

Towards Backpressure Routing in Wireless Mesh Backhauls for Dense LTE Deployments

Natale Patriciello[†], José Núñez-Martínez[‡], Jorge Baranda[†], Maurizio Casoni^{*}, Josep Mangues-Bafalluy[†]

^{*}Department of Engineering *Enzo Ferrari*,
University of Modena and Reggio Emilia,
maurizio.casoni@unimore.it

[†]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA),
08860 Castelldefels, Barcelona, Spain.

{natale.patriciello, jorge.baranda, josep.mangues}@cttc.cat

[‡] Telefonica I+D,
08019 Barcelona, Spain.
jose.nunezmartinez@telefonica.com

Abstract—Future 5G networks will rely on heterogeneous deployment of micro eNodeBs, and is expected that wireless transport interfaces, deployed together with each eNodeB, will provide high-speed backhaul connectivity through the possibility to create an ad-hoc wireless mesh between the deployed micro eNodeBs. Therefore, the backhaul routing strategy will play a fundamental role in exploiting the backhaul resources. In this context, we aim to investigate the advantages and limitations of qualitatively different backhaul routing strategies as UEs download TCP and UDP content. In this paper, we compare a backpressure-based strategy that takes *per-packet* or *per-flow* basis (BP-MR) and traditional single path routing (OLSR). Ns-3 simulations show that the *per-packet* BP-MR variant enhances UDP performance by circumventing congestion but degrades TCP because of reordering. The *per-flow* BP-MR variant minimizes retransmissions by limiting the excessive reordering, and outperforms OLSR performance for both UDP backhaul traffic and TCP traffic.

Index Terms—TCP, wireless mesh backhaul, LTE, backpressure, routing, load balancing.

I. INTRODUCTION

Bringing the network close to the users is the leading technique for improving coverage and spectrum efficiency in modern 5G deployments [1]. In fact, the rising demand of wireless data services can be accommodated by dense, and portable, Long Term Evolution (LTE) deployments. After a disaster, this is also one of the most effective strategy to build a temporary network to support the first responder operators during the emergency. One of the main challenges in this scenario is the traffic backhauling, a problem that can be divided in two design phases: (i) choose the transport technology and (ii) choose a routing protocol/load balancing scheme that can exploit the strengths of the transport technology.

The first phase, being unlikely to have fiber connections for every access node (in LTE terminology, it is called eNodeB), especially after a disaster, can be addressed with wireless transport nodes, a possibility already detailed in [2]. Each eNodeB will feature one or multiple Point-To-Point (PTP) backhaul interfaces, hence enabling the creation of a path

redundant backhaul Wireless Mesh Network (WMN). In this way, LTE user and control plane traffic will be appropriately carried from/towards the Evolved Packet Core (EPC). Coupled with the possibility of employing portable LTE infrastructures for establishing coverage and increasing capacity of existing networks, it is possible even to deploy self-organizing networks: each Small Cell (SC) (Evolved Node B (ENodeB) + wireless transport interface) can be assigned to a position, and then connected through near SCs forming a temporary ad hoc network topology. This allows people and devices to seamlessly communicate in areas with no pre-existing infrastructures (e.g., disaster recovery, battlefield environments and so on) and, coupled with a satellite backhaul [3], to receive data from/to an external network, such as Internet.

The second phase is centered about the routing strategy. In the aforementioned context, key features of a routing strategy are the load balancing capabilities (to fully exploit all deployed resources) and the network performance achieved when the traffic volume substantially increases. One of the most relevant performance parameters is the Transmission Control Protocol (TCP) behavior (e.g., TCP throughput), given its predominance in the Internet due also to the increasing popularity of on-line streaming services (e.g., YouTube, Netflix, Spotify). However, throughput is not the only parameter that affects the end users' experience: latency is important as well, if not critical in upcoming deployments.

A. Related Work

Load balancing could be conducted at the Radio Access Network (RAN) segment. Indeed, transport-layer protocols such as MultiPath TCP (MPTCP) can help balance traffic to User Equipment (UE)s in multi-homed environments (e.g., a UE equipped with both LTE and WiFi). However, we focus on the study of load balancing backhaul traffic in a single-homed RAN environment.

At the mobile backhaul segment, multi-path routing protocols operating at the IP layer can offer better resource

utilization, and better reliability, thanks to the possibility to exploit the path diversity. For interested readers, there is a complete survey on multi-path routing and provisioning in [4]. Nevertheless, in mobile backhaul networks state of the art routing, traditionally adopted, uses one single path between endpoints in the absence of failures (i.e. Multiprotocol Label Switching (MPLS) depicted in RFC 5921 or Optimized Link State Routing (OLSR) presented in RFC 3626). Moreover, congestion unawareness and unpredictability of mobile traffic patterns makes single-path routing not efficient for a resource-constrained WMN, as it does not exploit the multiple paths provided by the mesh. By looking to the literature, theoretical backpressure [5] offers the possibility to exploit all backhaul resources (given its throughput optimality), one of the main operator wishes towards a cost-efficient deployment.

For readers interested in the history of backpressure-based routing protocol, in [5] there is a comprehensive survey of backpressure state-of-the-art. The root concept consists in a centralized policy which routes traffic in a multi-hop network by minimizing the sum of the queue *backlogs* in the network among time slots. Basically, if we define as *backlog* the queue size at nodes, the main idea of backpressure is to give priority to links and paths that have higher differential *backlog* between neighboring nodes; from a network perspective, these strategies dynamically map the trajectory followed by each data packet to the more underutilized paths, hence making congestion-aware decisions. However, these decisions may potentially make the path followed by consecutive packets of the same flow disjoint. Despite the throughput optimality promise, many backpressure-based protocols (included the original one) present problems which limit their applicability in the real world, such as centralized control mode, high queuing complexity due to the maintenance of a per-flow queuing system, and poor delay performance. Recently, many proposals have been presented to alleviate the effect of these issues [5].

For the purpose of this paper, we are interested in the TCP performance, degraded by the reordering typical of backpressure algorithms. In [6], it is proposed a delayed reordering algorithm at the destination that eliminates TCP timeouts while keeping packet reordering to a minimum. Furthermore, in [7] it is shown that TCP experiences incompatibilities with backpressure strategies that maintain per-flow queues, hence leading to unfairness between flows. In contrast, in this work we keep the TCP layer unmodified and we employ *BP-MR* [8], a backpressure strategy which has a simpler and more scalable implementation, since it maintains per-interface queues. The per-flow variant of *BP-MR* was proposed in [9] with the objective to improve TCP performance in an emergency scenario with a satellite backhaul. In [10], the variants were compared over a regular mesh and an irregular ring-tree topology, and using many different TCP protocols. The work in [11] shows that this variant (per-flow) of *BP-MR* is also suitable for hybrid satellite-terrestrial LTE backhaul networks when using both UDP and TCP transmission protocols. The main difference between the variants is that *per-packet* computes the best possible next-hop for each packet forwarded, while *per-flow* calculates it just for the first packet of a flow, and then saving

the pair next hop/flow in a forwarding table.

B. Our Contribution

In this paper, assuming dense deployments formed by a LTE RAN backhauled by a WMN, we aim to investigate the advantages and limitations of qualitatively different routing strategies. In this direction, we use two variants of a backpressure-based routing protocol (*BP-MR* [8] per-flow and per-packet) and OLSR, comparing User Datagram Protocol (UDP) and TCP performance on top of these routing strategies, focusing on the scalability of the system (in the sense of evaluating the performance when the users increase and when the traffic increases).

The conducted experiments allow concluding that reducing the granularity from per-packet to per-flow decisions offers the best trade-off between resource utilization and TCP performance of LTE mobile device. More specifically, simulation results obtained with ns-3¹ reveal that the *per-flow BP-MR* variant offers the best TCP performance despite its higher Round-Trip Time (RTT), since it experiences fewer losses and reordering events, showing its potential in being a good routing protocol candidate in non-mobile (referred to access nodes, not users) dense deployments, without necessarily resorting to an unnecessary over-provisioning of backhaul resources.

The remainder of this paper is organized as follows. Section II presents the reference scenario and methodology before showing the evaluation of *BP-MR per-packet*, *per-flow*, and *OLSR* in Section III. Finally, Section IV concludes the paper.

II. REFERENCE SCENARIO

As mentioned in Section I, future large-scale/dense SC mesh topology will provide plentiful of resources and path redundancy. We have modeled a mesh network deployment covering 1Km^2 with inter-SC distance of two hundred meters, depicted in Figure 1. In our model twenty-five LTE eNodeBs are deployed with peak downlink throughput of 350 Mb/s, corresponding to an areal capacity of $8.8 \frac{\text{Gbps}}{\text{Km}^2}$ [12]. Each eNodeB includes a backhaul transport node equipped with different 500 Mb/s PTP interfaces, connected to form a plain grid. These links include 4 ms of processing and propagation delay, and are represented in the Figure 1 as wireless backhaul links. Afterwards, the mesh is connected to the LTE EPC through three PTP wired links, with 1 Gb/s of available bandwidth and 0.5 ms of propagation delay.

The connection between the EPC and the Internet is modeled through another PTP wired link (characterized by a bandwidth of 10 Gb/s and a propagation delay of 5 ms). The queue sizes of the network are dimensioned to be compliant with its bandwidth-delay product. Regarding the RAN, we used European frequencies and an Okumura-Hata propagation model: the LTE connection between UE and the eNodeB is modeled inside an urban environment of a medium city.

The plain grid topology has been chosen to ease the channel assignment problem, but the choice does not invalidate the generality of the obtained results. Moreover, it has been

¹<https://www.nsnam.org>

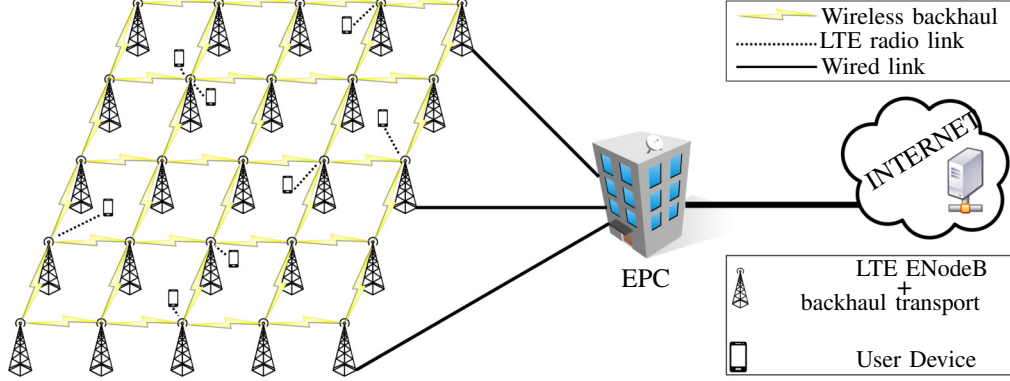


Fig. 1: Reference mesh SC topology (UE number can vary).

chosen to show the interaction of TCP and routing protocols in a multi-path scenario. For this reason, we have also abstracted the MAC layer issues (such as collisions or channel reservation) with an increased processing delay. For instance, mmWave technology is expected to provide a good quality and high bandwidth point to point link, comparable with the model used to define the mesh.

To evaluate the throughput and latency performance, we have modeled a TCP-based 10 MB file transfer from the remote node to the UEs, that represents the download of a data chunk from a remote streaming server. In all the evaluations, there is always one transfer per UE, and consequently the number of TCP flows is equivalent to the number of connected devices. For modeling the existing LTE traffic, we used a constant-rate model, between random SC source-destination pairs. In the description of the experiment, we refer to the aggregated UDP traffic in the entire network, instead of characterizing throughput performance of every flow. We will not consider the latency of UDP flows, because an in-depth analysis of UDP traffic was already conducted in [8]. We want to focus on TCP, and for it we analyzed experienced RTT and download finish time as a representative measure of end-to-end latency and throughput, respectively. Combining download finish time with the RTT values gives the better overall view of the network performance experienced with the combinations of L3/L4 protocols.

III. PERFORMANCE EVALUATION

A. $cWnd$ evolution in a single node

To explain the problems faced by TCP, we show in Figure 2 the congestion window ($cWnd$) evolution for a remote server that is delivering a 10 MB file to one UE, with concurrent 1200 Mb/s of UDP traffic in the mesh and 24 other UEs downloading. Due to the intrinsic properties of TCP, there are two events after which the $cWnd$ (and, in consequence, the transmission rate) is reduced: (i) packet loss and (ii) packet reordering at the destination.

With regard to OLSR, thanks to the knowledge of the drop-tail queue maximum size (340 packets), the backhaul load (1200 Mb/s), and the minimum number of hops inside the mesh required for the flow to reach the EPC (5), the consequence is that all the congestion events that reduce the

$cWnd$ (the maximum peak is 1117 packets) are due to packet losses. In fact, in a path composed by 5 hops, the total amount of available queue storage is 1700 packets, and with some space used for the UDP load a value of $cWnd$ near 1000 packets is perfectly understandable, before going out of capacity.

On the other hand, backpressure *per-packet* experiences the lowest congestion window (27 packets in the maximum peak), far away from the 1000 packets value experienced by OLSR. In this context, that $cWnd$ limit is clearly due to the high degree of reordering introduced by this strategy, giving the fact that the $cWnd$ value does not approach even the minimum-path queue storage limit. As a consequence, we can see that the transmission needs a longer time with respect to OLSR (even considering the path construction time, represented by the first 12 seconds where the report for OLSR is empty) to complete.

In contrast, backpressure *per-flow* experiences the highest $cWnd$ value (the peak is 2304 packets, that indicates that for this particular flow the number of hops is not less than 7) and faces fewer losses than OLSR, and no packet reordering, thus allowing the TCP transfer to finish in a very short amount of time, compared to the other variants.

The aforementioned results for a single UE can be generalized for all UEs in the considered scenario, and this (simple, for sake of clarity) report is the foundation of the result explanation presented in the following, where we analyze the download finish time and the RTT of the file transfers under various conditions.

B. Increasing Local Backhaul Traffic

In this section, we evaluate the routing strategies by analyzing the TCP performance of 25 LTE devices concurrently downloading a 10 MB file from the Internet. We present different set of results, gathered by keeping fixed the number of LTE devices but increasing UDP workload in the backhaul network. In Figure 3 and in Figure 4 we can see the download time values (i.e. the time that the TCP implementation takes to deliver completely and in-order all bytes to the application) and their average RTT, respectively. The values are represented in candlesticks, where the box stretches from the 20th to the 80th percentiles and the whiskers represent the maximum and minimum values. In Figure 3, we also plot the percentage of

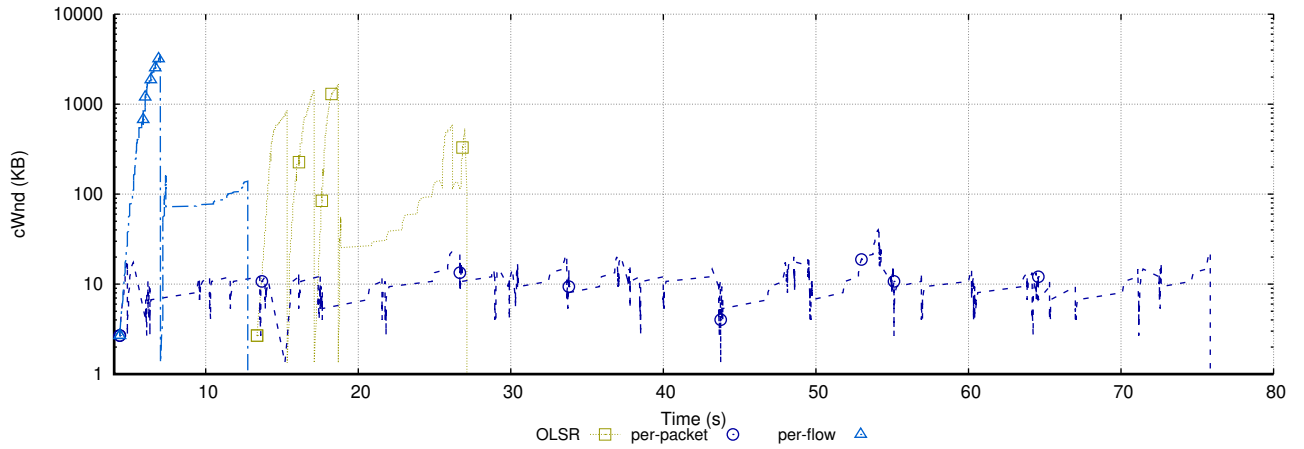


Fig. 2: TCP Congestion Window evolution with 1200 Mb/s of UDP and 24 additional TCP flows.

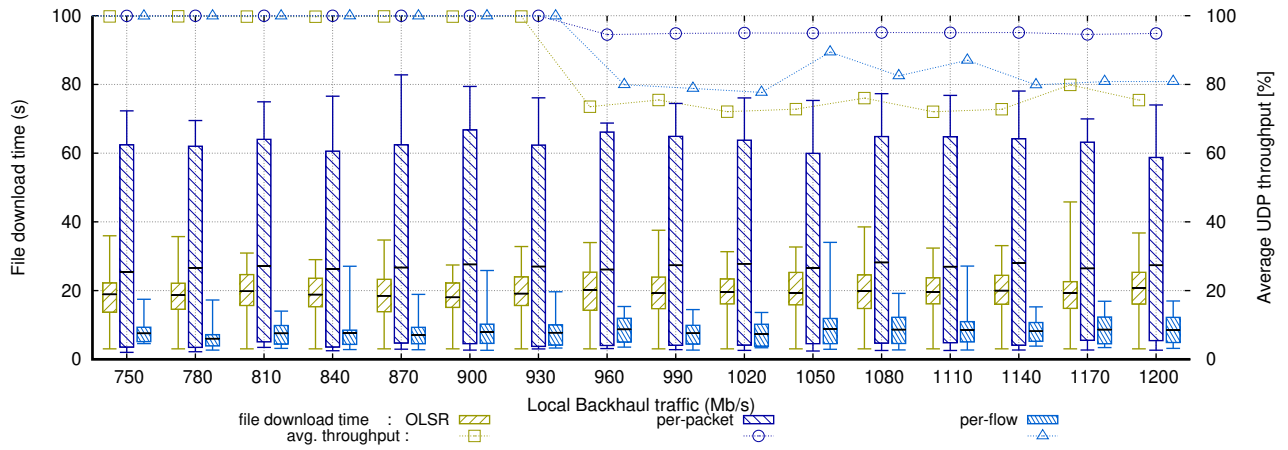


Fig. 3: File download time (TCP) and UDP average backhaul throughput, presented while increasing UDP backhaul traffic.

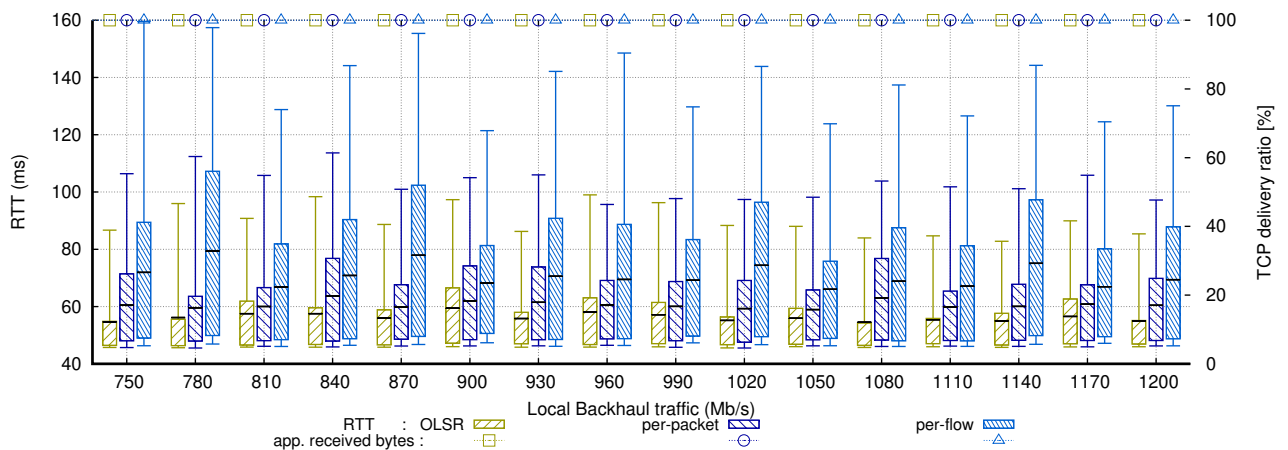


Fig. 4: TCP RTT and delivery ratio while increasing UDP backhaul traffic.

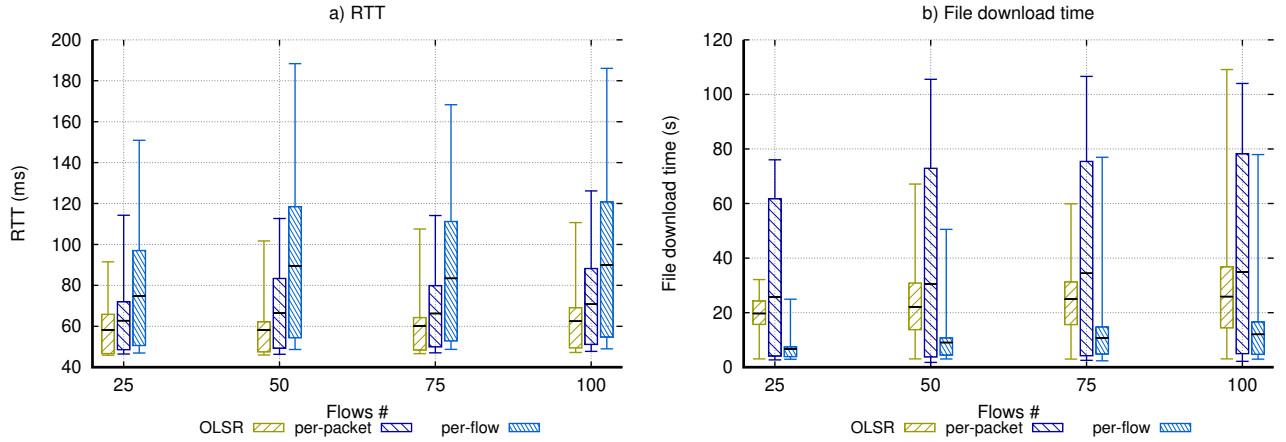


Fig. 5: File download time and RTT of up to 100 TCP file transfers with 600 Mb/s of UDP traffic.

achieved UDP traffic. For instance, a value of 80% means that, in average, only 80% of nominal UDP traffic has reached the destination. For sake of completeness, we added in Figure 4 the percentage of bytes correctly delivered to the applications (TCP delivery ratio). Since all values are fixed to 100%, it means that (for every flow) the TCP protocol has been able to entirely deliver, using also retransmissions, the 10 MB file.

OLSR. It represents the worst-case in terms of per-SC resource usage. In fact, the protocol specifications lead to routing the data always on the shortest path, irrespective of the congestion level. We observe in Figure 3 that when the UDP backhaul traffic is not sustainable anymore (i.e., from 960 Mb/s onward), OLSR experiences the lowest throughput values. A look to reference TCP traffic shows that OLSR experiences the lowest degree of RTT variability compared to *per-packet* and *per-flow* but higher download times than *per-flow*. The reason for this is the congestion unawareness of OLSR: as soon as the per-interface queue reaches its limit, packets begin to be dropped causing the TCP congestion control algorithm to limit TCP rate. Therefore, under high loads, statically choosing paths independently of the congestion level leads to inefficient resource usage (lower UDP throughput, increased packet drops) and constant delays, which are a function of the maximum queue size and the number of hops traversed. This is why the average file download time (black horizontal line in the candlesticks) does not change with the backhaul traffic load in Figure 3.

BP-MR per-packet. Taking backpressure routing decisions on a *per-packet* basis attains the best results for aggregated UDP throughput but high inefficiencies to complete all the TCP file transfers, regardless of the UDP workload. For instance, for an UDP workload of 1080 Mb/s *per-packet* variant attains a 90% of injected UDP traffic, while TCP transfers average finish time is 30 seconds. The finish time trend has a nearly constant average through all the UDP workload configurations but it presents a high degree of variability, represented by the bigger candlesticks, indicating a poor fairness between different TCP flows. There is more variability, with respect to OLSR, also in the RTT distribution. Average values are however close to the ones achieved with

OLSR, but we need to consider that, giving the high dynamism of the *per-packet* routing strategy, segments of the same TCP flow can in fact experience a different number of hops, leading to a reordering problem at the receiver, that is the main limiting factor for the original *BP-MR* strategy and many other backpressure strategies. To recap, *per-packet* routing strategy is able to utilize resources more efficiently in presence of congestion (higher backhaul UDP throughput) at the cost of penalizing TCP flows, due to serious reordering issues.

BP-MR per-flow. Taking decisions on a *per-flow* basis attains the best trade-off among UDP and TCP performance, showing significant TCP improvements compared to OLSR and *per-packet* flavors, experiencing also close-to the best results in terms of UDP backhaul traffic. This is because *per-flow* computes a new load balancing decision for every new flow injected in the network and sticks with this decision. The former feature allows calculating the less loaded fixed path, hence making the most out of the wireless backhaul resources, whereas the latter feature minimizes packet reordering. With less reordering and less packet drops, the finish time is in general really low and from Figure 3 it is clearly visible that it outperforms both OLSR and *per-packet*, even considering flow fairness (the candlesticks are condensed in a very tight space). For what regards RTT distribution, it is higher than the other two flavors, but we need to take into account the fact that lost, retransmitted, and out-of-order segments do not contribute to the RTT calculation in the senders, and thus they are not represented in that graph. With *per-flow* strategy, we can infer from the finish time performance a really low number of lost and out-of-order segments, and so a higher number of RTT samples. At the end, *per-flow* strategy is able to circumvent congestion (and deliver more backhaul traffic than OLSR) without penalizing TCP performance, as it can be expected from a strategy that blends the best from OLSR (fixed paths) and *per-packet* (dynamic congestion avoidance) routing strategy.

To achieve the same performance, in these loaded conditions with a static protocol such as OLSR, a network architect should have increased the allocated network resources. Using *per-flow* routing strategy however, the architect is able to

exploit resources even when the backhaul traffic is approaching the network saturation; in the following, we will analyze the protocol response with an increased number of connected devices, to see the scalability degree under such figure of merit.

C. Increasing LTE Devices Connected to the Internet

In the next analysis, we increase the number of connected device by steps of 25 UEs, presenting the file download time and RTT values for each step. The aggregate amount of backhaul traffic is fixed at 600 Mb/s, and evaluations are made over 25, 50, 75, and 100 connected devices, each one concurrently downloading a 10 MB file from the Internet. Results are presented under the form of candlestick, with the same meaning as in the previous section.

The RTT and finish time performance are reported in Figure 5. Finish time with OLSR is influenced by the traffic growth and shows lowest RTT values (but higher packet drops) compared to backpressure variants, showing its inability to handle congestion which leads to packets drops, generating more retransmissions. With *per-packet* strategy, TCP flows experience a higher finish time than OLSR or *per-flow* and a variable RTT distribution. The main reason for this finding is the *per-packet* high-degree of packet reordering, that limits the TCP performance. In contrast, we found that the *per-flow* backpressure strategy provides the lowest TCP download transfer times, at a cost of higher experienced RTT values as one should expect with the increase of traffic load and so congestion conditions. This indicates the compatibility of *per-flow* variant with the calculation of TCP congestion window, through choosing the less congested paths (and so limiting the packet losses) and eliminating the packet reordering issues typical of *per-packet*. Thus, results reveal that reducing the granularity to a *per-flow* decision allows the system to better handle a growing amount of TCP traffic, regardless of the existing backhaul load.

IV. CONCLUSION

In this paper we assumed an LTE scenario composed by a dense deployment of micro eNodeB, backhauled by point-to-point wireless interfaces to create a Wireless Mesh Network. Then, we discussed the response of three routing strategies, in terms of TCP and UDP performance, over an increasing load. We investigated OLSR, and then BP-MR *per-packet* and *per-flow* strategies, two variants of the backpressure-based BP-MR protocol. We compared their UDP throughput, TCP transfer times, and TCP RTTs.

Through ns-3 simulations we showed that is possible to efficiently utilize all the wireless mesh backhaul resources, evenly distributing the load in both UDP and TCP cases. The *per-packet* BP-MR variant enhances UDP performance by circumventing congestion, but TCP traffic suffers from reordering. The *per-flow* BP-MR variant minimize retransmissions by limiting the excessive reordering, and outperforms OLSR performance for both UDP backhaul traffic and TCP traffic.

We believe the presented results shed some light on the problem of conducting load balancing for dense small cell

deployments in the presence of increasing TCP/UDP traffic. In this sense, a scalable backpressure-based scheme with the proper degree of granularity when taking routing decisions might be a good routing protocol candidate without necessarily resorting to an unnecessary over-provisioning of the wireless backhaul resources.

As a future work, we plan to switch our scenario to the new 5G NR protocol stack, using [13].

V. ACKNOWLEDGMENT

This work was partially funded by Spanish MINECO grant TEC2017-88373-R (5G-REFINE) and Generalitat de Catalunya grant 2017 SGR 1195. While doing this work, Natale Patriciello was under the affiliation of University of Modena and Reggio Emilia, and José Núñez-Martínez under the affiliation of CTTC.

REFERENCES

- [1] W. Webb, *Wireless Communications: The Future*. John Wiley & Sons, 2007.
- [2] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: challenges and research advances," *IEEE Network*, vol. 28, no. 6, pp. 6–11, Nov 2014.
- [3] M. Casoni, C. A. Grazia, M. Klapez, N. Patriciello, A. Amditis, and E. Sdangos, "Integration of satellite and LTE for disaster recovery," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 47–53, 2015.
- [4] S. K. Singh, T. Das, and A. Jukan, "A Survey on Internet Multipath Routing and Provisioning," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2157–2175, Fourthquarter 2015.
- [5] Z. Jiao, B. Zhang, C. Li, and H. T. Mouftah, "Backpressure-based routing and scheduling protocols for wireless multihop networks: A survey," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 102–110, February 2016.
- [6] B. Radunović, C. Gkantsidis, D. Gunawardena, and P. Key, "Horizon: Balancing TCP over Multiple Paths in Wireless Mesh Network," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, ser. MobiCom '08. New York, NY, USA: ACM, 2008, pp. 247–258. [Online]. Available: <http://doi.acm.org/10.1145/1409944.1409973>
- [7] H. Seferoglu and E. Modiano, "TCP-aware backpressure routing and scheduling," in *Information Theory and Applications Workshop (ITA)*, 2014. IEEE, 2014, pp. 1–9.
- [8] J. Baranda, J. Núñez-Martínez, and J. Manges-Bafalluy, "BP-MR: Backpressure Routing for the Heterogeneous Multi-radio Backhaul of Small Cells," in *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*, Oct 2015, pp. 48–55.
- [9] N. Patriciello, C. Grazia, J. Núñez-Martínez, J. Baranda, J. Manges-Bafalluy, and M. Casoni, "Performance Evaluation of Backpressure Routing in Integrated Satellite-terrestrial Backhaul for PPDR Networks," in *Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016 IEEE 12th International Conference on, Oct 2016, pp. 110–117.
- [10] N. Patriciello, J. Núñez-Martínez, J. Baranda, M. Casoni, and J. Manges-Bafalluy, "TCP performance evaluation over backpressure-based routing strategies for wireless mesh backhaul in LTE networks," *Ad Hoc Networks*, vol. 60, pp. 40 – 51, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870517300458>
- [11] J. Baranda, J. Núñez-Martínez, J. Manges-Bafalluy, and N. Patriciello, "Evaluation of hybrid terrestrial-satellite suburban wireless mesh back-hauls for LTE networks," in *2017 European Conference on Networks and Communications (EuCNC)*, June 2017, pp. 1–6.
- [12] "Small Cell millimeter wave mesh backhaul," Whitepaper, INTERDIGITAL, December 2013. [Online]. Available: http://www.interdigital.com/research_papers/2013_01_25_small_cell_millimeter_wave_mesh_backhaul
- [13] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An E2E simulator for 5G NR networks," *Simulation Modelling Practice and Theory*, vol. 96, p. 101933, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1569190X19300589>