Elucidating the Instability of Random Access Wireless Mesh Networks

Adel Aziz
EPFL
Lausanne, Switzerland
adel.aziz@epfl.ch

David Starobinski
Boston University
Boston, MA
staro@bu.edu

Patrick Thiran
EPFL
Lausanne, Switzerland
patrick.thiran@epfl.ch

***Abstract*— We investigate both theoretically and experimentally the stability of CSMA-based wireless mesh networks, where a network is said to be stable if and only if the queue of each relay node remains (almost surely) finite. We identify two key factors that impact stability: the network size and the so-called "stealing effect", a consequence of the hidden node problem and non-zero propagation delays. We consider the case of a greedy source and prove, by using Foster's theorem, that 3-hop networks are stable, but only if the stealing effect is accounted for. On the other hand, we prove that 4-hop networks are always unstable (even with the stealing effect) and show by simulations that instability extends to more complex linear and non-linear topologies. We devise a stabilization strategy that throttles the source and prove that there exists a finite, non-zero rate at which the source can transmit while keeping the system stable. We run real experiments on a testbed composed of IEEE 802.11 nodes, which show the contrasting behavior of 3-hop and 4-hop networks and the effectiveness of our stabilization strategy.**

I. Introduction

A major challenge in multihop wireless networks is to devise efficient decentralized Medium Access Control (MAC) protocols. This is particularly relevant for Wireless Mesh Networks (WMNs) that promise to deliver ubiquitous, high speed Internet access at low cost. Quite a few of these prototypes have been deployed by both academia and the industry; they eliminate the expensive wired connections in the backbone by having their architecture composed of: an *Access Part* (AP) to deliver access to the users and a *backbone* to forward the traffic to/from a Wired Access Point (WAP) to the AP through multiple Transit Access Points (TAPs).

This paper deals with the backbone part of a WMN, whose multihop nature requires distributed solutions for the scheduling problem in order to work in practice. Most current MAC protocols use Carrier-Sense Multiple Access (CSMA), and its most popular implementation IEEE 802.11 [4], to solve the scheduling problem in a distributed manner. However, these protocols were initially designed for single-hop communication and therefore are not adapted to multihop scenarios, where multiple links have to cooperate to efficiently transport a single flow from its source to its destination. As a result, the buffer may overflow, leading to very poor throughput and large delays [11]. It is therefore critical to rigorously characterize the behavior of CSMA-like protocols in multihop scenarios and propose possible improvements when appropriate.

We focus our analytical investigation on one-dimensional backbone mesh networks, made of $K+1$ nodes $0, 1, \dots, K$. Node 0 is the WAP that we assume to be saturated (it always has traffic to send). The K other nodes are the TAPs, with nodes $1, \dots, K-1$ relaying traffic until the last node K. The transmission/receiving range is limited to the nearest neighbor transmissions; the interference range is limited to the second nearest neighbors. Such networks have been analyzed by [11, 19], when all the TAPs are bufferless, and with CSMA protocols modified accordingly to avoid or handle losses. We consider, instead, unmodified CSMA protocols, and TAPs with a large buffer (which is usually the case) that we approximate to be infinite in our model. The network is then lossless, but we are now confronted with a stability problem, as the buffer queues (and thus delays) may become infinite. As shown in this paper, even for linear topologies a rigorous analysis is far from trivial. Fortunately, simulations show that our analytical insights extend to more general topologies.

We prove that the network is stable or unstable, depending on its size and a phenomenon referred to as a *stealing effect* that results from the hidden node problem and non-zero propagation delays. The likelihood of this phenomenon is captured by the stealing effect probability $0 \leq p \leq 1$, a parameter explained in detail in Section IV.C. The stealing effect is absent (i.e., $p = 0$) only in the unrealistic case where RTS/CTS control messages are instantaneous. It becomes more significant when RTS/CTS are disabled. Quite surprisingly, the stealing effect plays a stabilizing role on the network.

After reviewing the related work in Section II, we provide experimental evidence of the different behavior in terms of stability of wireless IEEE 802.11b mesh networks in Section III, when $K = 3$ or $K = 4$. Multihop 802.11 networks are known to suffer from unfairness and starvation (see [12, 15, 21, 22]), but to the best of our knowledge, the (in)stability of multihop 802.11 networks has not be demonstrated either experimentally or analytically to date.

In Section IV, we introduce a discrete Markov chain model to capture this phenomenon. We demonstrate in Section V that if $K = 3$, then the network is stable if and only if the stealing effect is present ($p > 0$). When $K > 3$ however, the network is always unstable, as proven in Section VI for $K = 4$, and presumably so as well for larger K, as shown in Section VII with a formal proof for the case $p = 0$. Our analytical findings therefore support the ad-hoc observation that, with current hardware, it does not make sense to run mesh networks more than 3-hop long [2].

We therefore propose in Section VIII a *stabilization strategy* that keeps the network stable by throttling the source. This strategy is implementable in existing MAC cards by assigning to the WAP a larger minimum contention window than to the TAPs. The ratio between the values of the minimum contention windows is captured in our model by a throttling factor $0 \leq q \leq 1$, defined in Section VIII. We compute in the same section bounds on the threshold value of q ensuring stability.

The (in)stability of wireless mesh networks also arises in more general topologies, as ns-2 simulations show in Section IX. Finally, we summarize the results in Section X.

II. Related work

The unarguable success of the IEEE 802.11 [4] protocol in WiFi communications has lead to the current development of a new draft focusing on multihop networks such as WMNs, 802.11s [8, 10]. However, until the release of 802.11s, 802.11b/g remains the standard and it is therefore essential to understand its behavior. Towards this goal, previous works [6, 10, 20] present drawbacks of the current protocol in a multihop environment. In [15], Garetto et al. present a model to derive the throughput of flows in a multihop networks. Furthermore, Ng et al. identify the existence of an optimal offered load and propose source rate limiting at the application layer as a solution [20]. Our approach differs in the sense that we introduce an analytical model that focuses on buffer stability and therefore gives insight into the existence of a maximal feasible load. Furthermore our stabilization strategy uses solely the MAC layer and therefore does not impose any limiting requirement on the client side.

Tackling the congestion problem at the transport layer, e.g. TCP, is studied in [21, 22]. Shi et al. focus on inter-flow competition by studying the starvation occurring when a one-hop flow competes with a two-hop flow and propose a counter-starvation technique that solves the problem. Similarly Rangwala et al. propose an other rate-control protocol achieving better fairness and efficiency than TCP. Our works differs as we focus on the link competition taking place within a single flow and study the factors leading to transitions from stability to instability. Thus, rather than relying on the transport layer, we throttle congestion at the MAC (link) layer (in the OSI model, this functionality of the link layer is referred to as flow control [16]). Analytical and simulation results showing that a hop-by-hop congestion algorithm outperforms an end-to-end version are presented in [25]. Their findings reinforce the need to implement congestion control at the MAC layer.

The work in [18] studies the system stability of random access protocols in single-hop settings. Our work goes further by analyzing the multihop scenario where the buffer states of successive queues are dependent.

The distributed scheduling problem, which aims at ensuring stability and maximal throughput, has witnessed growing interest in the research community. The seminal work [23] introduces a back-pressure algorithm that uses global network buffer information to derive an optimal routing/scheduling policy and achieve stability and maximal throughput. Several extensions of this work have been conducted, e.g., [26] reduces the number of queues maintained at each node to enhance scalability. Further work on throughput and fairness guarantee can be found in [9], where Chapokar et al. introduce a distributed scheduling strategy that attains a guaranteed ratio of the maximal throughput. A more complete review of the stability problem in scheduling is presented in [24], where the tradeoff between complexity, utility and delay is discussed in depth. Finally a scheduling policy based on the buffer queue length is presented and studied analytically in [17]. These works propose conceptual scheduling solutions that keep the network stable, but which depart from 802.11 protocols to various extents, and for which no practical implementation exists to date. Our work differs from this previous body of work, as we focus on the stability of *existing* CSMA protocols, e.g. 802.11. To the best of our knowledge, we are the first to identify the key factors (network size and stealing effect) that affect the network stability. Furthermore, following our analytical study, we develop a practical stabilization strategy and validate experimentally our results with off-the-shelf hardware.

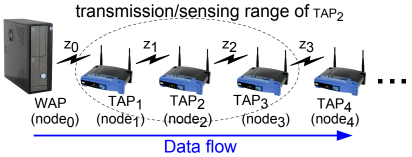

Fig. 1. Linear topology of a WMN backbone.

III. Experimental analysis

We show, through measurements on a testbed, an unexpected behavior of 802.11 multihop networks, which is the striking difference in stability between 3- and 4-hop networks.

A. Scenario

We consider a linear topology motivated by the backbone of WMNs, where each node hears only its direct neighbors (cf. Figure 1). A common scenario in these networks occurs when the gateway, the WAP, needs to send packets to an end-user that is beyond its direct coverage range. To achieve this goal, the packets transit through the wireless backbone, forwarded by multiple TAPs. As the backbone part is a key element of any WMN, we are interested in optimizing its performance as far as throughput and delay are concerned.

B. System description

Our testbed is composed of two laptops that act as the source and sink of the traffic and five wireless routers that act as the WAP (node 0) and TAP_i (node i with $1 \leq i \leq 4$).

Each laptop runs on Linux with the software *Iperf* [1] used to generate saturated UDP traffic with payload size of 1470 bytes. Each laptop is then connected through a wired cable to either the WAP or the last TAP (i.e., node 4).

The wireless routers are Asus WL-500gP running the version *Kamikaze 7.07* of the OpenWRT firmware [3]. We change the mini-PCI WiFi cards to Atheros cards in order to benefit from the flexibility of the MadWifi driver [5]. This allows the

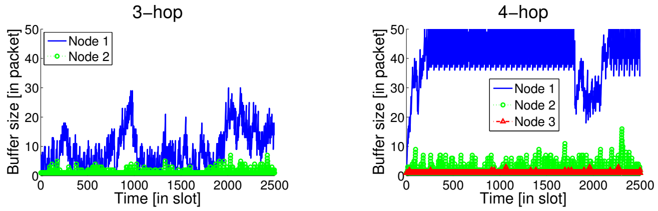

Fig. 2. Experimental results for the buffer evolution of each relay node in 3- and 4-hop topologies. A time slot corresponds to an event when the buffer size is recorded, that is every time a packet arrives at a node.

modifications of the driver source code to perform both buffer monitoring and the modification of the contention window. We then set the routers to run in ad-hoc mode on channel 13 of 802.11b at the data rate of 1Mb/s and without RTS/CTS. To avoid interference from neighboring networks, we perform our measurements in the basement of the BC building at EPFL, where no other wireless networks could be sensed. Finally, we set our topology to match our theoretical study: direct neighbors can communicate together but nodes separated by two hops or more cannot hear each other.

C. Measurement results

The major finding is the drastically different stability behavior of 3-hop and 4-hop topologies that can appear to be counter-intuitive. Figure 2 shows that the 3-hop topology is stable, but the 4-hop network is unstable. Furthermore, the 4-hop instability is due to node 1, whose queue length exhibits a transient behavior, i.e., which indefinitely grows until it reaches the hardware limit (50 packets).

IV. Model

A. MAC layer description

In order to understand the experimental stability results obtained in Section III, we introduce an analytical model that is inspired from the behavior of CSMA/CA protocols (e.g., 802.11-like protocols) with some necessary simplifications for the sake of tractability. We emphasize that, given the mathematical assumptions, our analysis is exact.

The first common assumption [9, 13, 18, 23, 26] is that of a slotted discrete time axis, that is, each transmission takes one time slot and all the transmissions occurring during a given slot start and finish at the same time. We consider a greedy source model, i.e., the WAP (gateway) always has new packets ready for transmission. Assuming a K-hop system, the packets flow from the WAP to TAP_K, via TAP_1, TAP_2, ..., TAP_{K-1} (see Figure 1). TAPs do not generate packets of their own. Each TAP is equipped with an infinite buffer.

We assume that the system evolves according to a two-phase mechanism: a *link competition phase* and a *transmission phase*. The link competition phase, whose length is assumed to be negligible, occurs at the beginning of each slot. During this phase, all the nodes having a non-empty buffer compete for the channel and a pattern of successful transmissions emerges, referred to as *transmission pattern* in this paper. Given the current state of buffers, the link competition process is assumed to be independent of competitions that happened in previous slots. This assumption is similar to the commonly used assumption of exponentially (memoryless) distributed backoffs. During this phase, non-empty nodes are sequentially chosen at random and added to the transmission pattern if and only if they do not interfere with already selected communications (with the notable exception of the *stealing effect* described below). The final pattern is obtained when no more nodes can be added without interfering with the others

The second phase of the model is fairly straightforward as it consists in applying the transmission pattern from the previous phase in order to update the buffer status of the system. This buffer status information is of utmost importance for our analysis because it is the parameter that indicates whether the network remains stable (no buffer explodes) or suffers congestion (one or more buffers build up).

B. Discrete Markov Chain Model

We now formalize the model just described mathematically. All packets are generated by the WAP (node 0), and are forwarded to the last TAP (node K) by successive transmissions via the intermediate nodes (TAPs) 1 to $K-1$. A time step $n \in \mathbb{N}$ corresponds to the successful transmission of a packet from some node i to its neighbor $i+1$, or if K is large enough, of a set of packets from different non interfering nodes $i, j, \ldots$ to nodes $i+1, j+1, \ldots$, provided these transmissions overlap in time (the transmitters and receivers must therefore not interfere with each other). We assume that node 0 always has packets to transmit (infinite queue), and that node K consumes immediately the packets, as it is the exit point of the backbone (its queue is always 0).

We are interested in the evolution of the queue sizes b_i of relaying nodes $1 \le i \le K-1$ over time, and therefore we adopt as a state variable of the system at time n the vector

$$\vec{b}(n) = [b_1(n)\ b_2(n)\ \ldots\ b_{K-1}(n)]^T ,$$

with T denoting transposition. We also introduce a set of K auxiliary binary variables z_i, $0 \le i \le K-1$, representing the i^{th} link activity at time slot n: $z_i(n) = 1$ if a packet was successfully transmitted from node i to node $i+1$ during the n^{th} time slot, and $z_i(n) = 0$ otherwise. Observing that $b_i(n+1) = b_i(n) + z_{i-1}(n) - z_i(n)$, we can recast the dynamics of the system as

$$\vec{b}(n+1) = \vec{b}(n) + A * \vec{z}(n) \tag{1}$$

where

$$\begin{aligned} \vec{z}(n) &= [z_0(n)\ z_1(n)\ z_2(n)\ \ldots\ z_{K-1}(n)]^T \\ A &= \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & 1 & -1 \end{bmatrix} . \end{aligned}$$

Finally, the activity of a link z_i depends on the queue sizes of all the nodes, which we cast as $z_i = g_i(\vec{b})$ for some random function $g_i(\cdot)$ of the queue size vector, or in vector form as

$$\vec{z}(n) = g(\vec{b}(n)). \tag{2}$$

The specification of $g = [g_0, \dots, g_{K-1}]^T$ is the less straightforward part of the model, as it requires to enter in some additional details of the CSMA/CA protocols, which we differ for the next sections. We will first expose it in Section V for a $K = 3$ hops network, and then move to the larger networks with $K = 4$ and $K \geq 5$ in the next two sections, as the specification of g comes with some level of complexity as K gets larger. Nevertheless, we can already mention here two simple constraints that g must verify: (i) node i cannot transmit if its buffer is empty, and therefore $z_i = g_i(\vec{b}) = 0$ if $b_i = 0$; (ii) nodes that successfully transmit in the same time slot must be at least 2 hops apart, as otherwise packet from node i would collide at node $i+1$ with the packet from node $i+2$. Hence

$$z_i z_{i+k} = 0 \text{ for } k \in \{-2, -1, 1, 2\}. \qquad (3)$$

We observe that (1) and (2) make the model a discrete-time, irreducible Markov chain. The (in)stability of the network coincides with its (non-)ergodicity.

C. Stealing effect phenomenon

The stealing effect phenomenon is a result of the well-known hidden node problem occurring in multihop topologies. Indeed, the existence of directional flow in the backbone of mesh networks, from WAP (node 0) to TAP_K (node K) may induce unfairness in a way that does not arise in single-hop scenarios. When node i first enters the link competition phase, node $i+2$ may be unaware of this transmission attempt. Node $i+2$ may therefore start a concurrent transmission as it senses the medium to be idle. As a collision occurs at node $i+1$, node i will experience an unsuccessful transmission while the transmission from node $i+2$ will succeed. We refer to this unfairness artifact as the stealing effect.

Definition 1 (Stealing effect): The stealing effect occurs when a downstream node $i+2$ successfully captures the channel from an upstream node i, even though it accesses the medium later. We define p to be the probability of the occurrence of the stealing effect.

In IEEE 802.11, the stealing effect corresponds to the event where node $i+2$ captures the channel even though it has a larger backoff value than node i. The probability of this event depends on the specific protocol implementation. If the optional RTS/CTS handshake is disabled, then $p \to 1$. If RTS/CTS is enabled, then p is typically much smaller, but still non-zero because RTS messages may collide [22]. Indeed, the transmission time of a control message (e.g., the RTS transmission time at the 1Mb/s basic rate is $352\mu s$) is non-negligible compared to the duration of a backoff slot ($20\mu s$).

In our model, the stealing effect is captured by having the function $g(\cdot)$ in (2) depending on p. As revealed by our analysis, a positive and somewhat counterintuitive consequence of the stealing effect is the promotion of a laminar packet flow, namely, a smooth propagation of packets. Indeed, by favoring downstream links over upstream ones, it creates a form of virtual back-pressure that prevents packets from being pushed too quickly into the network.

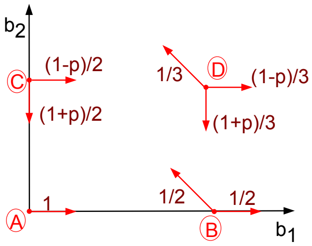

Fig. 3. Random walk in $\mathbb{Z}^2$ modeling the 3-hop network.

D. Stability definition

A buffer is stable when its occupancy does not tend to increase forever. More formally, we adopt the usual definitions of stability (see e.g. Section 2.2 of [7]).

Definition 2 (Stability): A buffer is stable when its evolution is ergodic (it goes back to zero a.s. in finite time). A network is stable when the queues of all forwarding nodes (i.e., all TAPs) are stable.

V. 3-HOP NETWORKS

Let us first analyze the 3-hop topology, which remains relatively simple because only one link can be active at a given time slot. Indeed, the only three possible transmission patterns $\vec{z}$ are $[1\ 0\ 0]^T$, $[0\ 1\ 0]^T$ and $[0\ 0\ 1]^T$. We can now complete the description of the function $g(\cdot)$, before analyzing the ergodicity of the Markov chain.

A. System evolution

The role of the stochastic function $g(\cdot)$ is to map a buffer status $\vec{b}$ to a transmission pattern $\vec{z}$ with a certain probability.

First, in the case of an idealized CSMA/CA model without the stealing effect ($p = 0$), all non-empty nodes have exactly the same probability of being scheduled. That is, if only node 0 and node 1 (or, respectively, node 2) have a packet to send, both patterns $[1\ 0\ 0]^T$ and $[0\ 1\ 0]^T$ (resp., $[0\ 0\ 1]^T$) happen with probability $1/2$. Similarly, when all three nodes have a packet to send, each of the three possible transmission patterns happens with probability $1/3$.

More generally, when we include the stealing effect, we capture the bias towards downstream links that are two hops away. When only node 0 and node 1 compete for the channel, nothing is changed and the probability of success remains $1/2$ as they are only separated by one single hop. However, when node 0 and node 2 compete together, there is a probability p that node 2 steals the channel.

This leads us to define function $g(\cdot)$ differently for each region of $\mathbb{Z}^2$ as shown in Figure 3. First, in region $A = \{b_1(n) = 0, b_2(n) = 0\}$, $g([b_1(n)\ b_2(n)]^T) = [1\ 0\ 0]^T$. In region $B = \{b_1(n) > 0, b_2(n) = 0\}$ we have that

$$g([b_1(n)\ b_2(n)]^T) = \begin{cases} [1\ 0\ 0]^T & \text{with probability } 1/2 \\ [0\ 1\ 0]^T & \text{with probability } 1/2. \end{cases}$$

In region $C = \{b_1(n) = 0, b_2(n) > 0\}$,

$$g([b_1(n)\ b_2(n)]^T) = \begin{cases} [1\ 0\ 0]^T & \text{with probability } (1-p)/2 \\ [0\ 0\ 1]^T & \text{with probability } (1+p)/2. \end{cases}$$

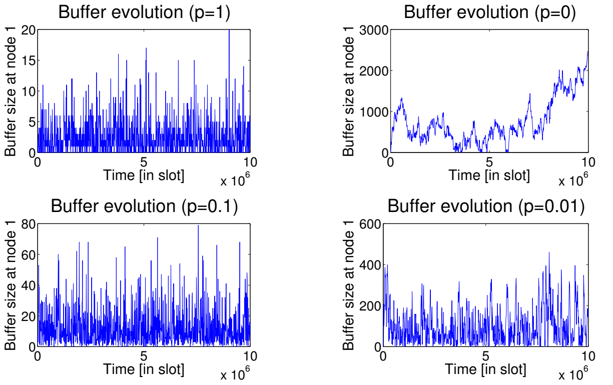

Fig. 4. Buffer evolution for 3-hop with different p values.

Finally, in region $D = \{b_1(n) > 0, b_2(n) > 0\}$, all three nodes compete, and node 2 can still steal the channel from node 0, hence

$$g([b_1(n)\ b_2(n)]^T) = \begin{cases} [1\ 0\ 0]^T & \text{with probability } (1-p)/3 \\ [0\ 1\ 0]^T & \text{with probability } 1/3 \\ [0\ 0\ 1]^T & \text{with probability } (1+p)/3. \end{cases}$$

B. Stability analysis

The buffer evolution from (1) is a random walk in $\mathbb{Z}^2$, as depicted in Figure 3. Theorem 1 shows the stabilizing influence of the stealing effect.

Theorem 1: A 3-hop network is unstable for the case $p = 0$ and it is stable for all $0 < p \leq 1$.

Proof: The instability of the case $p = 0$ is readily proved with the Non-ergodicity theorem ([14], p. 30) using the Lyapunov function

$$h(b_1, b_2) = b_1, \tag{4}$$

and setting the constants $c = d = 1$ in that theorem.

Next we prove the stability of the cases $0 < p \leq 1$ by using Foster's theorem ([14], p. 29) with the Lyapunov function

$$h(b_1, b_2) = b_1^2 + b_2^2 - b_1 b_2,$$

the finite set $F = \{0 \leq b_1, b_2 < 5/p\}$, and the notations

$$\begin{aligned} \mu_{b_1,b_2}(n) &= \mathbb{E}\left[h(\vec{b}(n+1)) \mid h(\vec{b}(n)) = h(b_1, b_2)\right] \\ \epsilon_{b_1,b_2}(n) &= \mu_{b_1,b_2}(n) - h(b_1, b_2), \end{aligned}$$

where $\epsilon_{b_1,b_2}(n)$ can be interpreted as the drift of the random walk at time n. Then we verify Foster's theorem for all the three regions of $\mathbb{Z}^2 \setminus F$. After some computations, we find that for Region $B \setminus F$, $\epsilon_{b_1,0}(n) = 2 - b_1(n)/2 < 0$. Likewise, for region $C \setminus F$, we get $\epsilon_{0,b_2}(n) = 1 - (3+p)b_2(n)/2 < 0$. Finally, for region $D \setminus F$, we have $\epsilon_{b_1,b_2}(n) = 5/3 - p(b_1(n) + b_2(n))/3 < 0$. Consequently, the two conditions of the theorem are satisfied and stability is proved. ■

Finally, we present in Figure 4 the effect of p on the buffer evolution through a simulation of our model. We also mention, that our theoretical results give insight to monitor the buffer of node 1 in order to assess the stability of the system (the function of (4) only considers buffer b_1 to prove instability).

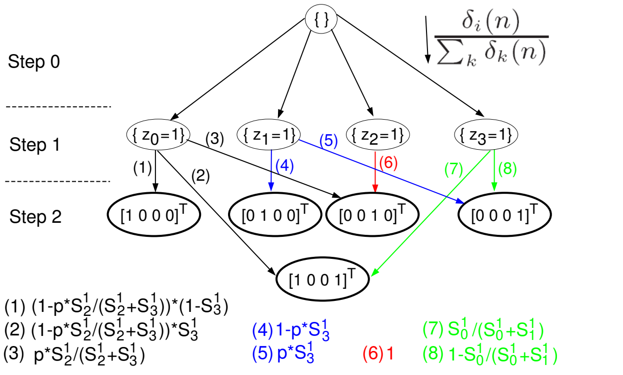

Fig. 5. *Decision tree* to obtain $\vec{z} = g(\vec{b})$ for the 4-hop model.

VI. 4-hop Networks

The 4-hop system is relatively similar to the 3-hop except that the function $g(\cdot)$ becomes more complex to derive. Indeed the five possible patterns $\vec{z}$ are now $[1\ 0\ 0\ 0]^T$, $[0\ 1\ 0\ 0]^T$, $[0\ 0\ 1\ 0]^T$, $[0\ 0\ 0\ 1]^T$ and $[1\ 0\ 0\ 1]^T$

A. System evolution

The drastic difference when moving to 4-hop topologies is that nodes that can transmit concurrently will *reinforce* each other and will increase their transmission probability [11, 12].

This inter-dependence makes the determination of $g(\cdot)$ less straightforward than in the 3-hop case. We capture this complexity by a *decision tree*, depicted in Figure 5, which maps all the sequential events that can occur for the selection of the transmission pattern (state in bold in Figure 5).

Before describing the exact mechanisms behind our decision tree, we introduce some necessary notations. First, we define the iteration step m that represents the step between two sequential events (an event corresponds to either the inclusion of a node in the transmission pattern or the removal of a node from the competition). As shown in Figure 5, the decision-tree process ends in two iterations ($m \in \{0, 1, 2\}$) and this is due to the fact that at most two links can be active concurrently in the transmission pattern of a 4-hop network.

Secondly, we introduce the two indicator vectors $\vec{\delta}(n)$ and $\vec{S}^m$. The four entries $\delta_i(n) = 1_{\{b_i(n)>0\}}$ indicate which buffers are occupied ($\delta_i(n) = 1$) or empty ($\delta_i(n) = 0$). The vector $\vec{S}^m = [S_0^m\ \ldots\ S_3^m]^T$, which is obtained through an iterative process, indicates the set of nodes that are still in competition for the channel at the iteration step m. Initially, all the nodes having a non-empty buffer compete for the channel at step 0 and therefore $\vec{S}^0 = \vec{\delta}(n)$. Then the indicator vector at step m, $\vec{S}^m$, is obtained by removing from $\vec{S}^{m-1}$ the node that was selected at the iteration step m and its direct neighbors. For example, if we start from the fully-occupied case $\vec{S}^0 = \vec{1}$ and follow the path where node 1 is selected (z_1 is set to 1), the nodes 0, 1 and 2 are removed from the competition and the new indicator vector becomes $\vec{S}^1 = [0\ 0\ 0\ 1]^T$ for this path.

The exact probabilities of each link of the decision tree are denoted in Figure 5. The intuition behind these probabilities is that at step m all nodes i that are still competing for the

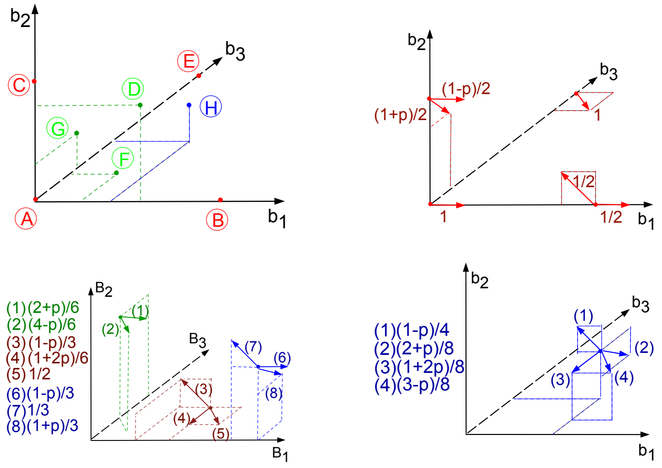

Fig. 6. Random walk in $\mathbb{Z}^3$ for a 4-hop network.

channel (i.e., $S_i^m = 1$) have an equal probability of being selected for transmission. Furthermore, if z_{i-2} is already set to 1 at step m, the selected node i has a probability p of successfully stealing the channel, in which case z_{i-2} is set to 0 and z_i is set to 1 instead. Otherwise, z_i is set to 0.

The computation of the different transmission patterns probability (i.e., the determination of the function $g(\cdot)$) is obtained by summing up the *path probability* of each of the paths leading to one of the five possible transmission patterns (state in bold in Figure 5). In other words, the probability of the pattern $[1\ 0\ 0\ 0]^T$ (resp., $[0\ 1\ 0\ 0]^T$) is the probability of having z_0 (resp. z_1) set to 1 at step 0, multiplied by the probability of keeping this selection at step 1 (i.e., no additional active link or stealing effect). Similarly, the probability of the pattern $[0\ 0\ 1\ 0]^T$ (resp., $[0\ 0\ 0\ 1]^T$) is obtained by adding: (i) the probability of having z_2 (resp. z_3) set to 1 at step 0, multiplied by the probability of having this selection maintained at step 1 and (ii) the probability of having z_0 (resp. z_1) set to 1 at step 0, multiplied by the probability of having the stealing effect at step 1. Finally, the probability of the pattern $[1\ 0\ 0\ 1]^T$ is obtained by adding: (i) the probability of having z_0 set to 1 at step 0 multiplied by the probability of having z_3 set to 1 at step 1 and (ii) the probability of having z_3 set to 1 at step 0 multiplied by the probability of having z_0 set to 1 at step 1. Similarly to Figure 3, Figure 6 summarizes the transmission patterns probability (i.e., $g(\cdot)$) for each of the 8 regions $A = \{0,0,0\}, \ldots, H = \{b_1(n) > 0, b_2(n) > 0, b_3(n) > 0\}$ of $\mathbb{Z}^3$.

B. Stability analysis

Similarly to the 3-hop network, we model the buffer evolution by the random walk in $\mathbb{Z}^3$ depicted in Figure 6. However, contrary to 3-hop case, the 4-hop case presents a structural factor that makes the system unstable either with or without the stealing effect as stated in Theorem 2.

Theorem 2: A 4-hop network is unstable for all $0 \leq p \leq 1$.

Proof: Starting with $p \neq 1$, we introduce the Lyapunov function

$$h(b_1, b_2, b_3) = b_1 + \frac{p}{1+p} b_3, \tag{5}$$

the constants $c = 3$, $d = 1$, $\epsilon = (1-p)/36$ and

$$k(i) = \begin{cases} 3 & \text{if } i \in \text{region } B \\ 2 & \text{if } i \in \text{region } D \\ 1 & \text{otherwise} \end{cases}, \tag{6}$$

Furthermore we introduce the notation

$$\begin{aligned} \mu_{k,b_1,b_2,b_3}(n) &= \mathbb{E}[h(\vec{b}(n+k)) | h(\vec{b}(n) = h(b_1, b_2, b_3))] \\ \epsilon_{k,b_1,b_2,b_3}(n) &= \mu_{k,b_1,b_2,b_3}(n) - h(b_1, b_2, b_3), \end{aligned}$$

where $\epsilon_{k,b_1,b_2,b_3}(n)$ is the drift of the k-step random walk, and verify condition 2 of the Transience theorem ([14], p. 31) in Table I.

Region	ϵ-value
$A \cap S_c$	$\epsilon_{1,0,0,0} = 1 \geq \epsilon$
$B \cap S_c$	$\epsilon_{3,b_1,0,0} = \frac{1-p}{36} \geq \epsilon$
$C \cap S_c$	$\epsilon_{1,0,b_2,0} = \frac{1-p}{2} + \frac{1+p}{2} \frac{p}{1+p} = \frac{1}{2} \geq \epsilon.$
$D \cap S_c$	$\epsilon_{2,b_1,b_2,0} = \frac{1-p}{24} + \frac{p^2}{12} \geq \epsilon$ for $b_2 > 1$ $\epsilon_{2,b_1,1,0} = \frac{1-p}{18} \geq \epsilon$
$E \cap S_c$	$\epsilon_{1,0,0,b_3} = \frac{1}{1+p} \geq \epsilon$
$F \cap S_c$	$\epsilon_{1,b_1,0,b_3} = \frac{1-p}{6(1+p)} \geq \epsilon$
$G \cap S_c$	$\epsilon_{1,0,b_2,b_3} = \frac{4+p+p^2}{6(1+p)} \geq \epsilon$
$H \cap S_c$	$\epsilon_{1,b_1,b_2,b_3} = \frac{p^2+1}{8(1+p)} \geq \epsilon$

TABLE I

PROOF OF CONDITION 2 OF THE TRANSIENCE THEOREM FOR $p \neq 1$.

Consequently, as conditions 1 and 3 are trivially satisfied, the system is unstable for $p \neq 1$.

A similar methodology is used for $p = 1$ beside the change in Lyapunov function. Indeed, function (5) suffers a zero-drift ($\epsilon_{1,b_1,0,b_3}(n) = 0$) in the region $F \cap S_c$ for $p = 1$. Furthermore, we have from Figure 6 that from region F (with $b_3 = x$) we need at least x-steps to leave F. As this violates condition 1 of the Transience theorem, the function (5) is unsuited and we therefore use a Lyapunov function strictly increasing in F

$$h(b_1, b_2, b_3) = b_1. \tag{7}$$

Then, even though the methodology of the proof remains the same, we use a computer-assisted proof due to the relatively high number of steps k needed to satisfy condition 2 of the Transience theorem . We finally prove instability for $p = 1$ by applying the Transience theorem with parameters: $c = 19$ (note that this reduces the proof to regions B, D, F and H), $d = 1$, $\epsilon = 0.003$ and

$$k(i) = \begin{cases} 17 & \text{if } i \in \text{region } B \\ 15 & \text{if } i \in \text{region } D \\ 1 & \text{if } i \in \text{region } F \text{ or } H. \end{cases} \tag{8}$$

■

Finally, we present in Figure 7 the simulation results showing instability independently of the p value. These results are fundamental for real networks as they reveal the tendency of CSMA to naturally produce instability for 4-hop topologies.

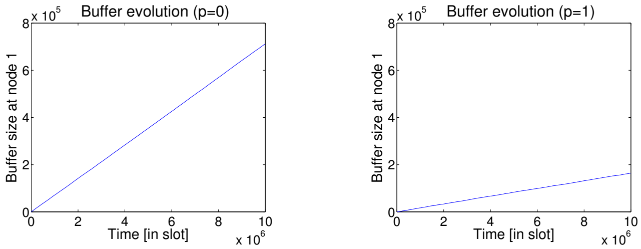

Fig. 7. Buffer evolution for 4-hop with different p values.

VII. General K-hop networks

The transition from stability (for $p > 0$) to instability occurs between $K = 3$ to $K = 4$ hops, as we just proved in the two previous sections. We conjecture that the system remains unstable for larger topologies as well, and we prove it for $p = 0$.

The description of function $g(\cdot)$ for a general $K > 3$ is obtained by directly extending the algorithm parsing the decision tree, described in the previous section. Following the same methodology as in Section VI, we confirm either by simulation ($K = 5$) or theoretically ($K = 6$) that the network remains unstable for larger topologies for all $0 \leq p \leq 1$. Moreover, in the case without the stealing effect ($p = 0$), we can easily prove the network instability for $K = 2$, and we just did it in the previous sections for $K = 3, 4$. When $p = 0$, the instability of a K-hop topology for any $K > 4$ follows then from the following lemma.

Lemma 1 (K-hop instability): If $p = 0$, a sufficient condition for a linear K-hop network to satisfy the conditions of the Non-ergodicity theorem ([14], p. 30) and thus to be unstable is that both the $(K-1)$ and $(K-3)$ hop networks satisfy the conditions of the Non-ergodicity theorem.

Proof: Let us denote the next step expectation of a K-hop network by $\mu_i^K(n) = \mathbb{E}[h(\vec{b}(n+1)) \mid h(\vec{b}(n)) = h(\vec{i})]$. Here $h(\vec{b}) = b_1$ and therefore we can write

$$\mu^K(n) = \alpha\mu_0^K(n) + (1-\alpha)\mu_1^K(n) \qquad (9)$$

where $\alpha = \mathbb{P}(z_{K-1}(n) = 0)$ and

$$\begin{aligned}
\mu_0^K(n) &= \mathbb{E}\left[b_1(n+1) \mid b_1(n) = b_1, z_{K-1}(n) = 0\right] \\
&= \mu^{K-1}(n) \\
\mu_1^K(n) &= \mathbb{E}\left[b_1(n+1) \mid b_1(n) = b_1, z_{K-1}(n) = 1\right] \\
&= \mathbb{E}\left[b_1(n+1) \mid b_1(n) = b_1, z_{K-3}(n) = z_{K-2}(n) = 0\right] \\
&= \mu^{K-3}(n)
\end{aligned}$$

where we have used (3) and the independence of $b_i(n+1) - b_i(n)$, $1 \leq i \leq K-3$, from $b_{K-2}(n)$ and $b_{K-1}(n)$, conditionally to $z_{K-3}(n) = z_{K-2}(n) = 0$. Therefore (9) becomes

$$\mu^K(n) = \alpha\mu^{K-1}(n) + (1-\alpha)\mu^{K-3}(n),$$

which implies that $\mu^K(n)$ verifies the inequalities of the Non-ergodicity theorem if $\mu^{K-1}(n)$ and $\mu^{K-3}(n)$ do. ■

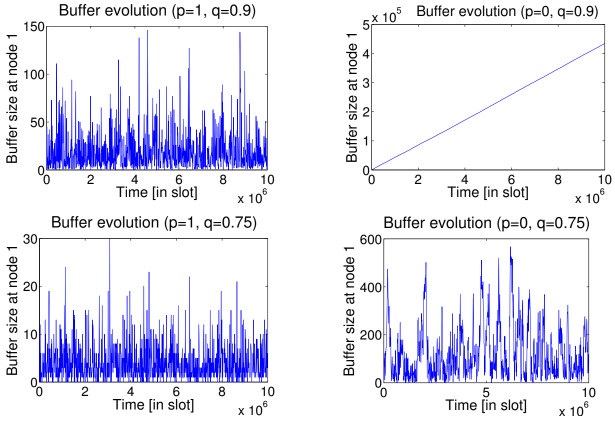

Fig. 8. Buffer evolution for 4-hop with different p and q values.

This structural instability of the MAC layer for linear topologies is of utmost importance for the design of future wireless mesh networks. Indeed, buffer instability means delay and packet drops. These artifacts are undesirable and a mechanism to prevent instability should be designed. Fortunately, some simple *stabilization strategies* are implementable with off-the-shelf hardware, as described in the next section.

VIII. Stabilization strategy

A. Source throttling

In the previous sections, we showed that system instability is caused by the node next to the source (i.e., node 1). The main reason for this problem is that, over the long run, node 0 is given more chance than node 1 to successfully transmit, thus resulting into a buffer build up at node 1.

Our MAC layer solution, called *penalty strategy*, works by reducing the odds of channel access by node 0. This strategy, besides its useful stabilization property, is easily implementable with off-the-shelf hardware. For instance, the odds of channel access for an IEEE 802.11 node can be affected using the minimum contention window parameter (CW_{min}). In particular, the higher CW_{min}, the lower the odds of channel access. Therefore our penalty strategy can be deployed by setting node 0 with a higher CW_{min} than the other nodes.

B. Penalty model

We model the penalty strategy by introducing a *throttling factor* $q \in [0, 1]$ that shows the degree at which the input rate is throttled at the source. That is, $q = 1$ means the input rate is not constrained and then node 0 is not penalized (similar to our previous model). On the opposite, $q = 0$ means node 0 is completely starved and the input rate is null.

A useful analogy to this strategy is a water pipe with the tap aperture being q. The higher the value of q, the higher the inflow into the network, but also the more likely that the flow will be turbulent. Mapping this back to our system, we deduce that there exists a throttling factor threshold q^* below which the system is stable.

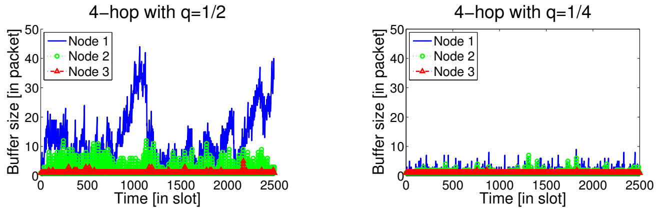

Fig. 9. Experimental results for the buffer evolution for the 4-hop topologies with different throttling factors q (we recall $q = 1$ is unstable).

The impact of the throttling factor is captured in our analytical model by choosing node 0 with probability $\frac{qS_0^m}{qS_0^m+\sum_{j>0}S_j^m}$ whereas node $i \neq 0$ is selected with probability $\frac{S_i^m}{qS_0^m+\sum_{j>0}S_j^m}$.

For instance, consider a K-hop network where at a given time slot only nodes 0 and 1 have packets to transmit: then node 0 wins with probability $q/(1+q)$ and node 1 wins with complementary probability $1/(1+q)$.

C. Theoretical analysis

We next analyze 3-hop and 4-hop networks, implementing the penalization strategy.

Starting with the previously unstable 3-hop case ($p = 0$), we prove the stabilization effect of the throttling factor and the existence of the threshold $q^* = 1$ by applying a similar methodology than in Section V.B. That is, using the function

$$h(b_1, b_2) = b_1^2 + b_2^2,$$

and the finite set

$$F = \left\{0 \le b_1 < \frac{3+q}{2(1-q)}\right\} \bigcap \left\{0 \le b_2 < \frac{1+q}{2}\right\},$$

we verify Foster's theorem ([14], p. 29) and therefore prove the stability of the 3-hop case for all $q < 1$.

Next, we consider 4-hop topologies. We observe that the stealing effect has an important impact on stability. Indeed, as depicted on Figure 8, the threshold q^* is highly dependent on the stealing effect probability p. The higher p, the higher q^*.

We derive theoretical bounds on q^* as follows. First, we obtain a lower-bound through a stability argument. The proof is obtained by a generalized version of Foster's theorem (Theorem 2.2.2, [14]) that uses k-step expectation instead of 1-step expectation. We apply it with the function $h = b_1+b_2+b_3$, taking $k = 18$.

The upper-bound is similarly obtained by the non-ergodicity theorem ([14], p. 30) with the function $h = b_1 + \frac{p}{1+p}b_3$ and the parameter $k = 15$.

Following this procedure for $p = 0$, we obtain $0.37 < q^* \le 0.884$. Similarly, when $p = 1$, we find $0.76 < q^* \le 0.964$. It should be noticed that the bounds on q^* could be made tighter by either investigating different Lyapunov functions or by increasing the value k (i.e., increasing the number of steps of the k-steps expectation at the expense of higher computational cost). However the current bounds already validate some results of Figure 8. For instance, $q = 0.75 < 0.76$ is stable for $p = 1$ and $q = 0.9 > 0.884$ is unstable for $p = 0$.

	Mean throughput	Standard deviation
3-hop with $q = 1$	240.5 kb/s	15.8
4-hop with $q = 1$	146.0 kb/s	16.7
4-hop with $q = 1/2$	157.5 kb/s	16.3
4-hop with $q = 1/4$	141.5 kb/s	14.5
4-hop with $q = 1/8$	143.3 kb/s	17.4
4-hop with $q = 1/16$	108.7 kb/s	11.8
4-hop with $q = 1/32$	96.9 kb/s	11.1
4-hop with $q = 1/64$	81.8 kb/s	13.4

TABLE II

MEAN THROUGHPUT AND STANDARD DEVIATION FOR NETWORKS OF DIFFERENT SIZES AND FOR DIFFERENT THROTTLING FACTORS q.

D. Experimental analysis

We test our stabilization strategy on our experimental testbed described in Section III. We implement the factor q in our testbed by setting a different value of CW_{min} at the WAP (Node 0) than at the other nodes. Thus, to implement $q = 1/2$, we use $CW_{min} = 2^6$ for the WAP and $CW_{min} = 2^5$ for all the other nodes. Note that our hardware forced us to set the contention window as a power of 2. The results of Figure 9 confirm experimentally the efficiency of the throttling factor in stabilizing the network. Indeed, a value of $q = 1/2$ suffices to change the network behavior from unstable to stable.

Table II presents mean throughput statistics for different network sizes and values of q. The results are obtained by running experiments for 400 s and measuring throughput over non-overlapping 5 s intervals. Each entry in the table thus represents an average of 80 samples. Corresponding standard deviations are also provided. We make the following observations. First, there is a significant throughput gap between 3-hop and 4-hop networks, which can be attributed to the distributed nature of 802.11 that does not always make best spatial reuse of the channel. Secondly, for the case of 4-hop networks, the mean throughput decreases with q, but only when q becomes very small (i.e., $q = 1/16$ or smaller). Up to that point, the overhead caused by using a larger contention window for the WAP is not too significant.

IX. SIMULATIONS ON MULTI-FLOWS TOPOLOGIES

Up to this point in the paper, we focused on single flow linear topologies as they are the building block of more general mesh topologies. However, to show that the stability problem also arises in more complex topologies, we present in this section the simulation results obtained with the ns-2 simulator.

We analyze the two multi-flows topologies depicted in Figure 10, where scenario 1 sees two concurrent flows and scenario 2 sees three. Furthermore, we set the simulator to use the standard parameters of 802.11 ad-hoc networks (RTS/CTS disabled, Tx range: 250 m, Cs range: 550 m) and let the simulations run for respectively 1100 s and 1600 s.

The two performance metrics we focus on are: (i) end-to-end delay (low delays means that the network is stable, whereas high delay is a symptom of saturated buffers) and (ii) throughput. Figure 11 ignores the first 100 s of the simulation and shows the average performance achieved by the network as a function of the throttling factor q. We compute the

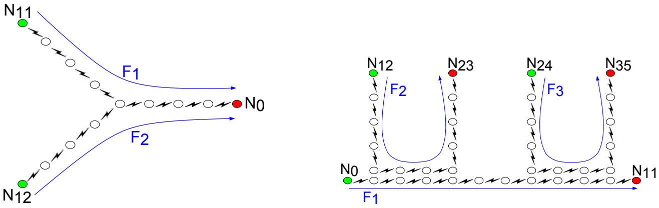

Fig. 10. Illustration of the two topologies tested by simulation.

throughput by measuring the average on disjoint 50 seconds intervals, thus obtaining 20 (30) measurement points. Then we plot the median value with the 95%-confidence intervals. We note that for both scenarios standard 802.11 (i.e. $q = 1$) performs poorly as expected, with high variance in throughput and high end-to-end delays. Furthermore, using an appropriate throttling factor that is larger than for the single-flow case (here $q = 1/128$), performance are significantly improved by achieving both negligible delay and higher global throughput due to a lower packet loss rate (as no buffer overflows in stable regime).

X. Conclusion

In this paper, we address the problem of network stability in CSMA-based linear wireless mesh network and provide three main contributions. First, we identify two key factors impacting the stability: the network size and an artifact that we called stealing effect. Second, we prove analytically and show experimentally that 3-hop networks are stable when we account for the stealing effect, but 4-hop networks (and presumably larger topologies) are not. Third, we devise and prove the effectiveness of a stabilization strategy that throttles the source at the MAC layer, preventing packets to be injected too quickly into the network. We note that this strategy penalizes only the source when other nodes also have packets to transmit. This desirable property allows to ensure both stability and high throughput. Our analysis and experiments have shown that selecting a value CW_{min} four to eight times larger at the source than at the relay nodes (i.e., $q = 1/4$ or $q = 1/8$) in the 4-hop case effectively achieves these goals. We also show that multi-flow networks can also be stabilized by using higher values of q. We will thus study in future work techniques to adapt q according to the network environment.

Acknowledgment

This work is supported in part by Deutsche Telekom Laboratories, in the framework of the Magnets project.

References

[1] *Iperf: The TCP/UDP bandwidth measurement tool.* available at http://dast.nlanr.net/Projects/Iperf/.

[2] *LUNAR- Lightweight Underlay Network Ad hoc Routing project.* available at http://cn.cs.unibas.ch/projects/lunar/.

[3] *OpenWRT firmware.* available at http://openwrt.org/.

[4] *IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Aug. 1999.

[5] *Madwifi/AtherosWireless Linux Driver Users Guide*, Nov. 2005.

[6] A. Aziz, R. Karrer, and P. Thiran. Effect of 802.11 adaptive exponential backoffs on the fluidity of downlink flows in mesh networks. In *Proc. of WinMee*, Berlin, Germany, Mar. 2008.

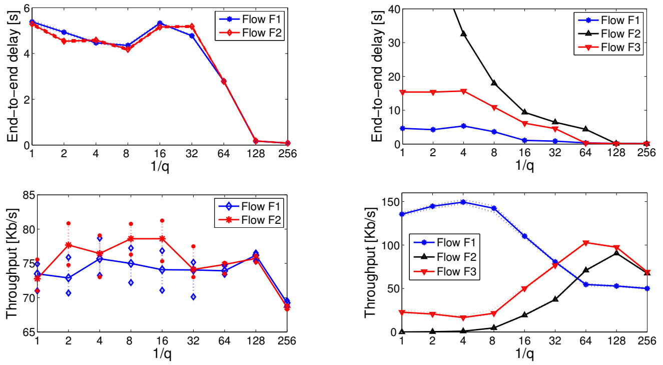

Fig. 11. Illustration of the evolution of the delay and the median of the averaged throughput (with confidence interval) depending on the throttling factor q for scenario 1 (left) and scenario 2 (right). We note that in both scenatios the value $q = 1/128$ stabilizes the network (i.e. low delay), while achieving a good throughput performance.

[7] S. Borst, M. Jonckheere, and L. Leskela. Stability of parallel queueing systems with coupled rates. *to appear in Journal of Discrete Events and Dynamic Systems.*

[8] J. Camp and E. Knightly. the ieee 802.11s extended service set mesh networking standard. *to appear in IEEE Communications Magazine.*

[9] P. Chapokar, H. Kar, X. Luo, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Transactions on Information Theory*, 54(2):572–594, Feb. 2008.

[10] D. Denteneer, S. Borst, P. van de Ven, and G. Hiertz. Ieee 802.11s and the philosophers' problem. *Statistica Neerlandica*, 62(3):283–298, 2008.

[11] O. Dousse. Revising buffering in CSMA/CA wireless multihop networks. In *Proc. of SECON*, San Diego, June 2007.

[12] M. Durvy and P. Thiran. A packing approach to compare slotted and non-slotted medium access control. In *Proc. of INFOCOM*, Barcelona, Spain, Apr. 2006.

[13] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *Proc. of INFOCOM*, Mar. 2005.

[14] G. Fayolle, V. A. Malyshev, and M. V. Menshikov. *Topics in constructive theory of countable Markov chains*. Cambridge University Press, 1995.

[15] M. Garetto, T. Salonidis, and E. W. Knightly. Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks. *IEEE/ACM Transactions on Networking*, 16(4):864–877, 2008.

[16] M. Gerla and L. Kleinrock. Flow control: A comparative survey. *IEEE Transactions on Communications*, 28(4):553–574, Apr. 1980.

[17] A. Gupta, X. Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. In *Proc. of INFOCOM*, Anchorage, Alaska, May 2007.

[18] W. Luo and A. Ephremides. Stability of n interacting queues in random-access systems. *IEEE Transactions on Information Theory*, 45(5):1579–1587, July 1999.

[19] P. Momcilovic and M. Squillante. On throughput in linear wireless networks. In *Proc. of Mobihoc*, Hong Kong, May 2008.

[20] P. C. Ng and S. C. Liew. Throughput analysis of IEEE 802.11 multi-hop ad hoc networks. *IEEE/ACM Trans. on Net.*, 15(2):309–322, Apr. 2007.

[21] S. Rangwala, A. Jindal, K.-Y. Jang, and K. Psounis. Understanding congestion control in multi-hop wireless mesh networks. In *Proc. of MobiCom*, San Francisco, USA, May 2008.

[22] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. Knightly. Measurement and modeling of the origins of starvation in congestion controlled mesh networks. In *Proc. of INFOCOM*, Apr. 2008.

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

[24] Y. Yi, A. Proutiere, and M. Chiang. Complexity in wireless scheduling: Impact and tradeoffs. In *Proc. of MobiHoc*, Hong Kong, May 2008.

[25] Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. *IEEE Trans. on Net.*, 15(1):133–144, Feb. 2007.

[26] L. Ying, R. Srikant, and D. Towsley. Cluster-based back-pressure routing algorithm. In *Proc. of INFOCOM*, Apr. 2008.