flush Message as in the conservative algorithm via nodes B (step 2) and C (step 3 and 5), but also a new message to itself via node C (steps 2 and 3), called ***Buffer Message***. These pair of messages are used to measure the actual delays experienced by data packets, taking account the asymmetric nature of the links. When the crossover node receives the Buffer Message (step 3), it updates the Routing Table and starts buffering data packets (step 4). The reception of the Flush Message is dealt with as before (step 6 and 7).

Using the same notation as above:

$$t_x=t_0+d_{(C,A)}+d_{(A,C)}+d_{(C,A)} \quad \text{(Eq. 22)}$$

$$t_y=t_0+d_{(C,A)}+d_{(A,B)}+d_{(B,C)}+d_{(C,A)} \quad \text{(Eq. 23)}$$

$$\Delta'=(t_y+d_{(A,C)})-(t_x+d_{(A,B)}+d_{(B,C)}) \quad \text{(Eq. 24)}$$

$$\Delta'=(t_0+d_{(C,A)}+d_{(A,B)}+d_{(B,C)}+d_{(C,A)}+d_{(A,C)})-(t_0+d_{(C,A)}+d_{(A,C)}+d_{(C,A)}+d_{(A,B)}+d_{(B,C)}) \quad \text{(Eq. 25)}$$

$$\Delta'=(d_{(A,B)}+d_{(B,C)}+d_{(C,A)}+d_{(A,C)})-(d_{(A,B)}+d_{(B,C)}+d_{(C,A)}+d_{(A,C)}) \quad \text{(Eq. 26)}$$

Without the need to assume symmetric links, equation 26 simplifies to:

$$\Delta'=0 \quad \text{(Eq. 27)}$$

Thus, the algorithm is always optimal regardless of the combination of asymmetric links: out-of-order packets are avoided, and no extra delay is incurred.

5 Simulation studies

5.1 Simulation scenario

To evaluate the proposed smooth de-triangulation algorithms, their behaviour was compared with the standard direct de-triangulation using simulation studies. The simulations were carried out using Network Simulator (NS) v2.26, where a simple mobility protocol which uses local temporary triangulations at each handover was modelled. A key addition to the simulator was the modification of the existing UDP LossMonitor object to evaluate and count the received out-of-order packets, besides the dropped packets that are already considered by the base object.

The simulation scenario is illustrated in **Fig. 5**. Here, a MN will be the receiver of a continuous stream of Constant Bit Rate (CBR) UDP probe packets sent by a CN located outside the domain, that generates 200 packets per second of 100 bytes each.

The MN will then make a series of roaming operations between the two existing Access Points, AP1 and AP2, being connected to the network by each one sequentially at a speed of 30 hand/min. The domain has a mobility anchor point (MAP) which receives the packets from the CN, and redirects them to the MN's current AP. Inside the domain the nodes are connected by wired links that feature sufficient bandwidth for the test probes.

Two separate scenarios are considered; in the former, all links, with the purposely different delays values as depicted in **Fig. 5**a, feature symmetric upstream and downstream propagation delays; an asymmetric scenario was also investigated, illustrated in **Fig. 5**b, where the crossover→AP1 downstream link delay is asymmetric from the upstream delay, in order to simulate local congestion.

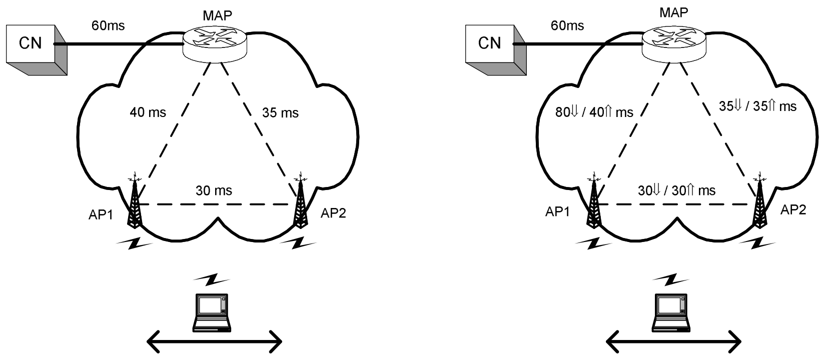

Fig. 5. Simulation Scenario: a) Symmetric case b) Asymmetric case

At each handover operation, the MN first creates a local tunnel between the involved APs, by sending an Update Message through the new AP destinated to the old AP. To avoid the existence of dropped packets at the old AP, the APs have sufficient transmission power to ensure a continuous coverage to the MN. Then, the handover operation is finalized by updating the MN's entry at the crossover node; in the *direct* model, this is achieved by sending an Update Message directly to the crossover node via the new AP; the other studied models (*smooth_conservative*, *smooth_symmetric* and *smooth_asymmetric*) additionally use the sequence of messages and buffering operations previously described.

In this scenario, each model was tested for a series of 100 handover operations, where the experienced packet losses, delay and buffer usage for the data and control packets were evaluated.

5.2 Loss and Delay Results

Fig. 6 and **Fig. 7** show the loss and delay results for all studied alternatives for both symmetric and asymmetric links. The loss graph divides the experienced losses into drops, late packets and out-of-order packets. The delay graph divides the experienced one-way delays into minimum, average and maximum values. As expected from the simulation scenario definitions, no packets are dropped in all cases.

Regarding the regular **direct mechanism**, a large number of packets are received out-of-order, confirming the de-triangulation problem previously defined. On the other hand, the data packets experience different minimum and maximum delays, depending on the stationary and handover situations. The minimum delay (~95 ms) is experienced when the MN is stationary at AP2, as the packets are routed directly from the crossover node to the AP2; the maximum delay is experienced for the triangulated packets, which are sent to AP2 triangulated via AP1 (~130 ms for symmetric test / ~170 ms for asymmetric test).

The **conservative mechanism** is able to solve the out-of-order packets phenomena, for both symmetric and asymmetric links. However, the mechanism is not optimal, as it greatly increases the maximum delay of the involved packets (~200 ms for symmetric test / ~240 ms for the asymmetric case).

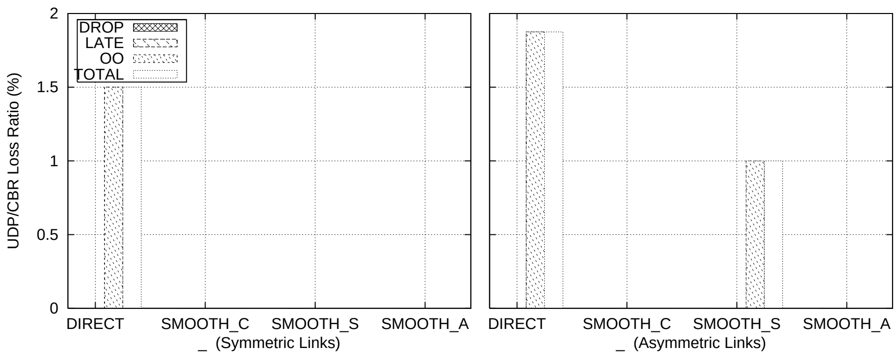

Fig. 6. Loss ratio as a result of Drops, Late and Out-of-Order packets

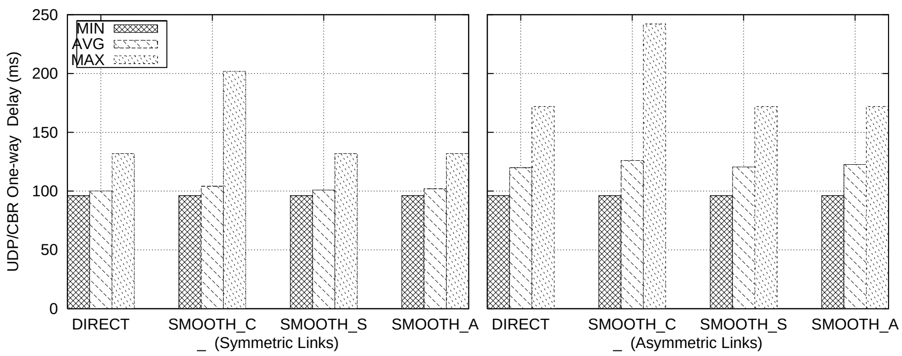

Fig. 7. One-way packet delay (minimum, average and maximum values)

The **symmetric mechanism** solves this problem, but only if the links are symmetric. In these conditions, no out-of-order packets occur, and the maximum delay is similar to the one experienced in the base direct mechanism (~130 ms). With asymmetric links, even though the maximum delay is not increased (~170 ms), a small number of out-of-order packets is experienced, being this value related to the sum of the delay asymmetries of the involved links, as defined by eq.20.

Finally, the **asymmetric mechanism** solves this last problem, for any combination of asymmetric links. In these conditions, no out-of-order packets occur, and the maximum delay is similar to the one experienced in the base direct mechanism (~130 ms for symmetric test / ~170 ms for asymmetric test).

5.3 Buffer utilization, Control load and Data load Results

Fig. 8, **Fig. 9** and **Fig. 10** show the buffer utilization, the signalling load and the data load for all studied alternatives, for both symmetric and asymmetric links. The buffer graph shows the average number of queued packets at the crossover node per handover; the control load shows the average number of forwarded control packets per handover, for all the used control packets (Update, Buffer and Flush messages); lastly, data load shows the ratio of data packets that are locally triangulated between the two APs.

Regarding the regular **direct mechanism**, the buffer queues are not used, as no packets are delayed at the crossover node. On the other hand, by only using a single Update message, it has the lowest control load requirements from all the alternatives. Finally, the data packets subject to triangulation for a certain time period, ending when the Update message reaches the crossover node.

The **conservative mechanism** requires the largest buffers at the crossover, which fully corresponds to the maximum delay studied in **Fig. 7**, needed to hold all the packets during the de-triangulation procedure. Additionally, by propagating its Flush Message through the whole triangle, this scheme has higher control load requirements than the previous case. However, it stops the triangulation effect at the same time as the direct case, by buffering the data packets at the crossover node; thus it has an equal data load as the previous case.

By imposing a much lower maximum delay, the **symmetric mechanism** requires a corresponding lower amount of buffer space at the crossover node. Additionally, the Flush message is simplified, which results in lower control load requirements. Again, the symmetric mechanism has a similar data load requirements than the base case, for the same reasons explained previously.

Finally, the **asymmetric mechanism** has the same low buffer usage for symmetric cases as the previous mechanism; however, it now correctly handles the asymmetric cases as well, by buffering the data packets for the necessary time. The downside of these capabilities is an increased control and data load of all the studied mechanisms. The former is related to the usage of additional messages to probe the delay differences; the latter is related to the maintenance of the triangulation effect for an additional small time period, in order to wait for the delay differences's measurements. However, it should be noted that all these control and data messages are always limited to the wired part of the network; thus, these mechanisms have the same efficiency as the base case regarding the usage of the limited wireless resources.

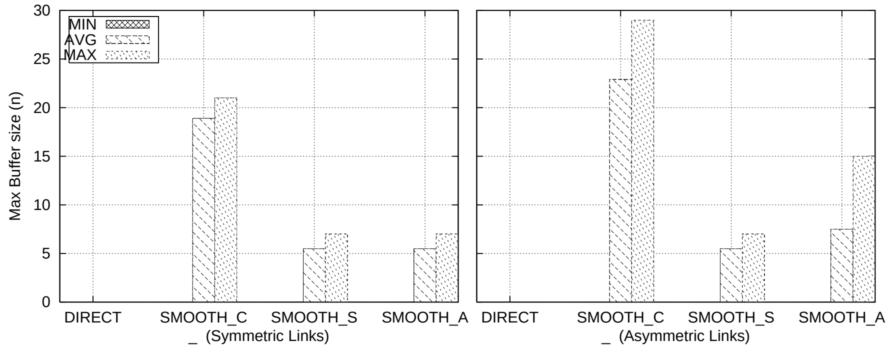

Fig. 8. Average Buffer size per Handover

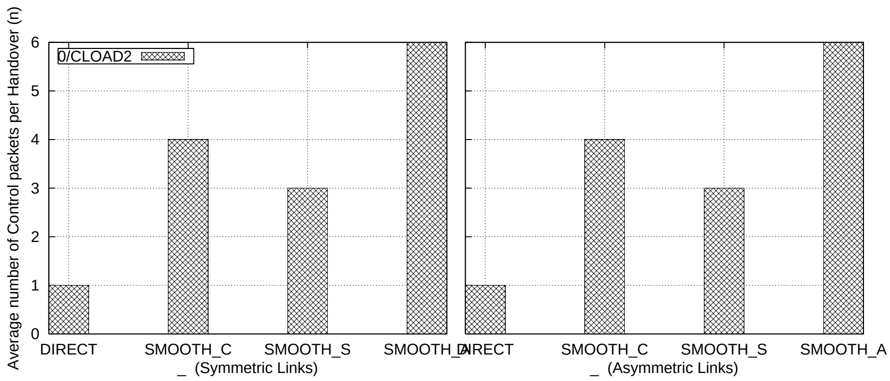

Fig. 9. Average Control load per handover

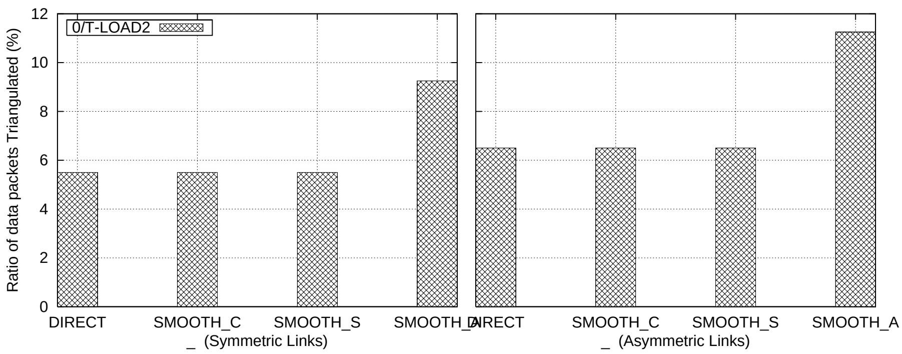

Fig. 10. Ratio of triangulated data packets

6 Conclusions

While the original MIP protocol keeps the proposed triangulation effect at all times, newer macro and micro mobility protocols solve the problem of average long delays, but suffer from a de-triangulation problem, where out-of-order packets are experienced at each de-triangulation event.

In this paper, we have identified the problem of de-triangulation, introduced a new analytical framework, and then used the framework to study the problem and to propose three increasingly complex de-triangulation schemes that feature different assumptions, results and signalling requirements.

The conservative approach guarantees packet ordering, but increases the maximum delay of the involved de-triangulation data packets; the symmetric approach maintains packet ordering without increasing the maximum delay, for any combination of link delays, but only if these are symmetric; the asymmetric approach guarantees packet ordering without maximum delay increases, for any combination of asymmetric link delays.

The proposed algorithm’s efficiency is evaluated using both analytic and NS2 simulation studies which measured the packet losses, packet delay, handover latency and control load; besides the ordering and delay results already mentioned, it was found that the buffer requirements are directly co-related to the maximum delay increase, and that the most complex mechanisms have increasing control load requirements, which are nonetheless limited to the wired part of the network.

All these generic algorithms can be easily applied to all mobility protocols that feature a de-triangulation component. An initial state-of-the-art review of this area showed that this can include, at least, the fMIP [1], MIPv4-RO [4], MIPv6-RO [6], BCMP [2], MEHROM [3] and MIP-SB [13] protocols.

References

[1] R. Koodli, Ed., “Fast Handovers for Mobile IPv6”, RFC 4068, July 2005

[2] C. Keszei, N. Georganopoulos, Z. Tyranyi, A., Valko, "Evaluation of the BRAIN candidate mobility management protocol. In Proceedings of the IST Mobile Summit (September 2001).

[3] L. Peters, I. Moerman, B. Dhoedt, P. Demeester, "MEHROM: Micromobility support with Efficient Handoff and Route Optimization Mechanism", published in 16th ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems, Antwerp, Belgium, August 31 - September 2, 2004, pp. 269-278

[4] C. Perkins and D. Johnson, “Route Optimization in Mobile IP”, draft-ietf-mobileip-optim-11.txt, IETF, September 2001.

[5] C. Perkins, Ed., "IP Mobility Support for IPv4", RFC-3320, IETF, January 2002.

[6] D. Johnson, C. Perkins and J. Arkko, “Mobility Support in IPv6”, RFC-3775, June 2004.

[7] A.Morton, L.Ciavattone, G.Ramachandran, S.Shalunov, J.Perser, “Packet Reordering Metric for IPPM”, draft-ietf-ippm-reordering-13.txt, May 2006

[8] J. Bennet, C. Partridge, N. Shectman, ”Packet reordering is not pathological behaviour”, IEEE/ACM Transactions on Networking, vol 7, issue 6, December 1999

[9] M. Laor and L. Gendel. “The effect of packet reordering in a backbone link on application throughput”, IEEE Network, 7(6), September 2002.

[10] X. Zhou and P. Mieghem. “Reordering of IP packets in Internet.” In Proc. Passive and Active Measurement, April 2004.

[11] A. Campbell, et al, “Design, Implementation and Evaluation of Cellular IP”, IEEE Personal Communications, Vol. 7 Nº4, August 2000.

[12] R. Hsieh, A. Seneviratne, “A comparison of mechanisms for improving mobile IP handoff latency for end-to-end TCP”, MOBICOM 2003, pages 29-41, September 2003

[13] K. El Malki, H. Soliman, “Simultaneous Bindings for Mobile IPv6 Fast Handovers”, draft-elmalki-mobileip-bicasting-v6-06 (work in progress), July 2005