Theoretical Computer Science 921 (2022) 20–35

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Stable routing scheduling algorithms in multi-hop wireless networks ☆

Vicent Cholvi [a,*,1], P. Garncarek [b,2], T. Jurdziński [b,2], Dariusz R. Kowalski [c,3]

[a] *Department of Computer Science, Universitat Jaume I, Castellón, Spain*
[b] *Instytut Informatyki, Uniwersytet Wrocławski, Wrocław, Poland*
[c] *School of Computer and Cyber Sciences, Augusta University, Augusta, GA, USA*

A R T I C L E I N F O

Article history:
Received 31 May 2021
Received in revised form 2 December 2021
Accepted 27 March 2022
Available online 31 March 2022
Communicated by L. Christoph

Keywords:
Wireless networks
Routing scheduling algorithms
Adversarial queuing
Interference
Stability
Packet latency

A B S T R A C T

Stability is an important issue in order to characterize the performance of a network, and it has become a major topic of study in the last decade. Roughly speaking, a communication network system is said to be *stable* if the number of packets waiting to be delivered (backlog) is finitely bounded at any one time.

In this paper we introduce a number of routing scheduling algorithms which, making use of certain knowledge about the network's structure, guarantee stability for certain injection rates.

First, we introduce two new families of combinatorial structures, which we call *universally strong selectors* and *generalized universally strong selectors*, that are used to provide a set of transmission schedules. Making use of these structures, we propose two *local-knowledge* packet-oblivious routing scheduling algorithms. The first proposed routing scheduling algorithm only needs to know some upper bounds on the number of links and on the network's degree, and is asymptotically optimal regarding the injection rate for which stability is guaranteed. The second proposed routing scheduling algorithm is close to be asymptotically optimal, but it only needs to know an upper bound on the number of links. For such algorithms, we also provide some results regarding both the maximum latencies and queue lengths. Furthermore, we also evaluate how the lack of global knowledge about the system topology affects the performance of the routing scheduling algorithms.

1. Introduction

Stability is an important issue in order to characterize the performance of a network, and it has become a major topic of study in the last decade. Roughly speaking, a communication network system is said to be *stable* if the number of packets waiting to be delivered (backlog) is finitely bounded at any one time. The importance of such an issue is obvious, since if one cannot guarantee stability, then one cannot hope to be able to ensure deterministic guarantees for most of the network performance metrics.

☆ A preliminary version of the paper appeared in [18].

* Corresponding author.
E-mail address: vcholvi@uji.es (V. Cholvi).

[1] This author was partially supported by the Ministerio de Ciencia, Innovación y Universidades grant PRX18/000163 and by the Spanish Ministry of Science and Innovation grant PID2019-109805RB-I00 (ECID) cofunded by FEDER.
[2] This author was partially supported by the Polish National Science Center (NCN) grant UMO-2017/25/B/ST6/02010.
[3] This author was partially supported by the Polish National Science Center (NCN) grant UMO-2017/25/B/ST6/02553.

https://doi.org/10.1016/j.tcs.2022.03.038

For many years, the common belief was that only overloaded queues (i.e., when the total arrival rate is greater than the service rate) could generate instability, while underloaded ones could only induce delays that are longer than desired, but always remain stable. However, this belief was shown to be wrong when it was observed that, in some networks, the backlogs in specific queues could grow indefinitely even when such queues were not overloaded [9,4]. These later results aroused an interest in understanding the stability properties of packet-switched networks, so that a substantial effort has been invested in that area. Stability of specific scheduling policies was considered for example in [7,17,23,26]. In [21,27,29], it was considered the impact of network topologies on injection rates that guarantee stability. A systematic account of some issues related to universal stability was given in [2].

Contrary to wireline networks, where a node can transmit data over any outgoing link and simultaneously receive data over any incoming link, the situation is different in wireless networks. In such scenarios, a node can transmit messages within its transmission range. Such a range is determined by the power of the transmitting device and the surrounding topography. However, nearby wireless signal transmissions that overlap in time can interfere with one another, to the effect that none can be transmitted successfully. As a result, this makes the study of stability in wireless networks more complex. The contents of transmissions get injected into nodes to be delivered to their respective destination nodes by traversing a number of nodes. A route indicates the nodes that a packet has to traverse upon its injection until its destination. Furthermore, a scheduling algorithm is run at each node independently and concurrently, and decides, at each time instant which packet to send and when.

Related work Similar to the wireline case, a substantial effort has been invested in investigating stability in that setting. Networks with bandwidth and delay parameters associated with wireline links were considered in [8] and [10]; such behavior can be considered as capturing some properties of wireless networks. Wireless networks were first studied by Andrews and Zhang [5,6] and Cholvi and Kowalski [19] without considering explicit interferences. Lim et al. [28] analyzed the stability of the max-weight protocol in wireless networks with interferences, assuming the existence of a set of feasible edge rate vectors sufficient to keep the network stable.

The stability and packet latency of broadcasting in single-hop radio networks understood as synchronous multiple access channels was studied by Chlebus et al. [15,16] and Anantharamu et al. [3]. Fernández Anta et al. [22] provided an adaptive transmission policy that guarantees a hearing latency of $\mathcal{O}(k^2 \log k)$, where $k-1$ is the maximum degree of the network. Garncarek et al. [24] studied stability of packet scheduling policies in a distributed system which are local in that nodes only access their local queues, and have no other feedback from the underlying distributed system. They developed an adaptive scheduling policy that is universally stable on a shared channel and proved that memoryless policies resorting only on the information about non-emptiness of queues could be stable with injection rates of $\mathcal{O}(1/\log n)$. In a subsequent work, Garncarek et al. [25] developed a local memoryless scheduling policy which is both adversarially and stochastically stable for injection rates $\Omega(1/\log n)$.

In [13], Chlebus et al. considered interactions among components of routing in multi-hop wireless networks with interferences, which included transmission policies, scheduling policies and control mechanisms to coordinate transmissions with scheduling. In [12], the authors demonstrated that there is no routing algorithm guaranteeing stability for an injection rate greater than $1/L$, where the parameter L is the largest number of nodes which a packet needs to traverse while routed to its destination. Furthermore, they also provided a routing algorithm that guarantees stability for injection rates smaller than $1/L$. However, their approach is not accurate for studying stability of longer-distance packets; therefore, in this work we study how the stability of routing depends of the *conflict graph* of the underlying wireless networks, which is independent of the lengths of the packets' routes. Roughly speaking, the conflict graph of a network models whether or not two transmission could interfere with each other.

Our results In this paper, we study the stability of dynamic *routing scheduling algorithms* in multi-hop radio networks with a specific methodology of adversarial traffic that reflects interferences. We focus on algorithms that do not take into account any historical information about packets or carried out by packets, which we call *packet-oblivious* algorithms. Such algorithms are important since practical forwarding protocols and corresponding data-link layer architectures are typically packet-oblivious.

First, we introduce two new families of combinatorial structures, which we call *universally strong selectors* and *generalized universally strong selectors*, that are used to provide a set of transmission schedules. Making use of these structures, combined with some known queuing policies such as Longest In System (LIS), Shortest In System (SIS), Nearest From Source (NFS) and Furthest To Go (FTG), we propose two *local-knowledge* packet-oblivious routing scheduling algorithms which work without using any global topological information about the networks on which they operate, that guarantee stability for certain injection rates.

The first proposed routing scheduling algorithm only needs to know some upper bounds on the number of links and on the network's degree, and is asymptotically optimal regarding the injection rate for which stability is guaranteed. The second proposed routing scheduling algorithm is close to be asymptotically optimal by a factor that depends logarithmically on the number of nodes, but it only needs to know an upper bound on the number of links. For such algorithms, we also provide some results regarding both the maximum latencies and queue lengths.

Furthermore and in order to evaluate how the lack of global knowledge about the system topology affects the performance of the routing scheduling algorithms, we also introduce a new *global-knowledge* routing scheduling algorithm and

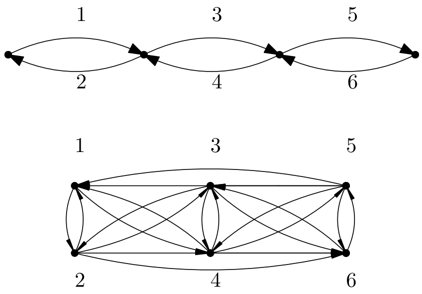

Fig. 1. Radio network G with 4 nodes and links labeled $1-6$ (up). Conflict graph $H(G)$ obtained from network G (down). Observe that each link i in network G corresponds to one node i in $H(G)$.

we show that it is optimal regarding the injection rate for which it guarantees stability. We show that the lack of global knowledge decreases the injection rate for which stability is guaranteed by a factor of e.

Paper structure The rest of the paper is structured as follows. Section 2 contains the technical preliminaries. In Section 3, we introduce a family of combinatorial structures, widely called *selectors* [20,14], that are the core of the deterministic local-knowledge routing scheduling algorithms presented in Sections 4 and 5. In Section 6 we also introduce a global-knowledge routing scheduling algorithm. In Section 7, we compare the maximum injection rates for which these routing scheduling algorithms guarantee stability. In Section 8, we extend the obtained results to a number of queueing scheduling policies. We end with some conclusions in Section 9. Some auxiliary technical proves are given in the Appendix.

2. Model and problem definition

2.1. Wireless radio network

We consider a *wireless radio network* represented by a directed symmetric network graph $G = (V_G, E_G)$. It consists of *nodes* in V_G representing devices, and directed edges, called *links*, representing the fact that a transmission from the starting node of the link could be directly delivered to the ending node. The graph is symmetric in the sense that if some $(i, j) \in E_G$ then $(j, i) \in E_G$ too. At this point, we note that we use symmetric directed graphs instead of undirected graphs since, in order to specify when transmissions interfere, we need to distinguish when a given node uses a link to transmit and when to receive.

Each node has a unique ID number and it knows some upper bounds on the number m of edges in the network and the network in-degree (i.e., the largest number of links incoming to a network node).[4]

Nodes communicate via the underlying wireless network G. Communication is in synchronous rounds. In each round a node could be either transmitting, receiving or waiting to receive. Node i receives a message from a node $j \neq i$ in a round if j is the only transmitting in-neighbor of i in this round and node i does not transmit in this round; we say that the message was successfully sent/transmitted from j to i.

2.2. Conflict graphs

We define the *conflict graph* $H(G) = (V_{H(G)}, E_{H(G)})$ of a network G as follows: (1) its vertices are links of the network (i.e., $V_{H(G)} = E_G$) and, (2) a directed edge $(u, v) \in E_{H(G)}$ if and only if a message across link $v \in E_G$ cannot be successfully transmitted while link $u \in E_G$ is being used to transmit. Note that, accordingly with the radio model, a conflict occurs if and only if the transmitter in u is also a receiver in v *or* the transmitter in u is a neighbor of the receiver in v (see Fig. 1 for an illustrative example). If network G is clear from the context, we skip the parameter G in $H(G)$ (i.e., we will use H). Note that, the links in our definition are directed in order to distinguish which transmission is blocked by which.

2.3. Routing scheduling algorithms

The contents of transmissions (i.e., messages or packets) get injected into nodes to be delivered to their respective destination nodes by traversing a number of nodes. A *route* indicates the nodes that a packet has to traverse upon its injection until its destination. In our case, we assume that they will be decided by an *adversary*.

A *routing scheduling algorithm*, denoted RoSA, is an algorithm that is run at each node independently and concurrently, and decides, at each time instant, which packet to send and when. Note that a RoSA should not to be confused with a routing algorithm, but it is one of the components that make it up.

[4] In which case the performance will depend on these known estimates, instead of the actual values.

All our RoSAs will be based on pre-defined *transmission schedules*, which will be circularly repeated — the properties of these schedules will guarantee stability for certain injection rates. These schedules will be different for different types of algorithms, due to the available information based on which these schedules could be created. We consider *packet-oblivious* RoSAs, that is, algorithms which only use their hardwired memory and basic parameters of the stored packets assigned to them at injection time (such as source, destination, injection time, route) in order to decide which packet to send and when.

2.4. Adversaries

We model dynamic injection of packets by way of an adversarial model, in the spirit of similar approaches used in [9,4,29,16,19,13,12]. An adversary represents the users that generate packets to be routed in a given radio network. The constraints imposed on packet generation by the adversary allow considering worst-case performance of deterministic RoSAs handling dynamic traffic.

Over time, an adversary injects packets to some nodes. The adversary decides on a path a packet has to traverse upon its injection. Our task is to develop a packet-oblivious RoSA such that the network remains *stable*; that is, the number of packets simultaneously queued is bounded by a constant in all rounds. Since an unbounded adversary can exceed the capability of a network to transmit messages, we limit its power in the following way: for any time window of any length T, the adversary can inject packets, with their paths, in such a way that each link is traversed by at most $\rho \cdot T + b$ packets, for some $0 \le \rho \le 1$ and $b \in \mathbb{N}^+$. We call such an adversary a (ρ, b)-*adversary*.

2.5. (ρ, T)-frequent schedules

Finally, we introduce the (ρ, T)-frequent schedule concept, which we will use throughout the rest of the paper.

Definition 1. *A* (ρ, T)-*frequent schedule* for graph G is an algorithm that decides which links of graph G transmit at every round in such a way that each link is guaranteed to successfully transmit (i.e., without radio network collisions) at least $\rho \cdot T$ times in any window of length T, provided at least $\rho \cdot T$ packets await for transmission at the link at the start of the window.

3. Selectors as transmission schedulers

In this section, we introduce a family of combinatorial structures, widely called *selectors* [20,14], that are the core of the deterministic local-knowledge protocols presented in Sections 4 and 5. In short, we will use specific type of selectors to provide a set of transmission schedules that assure stability when combined with suitable queuing policies.

There are many different types of selectors, with the more general one being described below:

Definition 2. Given integers k, m and n, with $1 \le m \le k \le n$, we say that a boolean matrix M with t rows and n columns is an (n, k, m)-*selector* if any submatrix of M obtained by choosing k out of n arbitrary columns of M contains at least m distinct rows of the identity matrix I_k. The integer t is referred as the size of the (n, k, m)-selector.

In order to use selectors as transmission schedules, the parameter n is intended to refer to the number of nodes in the network, k refers to the maximum number of nodes that can compete to transmit (i.e., $k = \Delta + 1$, where Δ is the maximum degree of the network), and m refers to the number of nodes that are guaranteed to successfully transmit during the t-round schedule. Therefore, each column of the matrix M is used to define the whole transmission schedule of each node. Rows are used to decide which nodes should transmit at each time slot: In the i-th time slot, node v will transmit iff $M_{i,v} = 1$ and v has a packet queued; the schedule is repeated after each t time slots.

Taking into account the above-mentioned approach, selectors may be used to guarantee that during the schedule, every node will successfully receive some messages.

An $(n, k, 1)$-selector guarantees that, for each node, one of its neighbors will successfully transmit during at least 1 round per schedule cycle. That is, that node will successfully receive at least one message. However, whereas the above use of selectors is helpful in broadcasting (since there is progress every time any node receives a message from a neighbor), it happens that many neighbors may have something to send, but only one of them has something for that node. Therefore, the above presented selector guarantees that each node will receive at least one message, but not necessarily will receive the one addressed to it.

An (n, k, k)-selector, which is known as *strong selector*, guarantees that every node that has exactly k neighbors will receive a message from each one of them. However, it has been shown that its size $t = \Omega(\min\{n, (k^2/\log k)\log n\})$ [20]. This means that k packets will be received, but during a long amount of time.

In order to solve the above mentioned problems with known selectors, now we introduce a new type of selectors, which we call *universally strong*. Formally:

Definition 3. An (n, k, ϵ)-*universally-strong selector* $\mathcal{S}$, also called an (n, k, ϵ)-*USS*, is a family of t sets $T_1, \ldots, T_t \subseteq [n]$ such that for every set $A \subseteq [n]$ of at most k elements and for every element $a \in A$ there exist at least $\epsilon \cdot t/k$ sets $T_i \in \mathcal{S}$ such that $T_i \cap A = \{a\}$.

In the context of wireless networks, the parameter n represents the number of nodes in the conflict graph H of a given graph G, k represents the upper bound on the number of links, whose transmissions can conflict with a transmission across any link e (e.g., $k = \Delta_{in}^H + 1$, where Δ_{in}^H is the in-degree of the conflict graph H). Then, ϵ/k represents the frequency of successful transmissions across any link.

Finally, we introduce one more new type of selectors that generalize (n, k, ϵ)-universally-strong selectors in the sense that they remove the restriction that the incoming set k is of bounded size.

Definition 4. An (n, ϵ)-*generalized-universally-strong selector* $\mathcal{S}$, also called an (n, ϵ)-*GUSS*, is a family of t sets $T_1, \ldots, T_t \subseteq [n]$ such that for every set $A \subseteq [n]$ and for every element $a \in A$ there exist at least $\epsilon \cdot t/|A|$ sets $T_i \in \mathcal{S}$ such that $T_i \cap A = \{a\}$.

3.1. Universally strong selectors of polynomial size

Clearly, universally strong selectors make sense provided they exist and their size is moderate. In the next theorem, we prove that, for any $\epsilon \leq 1/e$, there exists an (n, k, ϵ)-universally-strong selector of polynomial size.

Theorem 1. *For any $\epsilon \leq 1/e$, there exists an (n, k, ϵ)-universally-strong selector of size $O(k^2 \ln n)$.*

Proof. The proof relies on the probabilistic method.

Consider a random matrix M with t rows and n columns, where $M_{i,j} = 1$ with probability p and $M_{i,j} = 0$ otherwise. Given a row i and columns $j_1, \ldots, j_k$, the probability that $M_{i,j_1} = 1$ and $M_{i,j_2} = \cdots = M_{i,j_k} = 0$ (i.e., that node j_1's transmission is not interrupted by nodes $j_2, \ldots, j_k$ in round i) is $P = p(1-p)^{k-1}$ and is maximized with $p = 1/k$. In further considerations, we use matrix M generated with $p = 1/k$.

Given columns $C = \{j_1, \ldots, j_k\}$, let $X(C)$ be the number of "good" rows i such that $M_{i,j_1} = 1$ and $M_{i,j_2} = \cdots = M_{i,j_k} = 0$.

We will use the following Chernoff bound:

$$Pr[X(C) \leq (1-\delta)E[X(C)]] \leq exp(-E[X(C)]\delta^2/2) \ ,$$

for $0 \leq \delta \leq 1$.

Using $E[X(C)] = Pt$ and $\delta = (kP - \epsilon)/(kP)$, we obtain:

$$Pr[X(C) \leq \epsilon t/k] \leq exp(-Pt\delta^2/2) \ .$$

Consider a "bad" event $\mathcal{E}$ such that for at least one set of columns of size at most k, there are few good rows. More specifically, $X(C) \leq \epsilon t/k$ for at least one set of columns C, where $|C| = k$. The probability R of event $\mathcal{E}$ happening fulfills the following inequality:

$$R \leq k\binom{n}{k} exp(-Pt\delta^2/2) \ .$$

Therefore $R < 1$ if

$$exp(-Pt\delta^2/2) < 1/\left[k\binom{n}{k}\right]$$

$$-Pt\delta^2/2 < -\ln\left(k\binom{n}{k}\right)$$

$$Pt\left(\frac{kP-\epsilon}{kP}\right)^2/2 > \ln\left(k\binom{n}{k}\right) \ .$$

Let $c = kP$. Using $\binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$, provided $c \neq \epsilon$, we obtain the following:

$$t(c-\epsilon)^2/(2ck) > \ln k + \ln\left(\frac{ne}{k}\right)^k$$

$$t > \left[2ck\ln k + 2ck^2 \ln\left(\frac{ne}{k}\right)\right]/(c-\epsilon)^2 \ .$$

Therefore, as long as $0 \leq \delta = \frac{c-\epsilon}{c} \leq 1$ (so that we can use the Chernoff bound) and $\epsilon \neq c$, the probability of generating a random matrix M such that event $\mathcal{E}$ occurs is less than 1. Thus, there exists a matrix M such that, for every set of k

1. Let $d = \lceil \log_k n \rceil$ and $q = c \cdot k \cdot d$ for some constant $c > 0$ such that $q^{d+1} \geq n$.
2. Consider all polynomials P_i of degree d over field $[q]$. Notice that there are q^{d+1} of such polynomials.
3. Create a matrix M' of size $q \times q^{d+1}$. Each column will represent values $P_i(x)$ of each polynomial P_i for arguments $x = 0, 1, \ldots, q-1$ (corresponding to rows of M'). Next, matrix M'' is created from M' as follows: each value $y = P_i(x)$ is represented and padded in q consecutive rows of 0s and 1s, where 1 is on y-th position, while on all other positions there are 0s. Notice that each column of M'' has q^2 rows (q rows for each argument), thus M'' has size $q^2 \times q^{d+1}$.
4. Remove $q^{d+1} - n$ arbitrary columns from matrix M'', creating matrix M with exactly n columns remaining.
5. Each row of matrix M will correspond to one set T_i of a universally strong selector $\{T_i\}_{i=1}^{q^2}$ over the set $\{1, \ldots, n\}$ of elements.

Fig. 2. The Poly-Universally-Strong algorithm, given parameters n and k.

columns $j_1, \ldots, j_k$, there are at least $\epsilon t/k$ rows such that $M_{i,j_1} = 1$ and $M_{i,j_2} = \cdots = M_{i,j_k} = 0$. Trivially, such matrix M guarantees the above property for any set of at most k columns. Hence, M represents a (n, k, ϵ)-universally-strong selector, provided that $\epsilon < c = kP$. Next, we calculate which values of ϵ fulfill that inequality.

Consider a sequence $a_i = (1 + 1/i)^i$. a_i is known as a lower bound on the Euler's number e (i.e., $\forall i\ a_i < e$). Note that $c = kP = (1 - 1/k)^{k-1} = 1/a_{k-1} > 1/e$ for all $k \geq 2$. This implies that any $\epsilon \leq 1/e$ fulfills the requirement of $\delta > 0$ and results in the existence of an (n, k, ϵ)-universally-strong selector. □

3.2. Obtaining universally strong selectors of polynomial size in polynomial time

In the proof of Theorem 1, we have introduced a family of universally strong selectors of polynomial size. However, obtaining them by derandomizing would be very inefficient (all the approaches we know are, at least, exponential in n). Here, we present an algorithm, which we call Poly-Universally-Strong, that computes universally strong selectors of polynomial size in polynomial time They only have slightly lower values of ϵ comparing to the existential result in Theorem 1.

The algorithm, whose code is shown in Fig. 2, has to be executed by each node in the network taking the same polynomials, so that all nodes will obtain exactly the same matrix that defines the transmission schedule.

The next theorem shows that, indeed, it constructs an (n, k, ϵ)-universally-strong selector of polynomial size with $\epsilon = 1/(4 \log_k n)$.

Theorem 2. Poly-Universally-Strong *constructs (by using $c = 2$) an (n, k, ϵ)-universally-strong selector of size* $4 \cdot k^2 \cdot \lceil \log_k n \rceil^2$ *with* $\epsilon = 1/(4 \log_k n)$.

Proof. First, note that two polynomials P_i and P_j of degree d with $i \neq j$, can have equal values for at most d different arguments. This is because they have equal values for arguments x for which $P_i(x) - P_j(x) = 0$. However, $P_i - P_j$ is a polynomial of degree at most d, so it can have at most d zeroes. So, $P_i(x) = P_j(x)$ for at most d different arguments x.

Take any polynomial P_i and any k polynomials P_j still represented in M, excluding columns/polynomials removed from consideration in step 4. There are at most $k \cdot d$ different arguments where one of the k polynomials can be equal to P_i. So, for $q - k \cdot d$ different arguments, the values of the polynomial P_i are unique. Therefore, if we look at rows with 1 in column i of matrix M (there are q of those rows, one for each argument), at least $q - k \cdot d$ of them have 0s in chosen k columns. Since there are q^2 rows, so a fraction $(q - k \cdot d)/q^2$ of rows have the desired property (i.e., there is value 1 in column i and value 0 in the chosen k columns):

$$\frac{q - k \cdot d}{q^2} = \frac{(c-1) \cdot k \cdot d}{(c \cdot k \cdot d)^2} = \frac{c-1}{c^2 \cdot k \cdot d} \triangleq f(c) \ .$$

Let us find the value of c that maximizes the function f. To do it, we compute its differential

$$f'(c) = \left(\frac{c-1}{c^2 \cdot k \cdot d}\right)' = \frac{1 \cdot (c^2 \cdot k \cdot d) - (c-1) \cdot k \cdot d \cdot 2c}{c^4 \cdot k^2 \cdot d^2} = $$
$$= \frac{-c^2 \cdot k \cdot d + 2c \cdot k \cdot d}{c^4 \cdot k^2 \cdot d^2} = \frac{-c+2}{c^3 \cdot k \cdot d} \ .$$

Thus, $f'(c) = 0$ for $c = 0$ or $c = 2$. The value $c = 2$ maximizes f, giving $f(c) \leq f(2) = 1/(4k \cdot d) = 1/(4k \cdot \log_k n)$.

Therefore, we can construct an (n, k, ϵ)-universally-strong selector with $\epsilon = f(2) \cdot k = 1/(4d) = 1/(4 \log_k n)$ of length $4k^2 \cdot \lceil \log_k n \rceil^2$, which means that an $f(2) = 1/(4k \cdot \log_k n)$ fraction of the selector's sets have the desired property. □

3.3. Obtaining generalized universally strong selectors of polynomial size

Here, we show how to construct an (n, ϵ)-generalized-universally-strong selector $\mathcal{S}$, for admissible values of $\epsilon \in (0, 1/e)$, by interleaving specific universally strong selectors. Let $\gamma = \lfloor \log_{1+2\epsilon} n \rfloor$. Consider a family $\{\mathcal{S}_i\}_{i=0}^{\gamma-1}$ of $(n, \lfloor n/(1+2\epsilon)^i \rfloor, 2\epsilon)$-universally-strong selectors, each of size t'. They exist, cf., Theorems 1 and 2. We interleave them to obtain $\mathcal{S}$ of size $t = t' \cdot \gamma$. More specifically, we define $T_j \in \mathcal{S}$, the j-th set of $\mathcal{S}$, to be the $\lceil j/\gamma \rceil$ set of the universally strong selector $\mathcal{S}_{j \bmod \gamma}$.

1. Choose m and Δ such that $|E(G)| \leq m$ and $\Delta_{in}^{H} \leq \Delta$.
2. Obtain an $(m, \Delta+1, \epsilon)$-universally-strong selector (for some value of ϵ) of some length t and use it as the transmission schedule.
3. When there are several packets awaiting in a single queue, choose the packet to be transmitted according to ALG, breaking ties in any arbitrary fashion.

Fig. 3. The USS-PLUS-ALG RoSA for a network G.

Theorem 3.

1. *For any $\epsilon \leq 1/(2e \cdot \lfloor \log_{1+1/e} n \rfloor)$, there exists an (n, ϵ)-generalized-universally-strong selector of size $O(n^2 \ln^2 n)$.*
2. *For any $\epsilon \leq 1/(8 \log_2 n \cdot \lfloor \log_{1+1/(4\log_2 n)} n \rfloor) = O(1/\ln^3 n)$, an (n, ϵ)-generalized-universally-strong selector of size $O(n^2 \ln^4 n)$ could be constructed in polynomial time.*

Proof. First, we prove that $\mathcal{S}$ is an (n, ϵ)-generalized-universally-strong selector. Consider a set $A \subseteq [n]$; it holds $n/(1+2\epsilon)^{i+1} < |A| \leq n/(1+2\epsilon)^i$, for some $0 \leq i \leq \gamma$. Hence, for any element $a \in A$, there are at least

$$\frac{(2\epsilon)t'}{\lfloor n/(1+2\epsilon)^i \rfloor} \geq \frac{(2\epsilon)t'}{|A|(1+2\epsilon)} \geq \frac{(\epsilon/\gamma)\cdot(t'\gamma)}{|A|} \cdot \frac{2}{1+2\epsilon} \geq \frac{(\epsilon/\gamma)\cdot t}{|A|}$$

sets T_j in $(n, \lfloor n/(1+2\epsilon)^i \rfloor, 2\epsilon)$-universally strong selector $\mathcal{S}_i$ such that $T_j \cap A = \{a\}$. (Recall that t' is the size of the universally strong selector, $t = t'\gamma$ is the size of generalized universally strong selector, and $2\epsilon \leq 1/e$.) Hence, it is an $(n, \epsilon/\gamma)$-generalized-universally-strong selector, of length $O(t'\gamma)$, for admissible $\epsilon \leq 1/(2e)$ (i.e., such that for 2ϵ there are universally strong selectors) and $\gamma = \lfloor \log_{1+2\epsilon} n \rfloor$.

To obtain the first result, we apply to the above argument the bounds from Theorem 1 with $\epsilon = 1/e$ and $t' = O(k^2 \ln n)$ for values $k = \lfloor n/(1+2\epsilon)^i \rfloor$ as in the construction, to get the result for the $(n, (\epsilon/2)/\lfloor \log_{1+1/e} n \rfloor)$-generalized-universally-strong selector for $\epsilon/2 = 1/(2e)$ and corresponding $\gamma = O(\ln n)$; this generalized universally strong selector automatically works for smaller values than $(\epsilon/2)/\lfloor \log_{1+1/e} n \rfloor$.

To obtain the second result, we apply to the above argument the bounds from Theorem 2 with $\epsilon = 1/(4\log_2 n)$ and lengths $t' = O(n^2 \ln^2 n)$ (which occurs if we take $k \geq \lceil e \rceil$), to get the result for the $(n, (\epsilon/2)/\lfloor \log_{1+1/(4\log_2 n)} n \rfloor)$-generalized-universally-strong selector for $\epsilon/2 = 1/(8 \log n)$ and corresponding $\gamma = O(\ln^2 n)$; this generalized universally strong selector automatically works for smaller values than $1/(8\log_2 n \cdot \lfloor \log_{1+1/(4\log_2 n)} n \rfloor)$, and since $\log_{1+1/(4\log_2 n)} n$ is asymptotically $\Theta(\ln^2 n)$ then we have our result. □

As it can be observed, whereas generalized universally strong selectors do not make use of k (contrary to what happens with universally strong selectors), they have larger sizes; namely, $O(n^2 \ln^2 n)$ vs $O(k^2 \ln n)$ and $O(n^2 \ln^4 n)$ vs $O(k^2 \lceil \log_k n \rceil^2)$, with $k \leq n$. Furthermore, they also provide smaller frequencies of successful transmissions across any link; namely, $1/(2e \cdot \lfloor \log_{1+1/e} n \rfloor)$ vs $1/e$ and $1/(8\log_2 n \cdot \lfloor \log_{1+1/(4\log_2 n)} n \rfloor)$ vs $1/(4\log_k n)$.

4. The USS-PLUS-ALG RoSA

In this section, we introduce a local-knowledge packet-oblivious RoSA that makes use of the family of universally strong selectors introduced in Section 3 as *transmission schedules* (i.e., the time instants when packets stored at each one node must be transmitted to a receiving node). As it has been mentioned previously, local-knowledge RoSAs work without using any topological information, except for maybe some network's features that do not require full knowledge of its topology. In our particular case, that will consist of some upper bounds on the number of links and on the network's degree.

The code of the proposed RoSA, which we call USS-PLUS-ALG, is shown in Fig. 3. Given a graph G with a number of links bounded by m, and an in-degree of its conflict graph H (which we denote as Δ_{in}^{H}) bounded by $\Delta \geq 1$, it uses an $(m, \Delta+1, \epsilon)$-universally-strong selector as a schedule: assuming the selector is represented by matrix M with t rows, each link $z \in E_G$ will transmit at time i iff $M_{i \bmod t, z} = 1$. Notice that here each link is assumed to have an independent queue, and therefore they will act as a sort of "node" (in terms of selectors, such as it has been stated in the previous section). This means that each individual link will have its own schedule.

Next, we show that USS-PLUS-LIS (i.e., USS-PLUS-ALG where ALG is the Longest-In-System queueing scheduling policy) guarantees stability, provided a given packets' injection admissibility condition is fulfilled.

At this point, we note that the transmission schedules provided by our universally-strong selectors can be seen as (ρ, T)-frequent schedules, as shown in the lemma below.

Lemma 1. *The transmission schedule provided by an $(m, \Delta+1, \epsilon)$-USS of length t is a (ρ, T)-frequent schedule, with $\rho = \epsilon/(\Delta+1)$ and $T = t$.*

Proof. Assume that matrix M represents the used $(m, \Delta+1, \epsilon)$-USS of length t. Let us take any arbitrary link $z \in E_G$ and consider the set of all other links that conflict with the link z, of which there are at most Δ. This means that (according to

Definition 3) there exist at least $\epsilon \cdot t/(\Delta+1)$ rows i in M such that $M_{i,z}=1$ and $M_{i,c_1}=\cdots=M_{i,c_j}=0$, where $j \le \Delta$ and $\{c_1,\ldots,c_j\}$ is the set of links in conflict with the link z. Therefore, at time i, link z will transmit a message, and no link that conflicts with the link z will transmit. This guarantees that any link z will successfully transmit, at least, $\epsilon \cdot t/(\Delta+1)$ messages during any schedule of length t; i.e., this is an $(\epsilon/(\Delta+1), t)$-frequent schedule. □

Theorem 4. *Given a network G,* USS-PLUS-LIS *is stable against any (ρ, b)-adversary, for $\rho < \frac{\epsilon}{\Delta+1}$, where ϵ is the frequency parameter of the used USS and Δ is the upper bound on the in-degree of the conflict graph H of the graph G.*

Proof. According to Lemma 1, USS-PLUS-LIS uses an $(\epsilon/(\Delta+1), t)$-frequent schedule. By plugging the schedule into Lemma 5 (in Appendix A), we obtain the desired result. □

By using the selectors provided by the POLY-UNIVERSALLY-STRONG algorithm in USS-PLUS-LIS, we have the following result:

Corollary 1. *Given a network G,* USS-PLUS-LIS *using a universally strong selector computed by the* POLY-UNIVERSALLY-STRONG *algorithm is stable against any (ρ, b)-adversary, for $\rho < \frac{1}{4(\Delta+1)\log_{\Delta+1} m}$.*

If instead of the selectors provided by the POLY-UNIVERSALLY-STRONG algorithm, we use a selector from Theorem 1, we have that:

Corollary 2. *Given a network G, there exists a universally strong selector that, used in* USS-PLUS-LIS, *provides stability against any (ρ, b)-adversary, for $\rho < \frac{1}{e\cdot(\Delta+1)}$.*

We can use an $(m, \Delta+1, 1/e)$-universally-strong selector from Theorem 1 (unconstructive) as a (ρ', T)-frequent schedule (see Lemma 1) in Lemma 8 (in Appendix A), with $\rho' = \frac{1}{e(\Delta+1)}$ and T equal to the length of the $(m, \Delta+1, 1/e)$-universally-strong selector.

Corollary 3. *By using the $(m, \Delta+1, 1/e)$-USS from Theorem 1 (unconstructive), each packet spends at most $\frac{(b-1)}{\rho} \cdot \left(\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L} - 1\right)$ time slots in the system, where $\rho' = \frac{1}{e(\Delta+1)}$, L is the length of the longest simple directed path in the network and ρ is the used injection rate such that $\rho < \rho'$. Furthermore, each queue contains at most $\frac{b-1}{\left(1-\frac{\rho}{\rho'}\right)^L} + 1$ packets.*

Alternatively, we can use an $(m, \Delta+1, 1/(4\log_{\Delta+1} m))$-universally-strong selector from Theorem 2 (constructive) as a (ρ', T)-frequent schedule in Lemma 8 (in Appendix A).

Corollary 4. *By using the $(m, \Delta+1, 1/(4\log_{\Delta+1} m))$-USS from Theorem 1 (constructive), each packet spends at most $\frac{(b-1)}{\rho} \cdot \left(\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L} - 1\right)$ time slots in the system, where $\rho' = \frac{1}{4(\Delta+1)\log_{\Delta+1} m}$, L is the length of the longest simple directed path in the network and ρ is the used injection rate such that $\rho < \rho'$. Furthermore, each queue contains at most $\frac{b-1}{\left(1-\frac{\rho}{\rho'}\right)^L} + 1$ packets.*

4.1. On the locality of USS-PLUS-LIS

As it can be readily seen, USS-PLUS-LIS for a network G requires some knowledge of the value of the in-degree of its conflict graph H (i.e., of Δ_{in}^H). In order to obtain H it is necessary to gather the whole topology of G. However, as the next lemma shows, Δ_{in}^H can be bounded by the in-degree of the network G (denoted Δ_G).

Lemma 2. $\Delta_{in}^H \le \Delta_G^2 + \Delta_G - 1$, *provided $\Delta_G > 0$.*

Proof. If $\Delta_{in}^H = 0$, then the lemma is trivially true. Otherwise, consider a vertex e in H of maximum in-degree $deg(e) = \Delta_{in}^H$. Since $\Delta_{in}^H \neq 0$, there is at least one edge $(e', e) \in H$ such that, in G, e cannot successfully transmit at the same time instant when e' transmits. Let us denote $e = (u, v)$ and $e' = (u', v')$, and let us consider the different scenarios where e and e' may conflict.

Now, we make a case analysis regarding the possible conflicts in G (note that its in-degree is equal to its out-degree, since G is symmetric):

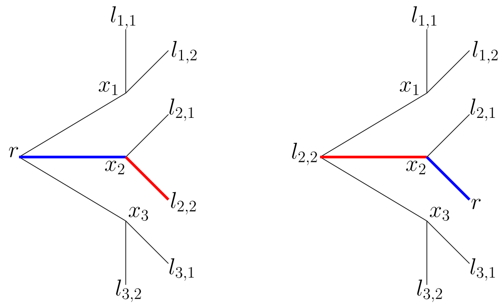

Fig. 4. Example of a tree T (on the left) and tree $T_{2,2}$ (on the right) for $\Delta = 3$. Nodes r and $l_{2,2}$ swapped places, which means that edges (x_2, r) and $(x_2, l_{2,2})$ (marked in blue and red, respectively) swapped their places as well. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

1. $u' = u$ and $v' \neq v$ (a node $u = u'$ cannot transmit messages to 2 different receivers): there are at most $\Delta_G - 1$ such links e', given fixed link e.
2. $u' = v$ (if u' transmits, it cannot listen at the same time): there are at most Δ_G such links e', given fixed link e.
3. $u' \neq u$ is a neighbor of v (i.e., v can hear both from u and u'): there are at most $\Delta_G - 1$ neighbors of v different than node u, and each of them has, at most, Δ_G different links. This gives $\Delta_G^2 - \Delta_G$ such links e', given fixed link e.

Therefore, in overall there are at most $(\Delta_G - 1) + \Delta_G + (\Delta_G^2 - \Delta_G) = \Delta_G^2 + \Delta_G - 1$ such links. □

The previous lemma shows that USS-PLUS-LIS can be seen as a local-knowledge RoSA, in the sense that it only requires some knowledge about two basic system parameters: the number of links and the network's in-degree.

4.2. *Optimality of* USS-PLUS-LIS

In the next theorem, we show an impossibility result regarding RoSAs, either based on selectors or not, that only make use of upper bounds on the number of links and on the network's degree.

Theorem 5. *No RoSA that only makes use of upper bounds on the number of links and on the network's degree guarantees stability for all networks of degree at most Δ, provided the injection rate $\rho = \omega(1/\Delta^2)$.*

Proof. Assume, to the contrary, that there exists an RoSA ALG such that, given any network of which it is aware of both its number of links and its degree, it guarantees that there are no more than Q_{max} packets in the system at all times against all adversaries with injection rate $\rho = \omega(1/\Delta^2)$. Note that Q_{max} could be a function on ρ, n, but a constant with respect to time.

Consider a complete Δ-regular tree T of depth 2, rooted at r. Let us denote the nodes at distance 1 from r as x_i, for $i = 1, \ldots, \Delta$ and leaves adjacent to x_i as l_i^j for $j = 1, \ldots, \Delta - 1$. Let us generate a family $\mathcal{F}$ of trees $T_{i,j}$ as follows: swap the root r of T with leaf $l_{i,j}$ of T (see Fig. 4). Note that edges (x_i, r) and $(x_i, l_{i,j})$ swapped places, edges (x_k, r) for $k \neq i$ were removed and in their place edges $(x_k, l_{i,j})$ appeared. Other edges, i.e., $(x_k, l_{k,a})$ for $a = 1, \ldots, \Delta - 1$ and $(x_i, l_{i,b})$ for $b \neq j$, remain in the same place in both T and $T_{i,j}$.

Note that edges $(x_k, l_{k,a})$ (for $k = 1, \ldots, \Delta$ and $a = 1, \ldots, \Delta - 1$) exist in every tree in $\mathcal{F} \cup \{T\}$. Let us denote the set of these edges as E.

Consider an adversary $\mathcal{A}$ that, starting from round 0, injects 1 packet into every edge outgoing from x_i (for $i = 1, \ldots, \Delta$) every $1/\rho$ rounds. Such adversary is a $(\rho, 1)$-adversary in each tree in $\mathcal{F} \cup \{T\}$.

Note that each packet injected into an edge incoming into the root of a tree $T' \in \mathcal{F} \cup \{T\}$ cannot be simultaneously transmitted with any other packet injected by $\mathcal{A}$. In particular, it cannot be simultaneously transmitted with any other packet on edges in E.

Consider a time prefix of length τ rounds. Consider any edge $e \in E$. Edge e is incident to the root in some tree $T' \in \mathcal{F} \cup \{T\}$. ALG must successfully transmit from e in T' in at least $\rho\tau - Q_{max}$ rounds during the considered prefix, since ALG is stable. This means that all other edges in E must not transmit in those rounds. Since there are $\Delta(\Delta - 1)$ possible choices of edge $e \in E$, each choice requiring all other edges in E not to transmit in $\rho\tau - Q_{max}$ rounds, we get that each edge in E must not transmit in $\Delta(\Delta - 1) \cdot (\rho\tau - Q_{max})$ rounds and must transmit in $\rho\tau - Q_{max}$ rounds, for a total of $\Delta^2 \cdot (\rho\tau - Q_{max})$ rounds in the prefix of length τ. Since $\rho = \omega(1/\Delta^2)$, we can choose τ such that $\Delta^2 \cdot (\rho\tau - Q_{max}) > \tau$, which gives us a contradiction. □

If we apply Theorem 5 to USS-PLUS-LIS, then our goal is to find how close to $\rho = O(1/\Delta_G^2)$ is its maximum injection rate for which it guarantees stability.

1. Choose m such that $|E(G)| \leq m$.
2. Obtain an (m, ϵ)-generalized-universally-strong selector (for some value of ϵ) of some length t and use it as the transmission schedule.
3. When there are several packets awaiting in a single queue, choose the packet to be transmitted according to ALG, breaking ties in any arbitrary fashion.

Fig. 5. The GUSS-PLUS-ALG RoSA for a network G.

If we consider Theorem 4 with $\Delta = \Delta^H_{in}$, we have that USS-PLUS-LIS can be stable for $\rho = O(1/\Delta^H_{in})$. Furthermore, by Lemma 2 we know that Δ^H_{in} can be as large as $\Theta(\Delta^2_G)$. Then, we have that USS-PLUS-LIS guarantees stability for $\rho = O(1/\Delta^2_G)$ for all networks G, which matches the result in Theorem 5. This proves that USS-PLUS-LIS is asymptotically optimal regarding the injection rate for which stability is guaranteed.

5. The GUSS-PLUS-ALG RoSA

In this section, we extend the results in the previous section so that the new RoSA (which we call GUSS-PLUS-ALG) does not need to have any knowledge of Δ, but only an upper bound on the number of links. The code of GUSS-PLUS-ALG, is shown in Fig. 5. As it can be seen, it is very similar to that in Fig. 3, but now it uses an (m, ϵ)-generalized-universally-strong selector.

Next, we show that GUSS-PLUS-LIS guarantees stability, provided a given packets' injection admissibility condition is fulfilled. At this point, we note that whereas the injection rates for which GUSS-PLUS-ALG guarantees stability depend on Δ (see Theorem 6), the RoSA itself does not use it.

Assuming that nodes do not know any (linear) estimate on Δ, observe that Lemma 5 (in Appendix A) still holds, as the proof is based on existence of a (ρ', T)-frequent schedule $\mathcal{S}$. In order to prove a counterpart of Theorem 4 in the setting without known Δ, we need to revisit the rate ρ' for which (ρ', T)-frequent schedule could be constructed.

Theorem 6. *Given a network G,* GUSS-PLUS-LIS *is stable against any (ρ, b)-adversary, for $\rho < \frac{\epsilon}{\Delta+1}$.*

Proof. Let us take any arbitrary link $z \in E_G$ and consider the set of all other links that conflict with link z, of which there are at most Δ. This means that there exist at least $\epsilon \cdot t/(\Delta+1)$ rows i in M such that $M_{i \bmod t, z} = 1$ and $M_{i \bmod t, c_1} = \cdots = M_{i \bmod t, c_j} = 0$. Therefore, at time i, link z will transmit a message, and no link that conflicts with the link z will transmit. This guarantees that each link will successfully transmit, at least, $\epsilon \cdot t/(\Delta+1)$ messages during any schedule of length t (i.e., we obtained an $(\epsilon/(\Delta+1), t)$-frequent schedule $\mathcal{S}$). Then, we can apply the result in Lemma 5 (in Appendix A) to deduce that such an algorithm is stable against any (ρ, b)-adversary, where $\rho < \frac{\epsilon}{\Delta+1}$. □

Now, by using the selectors provided in Theorem 3, we have the following results:

Corollary 5.

1. *Given a network G, there exists a universally strong selector that, used in* GUSS-PLUS-LIS, *provides stability against any (ρ, b)-adversary, for $\rho < \frac{1}{2e \cdot \lfloor \log_{1+1/e} n \rfloor \cdot (\Delta+1)}$.*
2. *Given a network G, there exists a universally strong selector that, used in* GUSS-PLUS-LIS, *provides stability against any (ρ, b)-adversary, for $\rho < \frac{1}{8 \log_2 n \cdot \lfloor \log_{1+1/(4 \log_2 n)} n \rfloor \cdot (\Delta+1)} = O(\frac{1}{\Delta \ln^3 n})$.*

Corollary 6.

1. *By using the first GUSS from Theorem 3 (unconstructive), each packet spends at most $\frac{(b-1)}{\rho} \cdot \left(\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L} - 1 \right)$ time slots in the system, where $\rho' = \frac{1}{(2e \cdot \lfloor \log_{1+1/e} n \rfloor) \cdot (\Delta+1)}$, L is the length of the longest simple directed path in the network and ρ is the used injection rate such that $\rho < \rho'$. Furthermore, each queue contains at most $\frac{b-1}{\left(1-\frac{\rho}{\rho'}\right)^L} + 1$ packets.*
2. *By using the second GUSS from Theorem 3 (constructive), each packet spends at most $\frac{(b-1)}{\rho} \cdot \left(\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L} - 1 \right)$ time slots in the system, where $\rho' = \frac{1}{8 \log_2 n \cdot \lfloor \log_{1+1/(4 \log_2 n)} n \rfloor \cdot (\Delta+1)} = O(\frac{1}{\Delta \ln^3 n})$, L is the length of the longest simple directed path in the network and ρ is the used injection rate such that $\rho < \rho'$. Furthermore, each queue contains at most $\frac{b-1}{\left(1-\frac{\rho}{\rho'}\right)^L} + 1$ packets.*

1. Use optimal coloring of graph H as the transmission schedule, and repeat it indefinitely.
2. When there are several packets awaiting in a single queue, choose the packet to be transmitted according to Alg, breaking ties in any arbitrary fashion.

Fig. 6. The Coloring-plus-Alg RoSA for graph G.

6. The Coloring-plus-ALG RoSA

In this section, we introduce a global-knowledge packet-oblivious RoSA, which we call Coloring-plus-Alg, that is based on using graph coloring as *transmission schedules*. Such an RoSA does not take into account any historical information. However, it has to be seeded by some information about the network topology (i.e., it is a global-knowledge protocol).

Next, we show that Coloring-plus-LIS (i.e., Coloring-plus-Alg where Alg is the Longest-In-System queueing scheduling policy), guarantees stability, provided a given packets' injection admissibility condition is fulfilled. But before we introduce Coloring-plus-Alg, we state the following fact regarding the relationship between vertex coloring in a conflict graph, and its use as a transmission schedule.

Note that every set of vertices of same color can be extended to a maximal independent set. The resulting family of independent sets is still a feasible schedule that guarantees no conflicts and is no worse than just coloring. In fact, it may allow some links to transmit more than once during the schedule, without increasing the length of the schedule.

Following, we show that coloring of a collision graph can be used to obtain a transmission schedule, where each link is guaranteed to regularly transmit.

Lemma 3. *A k-coloring of collision graph H provides a* $(1/k, k)$*-frequent schedule.*

Proof. Let us split the vertices V_H of the graph H into sets V_H^i for $i = 0, 1, \ldots, k-1$, where every vertex in V_H^i is assigned the i-th color in the vertex coloring of graph H. Each link in the graph G is represented by one vertex in V_H, and therefore each link is assigned a unique color. According to the definition of the conflict graph H, if there is no edge $(u, v) \in E_H$, then links $u \in E_G$ and $v \in E_G$ can deliver their packets simultaneously, without a collision. Therefore, if at a given round t only links of $(t \mod i)$-th color transmit, then no collision occurs. Since each link has a color $i \in \{0, 1, \ldots, k-1\}$ assigned to it, then each link will successfully transmit a packet once each k consecutive rounds (as far as there is one packet waiting in its queue). $\square$

Since $\chi(H)$-coloring is an optimal coloring of graph H, we have the following result.

Corollary 7. *An optimal coloring of collision graph H provides a* $(1/\chi(H), \chi(H))$*-frequent schedule.*

Once we have made it clear that coloring of a collision graph can be used to obtain a transmission schedule, the code of the Coloring-plus-Alg algorithm is shown in Fig. 6.

Now, we show that Coloring-plus-LIS (i.e., Coloring-plus-Alg where Alg is the Longest-In-System scheduling policy), guarantees stability, provided a given packets' injection admissibility condition is fulfilled.

Theorem 7. Coloring-plus-LIS *is stable provided* $\rho < 1/\chi(H)$*, where* $\chi(H)$ *is the chromatic number of the conflict graph H of the network G.*

Proof. We start the proof with referring to Corollary 7, which shows that coloring of a collision graph can be used to obtain a $(1/\chi(H), \chi(H))$-frequent schedule $\mathcal{C}$.

Let us take any $\rho = 1/\chi(H) - \epsilon$, for some $\epsilon > 0$. We can use Lemma 5 with $\mathcal{S} = \mathcal{C}$ (so, $\rho' = 1/\chi(H)$) to show that Coloring-plus-LIS is stable against any (ρ, b)-adversary in the radio network model. $\square$

Observe that, contrary to USS-plus-LIS, Coloring-plus-LIS requires global-knowledge of the structure of the graph: first, to construct H, and then to obtain its optimal coloring.

We can use the optimal coloring of the conflict graph H as a $(1/\chi(H), \chi(H))$-frequent schedule (see Corollary 7) in Lemma 8 (in Appendix A) to obtain the following result regarding both the maximum latencies and queue lengths provided by Coloring-plus-Alg.

Corollary 8. *By using* Coloring-plus-LIS*, each packet spends at most* $\frac{(b-1)}{\rho} \cdot \left(\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L} - 1 \right)$ *time slots in the system, where* $\rho' = 1/\chi(H)$*, L is the length of the longest simple directed path in the network and* ρ *is the used injection rate such that* $\rho < \rho'$*. Furthermore, each queue contains at most* $\frac{b-1}{\left(1-\frac{\rho}{\rho'}\right)^L} + 1$ *packets.*

Table 1
Maximum injection rates and required knowledge.

RoSA	Required knowledge	Maximum injection rate
USS-PLUS-LIS	Bounds on the number of links and on the network's degree	$O(1/(e \cdot \Delta^H))$ (asymptotically optimal)
GUSS-PLUS-LIS	Bound on the number of links	$O(1/(2 \cdot \lfloor \log_{1+1/e} n \rfloor \cdot e \cdot \Delta^H))$ (close to asymptotically optimal)
COLORING-PLUS-LIS	Full topology	$O(1/\Delta^H)$ (optimal)

6.1. *Optimality of* COLORING-PLUS-LIS

Now, we show that COLORING-PLUS-LIS is optimal regarding the injection rate, in the sense that no algorithm can guarantee stability for a higher injection rate than that provided by it.

Theorem 8. *No RoSA can be stable for all networks against a (ρ, b)-adversary for $\rho > 1/\chi(H)$.*

Proof. Let us consider a network graph G on n nodes that is a clique. For such network, the collision graph H is also a clique, since each link is in conflict with each other link. Collision graph H has $n^2 - n$ vertices and requires $n^2 - n$ colors to be colored, i.e., $\chi(H) = n^2 - n$.

Consider a $(1/\chi(H) + \varepsilon, 2)$-adversary for some $\varepsilon > 0$ that after every $\chi(H)$ rounds injects one packet into each link (starting in round 0) and simultaneously after each $1/\varepsilon$ rounds injects another packet into each link (starting in round 0). Therefore, in any prefix of $T = k \cdot \chi(H)$ rounds for $k \in \mathbb{N}$, the adversary injects $(k+1) + \lfloor T/\varepsilon \rfloor + 1$ packets into each link, i.e., $I = (k + \lfloor T/\varepsilon \rfloor + 2) \cdot (n^2 - n)$ packets into the system.

On the other hand, since G is a clique, any RoSA can successfully transmit at most 1 packet per round in the entire network. Therefore, in $T = k \cdot \chi(H) = k \cdot (n^2 - n)$ rounds at most $k \cdot (n^2 - n)$ packets can be transmitted. So, at the end of a prefix of length T, there are at least $I - k \cdot (n^2 - n) = (\lfloor T/\varepsilon \rfloor + 2) \cdot (n^2 - n)$ packets remaining in the system. For T approaching infinity, the number of packets remaining in the queues grows to infinity. This means that the queues are not bounded and the RoSA is not stable. □

7. Injection rates vs required knowledge

In this section, we compare the maximum injection rates for which the RoSAs introduced in the previous sections guarantee stability.

Regarding USS-PLUS-LIS, from the result in Corollary 2 we have that it can only guarantee stability for $\rho = O(1/(e \cdot \Delta^H_{in}))$. Analogously, from the result in Corollary 6 we have that GUSS-PLUS-LIS can only guarantee stability for $\rho = O(1/(e \cdot \Delta^H_{in} \cdot 2 \cdot \lfloor \log_{1+1/e} n \rfloor))$.

On another hand, it can be observed that the injection rate for which GUSS-PLUS-LIS guarantees stability is just $2 \cdot \lfloor \log_{1+1/e} n \rfloor$ times greater than the guaranteed by USS-PLUS-LIS, which shows that GUSS-PLUS-LIS is close to be asymptotically optimal regarding the injection of packets for which stability is guaranteed.

Furthermore, in Section 6 we have introduced a global-knowledge RoSA, which we called COLORING-PLUS-LIS, and we have shown that the maximum injection rate for which it guarantees stability is $1/\chi(H)$, where $\chi(H)$ is the chromatic number of the conflict graph H of the network (see Theorem 7). In addition, it has been also proved that this bound is optimal (see Theorem 8). By the Brooks' theorem [11], we have that $\chi(H) \le \Delta^H + 1$. Let $indeg^H(e)$ (and $outdeg^H(e)$) denote the indegree (outdegree) of node e in graph H. Recall that each edge in the network graph was replaced by two oppositely directed links. This means that, if a link e blocks $outdeg^H(e)$ other links, then the opposite link e' is blocked by $indeg^H(e') = outdeg^H(e)$ links. Therefore, $\Delta^H = \Theta(\Delta^H_{in})$. Then, Theorem 7 guarantees stability for $\rho = O(1/\Delta^H_{in})$. This implies that, by using COLORING-PLUS-LIS, it is possible to guarantee stability for a wider range of injection rates than by using the both GUSS-PLUS-LIS and GUSS-PLUS-LIS; namely, the injection rate for which stability is guaranteed is e times greater in the case of USS-PLUS-LIS, and $e \cdot 2 \cdot \lfloor \log_{1+1/e} n \rfloor$ times greater in the case of GUSS-PLUS-LIS.

Table 1 summarizes the results regarding the maximum injection rates and the required knowledge of each considered RoSA.

8. Extension of the results to other queueing scheduling policies

In this section, we show that the results obtained in Sections 4, 5 and 6 for routing combined with LIS (Longest In System) can be extended to other queueing scheduling policies; namely, NFS (Nearest-From-Source), SIS (Shortest-In-System) and FTG (Farthest-To-Go).

8.1. Reduction to the failure model

First, let us explain the (wired) failure model [1]. Given is a network graph G. A (ρ, b)-adversary in the failure model injects paths (packets) into G and generates failures in such a way that in any interval I the following inequality holds:

$$Arr_e(I) + Fail_e(I) \leq \rho|I| + b ,$$

where $Arr_e(I)$ is the number of packets injected during interval I that pass through edge e and $Fail_e(I)$ is the number of failures on edge e generated during interval I. Each link e that has some packets waiting in its queue can transmit a packet in every round, i.e., there are no collisions between edges.

There are known stable algorithms for packet queueing scheduling in the failure model, such as NFS (Nearest-From-Source), SIS (Shortest-In-System), or FTG (Farthest-To-Go) against (ρ, b)-adversary with any $\rho < 1$ [1].

Lemma 4. *Suppose we have a stable RoSA (called* Alg*) against any (ρ'', b)-adversary ADV_{fail} in the failure model on graph G. Suppose we have a (ρ', T)-frequent schedule $\mathcal{S}$. Then we can build a stable RoSA (called $\mathcal{S}$-*Plus-Alg*) against any (ρ, b)-adversary ADV_{RN} in the radio network model on graph G, for any ρ such that $\rho < \rho'$ and $\rho'' \geq 1 + \rho - \rho'$.*

Proof. The stable RoSA in each round has two steps:

1. Determine which links transmit, according to a (ρ', T)-frequent schedule $\mathcal{S}$ for some parameters ρ' and T,
2. Determine, for each link e, which packet awaiting in a queue of link e to transmit, according to Alg.

We can think of rounds when $\mathcal{S}$ does not successfully transmit a packet via link e due to a collision as failures on link e in the failure model. Schedule $\mathcal{S}$ guarantees that each link e has at most $(1 - \rho')T$ transmission blocked in any interval I of length T. This means that each link e has at most $Fail_e(I) \leq (1 - \rho')T$ failures during I. Furthermore, ADV_{RN} can inject at most $Arr_e(I) \leq \rho T + b$ packets passing through each edge e during I:

$$Arr_e(I) + Fail_e(I) \leq \rho T + b + (1 - \rho')T = T(1 + \rho - \rho') + b .$$

Therefore, the graph G with packet arrivals from ADV_{RN} and failures being collisions generated by $\mathcal{S}$ is an instance of the failure model with a $(1 + \rho - \rho', b)$-adversary. That means that using Alg to compute which packet to choose for each link at each round guarantees stability, provided $\rho'' \geq 1 + \rho - \rho'$. □

8.2. Stability results for NFS, SIS and FTG

Theorem 9. *Given a network G,* USS-plus-Alg *(where* Alg $\in$ *{NFS,SIS,FTG}) is stable against any (ρ, b)-adversary, for $\rho < \frac{\epsilon}{\Delta+1}$. The same result applies to* GUSS-plus-Alg *and* Coloring-plus-Alg

Proof. The proof is similar to these in Theorems 4, 6 and 7. The only difference is that, instead of Lemma 5 (in Appendix A), we can apply the results in Lemma 4 for NFS, SIS and FTG to deduce that such RoSAs are stable against any (ρ, b)-adversary, where $\rho < \frac{\epsilon}{\Delta+1}$. □

9. Conclusions

In this work, we studied the fundamental problem of stability in multi-hop wireless networks. We introduced a number of routing scheduling algorithms which, making use of certain knowledge about the network's structure, guarantee stability for certain injection rates.

We first introduced two new families of combinatorial structures, that were used to provide a set of transmission schedules. Making use of these structures, we proposed two *local-knowledge* packet-oblivious routing scheduling algorithms. The first proposed routing scheduling algorithm only needs to know some upper bounds on the number of links and on the network's degree, and it was shown to be asymptotically optimal regarding the injection rate for which it guarantees stability. The second proposed routing scheduling algorithms was close to be asymptotically optimal, but it only needs to know an upper bound on the number of links. For such algorithms, we also provided some results regarding both the maximum latencies and queue lengths. Furthermore, we also evaluated how the lack of global knowledge about the system topology affects the performance of the routing scheduling algorithms.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Stability results for the *Longest-In-System* queueing scheduling policy

In this section, we introduce some auxiliary results regarding the *Longest-In-System* queueing scheduling policy. First, we show that LIS, combined with a transmission schedule that guarantees a number of successful transmissions in some time interval, guarantees stability for a certain injection rate (these lemmas are adapted versions of analogous results about universal stability of the LIS protocol in wired network [4]).

Lemma 5. *If there exists a* (ρ', T)*-frequent schedule* $\mathcal{S}$*, then using LIS as the queueing policy guarantees stability against any* (ρ, b)*-adversary for* $\rho < \rho'$.

Before we prove this lemma, we will introduce some additional notations and auxiliary lemmas.

Let L be the length of the longest route in the system. Let us denote by class i the set of packets injected during i-th window. A class i is said to be *active* during a window w if and only if at some time during window w there is some packet in the system of class $i' \le i$.

Consider some packet p injected during window W_0, whose path crosses links $e_1, e_2, \ldots, e_L$, in this order. We use W_i to denote the window, during which p crossed link e_i. Let c_w denote the number of active classes during window w. We define $c = \max_{w \in [W_0, W_L)} c_w$. Then, we can bound the number of windows to deliver p.

Lemma 6.

$$W_L - W_0 \le \frac{1 - \left(1 - \frac{\rho}{\rho'}\right)^L}{\rho \cdot T} \cdot (b-1) + c \cdot \left[1 - \left(1 - \frac{\rho}{\rho'}\right)^L\right].$$

Proof. The packet p reaches link e_i for the first time in window W_{i-1}. Since p is in the system, during window W_{i-1} all classes $[W_0, W_{i-1}]$ are active. Therefore, according to the definition of c, there are at most $c - (W_{i-1} - W_0)$ active classes with packets older than packet p. Packets in those classes are the only packets that take priority over packet p on link e_i. The oldest such packet was injected during window $w_{first} = W_0 - [c - (W_{i-1} - W_0)] = W_{i-1} - c$. Since its injection, at most $(W_0 - w_{first}) \cdot \rho \cdot T + b = [c - (W_{i-1} - W_0)] \cdot \rho \cdot T + b$ packets older than p could be injected into the system. Therefore, there are at most $[c - (W_{i-1} - W_0)] \cdot \rho \cdot T + b - 1$ packets that will take priority over packet p on link e_i. Since each link transmits at least $\rho' T$ times per window, the number of windows until p transgresses link e_i is at most

$$W_i - W_{i-1} \le \frac{\rho \cdot T \cdot (c + W_0 - W_{i-1}) + b - 1}{\rho' \cdot T}.$$

Hence,

$$W_i \le \left(1 - \frac{\rho}{\rho'}\right) W_{i-1} + \frac{\rho}{\rho'}(c + W_0) + \frac{b-1}{\rho' \cdot T}.$$

Therefore, solving the recurrence, we get:

$$W_L \le W_0 + c\left[1 - (1 - \frac{\rho}{\rho'})^L\right] + \frac{1 - (1 - \frac{\rho}{\rho'})^L}{\rho \cdot T}(b-1),$$

which proves the lemma. □

Now we have a bound on how long packet p can be in the system, depending on value c. We will show that c is bounded by a constant, depending only on network and adversary parameters, i.e., L, ρ and b, and value ρ' from Lemma 5.

Lemma 7. *There are never more than*

$$(b-1) \cdot \frac{1 - (1 - \frac{\rho}{\rho'})^L}{(1 - \frac{\rho}{\rho'})^L \cdot \rho \cdot T}$$

active classes in the system.

Proof. Let $c' = (b-1) \cdot \frac{1-(1-\frac{\rho}{\rho'})^L}{(1-\frac{\rho}{\rho'})^L \cdot \rho \cdot T} + \frac{1}{(1-\frac{\rho}{\rho'})^L}$. Assume, by contradiction, that a window w is the first window during which there are at least $c' + 1$ active classes. Hence, at the end of window $w - 1$, there is a packet q that was in the system for c' windows, and no more than c' classes were active until the end of window $w - 1$.

According to Lemma 6, packet q is delivered in at most

$$c'\left[1-(1-\frac{\rho}{\rho'})^L\right]+\frac{1-(1-\frac{\rho}{\rho'})^L}{\rho\cdot T}(b-1)=$$
$$=c'\left[1-(1-\frac{\rho}{\rho'})^L\right]+\left(c'-\frac{1}{(1-\frac{\rho}{\rho'})^L}\right)\cdot(1-\frac{\rho}{\rho'})^L=$$
$$=c'-1$$

windows, which gives a contradiction. □

Now that we have proven that any packet p spends bounded time in the system, we can prove Lemma 5.

Proof of Lemma 5. In Lemma 7, it has been shown that c is bounded. By Lemma 7, this implies that $W_L - W_0$ is also bounded. This result guarantees that each packet spends a bounded time in the system. That means that such system is stable against any (ρ, b)-adversary, provided that $\rho' > \rho$, which completes the proof of the lemma. □

Regarding both the maximum latencies and queue lengths, we have the following result.

Lemma 8. *Assume we have a (ρ', T)-frequent schedule $\mathcal{S}$. Then, each packet spends at most $\frac{(b-1)}{\rho}\cdot\left(\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}-1\right)$ time slots in the system, where L is the length of the longest simple directed path in the network and ρ is the used injection rate such that $\rho < \rho'$. Furthermore, each queue contains at most $\frac{b-1}{\left(1-\frac{\rho}{\rho'}\right)^L}+1$ packets.*

Proof. From Lemmas 6 and 7, we have that:

$$W_L - W_0 \le \frac{1-\left(1-\frac{\rho}{\rho'}\right)^L}{\rho\cdot T}\cdot(b-1)+c\cdot\left[1-\left(1-\frac{\rho}{\rho'}\right)^L\right]\le$$
$$\le\frac{1-\left(1-\frac{\rho}{\rho'}\right)^L}{\rho\cdot T}\cdot(b-1)+(b-1)\cdot\frac{1-(1-\frac{\rho}{\rho'})^L}{(1-\frac{\rho}{\rho'})^L\cdot\rho\cdot T}\cdot\left[1-\left(1-\frac{\rho}{\rho'}\right)^L\right]=$$
$$=\frac{(b-1)}{\rho\cdot T}\cdot\left(1-\left(1-\frac{\rho}{\rho'}\right)^L\right)\left[1+\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}-1\right]=$$
$$=\frac{(b-1)}{\rho\cdot T}\cdot\left(1-\left(1-\frac{\rho}{\rho'}\right)^L\right)\left[\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}\right]=$$
$$=\frac{(b-1)}{\rho\cdot T}\cdot\left[\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}-1\right],$$

which provides the maximum number of windows in which a given packet is in the system. Since windows have a length of T then the maximum number of time slots that any packet spends in the system is:

$$T\cdot\frac{(b-1)}{\rho\cdot T}\cdot\left[\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}-1\right]=\frac{(b-1)}{\rho}\cdot\left[\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}-1\right].$$

Using the previous result, it is immediate to find that no queue contains more than

$$\rho\cdot\frac{(b-1)}{\rho}\cdot\left[\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}-1\right]+b=(b-1)\cdot\left[\frac{1}{\left(1-\frac{\rho}{\rho'}\right)^L}-1\right]+b=\frac{b-1}{\left(1-\frac{\rho}{\rho'}\right)^L}+1$$

packets. □

References

[1] Carme Àlvarez, Maria J. Blesa, Josep Díaz, Maria J. Serna, Antonio Fernández, Adversarial models for priority-based networks, Networks 45 (1) (2005) 23–35.
[2] Carme Àlvarez, Maria J. Blesa, Maria J. Serna, A characterization of universal stability in the adversarial queuing model, SIAM J. Comput. 34 (1) (2004) 41–66.
[3] Lakshmi Anantharamu, Bogdan S. Chlebus, Dariusz R. Kowalski, Mariusz A. Rokicki, Packet latency of deterministic broadcasting in adversarial multiple access channels, J. Comput. Syst. Sci. 99 (2019) 27–52.
[4] Matthew Andrews, Baruch Awerbuch, Antonio Fernández, Frank Thomson Leighton, Zhiyong Liu, Jon M. Kleinberg, Universal-stability results and performance bounds for greedy contention-resolution protocols, J. ACM 48 (1) (2001) 39–69.
[5] Matthew Andrews, Lisa Zhang, Scheduling over a time-varying user-dependent channel with applications to high-speed wireless data, J. ACM 52 (5) (2005) 809–834.
[6] Matthew Andrews, Lisa Zhang, Routing and scheduling in multihop wireless networks with time-varying channels, ACM Trans. Algorithms 3 (3) (2007) 33.
[7] Rajat Bhattacharjee, Ashish Goel, Zvi Lotker, Instability of FIFO at arbitrarily low rates in the adversarial queueing model, SIAM J. Comput. 34 (2) (2004) 318–332.
[8] Maria J. Blesa, Daniel Calzada, Antonio Fernández, Luis López, Andrés L. Martínez, Agustín Santos, Maria J. Serna, Christopher Thraves, Adversarial queueing model for continuous network dynamics, Theory Comput. Syst. 44 (3) (2009) 304–331.
[9] Allan Borodin, Jon M. Kleinberg, Prabhakar Raghavan, Madhu Sudan, David P. Williamson, Adversarial queuing theory, J. ACM 48 (1) (2001) 13–38.
[10] Allan Borodin, Rafail Ostrovsky, Yuval Rabani, Stability preserving transformations: packet routing networks with edge capacities and speeds, J. Interconnect. Netw. 5 (1) (2004) 1–12.
[11] Rowland Leonard Brooks, On colouring the nodes of a network, Math. Proc. Camb. Philos. Soc. 37 (2) (1941) 194–197.
[12] B.S. Chlebus, V. Cholvi, P. Garncarek, T. Jurdziński, D.R. Kowalski, Routing in wireless networks with interferences, IEEE Commun. Lett. 21 (9) (2017) 2105–2108.
[13] Bogdan S. Chlebus, Vicent Cholvi, Dariusz R. Kowalski, Universal stability in multi-hop radio networks, J. Comput. Syst. Sci. 114 (2020) 48–64.
[14] Bogdan S. Chlebus, Dariusz R. Kowalski, Andrzej Pelc, Mariusz A. Rokicki, Efficient distributed communication in ad-hoc radio networks, in: Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP), Part II, in: Lecture Notes in Computer Science, vol. 6756, Springer, 2011, pp. 613–624.
[15] Bogdan S. Chlebus, Dariusz R. Kowalski, Mariusz A. Rokicki, Maximum throughput of multiple access channels in adversarial environments, Distrib. Comput. 22 (2) (2009) 93–116.
[16] Bogdan S. Chlebus, Dariusz R. Kowalski, Mariusz A. Rokicki, Adversarial queuing on the multiple access channel, ACM Trans. Algorithms 8 (1) (2012) 5.
[17] Vicent Cholvi, Juan Echagüe, Stability of FIFO networks under adversarial models: state of the art, Comput. Netw. 51 (15) (2007) 4460–4474.
[18] Vicent Cholvi, Pawel Garncarek, Tomasz Jurdzinski, Dariusz R. Kowalski, Optimal packet-oblivious stable routing in multi-hop wireless networks, in: Structural Information and Communication Complexity - 27th International Colloquium, SIROCCO 2020, Paderborn, Germany, June 29 - July 1, 2020, Proceedings, Springer, 2020, pp. 165–182.
[19] Vicent Cholvi, Dariusz R. Kowalski, Bounds on stability and latency in wireless communication, IEEE Commun. Lett. 14 (9) (2010) 842–844.
[20] Andrea E.F. Clementi, Angelo Monti, Riccardo Silvestri, Distributed broadcast in radio networks of unknown topology, Theor. Comput. Sci. 302 (1) (2003) 337–364.
[21] Juan Echagüe, Vicent Cholvi, Antonio Fernández, Universal stability results for low rate adversaries in packet switched networks, IEEE Commun. Lett. 7 (12) (2003) 578–580.
[22] Antonio Fernández Anta, Miguel A. Mosteiro, Christopher Thraves, Deterministic recurrent communication in restricted sensor networks, Theor. Comput. Sci. 418 (2012) 37–47.
[23] David Gamarnik, Stability of adaptive and nonadaptive packet routing policies in adversarial queueing networks, SIAM J. Comput. 32 (2) (2003) 371–385.
[24] Pawel Garncarek, Tomasz Jurdziński, Dariusz R. Kowalski, Local queuing under contention, in: Proceedings of the 32nd International Symposium on Distributed Computing (DISC), in: Leibniz International Proceedings in Informatics, vol. 121, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 28.
[25] Pawel Garncarek, Tomasz Jurdziński, Dariusz R. Kowalski, Stable memoryless queuing under contention, in: Proceedings of the 33rd International Symposium on Distributed Computing (DISC), in: Leibniz International Proceedings in Informatics, vol. 146, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 17.
[26] Ashish Goel, Stability of networks and protocols in the adversarial queueing model for packet routing, Networks 37 (4) (2001) 219–224.
[27] Dimitrios Koukopoulos, Marios Mavronicolas, Sotiris E. Nikoletseas, Paul G. Spirakis, The impact of network structure on the stability of greedy protocols, Theory Comput. Syst. 38 (4) (2005) 425–460.
[28] Sungsu Lim, Kyomin Jung, Matthew Andrews, Stability of the max-weight protocol in adversarial wireless networks, IEEE/ACM Trans. Netw. 22 (6) (2014) 1859–1872.
[29] Zvi Lotker, Boaz Patt-Shamir, Adi Rosén, New stability results for adversarial queuing, SIAM J. Comput. 33 (2) (2004) 286–303.