

Stability in Routing: Networks and Protocols

Marios Mavronicolas*
Department of Computer Science
University of Cyprus
Nicosia, CY-1678, Cyprus

(MAY 15, 2001)

Abstract

We provide a state-of-the-art survey of results concerning *stability* issues for *routing* in *communication networks*, and corresponding *protocols*. Roughly speaking, a distributed routing protocol is *stable* on some particular network if the protocol maintains a bounded number of packets in the network at all times, under some suitable assumptions on the fashion of injecting packets into the network. Fundamental questions that pose themselves in such a setting include:

- What is the effect of the injection pattern on stability?
- Which protocols are stable on which networks?
- What can be said about an algorithmic characterization of stability?

In this survey, we summarize some of the known answers to these questions, and some questions for which no answers are yet known, as a collection of current challenges for the distributed computing community.

*Partially supported by funds from the Joint Program of Scientific and Technological Collaboration between Greece and Cyprus (research project "Efficiency and Performance of Distributed Systems"), and by funds for the promotion of research at University of Cyprus. Email: mavronic@ucy.ac.cy

1 Introduction

Transferring and delivering messages among different processing units is one of the most basic functionalities of a distributed *communication network*. This functionality is taken up by following some particular *routing protocol*, which decides, in a distributed fashion, which message to forward along which *link* of the network at each time step. This routing mechanism is invoked at every link of the network, and for each and every (discrete) time step, while it must ensure that all messages from all possible sources to all possible destinations are delivered in a timely fashion.

A basic challenge encountered by any distributed routing protocol is to come up with some policy for *prioritizing* packets that find themselves queued up for traversing the same link simultaneously. This policy must provide certain *QoS* guarantees on how much time it takes a packet to travel from source to destination (including the time waiting in queues). In order for such a guarantee to be possible, the situation where packets end up waiting in queues for an indefinite amount of time needs to be excluded in the first place. Thus, a protocol is *stable* on a particular communication network if the number of packets in the network remains, at all times, bounded from above by some quantity *independent* of time (but possibly dependent on characteristics of the individual network). So, issues of *stability* have recently attracted a lot of interest and attention within the broader community of the theory of computing; see, e.g., [1, 2, 3, 4, 5, 6, 7, 8, 14, 16, 17, 18, 19, 20, 27, 33] for a flavor of the obtained results.

In this survey, we advocate that the progress made so far on this fascinating subject presents a rich opportunity to the distributed computing community for making fundamental contributions in this area. It is the mission of this survey to provide a glimpse of the results obtained so far and the various open problems and research directions that naturally lend themselves in this rich context.

There are three distinct dimensions across which stability can be studied.

- The first dimension concerns the *traffic pattern*; in other words, the fashion in which packets are injected into the network. There are several models that have been studied so far; some of them (studied in the queueing theory and systems literature [22]) make probabilistic assumptions on the traffic pattern, while others (motivated by more general approaches adopted within the Theoretical Computer Science community) make no assumptions at all and naturally lend themselves to a *worst-case* analysis of stability, much in the same way computational algorithms have been traditionally analyzed (see, e.g., [10]).

For the latter class of models, a necessary condition of stability is that the rate of injecting traffic into the network not exceed the processing speed of the links (expressed as the number of packets a link may forward per unit of time); if each link can forward at most one packet at each time slot, the rate of traffic injection may not exceed one. Interestingly, recent research [7, 3] has revealed that this necessary condition fails to be sufficient for stability.

- The second dimension has to do with the particular *protocol* used. Central questions that follow this dimension include, e.g., the identification of possible trade-offs between the

simplicity of the distributed protocol and the time delay guarantee it provides. In general, one will prefer (simple and distributed) protocols that remain stable for *all* networks. Such protocols are said to be *universally stable*. Furthermore, a particular class of protocols that have attracted much attention are the *greedy* protocols which always forward a packet whenever they can. Recent research has provided interesting (and, in some cases, unexpected and even surprising) dichotomies among simple, distributed protocols with respect to their properties of universal stability.

- The last dimension considers the particular *network topology*, and aims to capture the effect to which network topology is responsible for stability or instability of any particular protocol (or, of a family of protocols). A desirable feature for a network is that it remains stable under *all* protocols (in a given class). Such networks are called *universally stable*; research results have contributed significantly to our understanding of the universal stability property for networks (and how it relates to structural properties of the network).

In a nutshell, this survey illuminates the interplay of these three, crucial dimensions with respect to stability. Some known answers are given, while many more questions are still awaiting in the research pipeline. It is hoped that the picture of this interplay that you will pick up when you read this survey will encourage you to consider working on some of the tantalizing questions that remain open.

The rest of this survey is organized as follows. Common routing protocols are summarized in Section 2. Section 3 provides a juxtaposition of traffic models studied in the context of stability in routing. Results on packet-switched networks are discussed in Section 4. Section 5 surveys results on the stability of networks made up of input-queued switches. Results on the decidability (and undecidability) of stability are surveyed in Section 6. Concluding comments and some general directions for future research are gathered in Section 7.

2 Protocols

In this section, we describe some common routing protocols whose stability properties will be reviewed in the rest of this survey. All of these protocols provide simple rules on how to choose which packet to forward on a given link among the waiting packets.

- *Furthest-to-Go*, abbreviated as FTG; it gives priority to the packet whose distance (in number of links) to its destination is *maximum*.
- *Nearest-to-Source*, abbreviated as NTS; it gives priority to the packet whose traversed distance is *minimum*.
- *Longest-in-System*, abbreviated as LIS; it gives priority to the packet injected the earliest.
- *Shortest-in-System*, abbreviated as SIS; it gives priority to the packet most recently injected.

- *First-in-First-out*, abbreviated as FIFO; it maintains the edge queue in *first-in-first-out* order.
- *Last-in-First-out*, abbreviated as LIFO; it maintains the edge queue in *last-in-first-out* order.
- *Nearest-to-Go*, abbreviated as NTG; it gives priority to the packet whose distance to its destination is *minimum*.
- *Furthest-from-Source*, abbreviated as FFS; it gives priority to the packet whose traversed distance from its source is *maximum*.

Curiously, it turns out that these protocols can exhibit quite different stability behavior on many common protocols.

3 Traffic Models

In this section, we describe some of the models of injecting traffic into the network. Before giving the basic definitions for the various models that appeared in the literature, we review some general terms and concepts.

3.1 Stochastic Networks

The model of *stochastic networks* has been widely studied in the literature on queueing theory and systems. A good starting point to follow related research work is the recent encyclopedic monograph by Serfozo [32]; the older book by Kelly [22] remains also a standard reference for the field of stochastic networks. The distinguishing feature of a stochastic network (which has been commonly criticized) is that of making *stochastic* assumptions about the various system parameters, such as packet sizes or service times; such assumptions vary widely, encompassing issues of stationarity, stochastic independence and stochastic ordering, to name a few.

A particular type of a stochastic network arises when we assume that the distribution of the *time service* at a server is *session-independent*; in other words, time services of different servers follow the same distribution. Such networks are known as *Kelly networks* [22]. An early result identifies conditions on a stochastic network for FIFO to be stable.

Theorem 3.1 (Kelly [21]) *Consider a Kelly network \mathcal{N} , where packet sizes and service times are exponentially distributed. Then, FIFO is stable for \mathcal{N} .*

The precise role of the particulars of the distribution of packet sizes and service times for FIFO to be stable for a Kelly-type network is not understood yet. Hence, we ask:

Open Problem 3.1 *Find a characterization of the probability distributions of packet sizes and service times in a Kelly network that imply stability of FIFO for such networks.*

No conditions are known for protocols other than FIFO to be stable (or unstable) in the stochastic network model, even when one is restricted to Kelly networks. Hence, one may ask:

Open Problem 3.2 *Identify probability distributions of packet sizes and service times in a Kelly network that make each of FTG, NTS, LIS, SIS, LIFO, NTG and FFS to be stable or unstable in the stochastic networks model.*

Some additional papers that treat stability issues for stochastic networks include [9, 13, 15, 24, 31].

3.2 The (σ, ρ) -Regulated Session Model

In the (σ, ρ) -regulated session model, packets are injected by an adversary along fixed *paths* in the network. Each path consists of a sequence of edges. We shall refer to these paths, together with the paths injected into them, as *sessions*.

Each session i is associated with a *rate* ρ_i and a *burst parameter* σ_i . Denote $A_i(t_1, t_2)$ the total size of the packets of session i injected into the network during the time interval $[t_1, t_2)$. We say that session i is (σ_i, ρ_i) -regulated if

$$A_i[t_1, t_2) \leq \rho_i(t_2 - t_1) + \sigma_i,$$

for all times t_1 and t_2 with $t_1 < t_2$.

Thus, the eventual rate at which traffic is injected into the network along session i is bounded by ρ_i . The term σ_i facilitates the injection of short *bursts* of packets into the network.* The *adversary* in the (σ, ρ) -regulated model is the abstract entity that arbitrarily injects packet into the network subject to the condition above.

3.3 Adversarial Queueing Model

The *adversarial queueing model* was introduced in a pioneering paper by Borodin *et al.* [7], in an effort to replace probabilistic assumptions on various system parameters, with (safer) worst-case ones.

In the adversarial queueing model, packets are injected into the network by an adversary subject to the following condition: for any edge e of the network, and for any time interval $[t_1, t_2)$, the total size (number) of the packets injected during the interval $[t_1, t_2)$ that wish to pass through edge e is at most $w + r(t_2 - t_1)$, for some $w \geq 0$ and $r \leq 1$. The *adversary* for the adversarial queueing model is defined much in the same way as the adversary for the (σ, ρ) -regulated model. It has been noted that the adversary for the adversarial queueing model is more powerful than the adversary for the (σ, ρ) -regulated model (cf. Andrews [2, Section 1]).

*In practice, one can use a "leaky bucket" to implement a session that is (σ_i, ρ_i) -regulated (see, e.g., [26]).

4 Packet-Switched Networks

In this section, we survey stability and instability results for the model of packet-switched networks. Networks with permanent sessions and temporary sessions are discussed in Sections 4.1 and 4.2, respectively.

4.1 Permanent Sessions

In the *permanent sessions* model, packets are routed over virtual circuits, called *sessions*, that remain fixed over time.

A fundamental result by Borodin *et al.* establishes some kind of a universal stability for any greedy protocol, when one is restricted to trees or arbitrary *directed* acyclic graphs (denoted as DAGs).

Theorem 4.1 (Borodin *et al.* [7]) *Consider the adversarial queueing model with permanent sessions, and assume that \mathcal{N} is a tree or a DAG. Then, any greedy protocol is stable for \mathcal{N} under any adversary \mathcal{A} with adversary rate $\rho_{\mathcal{A}} \leq 1$.*

Unfortunately, as the next result establishes, Theorem 4.1 does not extend to networks containing directed cycles.

Theorem 4.2 (Borodin *et al.* [7]) *Consider the adversarial queueing model with permanent sessions, and assume that \mathcal{N} is a ring on $n \geq 3$ nodes. Then, FIFO and LIS are unstable for \mathcal{N} under an adversary \mathcal{A} with adversary rate $\rho_{\mathcal{A}} = 1$.*

Theorem 4.2 dashes the possibility that FIFO and LIS be universally stable protocols. Most importantly, it also raises the question of identifying *instability thresholds* for such protocols, defined as the least adversary rate for which the protocols are unstable.

It is very interesting to note that Theorem 4.2 does *not* hold for *every* greedy protocol, since the next result identifies a protocol that is stable on the ring.

Theorem 4.3 (Borodin *et al.* [7]) *Consider the adversarial queueing model, and assume that \mathcal{N} is a ring on $n \geq 3$ nodes. Then, FTG is stable for \mathcal{N} under any adversary \mathcal{A} with adversary rate $\rho_{\mathcal{A}} \leq 1$.*

A class of universally stable networks was identified for the first time in a pioneering work by Andrews *et al.* [3] that immediately followed [7].

Theorem 4.4 (Andrews *et al.* [3]) *Consider the adversarial queueing model with permanent sessions. Then, FTG, NTS, SIS and LIS are universally stable.*

Stability and instability properties of FIFO have been further studied in a very recent paper by Diaz *et al.* [14]; this paper presents both positive and negative results. The positive result came in response to an open question posed by Andrews *et al.* [3]. More specifically, Andrews *et al.* [3] asked whether there exists some *universal* constant $r_{\text{FIFO}} > 0$, which is intrinsic to the particular FIFO protocol such that FIFO is stable on *every* network and under any adversary \mathcal{A} with adversary rate $\rho_{\mathcal{A}} \leq r_{\text{FIFO}}$. Diaz *et al.* [14] provide a partial answer to the question of Andrews *et al.*, showing that for every network, FIFO is stable against any adversary with a small injection rate, that depends on some parameters of the network.

Theorem 4.5 (Diaz *et al.* [14]) *Consider the adversarial queueing model. Then, for any network \mathcal{N} , there exists a constant $r_{\mathcal{N}}$, where $0 < r_{\mathcal{N}} < 1$, FIFO is stable on \mathcal{A} under any adversary \mathcal{A} with adversary rate $\rho_{\mathcal{A}} \leq r_{\mathcal{N}}$.*

Moreover, Diaz *et al.* [14] establish:

Theorem 4.6 (Diaz *et al.* [14]) *Consider the adversarial queueing model. Then, there exists a network \mathcal{N} and an adversary \mathcal{A} for which FIFO is unstable for any adversary rate $\rho_{\mathcal{A}} \geq 0.8357$.*

Theorem 4.6 provides the current record (0.8357) for the instability of threshold. Whether or not 0.8357 can be further improved remains a fascinating challenge.

Open Problem 4.1 *Determine the instability threshold of FIFO for packet-switched networks.*

Most studies of packet routing networks have been assuming that one packet can be forwarded on an edge in a single time step. This assumption makes sense for networks with *identical* communication links. In a very recent paper, Borodin *et al.* [8] pursue the direction where link capacities (the number of packets that can be forwarded at each time step) may vary. We sample a typical result from [8].

Theorem 4.7 *Consider the adversarial queueing model with permanent sessions and dynamic link capacities. There is a network \mathcal{N} with dynamic capacities and an adversary \mathcal{A} for which LIS is not stable. More specifically, if all dynamic link capacities are either 1 or C , for some some constant $C > 1$, then LIS is unstable for any adversary rate $\rho_{\mathcal{A}} > \frac{C}{2C-1}$.*

This line of research for networks with dynamic edge capacities is still in its infancy and numerous open problems remain. We sample one that is specifically related to Theorem 4.7.

Open Problem 4.2 *Consider the adversarial queueing model with permanent sessions and dynamic link capacities. Assume that all dynamic link capacities are either 1 or C , for some some constant $C > 1$. Determine the instability threshold of LIS as a function of C .*

See [8] for additional open problems in this direction.

Finally, we mention that Andrews [2] has extended Theorem 4.2 to the (σ, ρ) -regulated model, thus demonstrating that the instability of FIFO is not an artifact of the adversarial queueing model.

Theorem 4.8 (Andrews [2]) *Consider the (σ, ρ) -regulated model with permanent sessions. Then, there is a network \mathcal{N} and an adversary \mathcal{A} with adversary rate strictly less than 1 for which FIFO is unstable.*

4.2 Temporary Sessions

Not many things are known for the model with temporary sessions. We refer the reader to the work of Andrews and Zhang [5] for related results. Interestingly, we note that Andrews and Zhang provide *separation results* of different protocols with respect to their performance in the model with temporary sessions.

5 Input-Queued Switches

In this section, we describe results on the stability of networks with input-queued switches. This model is somehow different, but complementary, than those of packet-switching considered so far in this survey. We start with a description of the model.

5.1 Model

In an *input-queued switch*, packets accumulate forming queues at the input wires of the switch. At each discrete time step, at most one packet is sent from each input wire and at most one packet is forwarded to each output wire; in other words, a *matching* is formed between input and output wires at each step. An input-queued switch is *stable* if the number of packets waiting to traverse the switch remains bounded as long as no input or output wire is overloaded. We continue to explain what overloaded means.

Consider first the *permanent sessions* model (cf. Section 4.1); thus, sessions in the network are *fixed* over time in this model. It is assumed that traffic injection obeys the requirements in the (σ, ρ) -regulated model (see Section 3.2). An injection pattern is *admissible* if for every wire x (input or output) in the network,

$$\sum_{i \in \mathcal{S}_x} \rho_i \leq 1 - \epsilon,$$

where \mathcal{S}_x is the set of sessions traversing wire x and $1 - \epsilon$ is a switch parameter called *utilization*. (See Section 4 for the definition of ρ_i .)

Consider now the *temporary sessions model* (cf. Section 4.2). An injection pattern is *admissible* if for any wire x and for any time interval $[t_1, t_2)$,

$$A_x(t_1, t_2) \leq w + (1 - \varepsilon)(t_2 - t_1),$$

where w is the *burst size* and $1 - \varepsilon$ is as in the temporary sessions model.

It is noted that there are more admissible injection patterns for the temporary sessions model than for the permanent sessions model (see [6, Section I.A]). Thus, positive (resp., negative) results for the temporary sessions model (resp., permanent sessions model) immediately hold for the permanent sessions model (resp., temporary sessions model) as well.

5.2 The Protocol LQF

A very natural protocol for input-queued switches is the *Longest-Queue-First* protocol, abbreviated as LQF. The LQF protocol calculates a maximum-weight bipartite matching (cf. [23]) at each discrete step, where the weight of the edge from input wire i to output wire j is taken to be the number of packets that are waiting to be forwarded from i to j . LQF has been studied by McKeown *et al.* [25], who establish:

Theorem 5.1 (McKeown *et al.* [25]) *LQF is stable for an input-queued switch.*

One unpleasant feature of Theorem 5.1 is that it applies to a (trivial) network consisting of a single input-queued switch. The case of a general network was subsequently considered by Andrews and Zhang [6], who establish both negative and positive results. A somehow surprising, negative result in the work of Andrews and Zhang demonstrates that the behavior of LQF (with respect to stability) changes diametrically when one passes to non-trivial networks.

Theorem 5.2 (Andrews & Zhang [6]) *There exists a network \mathcal{N} of input-queued switches and an adversary A for which LQF is unstable for some injection pattern admissible by the adversary.*

To establish the general instability of LQF, Andrews and Zhang construct a simple network consisting of just 8 switches. It is interesting to know whether the network of Andrews and Zhang is the *smallest* possible network that allows instability for LQF.

Open Problem 5.1 *Construct a network with less than 8 input-queued switches for which LQF can be unstable (under an appropriate adversary), or show that no such network exists.*

A notable feature of the instability proof for Theorem 5.2 is that the proof does not specify how traffic sharing the same pair of an input and an output wire is specified. Hence, Theorem 5.2 hold regardless of the particular pattern of traffic sharing. Even more so, the proof uses a traffic pattern that remains fixed over time.

The network used to show the instability of LQF remains also an example to show the instability of a similar protocol called *Longest-Port-First* and abbreviated as LPF. The protocol LPF computes a maximum-weight bipartite matching, where the weight of the edge from input wire i to output wire j is taken to be the number of packets queued at input wire i plus the number of packets that have output wire j as their destination.

5.3 The Protocol LIN

Now for the good news. Andrews and Zhang [6] go on to describe a protocol called *Longest-in-Network* and abbreviated as LIN that remains stable for networks of input-queued switches. (The protocol LIN was originally introduced in [4].)

The protocol LIN is a *frame-based* protocol; roughly speaking, LIN partitions time into *frames* and computes matchings for each time frame, which it uses to decide which packets to forward from which input wire to which output wire. Each time frame used by LIN has length $\lceil 2w/\varepsilon \rceil$, where w and ε are the constants used to define the temporary sessions model for networks of input-queued switches.

Andrews and Zhang [6] analyze the performance of LIN in the model with permanent sessions; they pose as an open problem the improvement of its performance when restricted to networks with permanent sessions.

6 Is Stability Decidable?

In this section, we consider the fundamental question of algorithmically deciding stability of a given protocol in a given queueing system. Decidability and undecidability results are surveyed in Sections 6.1 and 6.2, respectively.

6.1 Decidability Results

In this section, we survey some decidability results for the adversarial queueing model. These results come from [3, 20].

We have already seen in Section 4 that there are networks that are universally stable, and that there are networks that are *not* universally stable. This apparent dichotomy establishes that universal stability of networks is a non-trivial property, and it naturally poses the question of characterizing networks that are universally stable, and (possibly) algorithmically recognizing such networks. Since universal stability of a network is a predicate quantifying over all protocols and adversaries, it may at first appear that universal stability is not a decidable property. One of the deepest results in the context of network stability in the adversarial queueing model establishes that, to the contrary, this is not the case. To describe this result, we need first some graph-theoretic definitions. (In giving these definitions, we will be using graphs and networks as synonyms.)

Let us recall the definition of *edge contraction* in a graph \mathcal{N} (cf. [34, Definition 2.2.7]).

In a graph \mathcal{N} , contraction of edge e with endpoints u and v is the replacement of u and v with a single vertex whose incident edges are the edges other than e that were incident to either u or v . The resulting graph $\mathcal{N} \setminus e$ has one edge less than \mathcal{N} .

Say that a network \mathcal{M} is a *minor* of a network \mathcal{N} if \mathcal{M} can be obtained from a subgraph of \mathcal{N} by zero or more applications of edge contraction. A set of networks \mathbf{N} is *minor-closed* if whenever $\mathcal{N} \in \mathbf{N}$, every minor of \mathcal{N} is also in \mathbf{N} . A result of Andrews *et al.* [3] in the direction of characterizing universally stable networks establishes that the class of universally stable networks is minor-closed:

Theorem 6.1 (Andrews *et al.* [3]) *Consider the adversarial queueing model. Assume that \mathcal{N} is universally stable. Then, so is every minor \mathcal{M} of \mathcal{N} .*

We discuss the algorithmic implications of Theorem 6.1. An important graph-theoretic result of Robertson and Seymour [28, 29] shows that if \mathbf{N} is any arbitrary minor-closed set of graphs, then there is a *finite* set of graphs $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k$ such that $\mathcal{N} \in \mathbf{N}$ if and only if none of $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k$ is a minor of \mathcal{N} . In other words, any minor-closed collection of graphs is defined by the exclusion of a finite set of graphs as minors. In addition, Robertson and Seymour [30] provide an $O(n^3)$ algorithm to test whether any fixed graph \mathcal{M} is a minor of an arbitrary graph \mathcal{N} with n vertices. Hence, it follows that there is an $O(n^3)$ algorithm to test for membership in any arbitrary minor-closed set of graphs. This has been further strengthened by Robertson and Seymour [28, 29, 30] to show that if a minor-closed set \mathbf{G} does not contain all planar graphs, then membership in \mathbf{G} can be decided in time $O(n^2)$.

What do these graph-theoretic results imply about deciding universal stability of networks? By Theorem 6.1, the class of universally stable networks is a minor-closed class. The network used to prove instability of FIFO in Theorem 4.6 is a planar graph (and there are more examples of planar graphs that have been used to demonstrate instability of specific protocols); hence, the class of universally stable networks does not contain all planar networks. In total, it follows that universal stability (for the adversarial queueing model) can be decided in time $O(n^2)$.

The approach of Andrews *et al.* [3] to establishing algorithmic decidability for the class of universally stable networks is, in some sense, *indirect*, since it requires tools from the theory of graph minors (developed in a series of papers by Robertson and Seymour), while it does not provide an explicit characterization of the class of networks that are universally stable. Goel [20] has filled this gap by providing an explicit algorithm to decide universal stability of any given network. In a very elegant piece of work, Goel presents three simple graphs $\mathcal{M}_1, \mathcal{M}_2$ and \mathcal{M}_3 , and shows that a (directed) graph \mathcal{N} is universally stable if and only if none of $\mathcal{M}_1, \mathcal{M}_2$ and \mathcal{M}_3 is a minor of \mathcal{N} . The nice contribution of the work of Goel is that it *explicitly* provides the class of forbidden minors for any universally stable graph, which are known to exist by the work of Andrews *et al.* [3] described above.

6.2 Undecidability Results

This section describes results from a very interesting and elegant, recent paper by Gamarnik [19]. Gamarnik considers two abstract models of queueing systems operating under some specific and

fixed queue management protocol which we continue to describe.

- The first model of a queueing system consists of a single server and a collection of buffers in which arriving packets are stored. An arriving packet may require to go through many buffers for completing its processing; each buffer represents a processing stage of the packet.
- The second model is a communication-type queueing network represented by a graph (as considered in other sections of this survey). Recall that an arriving packet (job) requests a simple path to follow.

In both models, the arrival pattern of the packets is completely deterministic: interarrival times are fixed and known; processing times are also deterministic. A *queue management protocol* specifies the rules under which arriving packets are processed in the queueing system. (Common queue management protocols include ones we already considered in this survey such as FIFO and LIFO.)

Gamarnik considers an abstract class of *generalized priority* protocols. The main result of the work of Gamarnik [19] establishes:

Theorem 6.2 *Checking stability of any given generalized priority protocol in any given queueing system (defined as in Gamarnik’s model) is undecidable.*

Interestingly, the undecidability proof of Gamarnik identifies connections with algorithmic problems consider homogeneous random walks in the nonnegative orthant. Natural questions that remain include (cf. [19, Section 7]):

Open Problem 6.1 *Establish the decidability or undecidability of common protocols such as FIFO and LIFO in Gamarnik’s model of a queueing system.*

Gamarnik [19, Section 7] conjectures that the answer to Open Problem 6.1 is negative.

7 Conclusion

Investigating the stability of communication networks, and corresponding distributed protocols, remains a fascinating challenge for the distributed computing community. In this survey, we have only touched upon this topic. We conclude this survey by mentioning some more general directions for future research that naturally present themselves and complement the various open problems that were mentioned in the text.

- Besides the particular results described in Section 6, decidability issues for stability properties are still poorly understood. In particular, the effect of the particulars of the model or the protocol still goes beyond our current understanding of the issue. We would like to demarcate the boundary between decidable and undecidable with respect to network stability properties as accurately as possible.

- The precise effect of *randomization* on stability properties remains still elusive. In particular, we would like to ask whether randomization can be used to either surpass instability results, or improve any of the stability results reviewed in this survey. (For uses of randomization in distributed packet switching, see the paper by Rabani and Tardos [27] and references therein.) For example, can randomization reduce the currently known instability threshold of FIFO [14]?
- Known results on stability are each tailored for a particular model or protocol. How do the various results relate to each other? For example, can there be any general techniques for transforming a result for some particular model to a result for a different model? The relation between the various models is not well understood yet, nor is the relation between the behavior of any *particular* protocol in one model to the behavior of the same protocol in a different model.

We hope that this survey will spawn an interest for the distributed computing community into this important problem of the stability of routing protocols in communication networks.

References

- [1] W. Aiello, E. Kushilevitz, R. Ostrovsky and A. Rosén, "Adaptive Packet Routing for Bursty Adversarial Traffic," *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 359–368, May 1998.
- [2] M. Andrews, "Instability of FIFO in Session-Oriented Networks," *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 440–447, January 2000.
- [3] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, A. T. Leighton and Z. Liu, "Universal Stability Results for Greedy, Contention-Resolution Protocols," *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pp. 380–389, October 1996.
- [4] M. Andrews and L. Zhang, "Stability of Networks with Input and Output Blocking," *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 369–377, May 1998.
- [5] M. Andrews and L. Zhang, "The Effects of Temporary Sessions on Network Performance," *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 448–457, January 2000.
- [6] M. Andrews and L. Zhang, "Achieving Stability in Networks of Input-Queued Switches," *Proceedings of IEEE INFOCOM 2001*, April 2001, to appear.
- [7] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan and D. Williamson, "Adversarial Queueing Theory," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 376–385, May 1996.
- [8] A. Borodin, R. Ostrovsky and Y. Rabani, "Stability Preserving Transformations: Packet Routing Networks with Edge Capacities and Speeds," *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2001.
- [9] M. Bramson, "Instability of FIFO Queueing Networks," *Annals of Applied Probability*, Vol. 4, No. 2, pp. 414–431, 1994.
- [10] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, McGraw Hill and The MIT Press, 1990.
- [11] R. L. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation," *IEEE Transactions on Information Theory*, No. 1, pp. 114–131, 1991.
- [12] R. L. Cruz, "A Calculus for Network Delay, Part I: Network Analysis," *IEEE Transactions on Information Theory*, No. 1, pp. 131–141, 1991.
- [13] J. G. Dai, "On the Positive Harris recurrence for Multiclass Queueing Networks: A Unified Approach via Fluid Models," *Annals of Applied Probability*, Vol. 5, pp. 49–77, 1995.

- [14] J. Diaz, D. Koukopoulos, S. Nikolettseas, M. Serna, P. Spirakis and D. Thilikos, "Stability and Non-Stability of the FIFO Protocol," *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures*, July 2001, to appear.
- [15] D. D. Down and S. P. Meyn, "Piecewise Linear Test Functions for Stability and Instability of Queueing Networks," *Queueing Systems*, Vol. 27, pp. 205–226, 1997.
- [16] U. Feige, "Nonmonotonic Phenomena in Packet Routing," *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pp. 583–591, May 1999.
- [17] D. Gamarnik, "Stability of Adversarial Queues via Fluid Models," *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pp. 60–70, October 1998.
- [18] D. Gamarnik, "Stability of Adaptive and Non-Adaptive Packet Routing Policies in Adversarial Queueing Networks," *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pp. 206–214, May 1999.
- [19] D. Gamarnik, "On Deciding Stability of Scheduling Policies in Queueing Systems," *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 467–476, January 2000.
- [20] A. Goel, "Stability of Networks and Protocols in the Adversarial Queueing Model for Packet Routing," Technical Note STAN-CS-TN-97-59, Stanford University, June 1997.
- [21] F. P. Kelly, "Networks of Queues with Customers of Different Types," *Journal of Applied Probability*, Vol. 12, pp. 542–554, 1975.
- [22] F. P. Kelly, *Reversibility and Stochastic Networks*, Wiley, 1979.
- [23] L. Lovász and M. D. Plummer, *Matching Theory*, Vol. 16 in *Annals of Discrete Mathematics*, North-Holland, 1979.
- [24] S. H. Lu and P. R. Kumar, "Distributed Scheduling Based on Due Dates and Buffer Priorities," *IEEE Transactions on Automatic Control*, Vol. 36, pp. 1406–1416, 1991.
- [25] N. W. McKeown, V. Anantharam and J. Walrand, "Achieving 100 % Throughput in an Input-Queued Switch," *Proceedings of IEEE INFOCOM'96*, pp. 296–302, March 1996.
- [26] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Network: The Single-Node Case," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, pp. 344–357, 1993.
- [27] Y. Rabani and É. Tardos, "Distributed Packet Switching in Arbitrary Networks," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 366–375, May 1996.
- [28] N. Robertson and P. D. Seymour, "Graph Minors V – Excluding a Planar Graph," *Journal of Combinatorial Theory, Series B*, Vol. 41, pp. 92–114, 1986.

- [29] N. Robertson and P. D. Seymour, "Graph Minors IV – Tree-Width and Well-Quasi-Ordering," *Journal of Combinatorial Theory, Series B*, Vol. 48, pp. 227–254, 1990.
- [30] N. Robertson and P. D. Seymour, "Graph Minors XIII: The Disjoint Paths Problems," *Journal of Combinatorial Theory, Series B*, Vol. 63, pp. 65–110, 1995.
- [31] A. Rybko and A. Stolyar, "On the Ergodicity of Stochastic Processes Describing Open Queueing Networks," *Problemi Peredachi Informatsii*, Vol. 28, pp. 3–26, 1992.
- [32] R. Serfozo, *Introduction to Stochastic Networks*, Springer, 1999.
- [33] P. Tsaparas, *Stability in Adversarial Queueing Theory*, M. Sc. Thesis, Department of Computer Science, University of Toronto, 1997.
- [34] D. B. West, *Introduction to Graph Theory*, Second Edition, Prentice Hall, 2001.