

# A Testbed for Developing, Simulating and Experimenting Multipath Aggregation Algorithms

Fábio Rico  
Rederia Innovation SA  
3840-385 Vagos, Portugal  
Instituto de Telecomunicações  
Universidade de Aveiro  
3810-193 Aveiro, Portugal  
Email: fabiorico@ua.pt

Pedro Fonseca  
Instituto de Telecomunicações  
Universidade de Aveiro  
3810-193 Aveiro, Portugal  
Email: pf@ua.pt

Amaro Sousa  
Instituto de Telecomunicações  
Universidade de Aveiro  
3810-193 Aveiro, Portugal  
Email: asou@ua.pt

**Abstract**—Today, electronic devices may have multiple possibilities to communicate, either through wired or wireless interfaces. Despite this diversity, devices still fail to fully use the available resources by not simultaneously using multiple channels to their full extent. This is especially true in wireless channels where the efficient aggregation of multiple channels has proved to be a difficult task, as shown in recent simulation based works. In this Work In Progress paper, we present a testbed suitable to the evaluation of aggregation algorithms under real network environments. The proposed testbed aims to simulate and experiment both existing and new aggregation algorithms, optimized for wireless heterogeneous communication channels that can be deployed in industrial environments. In order to illustrate the merits of the proposed testbed, we also describe its use in the performance assessment of two aggregation algorithms: *Linux Bonding Driver* and Multipath TCP.

## I. INTRODUCTION

As the diversity of electronic devices and communication mechanisms grows, the popularity of mobile devices, such as laptops, smartphones and Internet of Things (IoT) devices, is also growing. Today, these devices can easily integrate multiple wireless communication interfaces, including IEEE 802.11 (WiFi), 3G/4G, Bluetooth or other WAN/PAN technologies. Despite the existence of multiple communication channels, most of these devices only use one channel, while the remaining channels are reserved for failover or not used at all. The simultaneous data transmission over multiple channels (or paths) brings enormous benefits in terms of throughput and robustness. Regarding throughput, the capacity of the multiple channels can be aggregated into a virtual higher capacity channel that can serve bandwidth intensive connections and/or multiple low bandwidth connections. Considering robustness, by having a connection established over multiple paths, the perception of a link failure or wireless handover can be mitigated and made imperceptible to the connection. Also, it is possible to redundantly transmit the information over multiple paths, increasing the probability of the information reaching their destination when using unreliable paths.

Still, the development of an efficient multipath aggregation algorithm is not an easy task due to the high level of packet reordering. Some algorithms, directed at different layers of the TCP/IP protocol stack, have been proposed, but they are somehow conditioned by assumptions that do not correspond

to real network environments, as we will see in the Related Work section. Our goal is to evaluate several algorithms under real network environments and propose improvements. We also aim to develop, simulate and experiment new aggregation algorithms capable of enhancing the transmission efficiency when using wireless heterogeneous communication channels.

In the following sections, the main problem of multipath aggregation will be presented as well as some related works that address the problem in different layers of TCP/IP protocol stack. In the sequence, we will present a testbed for evaluating multipath aggregation algorithms, where we evaluate the *Linux Bonding Driver* and MPTCP.

## II. PROBLEM

Data transmission over multiple paths has a number of benefits, although the heterogeneity of path communication technologies and the inherent network variability makes difficult the process of efficiently transmitting data. A common approach to increase the throughput of a system consists in distributing data packets over multiple paths in proportion to their capacity. In an ideal scenario, the total throughput would be the sum of all individual paths throughputs. However, in a real scenario, the latency and capacity variability induces packet reordering and loss, which lowers the performance of reliable data transport protocols.

For example, in TCP Congestion Control algorithm [1], if a packet arrives more than three positions out of order, the sender assumes that the packet is lost and retransmits it (this is the Fast Retransmission mechanism). However, the packet could have just been delayed due to some unexpected network event and, therefore, the unnecessary transmissions will lower the transmission efficiency. Note that tweaking data transport protocols to allow a larger out of order reception of packets, without unnecessary retransmission, is a compromise that must be investigated between transmission efficiency and required buffering resources, which may be limited in some devices. A detailed analysis of packet reordering and its consequences in wireless networks can be seen in Kaspar et al. [2] and Chan et al. [3].

As a result, an efficient method for transmitting data over multiple paths must take into account the path heterogeneity

and variability, in order to efficiently distribute data in proportion to each path available bandwidth capacity and, at the same time, to mitigate reordering effects at the receiver devices.

### III. RELATED WORK

The main approach to simultaneously use multiple paths consists in distributing, under a certain criteria, the outgoing data flows by the available paths. For example, if the capacity of a certain path is saturated, a new outgoing data flow is assigned to a less saturated path. By choosing adequate criteria, the overall throughput can be increased. However, since each outgoing data flow is confined to a single path, the flow cannot take advantage of the other paths resources for throughput and robustness increase, and its maximum throughput is limited by the assigned path characteristics.

This limitation led us to focus on studying techniques and research efforts capable of establishing data flows yielding robustness and bandwidth increase due to multipath aggregation. In the following, we describe some known aggregation techniques that match these characteristics. In general, such techniques can be deployed in different layers of TCP/IP protocol stack. We do not include Application Layer techniques since they are highly platform dependent and resource hungry when compared to lower layers implementations.

#### A. Link Layer

At the Link Layer, the *Linux Bonding Driver* (LBD) [4] can be used to distribute evenly data frames over multiple paths (i.e using mode 0). If multiple point-to-point Link Layer paths between two nodes are available, the LBD is capable of aggregating the multiple physical interfaces into a single virtual interface in each host. The data packets delivered through the virtual interface are encapsulated into frames that are evenly split by LBD across the available links and merged at the destination.

Similarly to LBD, *Multilink PPP* [5] is also capable of sending data segments over various PPP channels. Moreover, it can also cope with different capacity channels using two possible approaches [5]. The first approach consists in dividing the packets into variable segment size in proportion to the transmission rates of the channels. The second approach consists in dividing the packets into many equal size fragments and distribute these fragments by the channels, the number being proportional to the transmission rate of each channel.

Both the LBD and *Multilink PPP* have good performance when running over paths with stable and predictable characteristics in terms of bandwidth and latency, such as wired paths. However, when dealing with wireless paths, with high latency and capacity variability, the packet reordering can affect the transmission efficiency. When using the LBD, the latency and capacity of the multiple paths must be equal, otherwise the packet reordering phenomena will occur. The same happens when using *Multilink PPP*, even if the capacity difference is compensated with appropriate segment distribution, but a latency variation exists, the reordering phenomena will occur. Thus, a Link Layer approach for aggregating multiple paths without any kind of feedback mechanism for compensating path characteristics variability is not a suitable solution.

#### B. Network Layer

A number of research works address the aggregation problem in the Network Layer. For example, Gurtovet et al. [6] work takes advantage of the Multihoming and advanced security features of Host Identity Protocol to implement a Fastest Path First scheduling algorithm. The authors managed to successfully simulate a TCP connection spread over multiple paths with minimal reordering, although the latency of the paths was constant, which does not correspond to a real scenario.

Chebrulu et al. [7] propose a proxy based reordering minimization algorithm for wireless paths. By introducing reordering buffers and estimating path bandwidth for conducting a weighted data distribution over the different paths, the authors present simulation results showing that their proposal can achieve good bandwidth aggregation. However, their method requires knowing the status of base station queues, which is not directly accessible in practice. Evenson et al. [8] use a similar approach but they consider a Delay Equaliser component, running at sender side, that compensates path latencies mismatch. Despite the emulation results show that packet reordering is significantly reduced [8] [9], the additional delay added to the fastest paths lowers the transmission efficiency, due to the increase of the round trip times.

#### C. Transport Layer

The majority of Transport Layer aggregation techniques are focused on the addition of multipath transmission capability to already standardized and vastly deployed protocols. This approach ensures compatibility with already existing upper layer protocols and network middleboxes (Firewall, NAT, etc.) A promising work and moving towards standardization is MultiPath TCP (MPTCP) [10], capable of splitting a single TCP connection over multiple heterogeneous paths. By establishing a TCP subflow per path, with independent sequence numbers and congestion algorithms, the transmission efficiency in each path is maximized. Moreover, each TCP subflow carries an additional Data Sequence Number used for reordering the data packets before they reach the receiving application [11]. MPTCP has already a Linux Kernel implementation [12] suitable for development and evaluation that we will use in Section IV-C.

Another example is MultiPath SCTP (MPSCTP) [13], one of many SCTP modifications that take advantage of the multiple IP failover capability to aggregate the resources of multiple paths. Under the assumption that all the acknowledgement messages are received, the simulation results show that MPSCTP is better than MPTCP [13], although the given assumption can not be guaranteed in practice, especially in wireless networks.

## IV. MULTIPATH AGGREGATION TECHNIQUES EVALUATION

All the previously identified techniques may achieve good results, but they are somehow conditioned, either by the stability of the paths, simulation environments or assumptions that do not correspond to real network environments. The goal of our work is to evaluate several techniques under real network environments and to propose improvements capable

of enhancing the transmission efficiency when heterogeneous communication channels are used. Below, we describe our Evaluation Testbed and evaluate the *Linux Bonding Driver* and the MPTCP using wireless communication channels.

### A. Evaluation Testbed

The evaluation testbed is shown in Figure 1. It consists on two computing nodes with networking capability, node A and node B, and two User Equipments (UE), UE A and UE B. Communication between UE A and UE B is to be conducted through regular data transport protocols (i.e. without any modifications). All aggregation algorithms run on nodes A and B and are completely transparent to both user equipments. By adopting a proxy based approach, UE A reaches UE B via aggregated channels, experiencing a faster and robuster connection.

Both node A and node B are based on the ALIXD13 (Figure 2) system boards with a 500 MHz AMD Geode LX800 CPU and 256 MB of RAM. Among other features, ALIX2D13 offers three 10/100 Ethernet ports, four USB ports and a miniPCI, enabling the interface with various telecommunication technologies, such as cable modems (via Ethernet ports), and 2G/3G/4G modems, WiFi or Bluetooth (via USB/miniPCI). Node A and B can run various embedded Linux flavours, providing a highly customizable and feature rich platform for developing and experimenting multipath aggregation algorithms. The use of embedded platforms will make us aware of the limited resources impact when testing existing and developing new aggregation algorithms. Therefore, besides validating aggregation algorithms, this testbed also aims at testing them in platforms with limited computational resources.

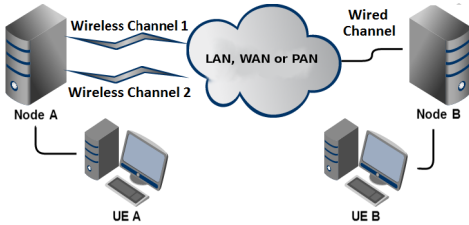


Fig. 1. Network of Evaluation Testbed



Fig. 2. ALIX2D13 platform used in Node A and B

### B. Linux Bonding Driver

To evaluate LBD under real wireless networks, Node A was connected to the Internet by two similar 3G USB modems (average transmission rates of 1.7 Kbps up link and 5.1 Mbps down link, measured at link layer) and Node B was connected to the Internet by a broadband access link (average

TABLE I. IMPACT OF PACKET REORDERING ON A TCP DATA FLOW

|                       | TCP flow over one wireless channel | TCP flow over two wireless channels |
|-----------------------|------------------------------------|-------------------------------------|
| No traffic shaping    | 3.6 Mbps                           | 2.62 Mbps                           |
| 1Mbps traffic shaping | 0.78 Mbps                          | 1.53 Mbps                           |

transmission rate always above 20 Mbps in both directions, measured also at link layer). In order to run LBD, we have created two independent VPN tunnels over the Internet, one per wireless channel, interconnecting node A and node B. The nodes were running Debian 6 with LBD module loaded. The measured Round Trip Time (RTT) of each tunnel was 110 ms.

In the first experiment, a file was downloaded from node B to node A over one wireless channel, achieving an average throughput of 3.6 Mbps. Afterwards, the same file was downloaded using LBD, and the average achieved throughput was 2.62 Mbps.

When dealing with wireless channels, even if they look similar (same operator and characteristics), the underlying wireless link retransmission mechanisms causes variability in the latency and capacity of the communication channel [3]. This variability causes packet reordering at the receiver and therefore the transmission throughput is lowered by the packet retransmissions, as described in Section II.

The simplest way to overcome the packet reordering negative effect consists in shaping (via token bucket) the sender throughput to a value smaller than the minimum instantaneous link capacity. In this way, the packets arrive in order and a throughput increase is expected when transmitting data simultaneously over two wireless channels. In a second experiment, traffic was shaped to a maximum of 1 Mbps in each link. We observed a throughput of 0.78 Mbps over one channel rising up to 1.53 Mbps when two channels were used, thus confirming our assumption. Table I presents the obtained results.

Despite the fact that traffic shaping overcomes the bandwidth aggregation problem, it is not a good approach, since the network resources are not fully utilized. An efficient algorithm operating at the Link Layer must have a frame scheduler that probes channel bandwidth, for weighted frame distribution, and probes the channel latency to predict the arrival instants and send reordered frames that will arrive in the right sequence due to latency mismatches between paths.

### C. MPTCP evaluation

We then proceeded to test MPTCP. The network configuration was the same as in the LBD evaluation. Node A was equipped with two 3G wireless modems and node B was well served by an internet broadband connection. Both nodes were connected through two independent VPNs and were running Debian 6 with MPTCP v0.88 Linux kernel [12] with the required changes in the routing tables.

To evaluate MPTCP, a data stream, lasting 60 seconds, was repeatedly sent from node A to node B and throughput was measured using *iperf* tool. We have conducted 30 consecutive measurements. The first 10 measurements were conducted with only the first channel active. The next 10 measurements were conducted with both channels active. The last 10 measurements were conducted with only the second channel active.

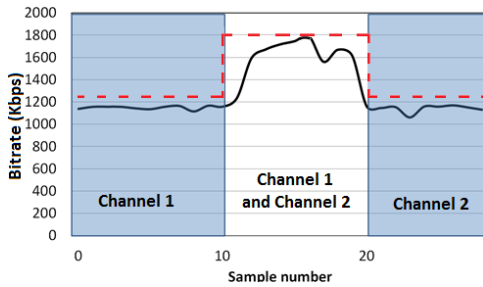


Fig. 3. Uplink Bitrate aggregation using MPTCP

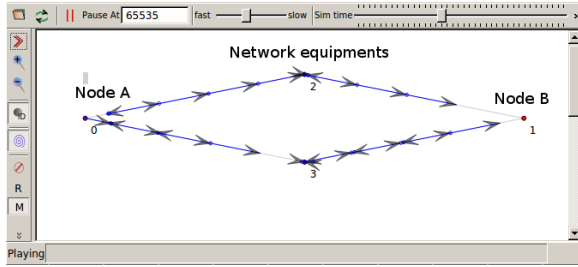


Fig. 4. NetAnim visualization of ns-3 DCE simulation network

This experiment was also evaluated through simulation with ns-3 and Direct Code Execution framework (ns-3 DCE). ns-3 DCE was configured to incorporate TCP/IP stack of MPTCP kernel instead of the standard Linux stack. Node A and B were connected by two point-to-point paths with a capacity of 1.7 Kbps and a RTT of 110 ms. In the middle of the paths we placed intermediate nodes that represent operator network equipments (Figure 4). These intermediate nodes were configured to induce 1% packet loss characteristic of the 3G wireless networks [3]. Under the simulation environment, the achieved results were 1.26 Mbps with one channel and 1.8 Mbps with the two aggregated channels.

The results are presented in Figure 3, where the straight line represents experimental measurements and the dashed line represents simulated measurements. As we can see, the achieved throughput when Channel 1 or Channel 2 is used is similar ( 1.18 Kbps on average). When both channels are used together with MPTCP, the data rate is increased to an average value of 1.68 Kbps and, therefore, throughput aggregation was obtained. Note, though, that the average aggregated throughput is not twice the average throughput of individual channels, which means that additional efficiency might be achieved with better aggregation algorithms.

The simulation results are similar to, but slightly better than, experimental measurements because the simulations only took into account packet loss, and did not consider path latency variability. In addition, the constant shape along samples in simulation results is due to the fact that simulation parameters were constant over simulation time.

When experimenting and simulating MPTCP, we noticed that throughput aggregation becomes significant only when long data streams are transmitted. When small files are transmitted, the achieved throughput is equal to a single path throughput. This behaviour was also documented by Chen et al. [14], who did a measurement based study of MPTCP

performance under different data flow sizes and concluded that most of the small file transfers are concluded before MPTCP can establish alternative paths.

## V. CONCLUSION AND FUTURE WORK

In this Work In Progress, we have presented a testbed for evaluating aggregation techniques under real network environments. We have used the testbed to evaluate a Link Layer approach (LBD) and a Transport Layer approach (MPTCP). From our evaluation, we have concluded that the LBD is not suitable for wireless link aggregation and MPTCP is inefficient when dealing with small size file transfers. We plan to continue evaluating and developing algorithms at different layers of communication stack and identify the problems caused by high variability networks. In parallel, we plan to develop, characterize and model various aggregation algorithms with ns-3 simulator. The goal is to be capable of deploying aggregation algorithms in different platforms, identifying the most adequate solutions.

## REFERENCES

- [1] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," IETF RFC Standard no. 5681, September 2009.
- [2] D. Kaspar, K. Evensen, A. Hansen, P. Engelstad, P. Halvorsen, and C. Griwodz, "An analysis of the heterogeneity and ip packet reordering over multiple wireless networks," in *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, July 2009, pp. 637–642.
- [3] M. C. Chan and R. Ramjee, "Tcp/ip performance over 3g wireless links with rate and delay variation," *Wireless Networks*, vol. 11, no. 1-2, pp. 81–97, 2005.
- [4] T. Davis, *Linux Ethernet Bonding Driver HOWTO*, apr 2011.
- [5] K. Sklower, B. Lloyd, G. McGregor, D. Carr, and T. Coradetti, "The ppp multilink protocol (mp)," IETF RFC Standard no. 1990, August 1996.
- [6] A. Gurtov and T. Polishchuk, "Secure multipath transport for legacy internet applications," in *Broadband Communications, Networks, and Systems, 2009. BROADNETS 2009. Sixth International Conference on*. IEEE, 2009, pp. 1–8.
- [7] K. Chebrolu, B. Raman, and R. R. Rao, "A network layer approach to enable tcp over multiple interfaces," *Wireless Networks*, vol. 11, no. 5, pp. 637–650, 2005.
- [8] K. Evensen, D. Kaspar, P. Engelstad, A. Hansen, C. Griwodz, and P. Halvorsen, "A network-layer proxy for bandwidth aggregation and reduction of ip packet reordering," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, Oct 2009, pp. 585–592.
- [9] D. Kaspar, "Multipath aggregation of heterogeneous access networks," Ph.D. dissertation, University of Oslo, 2011.
- [10] "MPTCP status pages," <http://tools.ietf.org/wg/mptcp>, accessed: 05-06-2014.
- [11] O. Bonaventure, M. Handley, and C. Raiciu, "An overview of multipath tcp," *USENIX login*, October 2012.
- [12] C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath tcp," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI'12), San Jose (CA)*, 2012.
- [13] S. Shailendra, R. Bhattacharjee, and S. Bose, "Mpsctp: A multipath variant of sctp and its performance comparison with other multipath protocols," in *Communication Systems (ICCS), 2012 IEEE International Conference on*, Nov 2012, pp. 280–284.
- [14] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of multipath tcp performance over wireless networks," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 455–468. [Online]. Available: <http://doi.acm.org/10.1145/2504730.2504751>