Multipath Algorithms and Strategies to Improve TCP Performance over Wireless Mesh Networks

David Gómez, Carlos Rabadán, Pablo Garrido, and Ramón Agüero

Universidad de Cantabria, Santander, Spain
dgomez,ramon@tlmat.unican.es
{carlos.rabadan,pablo.garrido}@alumnos.unican.es

Abstract. The remarkable growth at the worldwide wireless device sales, together with the cost reduction of the subjacent technologies, has lead to a situation in which most of this type of terminals carry more than one interface to access the network, through potentially different radio access technologies. This fact has fostered the interest of the research community to address new solutions to exploit the possibility of launching multiple simultaneous transmissions through multiple interfaces. In this work we evaluate three different routing algorithms (*link*, *node* and *zone* disjoint) that aim to discover the optimal route configuration of disjoint paths over a wireless mesh network. We use the obtained results to evaluate, by means of simulation, the performance of the MultiPath TCP (MPTCP) protocol, which allows the simultaneous delivery of traffic across multiple paths, showing that the aggregated performance is significatively higher than the one achieved by the traditional single-path and single-flow TCP.

Keywords: Wireless Mesh Networks, Multipath Routing Algorithms, MPTCP, Multi-homed devices.

1 Introduction

Wireless technologies are probably one of the most relevant elements in the current communication realm. Besides the legacy wireless devices (e.g. cellphones, laptops, etc.), a new batch of increasingly popular equipment is looming, such as smartphones or tablets, which shows the huge potential of this type of communications. In absolute terms and at the time of writing, the number of wireless devices sales easily surpass 10^9 units; in fact, it is more and more usual that an average user owns several gadgets/devices. This trend is likely to continue during the near future, and designers and manufacturers will develop new ways to use these technologies, easing the end users' life.

Some of these devices will be able to get interconnected amongst themselves, leading to the so-called Wireless Mesh Networks (WMNs). In this sort of topologies, it will be (most of the times) necessary using several hops to reach the destination, by means of intermediate relay nodes. In order to establish one (or more) paths, the routing algorithm shall provide the set of appropriate paths

D. Pesch et al. (Eds.): MONAMI 2013, LNICST 125, pp. 15–28, 2013.

to communicate two nodes (unicast transmissions). On the other hand, there are two main mechanisms and protocols using such algorithm, *reactive* (or "on-demand") and *proactive*. Those which belong to the first group only exchange discovery or maintenance messages when needed, whilst the second group periodically updates the routing tables, thus causing a higher overhead.

Likewise, the manufacturers tendency to include multiple interfaces into their devices has become a reality. This phenomenon calls the design and implementation of novel protocols to allow the simultaneous usage of all the available resources at the different "access elements". Although there are various solutions dealing with this functionality, MPTCP has deservedly become one of the most relevant ones, heavily supported by its own IETF working group, exclusively devoted to the accurate development of this protocol, as well as a set of extensions that were conceived to complement its basic features and performance. MPTCP is in fact an evolution of the legacy TCP, and it shares most of its architecture. MPTCP allows to divide the load between different interfaces (provided that at least one of the nodes has more than one active IP address), thus boosting the traditional TCP performance.

This work is structured in two clearly differentiated stages: first, we evaluate the behavior of three different routing algorithms (namely, *link*, *node* and *zone* disjoint) so as to find the optimal set of disjoint paths over a WMN; afterwards, using the results of the first phase, we assess, based on an extensive simulation campaign over the `ns-3` simulator, the MPTCP performance over this type of topologies, showing the enhancement compared to a traditional single-path TCP scheme.

The structure of this document is organized as follows: Section 2 briefly outlines the main related works, highlighting the novel aspects addressed in our work. Section 3 introduces the three routing algorithms that will be exploited for the MPTCP characterization; the operation of this protocol will discussed in Section 4. Section 5 depicts the most relevant results and discusses the potential benefits and drawbacks of multipath strategies. Finally, Section 6 concludes the paper and advocates some research lines that will be tackled in the future.

2 Related Work

In this work we exploit different routing algorithms to be used by multi-path strategies, assessing their potential benefits over WMNs. The use of multi-path communications might lead to a greater performance; moreover, they will bring about a more resilient connections, dynamically adjusting the load over the various paths, according to the particular network conditions.

As mentioned earlier, the classification of routing protocols for wireless multi-hop networks embraces two main groups. Both of them are based on mechanisms to discover and maintain routes over multi-hop networks. The *proactive* protocols (represented by *Optimized Link State Routing - OLSR* [5]), update the routing information by periodically flooding the network with topological information, thus introducing a remarkable overhead. On the other hand, the *reactive* or *on-demand* protocols (for instance, *Ad hoc On-demand Distance Vector routing - AODV* [14]) reduce, as much as possible, the exchange of control messages, triggering them only if needed.

The first generation of routing protocols was thought to operate with *single-path* strategies, where in case of a route break, the source node would need to start a new discovery process, in order to find alternative paths to reach the destination. The increasing interest on multi-path solutions during the last years opens new possibilities, and the implementation of new protocols is required. Most of the existing solutions are modifications of single-path protocols and can be classified according to how they select the alternative paths to the shortest one: *Link Disjoint (LD)*, which only excludes the links of the previously calculated routes (e.g. *Ad hoc On-demand Multipath Distance Vector - AODVM*), *Node Disjoint (ND)*, which does not allow any intermediate node to be active in two different routes (e.g. *Geographic Multipath routing Protocol - GMP*) and, finally, *Zone Disjoint (ZD)*, which inhibits the redundant participation of both the previously used nodes as well as their corresponding neighbors (e.g. *Zone Disjoint Multipath extension of the Dynamic Source Routing - ZD-MPDSR*).

Some of the most relevant works within this research line were carried out by Meghanathan [11,12], who, by means of graph theory, realizes a complete performance analysis of the link, node and zone disjoint algorithms over mobile ad hoc networks, where he thoroughly studies, through different simulation campaigns, different performance metrics that characterize the behavior of different routing schemes (e.g. number of routes found, average number of hops, average time between single/multipath route discoveries, etc.) Moreover, Waharte et al. [19] carry out another analysis that focused on LD and ND, paying special attention to the potential interferences between the different subflows (since they share the same channel), and estimate the resulting throughput as a function of the nodes' coverage area and their position within the scenario. Unlike Meghanathan's contribution, which only addresses the fundamental analysis of the routing algorithms, Waharte et al. apply end-user traffic (over UDP) to compare the performance of the different routing multipath solutions to that shown by a single-path scheme.

After finding the set of disjoint paths between a source node and a destination, we need to develop a solution to split a single connection into multiple subflows. Some proposals, based on the modifications of the legacy TCP operation have been already made (e.g. *mTCP* [22], *R-MTP* [10], *pTCP* [8]). The relevance of this type of communications is supported by the presence of standardization bodies, such as IETF, in the development of new protocols and techniques. In this sense, there are two working groups exclusively devoted to the design and implementation of the most relevant multi-path solutions: Stream Control Transmission Protocol (SCTP) [18] and MPTCP [7]. The former one uses multiple routes in order to provide some redundancy against failures, or to ease the mobility between different networks without breaking a session (at the transport level), but it does not support (yet) the simultaneous transmission over different paths; on the other hand, MPTCP focuses on the improvement of the TCP performance by multiplexing traffic load over different resources.

Regarding the analysis of MPTCP over wireless scenarios, we can highlight [9,15,13], all of them following complementary approaches, ranging from real scenarios (with emulated channels) over a Linux Kernel implementation [1], which shows a great improvement over the traditional TCP operation. On the other hand, the authors of [13] identify an important drawback if the physical attributes are rather different (e.g. IEEE 802.11 and 3G), due to the impact of the packet reordering algorithms. A common element of all these works is that they are based on rather simple topologies, consisting on one, or two hops.

Finally, it is worth highlighting the contribution of Chihani et al. [4], who implemented a fully-fledged MPTCP framework implementation for the `ns-3` simulator, which served as the basis for the work developed herein, since we ported it to a newer version of the simulator, adapting its operation so as to use it over wireless technologies. Chihani et al. analyzed the performance of different congestion control algorithms [16], and compared the behavior of the congestion window at each subflow, using an FTP transmission over a simple wired topology, encompassing two terminals which were directly connected through point-to-point links.

3 Multipath Strategies Routing Algorithms

The main goal of the different multipath routing algorithms consists in finding an optimal set of disjoint paths to simultaneously carry the traffic load using multiple subflows, over a WMN scenario.

In order to describe the operation of each of the algorithms (*link*, *node* and *zone disjoint*), we will employ a traditional graph theory notation, as shown below.

Let G(V,E) be the graph representing the scenario over which we want to get the set of paths (using the LD, ND or *ZD* algorithms)[1] between the source and the destination nodes (s and d, respectively). The set V represents the group of vertices (nodes) deployed within the scenario, and E (edges) is the set of existing links. We will establish a link between two nodes if the distance between them is shorter than the corresponding range of transmission. In this work we will use homogeneous nodes, and all of them will share the same coverage.

The first step to get the set of paths is the same for the three algorithms: the Dijkstra's algorithm is used to find the shortest path between s and d. If there is, at least, one route in G, it is stored in the corresponding set (P_L, P_N o P_Z, for the LD, ND or ZD algorithms, respectively). After that, the graph ($G \rightarrow G'$) is updated with the constraints imposed by each of the algorithms. Below we show the procedure followed by each of the solutions:

---

[1] In this work, since the nodes do not move, the subjacent topology will stay static during the simulation time; therefore, we only need to calculate the routes once.

- **Link Disjoint.** We will remove from G all the links that were found with the Dijkstra's algorithm, thus building a new graph G'(V, E^L). The procedure is repeated as many times as there is a route $s - d$ (Dijkstra's algorithm is used again), incorporating the resulting path to P_L. When the algorithm is finished, P_L contains the set of link disjoint paths of the original graph G.
- **Node disjoint.** In this case, the graph is modified by deleting the nodes belonging to the previously selected path. Therefore, after each iteration a new route is added to P_N and a modified graph G'(V^N, E^N) is built. The procedure is executed as long as s can discover a route to d.
- **Zone disjoint.** This is the most restrictive algorithm, since it severely limits the graph between successive iterations, deleting the nodes belonging to the previous route, as well as their neighbors; as a result, the original graph G is modified to G'(V^Z, E^Z). When it becomes impossible finding new routes, the algorithm returns the set of routes P_Z.

In this work the routing tasks have been performed on an external framework, outside the MPTCP implementation, using a proprietary tool developed in `C++`, which generates a random scenario to establish the graph G (V,E). Afterwards, the route selection procedure was performed by means of a single process.

Since the main objective of this work is to analyze the performance of MPTCP, which simultaneously delivers the information over multiple (disjoint) paths, we will only consider as valid those sets (P_L, P_N or P_Z) with more than one path between s and d.

4 MPTCP as a Multipath Transport Level Solution

MPTCP was conceived as an evolution of the TCP protocol, the most relevant transport level solution, although its performance over wireless links has been questioned. Its appearance is tightly linked with growing availability of devices with multiple interfaces[2].

The basic principle of MPTCP is rather simple: if a terminal has multiple points of connection (interfaces) this can be exploited, simultaneously dividing the traffic between different subconnections. Thanks to these multipath strategies, the overall performance is improved, as well as the robustness of the communication. In MPTCP, for instance, the traffic can be drifted from one subflow to another one after a link (or node) fault.

In order to ease the migration from legacy protocols, ensuring the backward compatibility with TCP, RFC 6824 [7] establishes that any MPTCP implementation must be able to support any non-MPTCP-aware application; in such cases, the services will not be able to differentiate between MPTCP and TCP transport level connections. In this sense, MPTCP can be seen as a modified TCP version, sharing most of its architecture and adding different extensions to cope with the most relevant features.

---

[2] They are usually referred to as “multi-homed” devices.

<table>
<tr><td colspan="2">Application</td></tr>
<tr><td colspan="2">MPTCP</td></tr>
<tr><td>Subflow (TCP)</td><td>Subflow (TCP)</td></tr>
<tr><td>IP</td><td>IP</td></tr>
</table>

Fig. 1. MPTCP architecture

Another requirement to be fulfilled, according to the aforementioned RFC, is that at least one of the edge nodes must have more than one IP address for the correct establishment of a multi-path session.

Once the core operation of MPTCP has been outlined in [7], its main challenge can be mapped onto the accomplishment of the following three goals:

1. *Improve throughput*: The performance of a multipath connection should be, at least, alike the one shown by legacy TCP (assuming the best available path is used).
2. *Do not harm*: An MPTCP subflow should not take more resources than the ones consumed by the traditional TCP using the same path.
3. *Balance congestion*: Upon a congestion situation, MPTCP should offload as much traffic as possible from the most congested paths.

Once we have described the most relevant functionalities of the MPTCP protocol and its main goals, Figure 1 depicts its architecture within the TCP/IP model. As can be seen, it is placed at the transport level and, at the same time, it includes two different sublayers: the first one handles the application-oriented issues (e.g. session initialization/finalization, subflow discovery/establishment, etc.); on the other hand, the lower level will embrace one instance per subflow established during the TCP initialization phase. Additionally, each of these instances will be associated to a different IP entity, to which they will send the outgoing packets down.

One of the major challenges that MPTCP has to face is the need to ensure an efficient "resource pooling" [20]). In order to accomplish this goal and, at the same time, fulfill the three previously described objectives, a congestion control algorithm needs to be used so as to provide a coupled operation of the various congestion windows. Although the protocol supports a number of solutions, in this work we will assume that there is an independent congestion window per subflow[3]. A congestion controller will monitor (congestion windows sizes) the aggregated throughput of the transmission, paying special attention to the fulfillment of the MPTCP three goals: provide a higher throughput than TCP (**Goal 1**), without taking more resources than necessary (**Goal 2**) and taking as much load as possible from the most congested paths (**Goal 3**). In order to estimate the load of a simple TCP flow, the control entity measures the packet

---

[3] It is worth mentioning that, while the (additive) increase congestion windows expressions are specific to MPTCP, it does not modify the legacy TCP operation upon a packet loss, which will lead to a (multiplicative) decrease of the congestion window.

loss rate and the Round Trip Time (RTT), providing a new congestion window value for each subflow. The relevance of this mechanism has made IETF to develop a recommendation just to address it [17].

5 Simulation Platform and Results

In this section we will depict the most relevant features of the different simulation campaigns and we will discuss the most outstanding results. These have been divided into two clearly different groups: first, we study and compare the behavior of the three different routing algorithms presented in Section 3; on the other hand, and using the paths found by those algorithms, a characterization of the MPTCP performance is carried out, showing that it can actually improve the behavior of the legacy TCP.

5.1 Multipath Routing Algorithms Behavior

As a previous step to carrying out the performance analysis of the MPTCP protocol, we used a proprietary software to analyze the operation of different multipath routing approaches. In particular, the tool takes the following steps: (1) deploy the nodes within the scenario, (2) execute the three multipath routing algorithms and (3) generate the output files that will be afterwards used on `ns-3` to perform the corresponding simulation campaign. We have established a set of aspects to be considered:

- The nodes will be deployed within a 100x100 meters squared area.
- Initially, disconnected graphs are discarded; i.e. only scenarios in which there is, at least, one path between any pair of nodes.
- In this work we do not consider node mobility, so nodes stay static during the simulation time.
- The coverage area of the nodes (disk radius model) is 20 meters.
- The source-destination nodes are selected so as to ensure same consistency to the multipath routes; by taking two points $(20, 50)$ and $(80, 50)$ as references (as shown in Figure 2); we select the source as the closest one to the first point and the receiver the closest to the latter reference point.

As an illustrative example, Figure 2 shows a random deployment of 16 nodes. In this particular topology we see that node 8 will take the transmitter role and node 3 will be the receiver. With regard to the selected routes, the three algorithms will provide the same result: the shortest path is $8 \rightarrow 13 \rightarrow 15 \rightarrow 3$, while the second option would be, for the three algorithms, $8 \rightarrow 11 \rightarrow 10 \rightarrow 12 \rightarrow 3$.

Figure 3 shows the percentage of multipath "feasible" topologies (those which there were two or more disjoint paths divided by the total number of runs)[4] as a function of the number of nodes. We can appreciate that *LD* always exhibits

---

[4] The experiment consisted in 1000 independent runs.

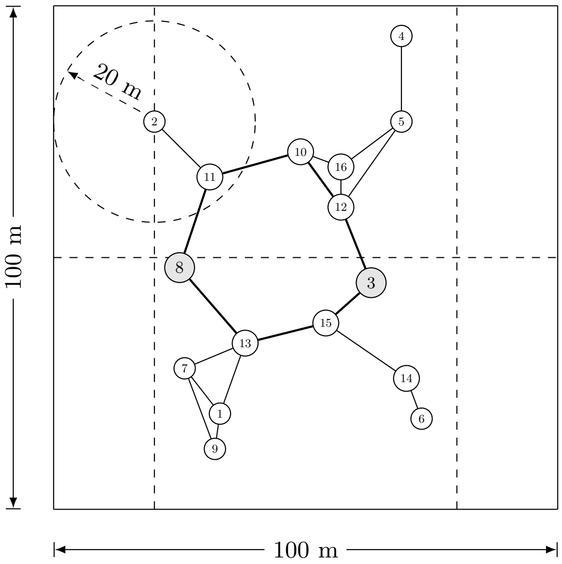


Fig. 2. Illustrative topology (16 nodes)

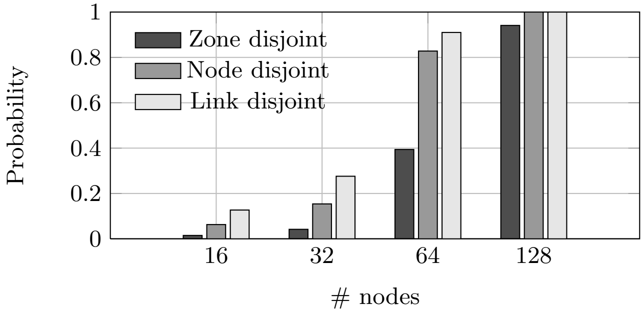
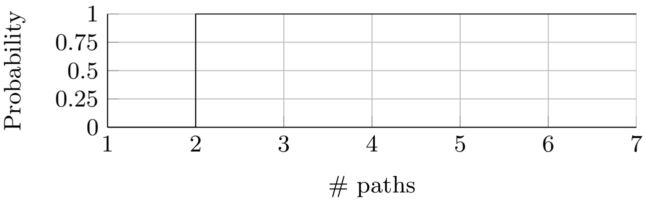


Fig. 3. Probability of finding a multipath strategy using the different routing algorithms

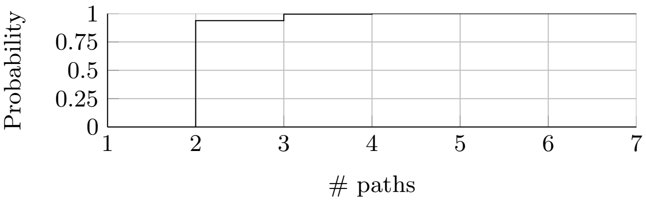
the best behavior, closely followed by *ND*; on the other hand, *ZD* appears as the most restrictive alternative.

After the first comparison, a new constraint was added. Only those topologies with, at least, two different routes (for the three routing algorithms) were considered. We used 32 nodes (all of them fulfilling the previous constraints) and generated 1000 scenarios. It is worth mentioning that, to get such a high number of deployments, many other scenarios were discarded, since, as shown in Figure 3, only 4.2% of the 32-node scenarios were multipath for the ZD algorithm (i.e. ZD found two or more paths between the two edge nodes).

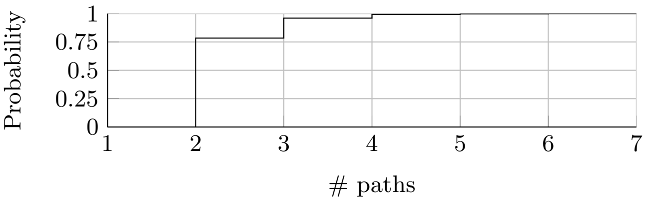
First, Figure 4 shows the cumulative distribution function (cdf) of the total number of routes found by each of the studied schemes. As could have been



(a) Zone disjoint



(b) Node disjoint



(c) Link disjoint

Fig. 4. *cdf* of the number of found routes for the different algorithms

expected, *LD* is the algorithm which provides the higher number of alternative paths, since it is the scheme which makes fewer changes to the graph between successive iterations. *ND* appears the intermediate solution, showing a non-negligible probability to discover three disjoint paths. On the other hand, the strong constraints imposed by *ZP* avoids finding more than two simultaneous routes.

Another insightful metric is the *cdf* of the number of hops of the two preferred routes, shown in Figure 5. As we can infer from the discussion given in Section 3, the shortest path (1st iteration) is the same for all the schemes, since all of them use the Dijkstra's algorithm to find it. However, the second alternate route length shows the same behavior as the previous statistic: *LD* finds, in the second iteration, the shortest path to reach the destination; *ND* appears again as the solution with the second shortest route, being *ZD* the scheme providing the longest paths. This is of outer relevance, since the number of hops will have a remarkable influence on the aggregated performance, since the greater the length of these paths the lower the throughput of the corresponding subflow.

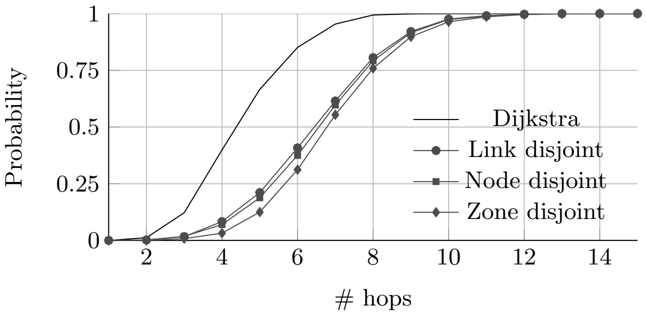


Fig. 5. Number of hops *cdf* for the two preferred routes

5.2 MPTCP Performance over Wireless Mesh Networks

After the analysis of the three different multipath routing algorithms over random deployments, we will now describe the simulation campaign, carried out over `ns-3` [2]. We use the outcome of the previous analysis, and nodes are connected by means of IEEE 802.11 links. It is worth recalling that we have ensured that all the simulated scenarios have, at least, two disjoint routes for each of the algorithms (we discarded these topologies not fulfilling this requirement). The input of this stage is the output of the previous one, in particular the following pieces of information: (1) the location of the nodes, and (2) the routes returned by the *LD*, *ND* and *ZD* algorithms (P_L, P_N and P_Z, respectively).

For each of the scenarios, we simulate the behavior of the following different transport level solutions, in order to study their performance:

1. **Single-path TCP.** It corresponds to the legacy behavior, and the path used (recall that we are using static routes) is the shortest one, which is alike for the three algorithms.
2. **Single interface MPTCP.** In this case, we configure two different IP addresses sharing the same interface. In this sense, the overall performance might get damaged, since both subflows share the same wireless channel, increasing the number of contending stations and the probability of suffering collisions.
3. **Multi-interface MPTCP.** This last configuration is expected to yield the highest performance, since we use different channels (non-overlapping) for the two subflows and, therefore, there will not be any interference between them. For this, both the transmitter and the receiver must have two different interfaces, each of them associated to a particular subflow.

To carry out the analysis, we have ported the MPTCP implementation provided by Chihani et al. [4], to a newer version of `ns-3` (`ns-3.13` instead of `ns-3.6`). This framework follows the IETF recommendations [7,6,17]. For this particular work, we have selected *Linked Increases* and *DSACK* as the congestion control and reordering schemes, respectively (for further information about them, the reader might refer to [4]). The MPTCP layer will be added at each node according to the corresponding configuration (1 or 2 interfaces), distributing the load between the two different subflows.

Besides, it is worth mentioning some additional aspects about the simulation setup:

1. The source node sends a 20 MB file to the destination (*unicast* traffic). Since the objective is to assess the upper bound performance for each of the configurations, we ensure that there is always a packet waiting to be delivered at the transmitter's buffer. In this saturated scenario, the wireless medium acts as the real bottleneck.
2. The subjacent technology is IEEE 802.11b (at 11 Mbps), setting a maximum number of transmissions per frame of four.
3. Since an external process is used to obtain the routes with the three algorithms, the routing scheme is based on static routes.
4. There is a single cause of packet losses: the collisions between simultaneous node transmissions. We will consider ideal channels, where the frame losses rate due to the wireless propagation effect is null.

Due to space constraints, we only report the results achieved with the routes provided by the LD algorithm, which correspond to highest performances.

Figure 6 shows the overall performance[5] for the three schemes. It represents the average, maximum and minimum values of throughput as a function of the number of hops used by the shortest path. We can clearly appreciate the improvement brought about by using the two channel scenario, achieving a higher aggregated throughput (e.g. 50% for the 2-hop scenario), compared to the traditional single-path TCP. However, we can find few cases with a lower performance, corresponding to these situations in which the second path needs many

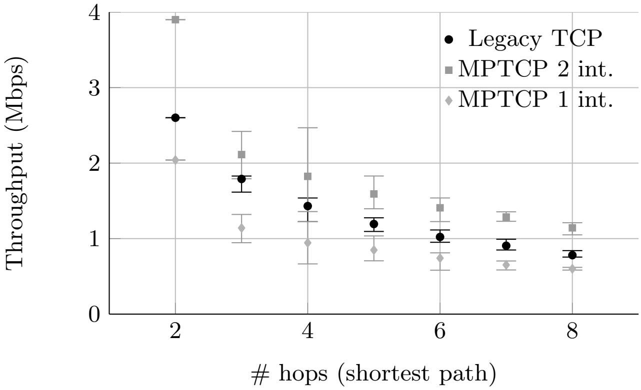


Fig. 6. Average throughput as a function of the number of hops of the first route

---

[5] In terms of throughput at the application level.

more hops than the first one. On the other hand, we can appreciate the limitation shown by the multipath strategy over a single channel WMN, since the contention caused by the high number of nodes contending for the channel leads to long idle times and a high probability of collision. The consequence is that the overall performance is quite lower than the one observed by the legacy TCP.

6 Conclussions and Future Work

In this work we have presented three different algorithms (link, node and zone disjoint) which were used to obtain the best set of disjoint paths over generic WMN topologies. We have focused on the use of multipath strategies over WMNs. We have compared their performance, in terms of feasibility (probability that there are two or more paths in a scenario), number of discovered paths and route length required to reach the destination node in such paths. According to the achieved results, the *ZD* algorithm seems too restrictive for the search of multiple disjoint paths.

Afterwards, using the outcomes of this first stage (the node deployment and the different routes between the source and the destination nodes), we compared the performance offered by the *MPTCP* protocol to the one exhibited by the legacy TCP, by means of a thorough simulation campaign carried out over the `ns-3` platform, leading to improvements of about $\sim 50\%$ in some of the cases.

The work undertaken so far opens a broad rante of aspects to be tackled in our future research. Below we briefly descuss the most relevant ones.

- Analyze different routing schemes, exploiting the presence of multi-channel devices by means of appropriate graph-theory models, as the one proposed by Yang *et al.* in [21].
- Increase the realism of the considered network environments, by introducing transmission errors over the wireless links. In this sense, it is well known that TCP performance heavily suffers from this type of losses, so it is interesting to see which is their effect over MPTCP. We would also like to introduce mobility to some of the nodes, analyzing the effect over the performance of various multi-path schemes.

Last, but not least, it is worth highlighting that all the MPTCP implementation, together with some additional documentation, can be found in [3].

Acknowledgements. This work has been supported by the Spanish Government by its funding through the projects **C3SEM**, *"Cognitive, Cooperative Communications and autonomous SErvice Management"* (TEC2009-14598-C02-01), and **COSAIF**, *"Connectivity as a Service: Access for the Internet of the Future"* (TEC2012-38754-C02-02).

References

1. MPTCP - Linux Kernel implementation, http://mptcp.info.ucl.ac.be/pmwiki.php?n=Main.HomePage
2. The ns-3 network simulator, http://www.nsnam.org/
3. Source code and documentation of the MPTCP implementation (ns-3.13), https://github.com/dgomezunican/multipath-ns3.13
4. Chihani, B., Collange, D.: A multipath TCP model for ns-3 simulator. CoRR abs/1112.1932 (2011)
5. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental) (October 2003), http://www.ietf.org/rfc/rfc3626.txt
6. Ford, A., Raiciu, C., Handley, M., Barre, S., Iyengar, J.: Architectural Guidelines for Multipath TCP Development. RFC 6182 (Informational) (March 2011), http://www.ietf.org/rfc/rfc6182.txt
7. Ford, A., Raiciu, C., Handley, M., Bonaventure, O.: TCP Extensions for Multipath Operation with Multiple Addresses. RFC (6824) (January 2013), http://www.ietf.org/rfc/rfc6824.txt
8. Hsieh, H.Y., Sivakumar, R.: A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts. In: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, MobiCom 2002, pp. 83–94. ACM, New York (2002), http://doi.acm.org/10.1145/570645.570656
9. Lim, M., Valdez, J.: MPTCP Wireless performance, http://reproducingnetworkresearch.wordpress.com/2012/06/06/mptcp-wireless-performance-draft/
10. Magalhaes, L., Kravets, R.H.: Transport level mechanisms for bandwidth aggregation on mobile hosts. In: Ninth International Conference on Network Protocols, pp. 165–171 (November 2001)
11. Meghanathan, N.: Stability and hop count of node-disjoint and link-disjoint multipath routes in ad hoc networks. In: Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMOB 2007, pp. 42–42 (2007)
12. Meghanathan, N.: Performance comparison of link, node and zone disjoint multipath routing strategies and minimum hop single path routing for mobile ad hoc networks. CoRR abs/1011.5021 (2010)
13. Nguyen, S.C., Nguyen, T.M.T.: Evaluation of multipath TCP load sharing with coupled congestion control option in heterogeneous networks. In: Global Information Infrastructure Symposium (GIIS), pp. 1–5 (2011)
14. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental) (July 2003), http://www.ietf.org/rfc/rfc3561.txt
15. Raiciu, C., Paasch, C., Barre, S., Ford, A., Honda, M., Duchene, F., Bonaventure, O., Handley, M.: How hard can it be? designing and implementing a deployable multipath TCP. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI 2012, pp. 29–29. USENIX Association, Berkeley (2012), http://dl.acm.org/citation.cfm?id=2228298.2228338
16. Raiciu, C., Wischik, D., Handley, M.: Practical congestion control for multipath transport protocols. UCL Technical Report (6824) (January 2009)
17. Raiciu, C., Wischik, M.H.D.: Coupled Congestion Control for Multipath Transport Protocols. RFC (6356) (January 2011), http://www.ietf.org/rfc/rfc6356.txt

18. Stewart, R.: Stream Control Transmission Protocol. RFC 4960 (Proposed Standard) (September 2007), (updated by RFC 6096) `http://www.ietf.org/rfc/rfc4960.txt`
19. Waharte, S., Boutaba, R.: Totally disjoint multipath routing in multihop wireless networks. In: IEEE International Conference on Communications, ICC 2006, vol. 12, pp. 5576–5581 (2006)
20. Wischik, D., Handley, M., Braun, M.B.: The resource pooling principle. SIGCOMM Comput. Commun. Rev. 38(5), 47–52 (2008), `http://doi.acm.org/10.1145/1452335.1452342`
21. Yang, Y., Wang, J., Kravets, R.: Interference-aware load balancing for multihop wireless networks. Technical Report (2005)
22. Zhang, M., Lai, J., Krishnamurthy, A., Peterson, L., Wang, R.: A transport layer approach for improving end-to-end performance and robustness using redundant paths. In: Proceedings of the Annual Conference on USENIX Annual Technical Conference, ATEC 2004, p. 8. USENIX Association, Berkeley (2004), `http://dl.acm.org/citation.cfm?id=1247415.1247423`