Universal Routing in Multi Hop Radio Networks

Bogdan S. Chlebus [*]
Department of Computer
Science and Engineering
University of Colorado Denver
Denver, Colorado, USA

Vicent Cholvi [†]
Department of Computer
Science
Universitat Jaume I
Castellón, Spain

Dariusz R. Kowalski [‡]
Department of Computer
Science
University of Liverpool
Liverpool, UK

ABSTRACT

We study dynamic routing in multi-hop radio networks in a specialized framework of adversarial queuing. We consider cross-layer interactions of the following three components of routing protocols: transmission policies for medium-access control, scheduling policies on the network layer, and hearing-control mechanisms through which transmissions interact with a scheduler.

We propose a model of adversarial queuing in radio networks in which transmission policies are delegated to oracles and adversaries control packet injection. For such a setting, we propose a definition of universal stability that takes into account not only how packets are injected, as in the wireline adversarial model, but also how transmission policies behave.

We investigate which scheduling policies are universally stable, depending on hearing control, and settle this question for many popular scheduling policies.

Keywords

Radio network; packet routing; adversarial queuing; universal stability; scheduling; transmission policy; hearing control

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication—*Radio networks*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Algorithms,Theory

1. INTRODUCTION

Wireless data communication involves multiple technologies interacting with each other. In order to study communication algorithms for wireless networks, one needs models that abstract from incidental details and capture the essential aspects of wireless networking.

A node of a wireless network can transmit messages within its transmission range. Such a range is determined by the power of the transmitting device and the surrounding topography. A possible approach to model this is through geometric wireless networks in which ranges are determined by distances assigned to nodes. A popular special case of geometric networks has all the ranges equal, so that the topologies of such networks are unit-disk graphs; see [18, 27, 32]. Another alternative is a signal-to-interference-plus-noise ratio (SINR) model which incorporates interference and noise explicitly in determining ranges of reliable transmissions. In a SINR setting, a transmission is successful when a suitable ratio of "good" to "bad" components of a received signal is above a threshold; see [16, 22, 23, 33].

Radio data networks model wireless communication in which just one channel is used for transmissions. Signal receptions at a node that overlap in time interfere with one another, so that none can be received successfully. The feasibility of a node-to-node direct transmissions determines how nodes can reach each other; reachability so defined is a relation on the nodes of a network. This suggests modeling the topology of a multi-hop network as an arbitrary connected graph where edges represent direct reachability; see [12, 13].

The model used in this paper abstract from geometrical constraints imposed on ranges of transmissions and represent the relation of reachability as a general graph. We consider simple graphs, that is, with symmetric bi-directional edges; see [25]. This is to make a dialog feasible between pairs of nodes the can communicate by direct transmissions. In particular, when a node receives a transmission then sending back a response is possible in principle.

Dynamic store-and-forward routing in wireless networks differs from the respective routing in wireline networks. In wireline networks, a scheduling policy, that manages queues of packets at nodes, is the only essential component of source routing when packets have their routes determined from the point of injection. This is because a node can transmit a packet per round over any outgoing link, and simultaneously accept a packet per round over any incoming link, these packets coming and going simultaneously. In wireless networks, coordinating timings of transmissions among nodes, with the goal to avoid collisions resulting from receiv-

[*] Supported by the NSF Grant 1016847.

[†] Supported by the CICYT Grant TIN2011-28347-C02-01.

[‡] Supported by the EPSRC Grant EP/G023018/1.

FOMC'14, August 11, 2014, Philadelphia, PA, USA.
Copyright 2014 ACM 978-1-4503-2984-2/14/08 ...$15.00.
http://dx.doi.org/10.1145/2634274.2634278.

ing overlapping transmissions, has the potential to improve performance of routing. Such coordination is handled by a transmission policy, which is another essential component of a routing protocol. These two policies need to cooperate with each other, which is delegated to a hearing control mechanism.

The structuring of routing protocols we consider can be related to the OSI model. Protocol stack separates subtasks of a communication algorithm into layers of their individual functionalities. Coordinating message transmissions among nodes has the purpose to avoid collisions of messages, so it can be considered to belong to the medium-access control sublayer of the data-link layer. A mechanism of exchanging control messages to provide hearing control belongs to the logical-link control sublayer of the data-link layer. A scheduling policy can be considered as operating on the network layer.

Cross-layer approach proposes to relax these functionality specifications to enhance efficiency. Such a relaxation is accomplished by providing additional interactions between layers. See [19, 21, 29, 31] for more on the motivation and guidelines in developing cross-layered algorithms.

Our results.

We study dynamic routing in multi-hop radio networks in the framework of adversarial queuing. We consider cross-layer interactions of the following three components of routing protocols: transmission policies for medium-access control, scheduling policies on the network layer, and hearing-control mechanisms through which transmissions interact with a scheduler.

We propose a model of adversarial queuing in radio networks in which transmission policies are delegated to oracles and adversaries control packet injection. For such a setting, we propose a definition of universal stability that takes into account not only how packets are injected, as in the standard wireline adversarial model, but also how transmission policies are integrated with scheduling.

We investigate which scheduling policies are universally stable, depending on hearing control, and settle this question for many popular scheduling policies.

Previous work.

Routing in radio networks was considered as early as in Gitman et al. [20]; for recent contributions see Kuhn et al. [26, 28]. For surveys of routing in wireless networks see Andrews [4], Rajaraman [30], and Urrutia [32]. In particular, paper [4] discusses methods of assigning wireless resources under scenarios when the rates at which users can receive data are time-varying and user depending, due to channel fading and user mobility. A difference between our radio adversarial model and the one addressed by Andrews in [4] is that we incorporate collisions into the model while they assume that a node can transmit to only one neighbor in a round. Adversarial queuing in general wireless networks was studied by Andrews and Zhang [7, 8], and by Cholvi and Kowalski [15]. Wang et al. [34] proposed an analytical model to study interplay between medium access control and packet routing disciplines.

Related work.

Adversarial queuing in wireline networks, as a methodology to study stability in the worst case abstracting from stochastic assumptions on traffic generation, was initiated by Borodin et al. [10] and Andrews et al. [5]. They also introduced the notions of universal stability of protocols and networks. A systematic account of issues related to universal stability in adversarial routing was given by Àlvarez et al. [2]. Extensions of adversarial queuing to address other aspects of networks were proposed in the following works. Àlvarez et al. [1] considered routing when packets have priorities. Networks with links that occasionally fail were studied by Àlvarez et al. [3]. Networks with bandwidth and delay parameters associated with links were considered in Blesa et al. [9] and Borodin et al. [11]; such behavior of networks can be considered as capturing some properties of wireless networks. Aspects of adversarial queuing in wireline networks were addressed by Andrews et al. [6], Cholvi and Echagüe [14], and Echagüe et al. [17].

2. TECHNICAL PRELIMINARIES

A network is modeled as a simple graph $G = (V, E)$, where V is the set of nodes and E is a set of edges. A node of the graph represents a transceiver that can act both as a sender and as a receiver. An edge (u, w) represents the property that the nodes u and w can transmit directly to each other, in the sense that there are two independent directed links from u to w and from w to u available for direct transmissions.

There are n nodes in the network. Each node is assigned a unique name, which is an integer in $[0, n-1]$. Every node knows n and its own name, in the sense that they can be used as a part of code of protocols.

Nodes have access to local clocks ticking at the same rate. Time is divided into time intervals of fixed length that we call *rounds*. Local computations at a node are considered to be of negligible duration.

Messages.

The contents of transmissions are structured into chunks of data that we call *messages*. Messages are of two kinds: packets and control messages. A *packet* carries a header that includes a destination address followed by this packet's contents. A *control message* carries a string of bits used to coordinate actions among the nodes. Control messages are significantly shorter than packet ones. We assume that transmitting a control message takes an insignificant amount of time compared to what is needed to transmit a packet of data. Rounds are scaled to the amount of time it takes to transmit a message with a packet. Packets and control messages are interleaved in executions of routing protocols.

Data transmissions.

A node may transmit exactly one message in a round or pause in this round. A message received successfully by a node is said to be *heard* by that node.

Radio networks are defined by the following two properties. First, when two messages arrive at a node v transmitted by its neighbor such that their receipt overlaps in time then they interfere with each other and none can be heard by v. Second, when only one neighbor of a node v transmits a message then v hears this message.

Radio networks are *single port* in the sense that a node can transmit at most one message in a round and hear at most one message in a round.

Routing.

A routing algorithm manages how packets traverse their respective assigned paths. We consider *distributed* routing algorithms, in which each node runs its code independently. Packets to be forwarded to a node's neighbors may need to wait for their turn to be transmitted. Each node contains a buffer space to temporarily store data to be transmitted in the future, which is organized as a queue. Each node maintains a single queue, rather than a dedicated queue for each outgoing link, as in the wireline adversarial queuing model. Protocols we consider operate under the principle that no packet is discarded until its delivery to the destination. To make this meaningful, we assume that the buffer space at a node can store an arbitrarily large number of packets, although we want to keep bounded queues at nodes.

Packets get injected into nodes to be delivered to their respective destination nodes by traversing paths. We consider *source routing* in which the entire path of a packet is known at the source where the packet is injected. When a packet traverses a link from v to w on such a path, by way of v transmitting the packet and w hearing it, then w is the *intended recipient* of the packet transmitted by v. A packet message carries a header which includes the packet's itinerary. This allows for a node that hears a packet message to decide how to process the message. The following are three cases of what occurs when a node v hears a message. One case is when v is not the intended recipient of the message: then v discards the message. The other occurs when v is the final destination for the message: then v consumes the message. The final case is when v is the message's recipient but not its final destination: then v enqueues the message to wait to be forwarded to a neighbor.

Hearing control.

What we call *hearing control* is a mechanism that coordinates scheduling with transmissions. One possibility is as follows: when a node wants to transmit in a round, it first obtains a list of neighbors that will hear the message in the round. For 802.11 networks, this can be realized by using the Carrier-Sense Multiple Access (CSMA) protocol, enhanced with a Request-to-Send/Clear-to-Send (RTS/CTS) mechanism [24]. An alternative to this hearing control is considered in Section 5.

Scheduling policies.

A scheduling policy is understood as a rule to select a packet from a group of packets at a node. It is coordinated with a hearing control. A scheduling policy selects a packet to transmit from these parked in the queue for whom one of the neighbors that will hear a message is their next stop on their path to a destination node. When the packet is eventually heard by the corresponding neighbor, then the sending node removes this packet from its queue.

Popular queueing policies include Furthest-From-Source (FFS), Furthest-To-Go (FTG), Nearest-To-Source (NTS), Nearest-To-Go (NTG), Shortest-In-System (SIS), and Longest-In-System (LIS).

Transmissions.

Some transmissions, during an execution of a routing protocol in a radio network, may not be successful, in transferring packets to their intended recipients, due to collisions. To mitigate this adverse effect, some nodes may be directed by a routing protocol to pause in a round to create more room in the only transmitting frequency available for the neighbors to possibly transmit and hear successfully.

A *transmission policy* indicates to a node whether to transmit in a round or rather to pause. Such an indication is either in the affirmative, meaning "do transmit in this round," or in the negative, meaning "do not transmit in this round." The ultimate goal of a transmission policy is to facilitate packet movement by avoiding collisions of packets transmitted by different neighbors of nodes.

We abstract from implementing transmission policies, but delegate such a task to a *transmission oracle* (or simply *oracle*) that will indicate each node whether or not to transmit in a round. This approach allows to abstract from specific transmission policies to consider qualities of scheduling policies independently of transmission policies.

The relevant desirable functionality such transmission oracles needs to provide is that each link can successfully transmit, when there is a packet ready to be transmitted, within a bounded number of rounds. We say that an integer $h_e > 0$ is a *hearing latency of link* e if link e is guaranteed to be able to successfully transmit at least one packet each h_e consecutive rounds; if such a number does not exist then e is said to have an *unbounded link hearing latency*, denoted $h_e = \infty$. We say that a transmission oracle *provides a link latency* h when hearing latencies of all links are upper bounded by the number h. We denote by $\mathcal{T}_h$ the class of transmission oracles that provide a link hearing latency of h.

Adversarial packet injection and stability.

Packets are injected by adversaries. An adversary is determined by a pair of numbers (b, r), called the *type* of the adversary, where *burstiness* b is a positive integer and *injection rate* r satisfies $0 \le r < 1$. We denote by $\mathcal{A}(b, r)$ an adversary of type (b, r). Such an adversary specifies for each injected packet its complete itinerary. Let $I(\tau, v)$ represent the number of packets that the adversary injects during time interval τ and has node v on its path. Adversary $\mathcal{A}(b, r)$ is constrained such that the inequality

$$I(\tau, v) \le r \cdot \mid \tau \mid + b$$

holds for any τ and v. When traffic demands are constrained this way, then we say that they are *admissible for rate* r *and burstiness* b.

A routing protocol based on a transmission oracle $\mathcal{T}$ and a scheduling policy $\mathcal{S}$ is denoted by $P(\mathcal{T}, \mathcal{S})$. Let there be given a routing protocol $P(\mathcal{T}, \mathcal{S})$ and adversary $\mathcal{A}$, and let $\mathcal{D}$ be an execution of protocol $P(\mathcal{T}, \mathcal{S})$ against $\mathcal{A}$ in a network G. For a positive integer t, let $Q_{\mathcal{D}}(t)$ be the number of packets simultaneously queued in all the nodes in round t of $\mathcal{D}$. An execution $\mathcal{D}$ of a routing protocol is *stable* when the numbers $Q_{\mathcal{D}}(t)$ are all bounded. Protocol $P(\mathcal{T}, \mathcal{S})$ is *stable against adversary* $\mathcal{A}$ if each execution of $P(\mathcal{T}, \mathcal{S})$ against $\mathcal{A}$ in any network G is stable. Finally, $\mathcal{S}$ with $\mathcal{T}_h$ is *stable against adversary* $\mathcal{A}$ if for any $\mathcal{T} \in \mathcal{T}_h$, protocol $P(\mathcal{T}, \mathcal{S})$ is stable against adversary $\mathcal{A}$.

Universal stability.

Given any transmission providing a link hearing latency h, the maximum injection rate one could expect to guarantee stability is $1/h$. Otherwise, instability can be created just by injecting packets passing through a link whose hearing latency is exactly h, at a rate higher than $1/h$. We say that $\mathcal{S}$

with $\mathcal{T}_h$ is *universally stable* when it is stable for any adversarial injecting rate that is smaller than $1/h$. A scheduling policy that is not universally stable is called *unstable.*

3. STABILITY OF SCHEDULING POLICIES

In this section we consider universal stability of scheduling in radio networks when using various scheduling policies. We say that a packet *leaves* a node v when it is successfully transmitted to the intended neighbor. We also say that *packet p has priority over packet q* if the policy used to assign priorities chooses p over q.

3.1 Shortest-In-System (SIS)

The *Shortest-In-System* (SIS) scheduling policy gives priority to the packet that has been in the system the shortest, with ties broken in an arbitrary manner. The adversarial injection rate r is always lower than $1/h$, because h is an upper bound on the hearing latencies of all links.

Lemma 1. *Consider SIS with a transmission policy in $\mathcal{T}_h$. For a node v and a packet p in its queue, v will transmit at least one packet with priority higher than that of p during any h rounds during the time interval form p's arrival to v until p is transmitted.*

Proof. Let e be the link through which packet p will be transmitted. Let T be the time interval since packet p arrives to v until it is transmitted. Recall that the scheduling policy chooses a packet to be transmitted from the set of links that are up at a round. So each time link e is up in T, a packet with priority over p will be transmitted; otherwise, packet p will be chosen before T, contradicting our assumption. Since all hearing link latencies are bounded by h, then at least one packet with priority over p will be transmitted each h rounds in T. □

Lemma 2. *Let p be a packet waiting in the queue of a node v at time instant t_0, whose scheduling policy is SIS and whose transmission policy is in $\mathcal{T}_h$. Suppose that at this time there are $k-1$ other packets in the queue of v that have priority over p. Then p will leave v within the next $\frac{k+b}{1-rh}\cdot h$ rounds, where $0 \leq r < 1/h$, and h is an upper bound on the hearing latencies of all links.*

Proof. We argue by contradiction. Suppose that p does not leave v in the next $\frac{(k+b)}{(1-rh)}\cdot h$ rounds. Then other packets different from p must have left the queue meanwhile. Because SIS is the scheduling policy, during that interval the only packets in the system that have priority over p are either those $k-1$ packets that were present at time t_0 or some other that have been injected meanwhile.

By Lemma 1, we have that v will transmit at least one packet with priority over p each h rounds until p is transmitted. Let us first consider a transmission scenario where only one packet is transmitted each h rounds. Since only one packet is guaranteed to be transmitted in each interval of h rounds, we have that the following two properties hold:

(1) the $k-1$ packets currently in the system will take $(k-1)\cdot h$ rounds to leave v, and

(2) the packets injected in the next $\frac{k+b}{1-rh}\cdot h$ rounds, which are $rh\cdot(\frac{k+b}{1-rh})+b$, will take at most $(rh\cdot(\frac{k+b}{1-rh})+b)\cdot h$ rounds to leave v.

Summing up, we obtain that the number of rounds p waits is at most

$$\big(k-1+rh\cdot\frac{k+b}{1-rh}+b\big)\cdot h$$
$$=\Big(\frac{k-1-rhk+rh+rhk+rhb+b-bhr}{1-rh}\Big)\cdot h\ ,$$

which is less than $\frac{k+b}{1-rh}\cdot h$. This results in a contradiction. We estimated the number of rounds assuming that only one packet is transmitted per interval of h rounds. If more than one packet is transmitted per a time interval of h rounds, then this decreases the relative number of rounds, so that the bound $\frac{k+b}{1-rh}\cdot h$ remains valid. □

Lemma 3. *Suppose SIS is the scheduling policy with ties broken arbitrarily. Define $k_1 = b$ and $k_{i+1} = \frac{k_i+b}{1-rh}$. When a packet p arrives at the ith queue v_i on its path then there are at most $k_i - 1$ packets requiring any queue in the path of p with a priority higher than that of p.*

Proof. The proof is by induction on i. Observe that, for any queue v, the only packets passing through v that initially could have priority higher than that of p are at most $b-1$ packets injected in the same round as p, which provides the base of induction. To show the inductive step, suppose that the claim holds for some i. By Lemma 2, p will arrive at the tail of v_{i+1} in at most another $\frac{k_i+b}{1-rh}\cdot h$ rounds, during which at most $rh\cdot(\frac{k_i+b}{1-rh})+b$ other packets requiring any queue v in the path of p with priority over p are injected. Thus, when p arrives at the tail of v_{i+1} the number of packets requiring any queue v that have priority higher that that of p is at most

$$k_i - 1 + rh\frac{k_i+b}{1-rh}+b$$
$$=\frac{k_i-1-rhk_i+rh+rhk_i+rhb+b-brh}{1-rh}$$
$$=\frac{k_i+b}{1-rh}+\frac{rh-1}{1-rh}$$
$$=k_{i+1}-1\ ,$$

so the claim holds. □

Theorem 1. *SIS with $\mathcal{T}_h$ is universally stable. No queue contains more than k_d packets, where d denotes the length of the longest simple directed path in the graph. No packet spends more than $\sum_{i=1}^{d}(\frac{k_i+b}{1-rh})\cdot h$ rounds in the system.*

Proof. We show first that no queue contains more than k_d packets. Let us assume that there are k_d+1 packets at some point all passing through the same queue. By Lemma 3, the packet with the lowest priority will contradict the property that no queue contains more than k_i-1 packets with priority above it. Therefore, the overall number of packets is bounded and therefore the system is stable. Furthermore, combining Lemma 3 with Lemma 2, we obtain that no packet spends more than $\sum_{i=1}^{d}(\frac{k_i+b}{1-rh})\cdot h$ rounds in the system. □

3.2 Longest-In-System (LIS)

The *Longest-In-System* (LIS) scheduling policy gives priority to a packet that has been longest in the system, with ties broken in an arbitrary manner. We will assume throughout that the adversarial injection rate r is smaller than $1/h$, where h is an upper bound on the hearing latencies of all links. Let us consider a system whose scheduling policy is LIS and whose transmission policy is in $\mathcal{T}_h$. For a round c, we denote by *class* c the set of packets injected at round c. A class c is said to be *active* at the end of round t if and only if at that round there is some packet in the system of class $c' \leq c$. Consider some packet p, injected at time T_0, and whose path contains queues $v_1, v_2, ..., v_d$, in this order. We denote by T_i the round in which p leaves v_i, and by t some round in $[T_0, T_d)$. Let a_t denote the number of active classes at the end of round t, and define $a = \max_{t \in [T_0,T_d)} a_t$. In such a situation, we will say that p has a active classes while in the system.

Lemma 4. *The inequality* $T_d - T_0 \leq \frac{(r \cdot a+b) \cdot h \cdot (d-1)}{1+r \cdot h \cdot (d-1)}$ *holds.*

Proof. Packet p reaches the tail of queue v_i at time T_i. Since p is still in the system at round T_i, all classes formed by packets injected in $[T_0, T_{i-1}]$ are active at the end of that round. From the definition of a, there are at most $a - (T_{i-1} - T_0)$ active classes of packets that can block p in the queue of v_i. As LIS is the scheduling policy, packets injected after p can not block it, because they are in classes after the class of p.

Observe that all active classes are consecutive. Indeed, if a class is active then all the subsequent classes are active; so, take the lowest active and all the subsequent classes will be also active.

There are at most $r \cdot (a - T_{i-1} + T_0) + b$ packet in these classes. And since p is one of these packets, at most

$$r \cdot (a - T_{i-1} + T_0) + b - 1$$

packets can block p. Therefore, since h is a bound on the queue latency, we have the following estimates:

$$\begin{aligned} T_i &\leq T_{i-1} + (r \cdot (a - T_{i-1} + T_0) + b) \cdot h \\ &\leq T_{i-1}(1 - r \cdot h) + (r \cdot (a + T_0) + b) \cdot h \\ &\leq T_{i-1} + (r \cdot (a + T_0) + b) \cdot h. \end{aligned}$$

Solving the recurrence results in the following estimate:

$$\begin{aligned} T_d &\leq (r \cdot (a + T_0) + b) \cdot h) \cdot (d-1) + T_0 \\ &= (r \cdot a + b) \cdot h \cdot (d-1) + T_0 \cdot (1 + r \cdot h \cdot (d-1)). \end{aligned}$$

We conclude with this inequality: $T_d - T_0 \leq \frac{(r \cdot a+b) \cdot h \cdot (d-1)}{1+r \cdot h \cdot (d-1)}$. □

Theorem 2. *LIS with* $\mathcal{T}_h$ *is universally stable. No queue contains more than* $r \cdot ((b+r) \cdot h \cdot (d-1) + 1) + b$ *packets. No packet spends more than* $(b+r) \cdot h \cdot (d-1) + 1$ *rounds in the system.*

Proof. We show that there are always at most

$$(b+r) \cdot h \cdot (d-1) + 1$$

active classes in the system, where d is the length of the longest simple directed path. Let $a = (b+r) \cdot h \cdot (d-1) + 1$ and assume that the end of round t is the first where there are exactly $a+1$ active classes. We show next how to arrive at a contradiction. At the end of a round a, there are packets that have been in the system for $a+1$ rounds, and during the first a of these rounds no more than a classes were active. From Lemma 4, any packet that has at most a active classes while in the system, with a possible exception of the last round, reaches its final destination in a number of rounds that is at most as large as the following estimate:

$$\begin{aligned} &\frac{(r \cdot a + b) \cdot h \cdot (d-1)}{1 + r \cdot h \cdot (d-1)} + 1 \\ &= \frac{(r \cdot ((b+r) \cdot h \cdot (d-1) + 1) + b) \cdot h \cdot (d-1)}{1 + r \cdot h \cdot (d-1)} + 1 \\ &= (b+r) \cdot h \cdot (d-1) + 1 \ . \end{aligned}$$

This bound is less than $a+1$, which yields a contradiction. □

4. INSTABILITY OF SCHEDULING POLICIES

In order to show that a given scheduling policy $\mathcal{S}$ is unstable with a transmission policy $\mathcal{T}$, we need to prove that there is an unstable execution of protocol $P(\mathcal{T}, \mathcal{S})$ against some adversary.

We begin by introducing a model of wireless networks in which a node can hear messages transmitted by its neighbors while two or more of them transmit concurrently. A node can still transmit only one message per round. In this model, we consider a transmission policy which has every node transmit in each round; this transmission policy is denoted $\mathcal{T}^{greedy}$. This transmission policy provides a node latency of one round. It is the only transmission policy considered for this model, and the only object of consideration when we depart from the radio model. In what follows, whenever we refer to $\mathcal{T}^{greedy}$, we mean this particular model.

Given a wired network G we define its *equivalent network* G^* as follows:

(1) For each link e in G, and so for each queue in G, create a node in G^* containing a single queue, denoted e^*.

(2) For each pair of links $e = (-, u)$ and $f = (u, -)$ in G, connect e^* to f^* in G^*.

We may observe that any simple path in G, with respect to queues, can be created in G^* by replacing each queue e by e^*. The queues e and e^* are called *equivalent*, and the path followed by p^* is called the *equivalent path* of the one followed by p.

Given an adversary $\mathcal{A}$, we define $\mathcal{A}^*$ as follows: for each packet p injected by $\mathcal{A}$ at some round, $\mathcal{A}^*$ injects another packet p^* at the same round following the equivalent path of the one followed by p. Packet p^* may be absorbed at any node pointed by its last traversed queue.

Let $\mathcal{D}$ be an arbitrary execution in system $(G, \mathcal{A}, \mathcal{S})$, and let $\mathcal{D}^*$ denote the execution of $P(\mathcal{T}^{greedy}, \mathcal{S})$ against adversary $\mathcal{A}^*$ in network G^*.

Lemma 5. *The rounds when* p *is in* e *in* $\mathcal{D}$ *are the same as the rounds when* p^* *is in* e^* *in* $\mathcal{D}^*$.

Proof. For each queue in G there is an equivalent one in G^*, so that two queues in G^* are connected provided their equivalent ones are also connected in G. Packets follow the original path in G and its equivalent one in G^*. Greedy transmission policies are used in each case. □

Lemma 6. *If* $\mathcal{S}$ *is unstable in wired networks then* $\mathcal{S}$ *with* $\mathcal{T}^{greedy}$ *is also unstable.*

Proof. Let $\mathcal{D}$ be an arbitrary execution in $(G, \mathcal{A}, \mathcal{S})$ in wired networks. By Lemma 5, there is an execution $\mathcal{D}^*$ of $P(\mathcal{T}^{greedy}, \mathcal{S})$ against adversary $\mathcal{A}^*$ in network G^*, such that the rounds when each packet p is in e in $\mathcal{D}$ are the same as the rounds when p^* is in e^* in $\mathcal{D}^*$. Therefore, if $\mathcal{D}$ is unstable then $\mathcal{D}^*$ is such as well. □

By Lemma 6 and by the instabilities shown in [5, 10], we can conclude that FIFO, NTG, FFS and LIFO are all unstable with $\mathcal{T}^{greedy}$.

4.1 Instability with a round-robin transmission policy

Let us consider a set of transmissions policies $\mathcal{T}^{cyclic}$ where nodes are indicated to transmit using a token traveling along a logical ring that includes the whole set of nodes. When a node obtains the token, it is eligible to transmit for a fixed number of rounds, which may vary from node to node. For each node, we define its *speed* as the number of rounds when such a node is indicated to transmit, once it obtains the token.

For a node v, let us call *turns* of v the rounds when v receives the token but not the subsequent rounds when it may be eligible to transmit. Given a packet p arriving to node v at time t, we say that its *turn* at v is the turn of v immediately after t; if such a turn has not yet occurred, we say that it is not p's turn yet at v.

We define $\mathcal{S}^{turn}$ as the scheduling policy that behaves as $\mathcal{S}$, and with the following additional stipulations:

(1) $\mathcal{S}^{turn}$ chooses packets to be transmitted from the set of packets that already have a turn, and

(2) $\mathcal{S}^{turn}$ treats packets as if they arrived at a given node v exactly at their turns at v.

Let us consider a transmission policy $\mathcal{T}^{one} \in \mathcal{T}^{cyclic}$ where each node has a speed of one round, which is that same as a round-robin policy. Such a transmission policy provides a node hearing latency of n, where n is the number of nodes.

Theorem 3. *If $\mathcal{S}$ with $\mathcal{T}^{greedy}$ is unstable then $\mathcal{S}^{turn}$ with $\mathcal{T}^{one}$ is unstable as well.*

Proof. If we normalize the length of rounds to n, that is, the time interval between two turns, then the behavior of $\mathcal{S}^{turn}$ with $\mathcal{T}^{one}$ is the same as the behavior of $\mathcal{S}$ with $\mathcal{T}^{greedy}$. Now it is sufficient to resort to Lemma 6. □

From Theorem 3 and by the instabilities shown in [5, 10], we conclude the following fact.

Corollary 1. *Scheduling policies FIFO^{turn}, NTG^{turn}, FFS^{turn} and LIFO^{turn} are all unstable with $\mathcal{T}^{one}$.*

4.2 Instability depending on the speed of nodes

Algorithmic queuing was extended by Blesa et al. [9] to a situation when links may have different bandwidths. We refer to this model as continuous adversarial queuing. In this model, the reasoning in the proof of Lemma 6 remains valid, assuming that queues in $\mathcal{T}^{greedy}$ may have different bandwidth. This implies the following fact.

Corollary 2. *If $\mathcal{S}$ is unstable under continuous adversarial queuing then $\mathcal{S}$ with $\mathcal{T}^{greedy}$ is unstable.*

We define *Slowest-Previous-Node* (SPN) as the scheduling policy which gives priority to the packets whose last visited node had the lowest bandwidth, with ties broken according to the Nearest-To-Source (NTS) policy.

Theorem 4. *SPN^{turn} with $\mathcal{T}^{cyclic}$ is unstable.*

Proof. In the continuous adversarial queuing, the scheduling policy SPL gives priority to a packet whose last link through which it has been transmitted had the lowest bandwidth, with ties broken according to NTS. It was shown in [9] that SPL was not stable in continuous adversarial queuing. Therefore, from Corollary 2, we obtain that SPL is unstable with $\mathcal{T}^{greedy}$. When we consider SPL with $\mathcal{T}^{greedy}$, we must take into account that queues are located at links. We can transform queues located at links into queues located at nodes to obtain that SPN is unstable with $\mathcal{T}^{greedy}$. Given a transmission policy $\mathcal{T} \in \mathcal{T}^{cyclic}$, if we normalize the length of the rounds to the smallest speed of any node, then we have that the behavior of the SPN^{turn} with $\mathcal{T}$ is the same as the behavior of SPN with $\mathcal{T}^{greedy}$, assuming that queues in $\mathcal{T}^{greedy}$ could have different bandwidths.

A similar reasoning can be applied, starting from SPN with $\mathcal{T}^{greedy}$ to SPN^{turn} with $\mathcal{T} \in \mathcal{T}^{cyclic}$. We obtain that SPN^{turn} with some $\mathcal{T} \in \mathcal{T}^{cyclic}$ is unstable, where an adversarial transmission policy $\mathcal{T}$ makes $\mathcal{T}^{greedy}$ unstable. □

Andrews et al. [5] showed that NTS is universally stable in wired networks. SPN^{turn} is unstable with $\mathcal{T}^{cyclic}$, while it is universally stable in wired networks. This is because scheduling policy SPN^{turn} behaves as NTS in wired networks, as links have the same bandwidth.

5. AN ALTERNATIVE HEARING CONTROL

Hearing control introduced in Section 2 is such that when a node wants to transmit in a round, it first obtains a list of neighbors that will hear the message in the round. In such an arrangement, a scheduler selects a packet from those parked in the queue that have one of these neighbors on their paths to traverse. This hearing control allows to avoid unexpected collisions, but its implementation requires a handshaking which adds a significant overhead (see for instance [24]).

An alternative approach is to have a hearing control such that immediately after a transmission a mechanism is invoked to detect if the intended recipient node has heard the message. This means that a scheduler learns about the effectiveness of selection after a transmission. The packet is retransmitted until it is eventually heard by the corresponding neighbor. Whereas now transmissions may suffer from collisions, it has the advantage of requiring an overhead much smaller than in the previous case.

To distinguish between these scenarios, we will denote by $\mathcal{S}_{pre}$ the scheduling policy $\mathcal{S}$ working in a setting when, prior to transmitting, a list of neighbors that will hear the message in the round is obtained by a node. We will also denote by $\mathcal{S}_{post}$ the scheduling policy $\mathcal{S}$ working in a setting where it is only detected if the intended recipient node has heard the message.

Many scheduling policies assign priorities based on parameters that can produce two packets with the same priority. Examples of such policies are these who assign priorities based on the injection times into the system (e.g., Shorter-In-System and Longest-In-System), on the traveling path of the packet (e.g., Farthest-To-Go, Nearest-To-Source, Nearest-To-Go and Farthest-From-Source). In order to resolve ties between packets in such a situation, we allow an arbitrary assignment of priorities to packets, which includes the worst-case solution of the tie in terms of system's stability; this results in breaking ties arbitrarily. The next Theorem 5 applies to such scheduling policies.

Theorem 5. *If $\mathcal{S}$ breaks ties arbitrarily then $\mathcal{S}_{pre}$ with $\mathcal{T}_h$ is unstable, regardless of injection rate.*

Proof. Consider a scenario involving three nodes: u, v_1 and v_2, where u is connected to both v_1 and v_2. Let us inject two packets p_1 and p_2 at the same time into node u so that p_1 is addressed to node v_1 and p_2 is addressed to node v_2. Assume that the link (u, v_1) and the link (u, v_2) are up alternately, so that the two links are never both up in the same round. Since both packets are injected at the same time into the same node and have one hop to travel, and since ties are arbitrarily broken, the scheduling policy can choose any one packet at any round, as far as they are both in node u. If the scheduling policy chooses p_2 when link (u, v_1) is up, and p_1 when link (u, v_2) is up, then no packet will be successfully transmitted in any round. □

The next fact demonstrates that, regardless of how ties are broken, the fact that $\mathcal{S}_{post}$ with $\mathcal{T}_h$ is universally stable does not imply that $\mathcal{S}_{pre}$ with $\mathcal{T}_h$ is universally stable.

Theorem 6. *SIS_{pre} with $\mathcal{T}_h$ is unstable against adversary $\mathcal{A}(b, 1/(2h-4))$, regardless of how ties are broken, where $h \geq 4$.*

Proof. We consider a network topology in which there is a node with two outgoing links to nodes u and v. Packets are injected directly into queues by an adversary $\mathcal{A}(b, r)$. We consider an execution that consists of two phases. This execution is represented in Figure 1. The phases are specified as follows.

In Phase 1, link u is up each k rounds, and link v is up each $k-1$ and $k+1$ rounds alternately. We inject one packet at rounds $k-1, 2k, 3k-1, 4k, 5k-1, \ldots, 2b \cdot k$ to traverse the links to u and v alternately, starting with the link to v. Let us assume that packets that need to traverse the link to v correspond to the injection-rate component of the adversary's type, and packets that need to traverse the link to u correspond to the burstiness component of the adversary's type.

The adversary injects one packet corresponding to the injection-rate component each $2k$ rounds, until the b packets representing the burstiness are injected. A packet can be transmitted starting from the next round after it has been injected into a queue. Because SIS is the scheduling policy, during Phase 1 no packet is transmitted, and at the end of Phase 1 there are $2b$ queued packets.

Phase 2 phase starts at the same round when Phase 1 ends. In this phase, both links are up at the same time each $k+2$ rounds for b rounds, and no packet is injected. Therefore, at the end of Phase 2, some b packets have been transmitted and some b packets remain queued. Moreover, the adversary can again inject a number of packets corresponding to the burstiness component of its type.

Observe that the latency of the links to u and v is $k+2$, which means that $h = k+2$. Furthermore, the actual injection rate is bounded by $1/2k$, that is, $1/(2h-4)$. This, for $h \geq 4$, is lower or equal than $1/h$, and consequently fulfills the admissibility condition regarding the injection of packets. At the end of Phase 2 we are in the same situation as at the begin of Phase 1, except that now b packets remain queued. Therefore, we can iterate the same injection pattern to create instability. □

Transmission policies with bounded node hearing latencies.

It is a natural question to ask if the fact that $\mathcal{S}_{post}$ with $\mathcal{T}_h$ is universally stable implies that $\mathcal{S}_{pre}$ with $\mathcal{T}_h$ is universally stable as well.

We say that an integer $h_v > 0$ is a *hearing latency of a node* v if this node v can transmit successfully, using any link, at least one packet during each h_v consecutive rounds. When such a number $h_v > 0$ does not exist then v is said to have *unbounded node hearing latency*, which is denoted by $h_v = \infty$. We say that a transmission oracle *provides a node latency* h when hearing latencies of all nodes are upper bounded by h.

We denote by $\mathcal{T}_h^{node}$ the class of transmission oracles that provide h as a node hearing latency. Along the same line, we denote by $\mathcal{T}_h^{link}$ the class of transmission oracles that provide h as a link hearing latency. In the proof of the next fact we rely on Theorems 1 and 2.

Lemma 7. *If $\mathcal{S}_\phi$ with $\mathcal{T}_h^{link}$ is stable against adversary $\mathcal{A}$ then $\mathcal{S}_\phi$ with $\mathcal{T}_h^{node}$ is stable against $\mathcal{A}$.*

Proof. Any $\mathcal{T} \in \mathcal{T}_h^{node}$ fulfills that $\mathcal{T} \in \mathcal{T}_h^{link}$. So, by our assumption, $\mathcal{S}_\phi$ with $\mathcal{T}_h^{node}$ will be stable against $\mathcal{A}$. □

Lemma 8. *If $\mathcal{S}_{post}$ with $\mathcal{T}_h^{node}$ is stable against an adversary $\mathcal{A}$ then $\mathcal{S}_{pre}$ with $\mathcal{T}_h^{node}$ is also stable against $\mathcal{A}$.*

Proof. First we show that if $\mathcal{S}_\phi$ with $\mathcal{T}_h^{link}$ is stable against adversary $\mathcal{A}$ then $\mathcal{S}_\phi$ with $\mathcal{T}_h^{node}$ is stable against $\mathcal{A}$. Observe that $\mathcal{T} \in \mathcal{T}_h^{link}$ for any $\mathcal{T} \in \mathcal{T}_h^{node}$. By the assumption, $\mathcal{S}_\phi$ with $\mathcal{T}_h^{node}$ is stable against $\mathcal{A}$. Next we show that if $\mathcal{S}_{post}$ with $\mathcal{T}_h^{node}$ is stable against adversary $\mathcal{A}$ then $\mathcal{S}_{pre}$ with $\mathcal{T}_h^{node}$ is stable against $\mathcal{A}$. To this end, let us consider an arbitrary execution $\mathcal{D}$ of protocol $P(\mathcal{T} \in \mathcal{T}_h^{node}, \mathcal{S}_{pre})$ against $\mathcal{A}$. Let us also consider an execution $\mathcal{D}'$ of protocol $P(\mathcal{T}', \mathcal{S}_{post})$ against the same adversarial packet injection as in $\mathcal{D}$, but such that $\mathcal{T}'$ makes the node v to transmit in each round when node v is to transmit in $\mathcal{D}$ and it succeeds in the packet being heard by the recipient node; this can also include the case where there are no queued packets. Observe that $\mathcal{T}'$ provides a node hearing latency h, that is, $\mathcal{T}' \in \mathcal{T}_h^{node}$. We obtain that nodes transmit the same packets at the same rounds in each $\mathcal{D}$ and $\mathcal{D}'$. The execution $\mathcal{D}$ is stable because $\mathcal{D}'$ is such. □

Theorem 7. *Scheduling policies SIS$_{post}$, SIS$_{pre}$, LIS$_{post}$, and LIS$_{pre}$, all with with $\mathcal{T}_h^{node}$, are universally stable.*

Proof. We use Lemmas 7 and 8 combined with Theorems 1 and 2. □

6. REFERENCES

[1] C. Àlvarez, M. J. Blesa, J. Díaz, M. J. Serna, and A. Fernández. Adversarial models for priority-based networks. *Networks*, 45(1):23–35, 2005.

[2] C. Àlvarez, M. J. Blesa, and M. J. Serna. A characterization of universal stability in the adversarial queuing model. *SIAM Journal on Computing*, 34(1):41–66, 2004.

[3] C. Àlvarez, M. J. Blesa, and M. J. Serna. The impact of failure management on the stability of communication networks. In *Proceeding of the 10th IEEE International Conference on Parallel and Distributed Systems (ICPAD)*, pages 153–160, 2004.

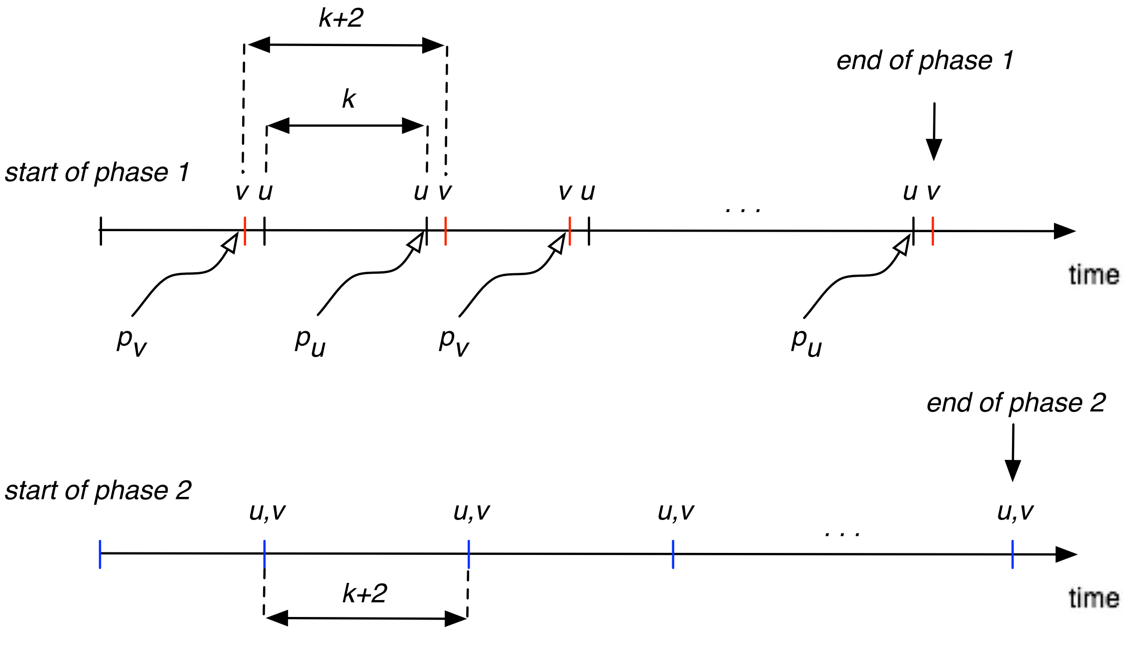

Figure 1: A visualization of the execution used in the proof of Theorem 6.

[4] M. Andrews. A survey of scheduling theory in wireless data networks. *IMA Volumes in Mathematics and its Applications*, 143:1–18, 2007.

[5] M. Andrews, B. Awerbuch, A. Fernández, F. T. Leighton, Z. Liu, and J. M. Kleinberg. Universal-stability results and performance bounds for greedy contention-resolution protocols. *Journal of the ACM*, 48(1):39–69, 2001.

[6] M. Andrews, A. Fernández, A. Goel, and L. Zhang. Source routing and scheduling in packet networks. *Journal of the ACM*, 52(4):582–601, 2005.

[7] M. Andrews and L. Zhang. Scheduling over a time-varying user-dependent channel with applications to high-speed wireless data. *Journal of the ACM*, 52(5):809–834, 2005.

[8] M. Andrews and L. Zhang. Routing and scheduling in multihop wireless networks with time-varying channels. *ACM Transactions on Algorithms*, 3(3):33, 2007.

[9] M. J. Blesa, D. Calzada, A. Fernández, L. López, A. L. Martínez, A. Santos, M. J. Serna, and C. Thraves. Adversarial queueing model for continuous network dynamics. *Theory of Computing Systems*, 44(3):304–331, 2009.

[10] A. Borodin, J. M. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson. Adversarial queuing theory. *Journal of the ACM*, 48(1):13–38, 2001.

[11] A. Borodin, R. Ostrovsky, and Y. Rabani. Stability preserving transformations: Packet routing networks with edge capacities and speeds. *Journal of Interconnection Networks*, 5(1):1–12, 2004.

[12] B. S. Chlebus, D. R. Kowalski, and A. Pelc. Electing a leader in multi-hop radio networks. In *in Proceedings of* 16*th International Conference on Principles of Distributed Systems (OPODIS)*, LNCS 7702, pages 106–120, 2012.

[13] B. S. Chlebus, D. R. Kowalski, A. Pelc, and M. A. Rokicki. Efficient distributed communication in ad-hoc radio networks. In *Proceedings of the* 38*th International Colloquium on Automata, Languages and Programming (ICALP), Part II*, LNCS 6756, pages 613–624, 2011.

[14] V. Cholvi and J. Echagüe. Stability of FIFO networks under adversarial models: State of the art. *Computer Networks*, 51(15):4460–4474, 2007.

[15] V. Cholvi and D. R. Kowalski. Bounds on stability and latency in wireless communication. *IEEE Communication Letters*, 14:842–844, 2010.

[16] S. Daum, S. Gilbert, F. Kuhn, and C. C. Newport. Broadcast in the ad hoc SINR model. In *Proceedings of the* 27*th International Symposium on Distributed Computing (DISC)*, LNCS 8205, pages 358–372, 2013.

[17] J. Echagüe, V. Cholvi, and A. Fernández. Universal stability results for low rate adversaries in packet switched networks. *IEEE Communication Letters*, 7(12):578–580, 2003.

[18] Y. Emek, L. Gąsieniec, E. Kantor, A. Pelc, D. Peleg, and C. Su. Broadcasting in UDG radio networks with unknown topology. *Distributed Computing*, 21(5):331–351, 2009.

[19] F. Foukalas, V. Gazis, and N. Alonistioti. Cross-layer design proposals for wireless mobile networks: A survey and taxonomy. *IEEE Communications Surveys and Tutorials*, 10(1-4):70–85, 2008.

[20] I. Gitman, R. Van Slyke, and H. Frank. Routing in packet-switching broadcast radio networks. *IEEE Transactions on Communications*, 24(8):926–930, 1976.

[21] R. Jurdak. *Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspecitve*. Springer-Verlag, 2007.

[22] T. Jurdzinski, D. R. Kowalski, M. Rozanski, and G. Stachowiak. Distributed randomized broadcasting in wireless networks under the SINR model. In *Proceedings of the 27th International Symposium on Distributed Computing (DISC)*, LNCS 8205, pages 373–387, 2013.

[23] T. Jurdzinski, D. R. Kowalski, and G. Stachowiak. Distributed deterministic broadcasting in wireless networks of weak devices. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP), Part II*, LNCS 7966, pages 632–644, 2013.

[24] P. Karn. MACA : A new channel access method for packet radio. In *Proceedings of the APRL / CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, 1990.

[25] D. R. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. *Distributed Computing*, 18(1):43–57, 2005.

[26] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 63–72, 2003.

[27] F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad hoc networks beyond unit disk graphs. *Wireless Networks*, 14(5):715–729, 2008.

[28] F. Kuhn, R. Wattenhofer, and A. Zollinger. An algorithmic approach to geographic routing in ad hoc and sensor networks. *IEEE/ACM Transactions on Networking*, 16(1):51–62, 2008.

[29] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, 2006.

[30] R. Rajaraman. Topology control and routing in ad hoc networks: A survey. *SIGACT News*, 33(2):60–73, 2002.

[31] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41(10):74 – 80, 2003.

[32] J. Urrutia. Routing with guaranteed delivery in geometric and wireless networks. In I. Stojmenovic, editor, *Handbook of wireless networks and mobile computing*, pages 393–406. John Wiley & Sons, 2002.

[33] P. von Rickenbach, R. Wattenhofer, and A. Zollinger. Algorithmic models of interference in wireless ad hoc and sensor networks. *IEEE/ACM Transactions on Networking*, 17(1):172–185, 2009.

[34] X. Wang, J. J. Garcia-Luna-Aceves, and H. R. Sadjadpour. Understanding the interaction between packet forwarding and channel access in multihop wireless networks. *IEEE Transactions on Mobile Computing*, 11(4):679–691, 2012.