Experimental Evaluation of Self-organized Backpressure Routing in a Wireless Mesh Backhaul of Small Cells

José Núñez-Martínez, Jorge Baranda, Josep Mangues-Bafalluy

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
Av. Carl Friedrich Gauss, 7
08860 Castelldefels (Barcelona), Spain
e-mail: [jose.nunez, jorge.baranda, josep.mangues]@cttc.cat

Abstract

Small cells (SC) are low-power base stations designed to cope with the anticipated huge traffic growth of mobile communications. These increasing capacity requirements require the corresponding backhaul capacity to transport traffic from/to the core network. Since it is unlikely that fiber reaches every SC, a wireless mesh backhaul amongst SCs is expected to become popular. These low-cost deployments require to balance resource consumption amongst SCs, however, current routing protocols were not designed to fulfill this requirement. To tackle this challenge, we presented and developed with ns-3 a self-organized backpressure routing protocol (BP), designed to make the most out of the backhaul resources. This paper provides the evaluation of BP exploiting built in ns-3 emulation features to allow rapid prototyping under real-world conditions and through controlled ns-3 simulations. Through a novel evaluation methodology based on ns-3 emulation, we evaluate BP in a 12 SC indoor wireless mesh backhaul testbed under different wireless link rates and topologies, showing Packet Delivery Ratio (PDR) gains of up to 50% with respect to shortest path (SP). Through simulations, we show BP scalability properties with both the size of the backhaul and the number of backhaul radios per SC. Results in single- and multi-radio deployments show TCP traffic gains with BP of up to 79% and 95%

[☆] Fully documented templates are available in the elsarticle package on CTAN.
[1] Since 1880.

compared to SP in terms of throughput and latency, respectively.

Keywords: wireless mesh network, backpressure, experimentation, ns-3, performance evaluation, small cell

1. Introduction

The ever increasing demand for wireless data services has given a starring role to dense deployments of low-power base stations referred to as small cell (SCs), as increasing frequency re-use by reducing cell size has historically been the most effective and simple way to increase capacity. Such densification entails challenges at the Transport Network Layer (TNL), since hard-wired backhaul deployments of SCs prove to be cost-prohibitive and inflexible. The main challenge is to provide cost-effective and dynamic TNL solutions for dense and semi-planned SC deployments. An approach to decrease costs and augment the dynamicity at the TNL is the creation of a wireless mesh network [1] amongst SCs to carry control and data plane traffic to/from the core network [2].

A wireless mesh backhaul requires of practical routing schemes realizing an even resource consumption to ease the mentioned capacity crunch. An identified requirement when steering traffic is to dynamically grow or shrink the pool of SC resources according to network conditions, thus, exploiting the capacity offered by the wireless mesh backhaul. However, the IEEE 802.11s [3] mesh standard specifies Hybrid Wireless Mesh Protocol (HWMP) (see section 2), a tree-based protocol oriented to provide network connectivity rather than the exploitation of resources. Aiming for capacity, the original centralized backpressure algorithm [4] proved to be throughput optimal in theory. However, its implementation under real-world conditions showed scalability problems with the number of flows and forces the wireless network to operate on a Time Division Multiple Access (TDMA) MAC [5]. To counteract these issues, in [6, 7], we presented and evaluated through ns-3 simulations [8] a self-organized backpressure routing protocol (BP) for the TNL that dynamically grows or shrinks SC resources according to network conditions (see section 2). Specifically [6] focuses on multi-gateway SC deployments, whereas in [7] the focus is on sparse wireless mesh backhaul deployments. Additionally, in [9] we presented the integration details and preliminary experimental results of our scheme using ns-3 emulation features (see section 2). However, the evaluation in [6] was merely based on simulations in a single-radio single-channel backhaul deployments, and the experimental evaluation in [9] was scarce.

The contribution of this paper is two-fold. First, we detail the configuration of the experimental platform (see section 3) using ns-3 emulation composed by 12 SCs endowed with 3G for the Radio Access Network (RAN), and an additional WiFi card to form a WiFi-based mesh backhaul amongst them (see Figure 1), thus, forming an all-wireless Network of SCs (NoS). Prior to the evaluation of BP, we characterize the NoS testbed by analyzing the quality of the backhaul WiFi links composing the prototype. Additionally, we revealed that the ns-3 emulation rolled out in each SC does not introduce performance degradation in terms of throughput by showing that it can saturate the WiFi cards underneath (see section 4).

Second, we tested BP in a wide variety of realistic wireless mesh backhaul conditions (see section 5). In particular, using ns-3 emulation we demonstrate the operation of BP under different wireless link configurations (wireless link rate, ambient noise reduction techniques) and showed how by switching on and off SCs in the testbed, BP adapts to dynamic wireless mesh backhaul deployments. We observed gains of around 50% in terms of PDR with regards to a shortest path (SP) routing policy. Additionally, we demonstrated through ns-3 simulations the scalability of BP with both the number of SCs, and the number of backhaul radios per SC forming a multi-radio multi-channel wireless mesh network. We provide first simulation results of BP with TCP traffic showing promising advantages. In particular, BP showed significant gains over SP of up to 79% and 95% in terms of throughput and latency, respectively, with respect to SP routing.

We conclude the paper and provide future possibilities with section 6.

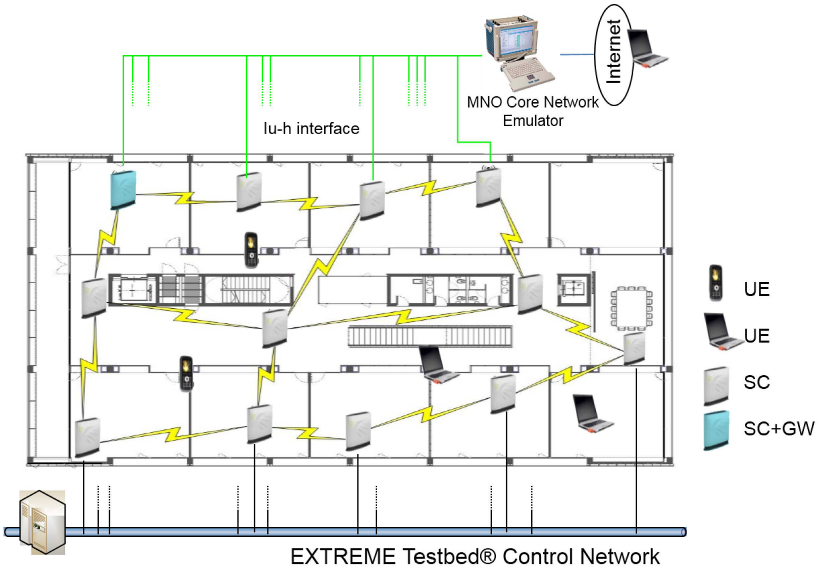


Figure 1: Architecture of the NoS testbed

2. Background and Related Work

2.1. Emulation

Fall [10] classifies emulation into two types. In environment emulation, an implementation environment is built so that real protocol implementations can be executed in a simulator, whilst in network emulation simulated components interact with real world implementations. As for the former, Network Simulation Cradle (NSC) [11] is a pioneer of introducing real stacks into network simulators. Recently, in [12] a framework for executing Linux kernel code in the ns-3 simulator was presented. As for the later, which is the focus in this paper, the ns-3 simulator [8] provides emulation capabilities for evaluating network protocols in real testbeds [13]. Figure 2(a) illustrates the main architecture of a node (in this case a SC) running ns-3 emulation. The ns-3 system allows executing ns-3 IP stacks over real physical devices. In particular, the EmuNetDevice class provides the interface to send/receive packets to/from the real physical device underneath through a raw socket. The EmuNetDevice lying on top of the real physical device uses MAC spoofing to avoid conflicts between the guest ns-3 IP stack running in user space and the native IP stack. Therefore, the EmuNetDevice can send over the physical device packets that have assigned MAC addresses different from those identifying the real device. In this way, packets received by the EmuNetDevice are sent to the ns-3 IP stack whenever the MAC destination address corresponds to the spoofed ns-3 MAC address.

2.2. Routing for WMNs

Transport schemes for the mobile backhaul, such as [14], assume that they will be rolled out with a highly reliable infrastructure (e.g., fiber) underneath. This is not the case of low-cost wireless mesh networks [1] that are unstable, unreliable, and have scarce resources. With such constraints, finding routing paths is a fundamental problem for a WMN. The IEEE 802.11s [3] mesh standard specifies the Hybrid Wireless Mesh Protocol (HWMP), a tree-based protocol oriented to provide network connectivity rather than the required capacity. The comprehensive survey in [15] classifies routing protocols for WMNs into two main groups: multi-radio and opportunistic routing protocols. Although multi-radio routing protocols aim to exploit the added capacity brought by multi-radio multi-channel WMNs, they rely on the establishment of fixed single/multiple end-to-end paths assuming a high network stability regarding path qualities. Such an assumption may be inappropriate under unreliable and low-cost WMN deployments. On the other hand, opportunistic routing protocols take routing decisions on a hop-by-hop basis. Their design is characterized by deferring the next-hop decision after the packet has been transmitted. Although these protocols aim to address the inherent unreliability of WMNs, they incur into an extra coordination overhead amongst the possible set of forwarders, and potential generation of packet duplicates .

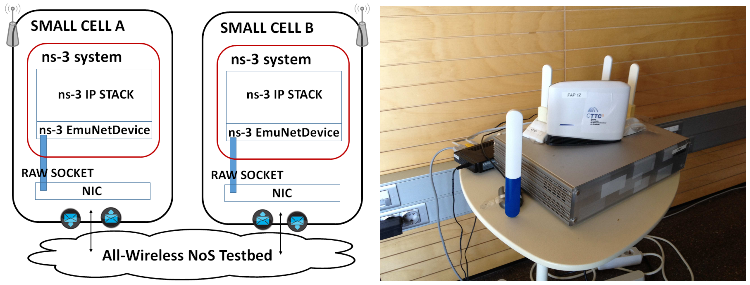
In terms of methodologies, the evaluation of routing schemes under real world conditions is of primal importance. Researchers have recently built a number of WMN testbeds to evaluate the sources of performance degradation of routing protocols in real environments. Previous work [16] characterized the sources of noise in an indoor deployment using 802.11b/g evaluating different routing protocols. In outdoor deployments [17], sources of noise are more frequent due to the interference of external interferences of non WiFi devices. Our work in terms of testbed deployment in this paper is similar to that in [18] in the sense that we also deploy an indoor WMN using 802.11a technology as backhaul.

2.3. Self-organized BP for the Wireless Mesh Backhaul

Unlike routing families previously described, our scheme [6], referred to as self-organized Backpressure Routing (BP), is based on a decentralized flavor of the original centralized backpressure algorithm [4] that promises throughput optimality assisted by scalable geographic routing [19]. As observed in [5] the centralized backpressure algorithm limits its implementation to small centralized wireless networks. First, it maintains centralized routing tables, and a queue per every flow in every node. In addition, it does not provide any guarantees in terms of latency and it forces the wireless network to operate on a Time Division Multiple Access (TDMA) MAC, as it is originally proposed in theory. In contrast, as detailed in [6], BP relies on a single queue backlog per node and geolocation information to take per-packet routing decisions that dynamically grow or shrink path utilization according to traffic conditions. For each packet being routed in SC_i, wireless link weights with all potential SC neighbors j according to $w_{ij} = \Delta Q_{ij} - V \times cost$. The wireless link selected for transmission is the one that maximizes the computation of w_{ij}. Note also that the protocol is agnostic to the wireless technology underneath. All the necessary information to compute weights (queue backlog and geolocation) is exchanged amongst SCs by means of periodic emission of HELLO broadcast messages. In essence, the protocol aims to transmit packets to the less loaded SC, however, such a decision may incur a cost based on the geographic progress towards the intended destination and the importance assigned to this cost function denoted by the value assigned to the V parameter.

The importance of this cost function with respect to minimization of queue backlog differentials is determined by the value of the V parameter. BP includes a self-organized algorithm to calculate the V value on a per-packet basis. This algorithm aims to find the proper trade-off between balancing resource consumption (based on the minimization of queue backlog differentials) and the cost function (based on geographic progress to the destination). As detailed in [6], for each packet being routed in SC_i, V_i is calculated as $Q_{max} - max\{Q_j(t)\}$, where j includes the queue backlogs of the SCs in a 1-hop distance plus the queue backlog of SC_i. In this way, to mitigate congestion SC_i decreases V_i, hence, decreasing the importance associated to the cost function, whereas under no congestion SC_i would increase V_i, hence, increasing the importance of the geographic cost function.

We use geographic greedy routing as cost function to enable the scalability BP needs for its practical implementation. The resulting scalable BP scheme showed gains in terms of adaptability to wireless mesh backhaul dynamics in the preliminary work in [7] and in uplink traffic communications in [6] showing not only promising gains in throughput but also in latency. Results to support these features are, hitherto, obtained through ns-3 [8] simulations, which are insufficient by themselves.



(a) NS-3 Emulator Architecture

(b) 3G+WiFi Small Cell

Figure 2: Main entities of the NoS Testbed.

3. Testbed Configuration

This section first describes the indoor all-wireless Network of SCs (NoS) [2] developed in the first floor of the CTTC building over an approximate indoor area of 1200 square meters (see Figure 3). In second subsection, we describe how we used the ns-3 emulation to run BP in the testbed. Third subsection provides configuration details of the WiFi mesh backhaul.

3.1. Testbed Description

Figure 1 illustrates the main entities of the all-wireless NoS testbed. A description of the main hardware entities enabling the indoor all-wireless NoS proof-of-concept follows:

- **SC**: The SCs are based on two components. One component is the RAN 3G device implementing an Iuh [20] interface to enable interoperability with the Iuh interface implemented by the emulated core network. The 3G RAN device is connected via Ethernet to a WiFi mesh router (see Figure 2(b)). Wifi mesh routers, based on a mini-ITX board that can endow up to four CM9 WiFi cards (802.11abg), merely entail TNL functionalities. Thus the mesh routers build the TNL, whereas the 3G SC part includes both Mobile Network Layer (MNL) and TNL functionalities.

- **Core Network**: The Core Network Emulator in Figure 1 implements, besides other MNL functionalities, the Iuh interface to which the 3G RAN part of the SC can connect through a WiFi mesh backhaul.

- **TNL GW**: One SCs acts also as a TNL GW [6]. This component is in charge of pulling/injecting packets from/to the wireless mesh backhaul.

- **User Equipment**: The User Equipment (UE) are based on laptops equipped with an UMTS PCMCIA card attached to the 3G SCs.

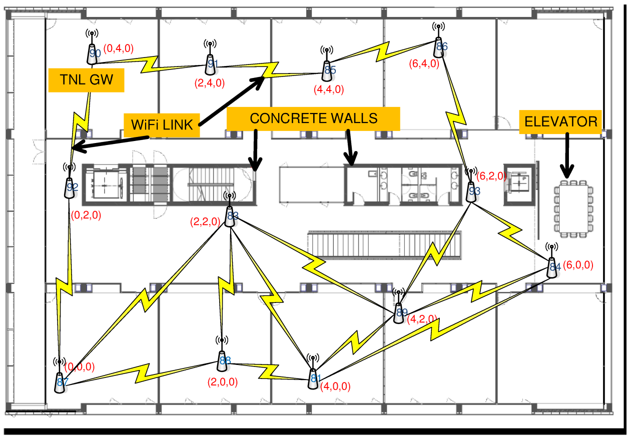


Figure 3: WiFi-based Mesh Backhaul Testbed

3.2. Configuration of BP using ns-3 emulation

As previously noted, each SC is directly connected to a WiFi router through Ethernet, hence forming a 3G SC with WiFi as backhaul interface. On top of the WiFi backhaul interface, we associate a single ns-3 IP stack including the implementation of BP at the ns-3 routing layer. Through the ns-3 EmuNetDevice (see Figure 2(a)), we associate to every ns-3 IP stack a different MAC address from that identifying the WiFi card underneath. In this way, ns-3 packets can be send to the real testbed, and real packets can be also intercepted by the ns-3 stack. With this particular method any network protocol can be implemented in ns-3 without modifications, as long as it works above layer 2. This method allowed to evaluate BP in a real testbed re-using the ns-3 code developed in our previous work [6, 7] in a real SC. Note also that BP needs geolocation information in order to compute the weights of every link. Geographic coordinates are statically assigned to each ns-3 IP stack so as to form a 4x3 grid (see Figure 3) with a step of size 2.

One lesson learned from using the ns-3 emulation framework on top of a WiFi card is the importance in the specification of the MAC address associated to the ns-3 EmuNetDevice class. In the MadWiFi driver, which is used in our platform to manage WiFi cards, the Basic Service Set Identifier (BSSID) mask specifies the common bits that a MAC address must match in order to process a receiving packet. Therefore, the spoofed MAC address associated to the EmuNetDevice object in the ns-3 emulator must comply the restrictions imposed by the BSSID mask specified in the MadWiFi driver, whereas an Ethernet card does not pose such restrictions. Otherwise, packets received by the MadWiFi driver with a MAC address not following BSSID mask restrictions are discarded by the WiFi card. Thus, they cannot be captured by the packet (raw) socket in the ns-3 class defined by the EmuNetDevice. The mask allows specifying a different MAC address to the ns-3 emulated device by changing the most significant bits of the usual MAC identifier.

3.3. Configuration of the WiFi Mesh Backhaul

Every WiFi mesh router runs ns-3 emulation on top of a virtual interface (in our Linux setup labeled as ath0 interface) associated to one physical WiFi card, labeled as Wifi0 interface. A second WiFi virtual interface (in our specific Linux setup labeled as ath1 interface) associated to the same physical WiFi card (e.g., Wifi0) is configured in each node in monitor mode for tracing purposes.

The 12 SCs are deployed in a confined space of around 1200 square meters. In addition to building obstacles (e.g., brick walls, plants, elevator), there are mobile obstacles during working hours (e.g., workers moving inside the building). Additionally, the 12 SCs have assigned the same channel in a very confined space, which allows most nodes to be in carrier sense range of each other, hence minimizing interference due to hidden nodes. In addition to building obstacles (e.g., brick walls, plants, elevator), there are mobile obstacles during working hours (e.g., workers moving inside the building). Additionally, the backhaul radio of the 12 SCs have assigned the same WiFi channel in a very confined space, which allows most nodes to be in carrier sense range of each other.

The layer 3 (L3) topology is determined by the set of nodes receiving HELLO messages. After an initial set of experiments, we observed that, in a so confined space, HELLO broadcast messages sent by every node are received by almost all the nodes in the testbed. This is because HELLO broadcast messages are usually transmitted at the minimum data rate allowed by the WiFi card, leading to a high decoding probability at all nodes in such a confined space. To establish L3 multihop topologies, we extend the driver of the WiFi cards so that one can increase the physical rate at which broadcast HELLO messages can be transmitted. After a set of empirical tests, we found that increasing the physical data rate of broadcast packets to 54Mbps is a conservative and simple way of detecting links with high quality. Note that an accurate measurement of wireless link quality can be much more complex. In particular, the selection of the broadcast rate of the probing packets and the measurement method (active or passive) are some of the key parameters to take into account [15]. In Figure 3, we show the resulting L3 connectivity patterns when sending HELLO broadcast messages at a physical rate of 54Mbps. We observed that every SC has at least two direct neighbors and no more than four.

4. Testbed Characterization

Prior to evaluate BP in the testbed, we need to characterize the particularities of our designed prototype, such as, the quality of the WiFi links and the performance provided by the ns-3 emulator.

4.1. WiFi Link Characterization

We use HELLO messages generated by BP, which is rolled out in each SCs. We measure the quality of WiFi backhaul links by calculating the PDR of HELLO messages sent at a fixed rate of 54Mbps every 100ms between each pair of SCs. HELLO messages determine which SCs are within direct communication (i.e., the set of 1-hop neighbor SCs at which one SC can directly transmit a data packet).

Usually, HELLO packets to probe the quality of WiFi links are transmitted with fixed HELLO rates of 6Mbps, while the the actual data rate of packets can be up to 54Mbps. HELLO messages are sent at the maximum rate (i.e., 54Mbps)

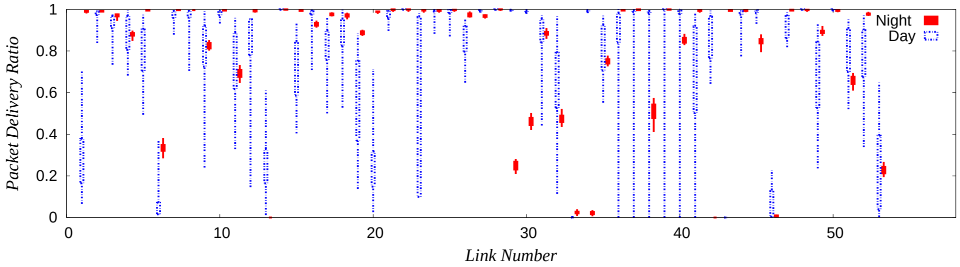


Figure 4: Wireless Link Quality: Day (i.e., Working) Hours vs. Night Hours

not only to increase the number of hops in a confined space, as described in previous section, but also to maximize the rate at which SCs can successfully decode received data packets. In this way, the HELLO probing data rate is equivalent to the maximum rate at which data packets can be decoded.

The set of experiments that characterize our WiFi mesh testbed evaluate the PDR of WiFi links in two different time frames. The first time frame starts at working hours (i.e., 11AM), whilst the second time frame starts at night hours (i.e., at 10PM without workers in the building). For each time frame under evaluation, the testbed calculates the average PDR of HELLO during 100 seconds. The testbed repeats this experiment 15 times, with a pause time of 100 seconds between every replication.

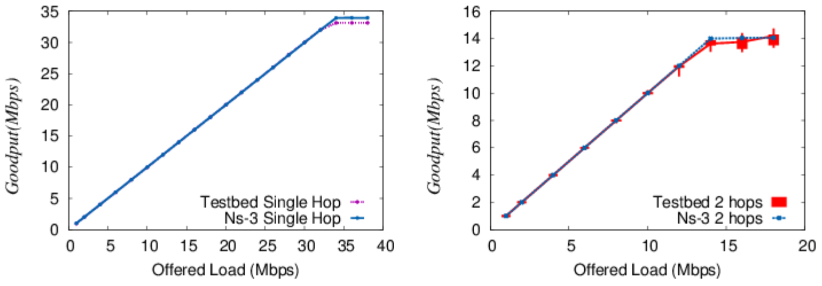
One observation characterizing WiFi mesh links configured in the 5Ghz band is the impact of the time of day in the wireless link quality. We observe that WiFi links during day hours are more unstable than during night hours. Interestingly, this fact was also revealed in [16] for WiFi mesh links configured in the 2.4Ghz band. The more variable environmental conditions during working hours can even help some packets reach more distant SCs through direct communication. Precisely, we observed 57 stable links during the night, whereas up to 69 lossy links showed a higher degree of dynamics during the day. In fact, during working hours just a 36% of all the detected links were present in all the 15 replications.

Figure 4 plots the subset of WiFi links with a PDR bigger than zero during the 15 replications of the experiment in at least one of the two times frames (i.e., day and night) under evaluation. Thus, we compare the PDR of those selected WiFi links in both time frames. During working hours the reported PDR of WiFi links shows a high degree of variability, whilst during night hours WiFi links show more stability as shown by the size boxplots in Figure 4. As expected from previous work [16, 18, 21], the environmental conditions can be considered constant during the 15 replications since during night hours there are no people in the building. Thus, we can conclude that during night hours, no matter the PDR shown by a WiFi link (i.e., good or bad), link quality shows a high degree of stability.

4.2. Emulation Characterization

The goal here is to assess the type of traffic patterns that the ns-3 emulator can handle in the testbed without introducing performance degradation issues that could bias the interpretation of evaluation results. Note that as recently showed in [13], ns-3 emulation may experience performance degradation problems under certain conditions. We run two different experiments 20 times with the routing protocol rolled out in the NoS testbed. The first experiment injects a one-hop UDP flow of maximum packet size, whereas the second injects a two-hop UDP flow of maximum packet size in the testbed using the ns-3 emulator. We also replicated the aforementioned scenario with the ns-3 simulator for comparison purposes.

As depicted by Figure 5(a) and Figure 5(b), goodput results with the testbed running BP in ns-3 emulation mode match those obtained with the ns-3 network simulator. For the two-hop case, as can be shown in Figure 5(b), though there is higher variability in the testbed (up to 1Mbps), goodput results are also similar to those obtained with the ns-3 network simulator. These tests revealed that in terms of packet generation, forwarding, and packet reception the ns-3 emulator does not introduce packet processing performance issues when handling UDP flows of maximum packet size.



(a) 1-hop case

(b) 2-hop case

Figure 5: Reference Scenario to evaluate the performance of the ns-3 emulation framework

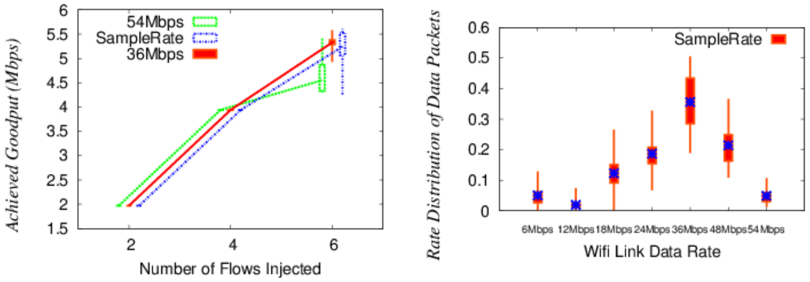
5. Performance Evaluation

We evaluated BP under a wide range of real and simulated conditions. In all the experiments, BP uses the variable-V algorithm presented in [6], which is in charge of determining the proper trade-off between backpressure (i.e., minimization of queue backlog differentials) and proximity to the destination. The data queue size limit denoted by Q_{max} in each SC is of 200 packets.

The first and second set of experiments are conducted in the NoS testbed. Their objectives are to show the impact of the WiFi link rate configuration in BP results and to illustrate the inherent properties of BP to adapt to wireless backhaul topology seamlessly. According to the results revealed in the previous section, note that a) the experiments in the NoS testbed have been executed during night hours and b) the traffic injected consists of unidirectional UDP traffic with maximum packet size. The third set of experiments analyzes through ns-3 simulations the interaction BP with an unmodified TCP stack and BP scalability with both the size of the backhaul and the number of backhaul radios per small cell.

5.1. Impact of WiFi Link Rate

The goal is assess the impact of the backhaul WiFi link rate. We inject 2,4, and 6 UDP traffic flows of 1Mbps originated from random SCs sources and headed towards the TNL GW (see Figure 3). For every WiFi link rate configuration under evaluation, we repeat the experiment 16 times with a duration of 80s. Other parameters of the WiFi cards such as the contention window and the transmission power (6dBm) in all the WiFi cards are static during the whole experiment. Note that for all the experiments the HELLO rate is fixed to 54Mbps to preserve the same multihop connectivity patterns in the wireless mesh backhaul.



(a) Achieved Goodput.

(b) Data Rate Distribution.

Figure 6: Impact of WiFi link rate in the measured Goodput with BP routing.

Figure 6(a) compares BP goodput results obtained with different WiFi link rate configurations in the testbed, namely a fixed rate of 36Mbps, 54Mbps, and the SampleRate [22] autorate algorithm. We observe that BP does not suffer from performance degradation with the injection of 2 or 4 flows regardless of the WiFi link rate configuration. However, there is a significant mismatch between the workload and the achieved goodput measured at the receiver for 6 UDP flows regardless the WiFi link rate configuration. This experiment reveals that due to both the collisions frequent in a single channel mesh network and poor channel quality, the attained goodput is lower at a fixed data rate of 54Mbps than that attained with a fixed data rate of 36Mbps. In fact, best goodput results with all the backhaul links configured to a data rate of 36Mbps. Figure 6(b) confirms that the predominant WiFi link rate attempted by the SampleRate autorate algorithm is that of 36Mbps. Note that the Sample rate intends to estimate the maximum link rate experiencing a high reception probability. We also see from Figure 6(a) that the SampleRate autorate algorithm experiences the higher variability due to the autorate configuration experienced by all WiFi cards.

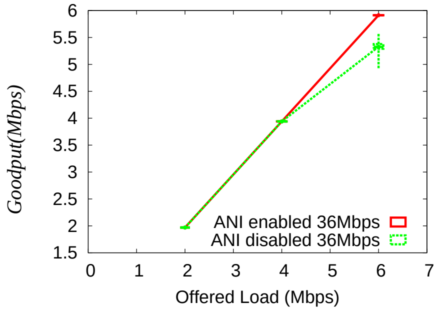


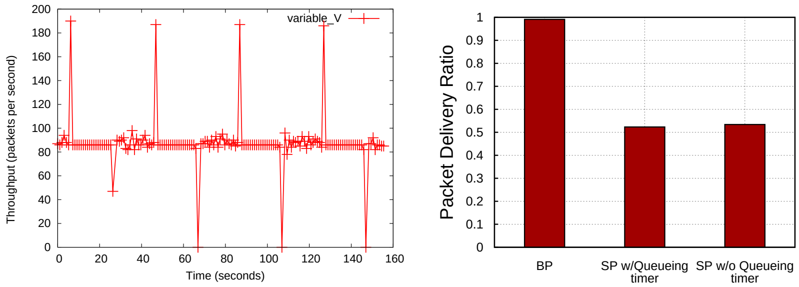
Figure 7: Impact of Ambient Noise Immunity (ANI) in Goodput.

We repeat the experiments conducted with WiFi link rates configured to 36Mbps, but activating the Ambient Noise Immunity algorithm (ANI), which is a MadWiFi proprietary algorithm disabled by default. The ANI algorithm manages the sensitivity of wireless cards to discard potential external noise, reducing the impact of the interfering sources in a given environment. The algorithm resides on the Hardware Abstraction Layer (HAL) of the MadWiFi driver. (Recall that the HAL acts like a wrapper with the card hardware registers and is distributed in its binary form.) The algorithm is recommended to be activated in an indoor environment such as ours. As Figure 7 depicts, BP attains an equivalent goodput for the case of 2, or 4 flows independently of the activation or deactivation of the ANI algorithm. However, when the load if of 6 UDP flows, the ANI algorithm achieves a higher goodput than that with the ANI algorithm disabled.

5.2. Adaptability to the Wireless Backhaul Topology

We demonstrate the adaptability of BP to sudden changes introduced in the wireless backhaul topology of the NoS testbed. Note that it may be interesting to switch off some SCs to bring energy cost savings or decrease interference. The experiment injects a CBR UDP traffic flow of 1Mbps (around 85 packets per second) from the SC in position (6,4,0) towards the TNL GW, that is, the SC in position (0,4,0) in Figure 3. The SC in position (0,4,0) has two SC neighbors available. One neighbor is closer to the destination and the other is farther from the destination than the source SC. According to Figure 3, choosing the closer neighbor supposes reaching the TNL GW in exactly three hops, whereas using the farther neighbor as next-hop can increases the path length to reach the TNL GW (i.e., 4, 5, or even 6 hops).

We periodically switch off and on the interfaces composing the WiFi backhaul to generate dynamic backhaul topologies in the testbed. To have awareness of the real WiFi card status (i.e., up or down) at the ns-3 stack running on top of the WiFi devices, we extended the implementation of the EmuNetDevice class of the ns-3 emulator, as the current implementation relies on a default up status for real cards underneath. During a execution time of 160s, every 20s we switch on and off the WiFi card SCs in positions (4,4,0) and (2,4,0) in Figure 3, hence disabling the shortest path to reach the TNL GW. We repeat the experiments for BP and single fixed path (SP) with and without queuing timers in their data queue structure. The SP protocol computes the fixed shortest path routes in terms of number hops at the beginning of the experiment.



(a) TNL GW Throughput with BP

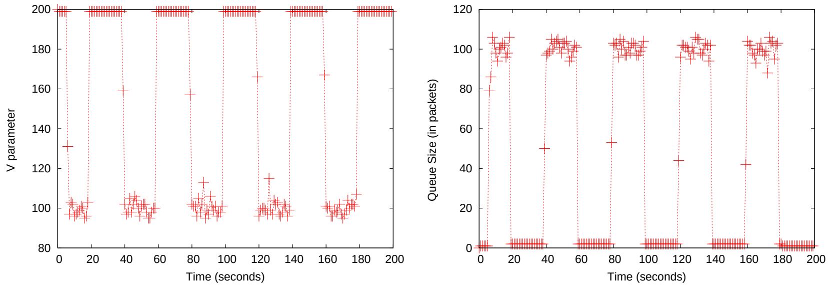
(b) PDR

Figure 8: Reaction under dynamism.

Figure 8(a) shows the projection over time of the attained throughput at the TNL GW when using the variable-V algorithm. From this figure and reported

PDR results illustrated in Figure 8(b), we can reveal some remarkable aspects.

First, BP serves practically all the traffic, with a PDR close to 1, using either the shortest path or multiple longer paths. While the number of packets received by the destination is constant in time when it uses the shortest path, the number of packets received every second experiences a slight variation with the use of multiple paths. PDR is not exactly 1 due to the abrupt switch off of interfaces being switched off, losing packets being routed in these interfaces.



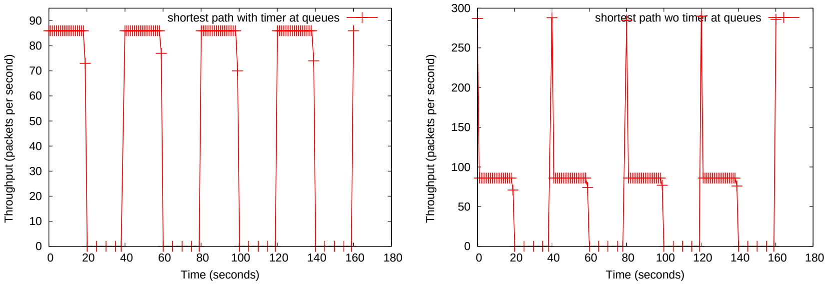
(a) V parameter evolution

(b) Queue evolution with Variable-V

Figure 9: BP behavior under network dynamism.

Second, we observe very short instants in which BP experiences a dramatic degradation of throughput. Figure 9(a) reveals that this corresponds to the time required for BP to decrease its V value so that the SC forming the link with the highest BP (i.e., differential of queue backlogs) is chosen as next hop by the source SC. Figure 9(b) reveals how the source SC never overflows, since the variable-V algorithm leverages the multiple available paths offered by the mesh backhaul testbed. The reconfiguration time of the V parameter is function of the rate at which the data queues in source SC fills up, which is around 1s in this experiment (one flow of 1Mbps) to choose a longer path. In fact, this time would be substantially reduced with the increase of the rate in the source SC. Nonetheless, in [7], we have started to propose some modifications in the BP protocol to reduce the extra latency introduced by this reconfiguration time.

Third, BP periodically shows throughput peaks when the shortest path to the destination is available again. Such a throughput peak is generated by the aggregation of newly generated traffic (at a rate of 85 packets per second) and all the data packets accumulated at the data queue in source SC. Once the queue is empty, the TNL GW serves 1Mbps. As showed by Figure 9(a), the V parameter in source SC evolves by increasing its value implying the use of geolocation information in routing decisions.

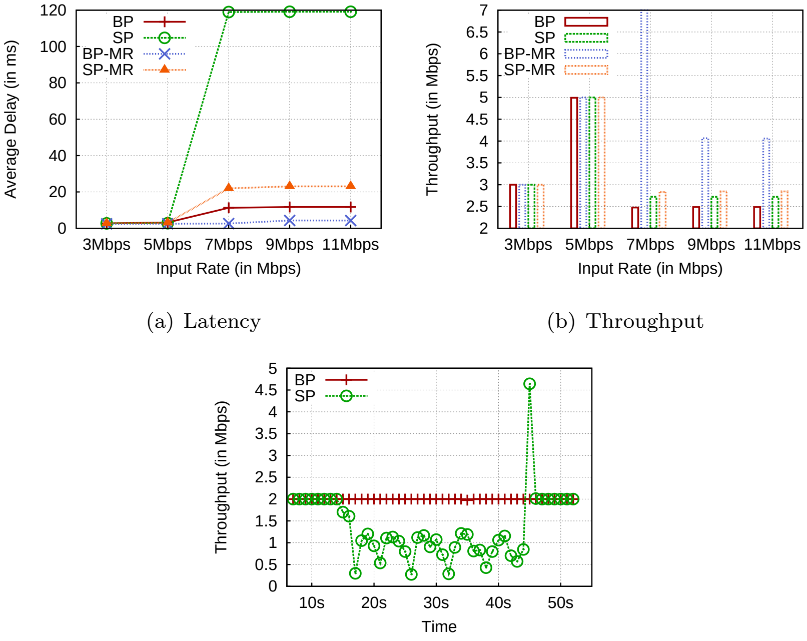


(a) With Queuing Timer limit.

(b) Without Queuing Timer limit.

Figure 10: TNL GW throughput of SP under dynamic network conditions.

With SP, the source SC is unable to transmit packets whenever the SCs belonging to the shortest path are switched off. Therefore, during a time interval of 20 seconds, the TNL GW does not receive data packets (PDR around 0.52). Indeed, the TNL GW starts receiving packets when SCs closer to the destination are available by the source (around 85 packets per second). Note that in Figure 10(a) periodic throughput peaks do not appear, whereas with a timer disabled (see Figure 10(b)) throughput peaks appear. SP without queuing expiration timer transmits all data packets accumulated at the FIFO in source SC (i.e., 200 packets) jointly with the constant rate offered to the network (i.e., 85 packets per second). This explains the throughput peaks of around 285 packets per second observed in Figure 10(b) when the SP is available again. Since the packet timestamp is checked at transmission time and the switch off period is of 20s and queuing timer is of 5s, when the SCs belonging to the path closer to the destination are switched on, there are no data packets valid to transmit accumulated at the FIFO data queue, and they are dropped.



(a) Latency

(b) Throughput

(c) Background traffic during 30s

Figure 11: Single TCP Flow Case.

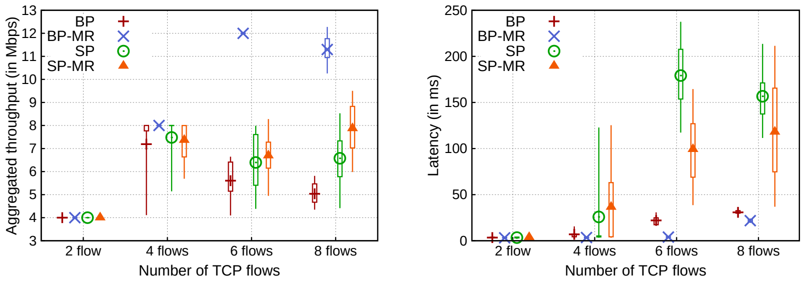
5.3. Interaction with TCP Traffic

The goal of these experiments is to illustrate through ns-3 simulations how BP scales and interacts with TCP traffic. In particular we simulate larger SC deployment areas thus, complementing previous results obtained in the testbed deployed in a confined indoor space. Simulation allowed us not only to show scalability with the size of the network but with the number of backhaul radios per SC. In particular, we tested single-radio and multi-radio SCs, including the multi-radio backhaul two backhaul radios and two orthogonal channels per SC. By its very design, BP can be deployed without changing the TCP stack. The fundamental goal of TCP, which applies to all TCP variants, is to provide a reliable datagram service by means of end-to-end ACK packets. Based on the received ACKs, TCP determines whether and how many data packets should be injected into the network by updating its window size.

5.3.1. Single TCP Flow Case

Our first experiment compares BP and SP with a single TCP flow in a 4x4 grid mesh backhaul. From Figure 11(a), we can observe significant latency gains of BP with respect to SP. The use of all possible paths increases the level of contention in the network in BP with respect to SP, which explains the lower throughput obtained when distributing traffic, achieving 2.5Mbps when injecting 7Mbps, 9Mbps, and 11Mbps. Note that contention would decrease with the use of multiple channels in the network, which will be studied in next section. While contention leads to throughput in both protocols, SP suffers also from a dramatic increase in latency due to the increase of queue backlogs. As depicted in Figure 11(b) BP does not experience such a latency increase since it balances the use of all possible paths to improve network resource utilization. An interesting observation is that when introducing and additional non-overlapping WiFi radio for backhauling in every SC is how the decrease of contention and additional capacity is exploited by both BP and SP, which are labeled as BP-MR and SP-MR in multi-radio multi-channel experiments. However, gains are more significant with BP rather than with SP since contention is the parameter more relieved with the addition of an orthogonal channel in the backhaul.

We analyzed a second experiment where we consider a reference 2Mbps TCP traffic flow sent during 50s. We left background traffic during 30s (from instant $t = 15s$ to $t = 45s$) that overlaps with the fixed path taken by SP and a subset of the paths taken by the reference TCP flow with BP. Figure 11(c) shows the throughput of a single flow while injecting background traffic. While BP reacts and starts using non overlapping paths, SP experiences performance degradation decreasing its rate up to 0.3Mbps since both reference and background traffic are sharing paths. When the background TCP traffic stops, SP experiences an abrupt increase of its TCP performance since the fixed single path can be fully used by the 2Mbps reference TCP flow.



(a) Throughput

(b) Latency

Figure 12: Several TCP Flows Case.

5.3.2. Several TCP Flows Case

The third experiment compares the performance of BP and SP when injecting a variable amount of TCP flows of 2Mbps in a 7x7 grid mesh backhaul. We show the scalability properties of BP with both the number of backhaul radios per SC and the size of the backhaul. For each amount of TCP flows considered, random SCs injects traffic towards the single TNL GW placed in one corner of the 7x7 grid mesh backhaul. Each experiment is repeated 20 times varying the set of TCP flows injected. We use average values and boxplots to plot the statistical distribution of the measured metrics. The box stretches from the 25th to the 75th percentiles, and the whiskers represent the 5th and 95th percentiles.

Figure 12(a) and Figure 12(b) present similar trends as those observed in the single flow case. In the case of single-radio and single-channel, BP suffers from more contention but lower queuing delays than SP with the increase of the number of TCP traffic flows. Contention is alleviated when BP and SP, labeled as BP-MR and SP-MR in Figures 12(a) and 12(b)) are tested with two radios and two channels per SC. Comparison results show that BP-MR better exploits these added resources than SP-MR. When injecting 6 TCP traffic flows BP-MR shows substantial gains of up to 79% and 95% in terms of throughput and latency, respectively. Indeed, both SP and, in a minor extent, SP-MR suffer from contention and higher queuing as TCP flows share their path towards destination. In the multi-radio case, SP-MR is unable to exploit the added resources brought by multiple radios as BP-MR does. In particular, BP-MR showed gains of up to 114% and 81% compared to BP, while SP-MR showed gains of up to 4% and 44% compared to SP in terms of throughput and latency, respectively. These results indicate the convenience of BP over SP in both single-radio and multi-radio networks.

6. Conclusions

This paper has presented an extensive performance evaluation of BP both experimentally and by simulation. Our experimental methodology is based on ns-3 emulation, which allowed us to rapidly run our ns-3 implementation of BP practically unmodified in a WiFi-based mesh backhaul of small cells prototype. We tested BP under a wide variety of conditions in the testbed. With dynamic backhaul configurations BP showed fast adaptability to topology conditions leading to PDR improvements of up to 50% against a SP routing policy. We have demonstrated the scalability of BP with the size of the network and the number of backhaul radios through simulation showing substantial benefits over a SP routing policy. In multi-radio multi-channel backhaul deployments, we observed BP improvements with respect to SP for TCP traffic of up to 79% and 95% in terms of throughput and latency, respectively.

7. Acknowledgements

This work has been funded by the Spanish Ministry of Science and Innovation under grant TEC2011-29700-C02-01. We would also like to thank the reviewers for their constructive and helpful comments.

References

[1] I. F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: A survey, Elsevier Computer Networks 47 (4) (2005) 445–487.

[2] J. Ferragut, J. Mangues-Bafalluy, J. Núñez Martínez, F. Zdarsky, Traffic and mobility management in networks of femtocells, ACM/Springer Mobile Networks and Applications Journal 17 (5) (2012) 662–673.

[3] Draft Standard for Information Technology: Amendment 10: Mesh Networking, IEEE P802.11s/D4.0, Dec. 2009.

[4] L. Tassiulas, A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, IEEE Trans. on Automatic Control 37 (12) (1992) 1936–1948.

[5] R. Laufer, T. Salonidis, H. Lundgren, P. Le Guyadec, Xpress: a cross-layer backpressure architecture for wireless multi-hop networks, in: Proc. of the 17th MobiCom, ACM, 2011, pp. 49–60.

[6] J. Núñez-Martínez, J. Mangues-Bafalluy, J. Baranda, Anycast backpressure routing: Scalable mobile backhaul for dense small cell deployments, IEEE Communications Letters 17 (2013) 2316–2319.

[7] J. Núñez-Martínez, J. Baranda, J. Mangues-Bafalluy, Backpressure routing for the backhaul in sparse small cell deployments, in: Proc. of the 32nd IPCCC, IEEE, 2013, pp. 1–2.

[8] The ns-3 network simulator, available at: `http://www.nsam.org`.

[9] J. Núñez-Martínez, J. Mangues-Bafalluy, A case for evaluating backpressure routing using ns-3 emulation in a wifi mesh testbed, in: Proc. of the 7th ACM WinTECH, ACM, 2012, pp. 27–34.

[10] K. Fall, A network emulation in the vint/ns simulator, in: Proc. of the IEEE Symp. on Computers and Communications, IEEE, 1999, pp. 244–250.

[11] S. Jansen, A. McGregor, Simulation with real world network stacks, in: Proc. of Winter Simulation Conference, IEEE, 2005.

[12] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, W. Dabbous, Direct code execution: Revisiting library os architecture for reproducible network experiments, in: Proc. of the 9th ACM CoNEXT, 2013, pp. 217–228.

[13] G. Carneiro, H. Fontes, M. Ricardo, Fast prototyping of network protocols through ns-3 simulation model reuse, Simulation Modelling Practice and Theory 19 (2011) 2063–2075.

[14] M. Bocci, A framework for mpls in transport networks, Internet RFC 5921.

[15] J. Núñez-Martínez, J. Mangues-Bafalluy, A survey on routing protocols that really exploit wireless mesh network features, Journal of Communications 5 (3) (2010) 211–231.

[16] J. Friginal, J.-C. Ruiz, D. de Andrés, A. Bustos, Ambient noise in wireless mesh networks: Evaluation and proposal of an adaptive algorithm to mitigate link removal, Journal of Network and Computer Applications 41 (2014) 505–516.

[17] D. Gokhale, S. Sen, K. Chebrolu, B. Raman, On the feasibility of the link abstraction in (rural) mesh networks, in: Transactions on Networking, IEEE, 2009.

[18] P. Serrano, C. J. Bernardos, A. De La Oliva, A. Banchs, I. Soto, M. Zink, Floornet: Deployment and evaluation of amultihop wireless 802.11 testbed, EURASIP Journal on Wireless Communications and Networking 2010.

[19] F. Cadger, K. Curran, J. Santos, S. Moffett, A survey of geographical routing in wireless ad-hoc networks, IEEE Communications Surveys Tutorials 15 (2013) 621–653.

[20] 3GPP, TS 25.444 v. 10.0.0; UMTS Iuh data transport.

[21] K. Papagiannaki, M. D. Yarvis, W. S. Conner, Experimental characterization of home wireless networks and design implications., in: Proc. of the 25th INFOCOM, IEEE, 2006.

[22] J. Bicket, D. Aguayo, S. Biswas, R. Morris, Architecture and evaluation of an unplanned 802.11 b mesh network, in: Proc. of the 11th MobiCom, ACM, 2005, pp. 31–42.