From Static to Dynamic Routing: Efficient Transformations of Store-and-Forward Protocols*

Christian Scheideler† Berthold Vöcking‡

Abstract

We investigate how static store-and-forward routing algorithms can be transformed into efficient dynamic algorithms, that is, how algorithms that have been designed for the case that all packets are injected at the same time can be adapted to more realistic scenarios in which packets are continuously injected into the network. Besides describing specific transformations for well-known static routing algorithms, we present a black box transformation scheme applicable to every static, oblivious routing algorithm. We analyze the performance of our protocols under a stochastic and an adversarial model of packet injections.

One result of our specific transformations is the first dynamic routing algorithm for leveled networks that is stable for arbitrary admissible injection rates and that works with packet buffers of size depending solely on the injection rate and the node degree, but not on the size of the network. Furthermore, we prove strong delay bounds for the packets. Our results imply, for example, that a throughput of 99% can be achieved on an n-input butterfly network with buffers of constant size while each packet is delivered in time $O(\log n)$, with high probability.

Our black box transformation ensures that if the static algorithm is pure (i.e., no extra packets apart from the original packets are routed), its dynamic variant is stable up to a maximum possible injection rate. Furthermore, in the stochastic model, the routing time of a packet depends on local parameters such as the length of its routing path, rather than on the maximum possible path length, even if the static algorithm chosen for the transformation does not provide this locality feature and is not pure. In the adversarial model, the delay bound of the packets is closely related to the time bound given for the static algorithm.

*A preliminary version appears at the 31st STOC, 1999, see [18].

†Department of Mathematics and Computer Science, and Heinz Nixdorf Institute, Paderborn University, Germany. Email: chrsch@uni-paderborn.de. Supported in part by the DFG-Sonderforschungsbereich 376 and the EU ESPRIT Long Term Research Project 20244 (ALCOM-IT).

‡International Computer Science Institute, Berkeley, CA 94704-1198. Email: voecking@icsi.berkeley.edu. Supported by a stipend of the "Gemeinsames Hochschulsonderprogramm III von Bund und Ländern" through the DAAD. This research was conducted in part while he was staying at the Heinz Nixdorf Institute, with support provided by the DFG-Sonderforschungsbereich 376.

1 Introduction

Many static routing protocols have been developed in recent years (see, e.g., [9, 8, 10, 13, 6, 12]). These protocols aim to route some initially given set of packets along predetermined paths in a network as fast as possible. In practice however, networks are rarely used in this static fashion but packets are injected dynamically into the network. Since much less is known in the area of dynamic routing (see e.g. [5, 15, 17]) than in the area of static routing, it would be highly desirable to transfer the results gathered for static routing to dynamic routing. So far, however, not much is known about how to transform static routing protocols into stable dynamic protocols, or how efficient dynamic variants of important static routing protocols are.

In this paper we present transformations for *oblivious* algorithms, i.e., the path of a packet is already fixed when the packet is injected into the system. We investigate how static, oblivious routing algorithms can be transformed into dynamic routing algorithms that are stable and efficient under a stochastic or adversarial model of packet injections. In particular, we will show that the ghost packet protocol [8, 14] and the growing rank protocol [10, 11] can be transformed into dynamic routing protocols that are stable up to a maximum possible injection rate. Furthermore, we will present a simple and elegant scheme that transforms any static protocol into an efficient dynamic protocol that is also stable up to a maximum possible injection rate if the protocol is pure. Besides showing the stability of these protocols, we will prove bounds on the routing time of the packets. For the protocols derived by the black box transformation we further prove that they recover quickly from any worst case scenario, that is, packets generated a certain amount of time after a bad event are not influenced by this event anymore.

1.1 Models and problems

We consider arbitrary network topologies modeled as undirected graphs of n nodes. The nodes represent switches, and the edges represent bidirectional communication links, unless otherwise stated, with buffers for incoming packets on either side. These buffer are called *edge buffers*, and bounds on the buffer size always refer to the maximum number of packets that these buffers can hold. Additionally, every node contains an *injection buffer* in which the initial packets, in case of static routing, or the newly injected packets, in case of dynamic routing, are stored. Routing is performed in a synchronous "store and forward" manner, that is, in every step, each edge can be crossed by at most one packet in each direction. (For simplicity, we assume that time proceeds in discrete time steps.) Once a packet reaches its destination it is discarded.

We present routing protocols in which the nodes *locally* decide which packets to move forward in each step, i.e., a decision only depends on the routing information carried by the packets and on the local history of the execution. These algorithms are called *local control* or *on-line* algorithms. In general, a packet routing scheme consists of two (not necessarily independent) parts: the first part is responsible for selecting a path for each packet, and the second part is responsible for scheduling the packets along their chosen paths. We assume that some suitable strategy for the path selection is given. Hence, in the following we only concentrate on the question of how to schedule the packets along their fixed paths. We use the following models.

1.1.1 Static packet routing

Here we assume that a fixed collection of paths is given with *congestion* C and *dilation* D, that is, C denotes the maximum number of paths crossing an edge, and D denotes the maximum length of a path. Along each of these paths a packet has to be sent. Further, let M denote the *complexity* of the routing problem, i.e., M is defined to be the maximum of the number of edges, the number of paths, and the dilation. Let us give some examples of known results on the *routing time* for static packet routing, i.e., the time needed to deliver all packets:

- $C \cdot D$: trivial upper bound for any greedy protocol in case of unlimited buffers, i.e., protocols in which a packet is only delayed because other packets move along the next edge on the packet's routing path;

- $(1+\kappa) \cdot C + O(D \cdot \log M)$, w.h.p.[1], for any constant $\kappa > 0$: upper bound for arbitrary paths in arbitrary networks with unbounded buffers [9].
- $O(C + D + \log M)$, w.h.p.: upper bound in leveled networks with bounded buffers of constant size, where D is the depth of the network [14, 8], and upper bound for routing along shortest paths in arbitrary networks with unbounded buffers [11, 10];
- $(1+\kappa) \cdot C + (\log^* M)^{O(\log^* M)} D + O((\log M)^6)$, w.h.p., for any constant $\kappa > 0$: upper bound for routing along *simple paths*, i.e., paths without cycles, in arbitrary networks with unbounded buffers [13];
- $O(C + D + \log^{1+\kappa} M)$, w.h.p., for any constant $\kappa > 0$: upper bound for routing along simple paths in arbitrary networks with unbounded buffers [12].

We will come back to some of these results when using our black box transformations.

1.1.2 Dynamic packet routing

The most commonly used injection models in the dynamic setting are the *stochastic* and the *adversarial* injection model.

The stochastic model. Here the packets are injected by a set of *generators*, each of them mapped to one of the nodes in the network. We allow any relationship between the number of generators and the number of nodes in the network. Furthermore, we place no restrictions on the distribution of the generators among the nodes. That is, one node could have several generators, whereas another node may have none. So a generator may represent a user thread or process, whereas a node may represent a processor. In each time step, each generator g placed on a node v injects a packet with some probability p_g. This probability is called the *injection rate* of g. For each packet, the generator randomly selects a destination and a routing path from v to this destination according to an arbitrary, fixed probability distribution. We assume that each generator is operating independently from other generators, and the injection of a packet and its routing path is independent from injections in previous time steps. Note that we do not demand that the destinations are chosen uniformly from the set of all nodes, or that packets with the same source and destination node follow the same routing path. Finally, we define the *(overall) injection rate*, which is denoted by λ. Define λ_e to be the expected number of messages generated in a time step that contain the edge e in their routing paths. Then λ is defined to be the maximum λ_e over all edges.

The adversarial model. An adversary is allowed to demand network bandwidth up to a prescribed injection rate. For any $w, \lambda > 0$, an adversary is called a *bounded adversary of rate* (w, λ) if for all edges e and all time intervals I of length w, it injects no more than $\lambda \cdot w$ packets during I that contain edge e in their routing paths. As in the stochastic model, λ is defined to be the *injection rate*. (We use the adversarial model as defined by Andrews *et al* [1] rather than the original model introduced by Borodin *et al* [3] because this model avoids calculating with floors and ceilings. Apart from minor changes in constants, however, all our results hold in the original model of Borodin *et al*, too.)

For both injection models, a protocol is called *stable* for a given injection rate λ if the number of packets in the injection and edge buffers does not grow unboundedly with time. We are interested in protocols that are stable for high injection rates. Of course, since an edge can transport at most one packet per step, λ can be at most 1. Our aim is to construct algorithms that are stable under injection rates that are close to 1. Additionally, we are interested in short delays for the packets, i.e., we aim to minimize the time from injection to service for each packet.

[1] Throughout this paper, the term "w.h.p." means "with high probability", i.e., with probability at least $1 - M^{-\alpha}$, where $\alpha > 0$ is an arbitrary constant term and M denotes the complexity of the routing problem.

Apart from the stability and the routing time we will consider another property of dynamic routing protocols, the recovery from worst case scenarios. Although our bounds on the routing time guarantee that bad configurations are very unlikely, they eventually occur from time to time when the routing protocol runs for an infinite number of time steps. Let a *worst case scenario* denote an arbitrarily bad configuration of the network. Then the *recovery time* with regard to some property $\mathcal{P}$ of the routing protocol is defined as the time that has to pass by after the occurence of a worst case scenario until $\mathcal{P}$ holds again. (In our case, we are interested in properties such as the expected routing time of a packet and time bounds that hold w.h.p.)

As in the static model we define the complexity of a dynamic routing problem to be a value capturing all relevant parameters. In particular, the *complexity* is defined to be the maximum of the number of edges, the number of generators, the maximal possible length of a routing path, and $1/(1-\lambda)$. (The number of generators in the adversarial model is defined to be w times the number of edges, which corresponds to the maximum number of packets that can be injected in a single step.)

In the following sections, we will prove results for both the stochastic and the adversarial injection model.

1.2 Previous Results

In the last two years a new model called *adversarial queuing theory* emerged. This approach was introduced by Borodin *et al* in [3]. Most research in this model focuses on the stability of routing protocols and networks. For example, Borodin *et al* [3] show several stability results for greedy protocols on DAGs and directed cycles. Andrews *et al* [1] extend their results by showing that there exist simple greedy protocols (such as longest-in-system, shortest-in-system and farthest-to-go) that are stable against any adversary for all networks. However, the delay of the packets and the number of packets stored in a queue might get exponential in the length of the longest path.

Furthermore, Andrews *et al* [1] present a transformation of the static protocol presented in [9] into a dynamic protocol that is stable for any injection rate < 1 and fulfills the following constraint on the buffer size: For any fixed time step t, at most $(D \cdot \log m)^k$ packets are stored in any queue at time t, w.h.p., where D denotes the longest routing path, m the number of edges, and k is a suitable constant. Note that this result implies that the delay of the packets is also bounded by $(D \cdot \log m)^k$, w.h.p. However, as the bound on the buffer size does not hold deterministically, any buffer of fixed size will overflow eventually.

Rabani and Tardos [13] present a transformation scheme which yields much better routing times. Assuming there is a static algorithm that delivers all packets in $(1+\kappa)C + g(M)D + f(M)$ steps for some constant $\kappa > 0$, their tansformation yields a dynamic algorithm that delivers each packet to its destination, w.h.p., in $O(w + g(N)D + f(N) + \log N)$ against an adversary of rate $(w, \Theta(\kappa))$, where M and N denote the complexity of the static and dynamic routing problems, respectively. The stability of the dynamic algorithms, however, is not shown. In fact, although most packets will be delivered fast, some packets will never reach their destination and queues will grow to infinity assuming the stochastic model or the adversarial model using a randomized, static algorithm.

Broder *et al* [4] introduce a general approach to dynamic packet routing with bounded buffers in the stochastic and adversarial model. They show sufficient conditions for the stability of dynamic packet routing algorithms and investigate how some well-known static routing protocols for the butterfly network can be transformed into dynamic algorithms that fulfill these conditions. In particular, they present a dynamic routing algorithm for the butterfly that is stable for a small constant injection rate, and they show that the expected routing time for each packet is $O(\log n)$, with n denoting the number of nodes on a level.

Andrews *et al* [2] investigate another, more restrictive dynamic routing model. In contrast to the stochastic and the adversarial model, the packets are injected regularly in "sessions". For each session i, packets are injected at a rate r_i to follow a fixed path of length d_i. They describe a schedule that delivers each packet in a time depending only on local paramaters, that is, each packet reaches its destination in time $O(d_i + 1/r_i)$, which is worst case optimal. We will see that similar local properties, i.e., the routing time depends on d_i, can

be achieved also in the stochastic model.

1.3 New Results

In this paper, we present specific transformations of well-known routing protocols and introduce a powerful black box transformation scheme applicable to every static, oblivious routing protocol.

In the following, N denotes the complexity of the routing problem (as defined in Section 1.1), D denotes the maximum length of a routing path, and $\epsilon = 1 - \lambda$, where λ denotes the injection rate. Further, we define $\epsilon^* = a \cdot \epsilon^b$, for suitable constants $a, b > 0$. For simplicity, we only state our results for networks of constant degree. For more detailed results the reader is referred to the following sections.

In this paper, we give three specific transformations of well-known routing protocols.

- In Section 2, we present a dynamic variant of the ghost packet protocol [8, 14] for leveled networks that is stable for any $\lambda < 1$ in the stochastic model and $\lambda \leq 1$ in the adversarial model, given a sufficiently large but fixed buffer size of $1/\epsilon^*$ in the stochastic model and $2\lambda \cdot w + 2$ in the adversarial model. In the stochastic model, each individual packet is delivered in expected time L/ϵ^*, and in time $(L + \log N)/\epsilon^*$, w.h.p., where L denotes the depth of the network. In the adversarial model, each packet reaches its destination in at most $L + \lambda \cdot w \cdot L - 1$ time steps.

  For example, the tuned ghost packet protocol achieves a throughput of $1 - \epsilon$, for any $\epsilon > 0$, on an n-input butterfly network with buffers of size $1/\epsilon^*$ if we place two generators on each node of level 0 each of which injects packets that are sent to randomly selected nodes of level $\log n$, using a rate of $\lambda = 1 - \epsilon$. Furthermore, the algorithm delivers each individual packet in time $\log n/\epsilon^*$, w.h.p.

  Previous results on routing with bounded buffers in leveled networks obtain stability only for constant injection rates $\lambda \ll 1$ [4, 16] (stochastic model) or require buffers whose size is exponential in the depth of the network [1] (adversarial model).

- In Section 3, we present a dynamic routing protocol for arbitrary networks that is stable for any injection rate $\lambda < 1$, assuming buffers of fixed size D/ϵ^*. We prove an expected routing time of $(D^2 + w)/\epsilon^*$, and $(D^2 + D \cdot \log N + w)/\epsilon^*$, w.h.p., for every individual packet. These bounds hold both for the adversarial model and for the stochastic model (with $w = 0$).

  To the best of our knowledge, this is the first protocol that is stable for buffers of small fixed size under any injection rate $\lambda < 1$. Previous results in the stochastic model with bounded buffers require $\lambda \ll 1$ and besides assume that packets can be dropped and reinjected in later time steps [16]. Previous results in the adversarial model require buffers whose size is exponential (or polynomial, w.h.p.) in D [1]. Note that a bound on the buffer size that does not hold with certainty leaves open the question of what to do in the rare case of a buffer overflow (e.g., dropping or blocking incoming packets), and hence does not guarantee stability for networks with a fixed buffer size.

- In Section 4, we describe a dynamic variant of the growing rank protocol [10] for shortest paths in arbitrary networks. The dynamic protocol is stable for any $\lambda < 1$ if unbounded buffers are given. In the stochastic model, each individual packet p with a routing path of length d_p is delivered in time d_p/ϵ^*, expected, and $(d_p + \log N)/\epsilon^*$, w.h.p. In the adversarial model, the d_p in the time bound has to be replaced by $D + w$. In contrast to the results above, these results hold independent of the node degree.

  Previously, similar results have only been shown for $\lambda < 1/\mathrm{e}$ in the stochastic model [16].

Furthermore, in Section 5, we present a powerful black box transformation scheme that is applicable to every static, oblivious routing algorithm in networks with unlimited buffers. Basically, we combine the ideas of Rabani and Tardos [13] for the fast delivery of packets with the universal stability of the shortest-in-system

protocol originally shown by Andrews *et al* [1] for the adversarial model. The major problem that we solve is merging these two approaches so that we obtain dynamic protocols that are stable up to some injection rate depending on the static protocol without any slow down due to the inefficiency of the shortest-in-system protocol.

Let $\mathcal{S}$ denote any set of paths, e.g., the set of all simple or all shortest paths in the network. Suppose we are given a static routing algorithm that routes all packets in $\gamma \cdot C + \delta \cdot D + O(\log^{\alpha} M)$ steps, with high probability (or even with certainty), for any collection of paths or subpaths in $\mathcal{S}$ with congestion C, dilation D, and complexity M. Assume that in the dynamic setting only paths in $\mathcal{S}$ are allowed to be generated. Then our black box transformation yields a dynamic variant of this protocol with the following properties. (In the following overview, we only describe the results for the case that γ and δ do not depend on C or D and $\alpha \geq 1$ is a constant. Similar results will be shown for other choices of α, γ and δ.)

- If the given static protocol is *pure* (i.e., no control messages or copies of packets are allowed), the dynamic algorithm is stable for any injection rate $\lambda < 1$. Otherwise, it is stable for any injection rate $\lambda < 1/\gamma$.
- If $\lambda < 1/\gamma$ then the algorithm guarantees that any packet p that has to travel a distance of d is delivered in time $O(\delta \cdot d + \log^{\alpha} N)$, w.h.p., in the stochastic injection model, and in time $O(\delta \cdot D + w + \log^{\alpha} N)$, w.h.p., in the adversarial injection model.
- The algorithm recovers from any worst case scenario in $O(\delta \cdot D + w + \log^{\alpha} N)$ time steps (with $w = 0$ in the stochastic model).

The bound on the routing time implies that it might be important for static routing protocols to know the exact factor γ in front of the C since this can be decisive for the performance of their dynamic counterparts. Interestingly, in the stochastic injection model, the dynamic variant is able to exploit locality, whereas the static algorithm does not need to provide this feature. For example, the transformation of a well-known static routing algorithm (see, e.g., [9]) that delivers all packets in time $(1+\kappa) \cdot C + O(D \cdot \log M)$, w.h.p., for any constant $\kappa > 0$, yields a dynamic algorithm that delivers each packet p in time $O(d \cdot \log N)$, w.h.p., for any constant injection rate $\lambda < 1/(1+\kappa)$, where d is the length of p's path.

We will intially show most of our results only for the stochastic model. For this, we will frequently apply Chernoff bounds.

Lemma 1.1 (Chernoff) *Let $X_1, \ldots, X_n$ be n independent random variables with $X_i \in \{0,1\}$ for all $i \in \{1, \ldots, n\}$. Furthermore, let $X = \sum_{i=1}^{n} X_i$ and $\mu \geq \mathrm{E}[X]$. Then it holds for all $\epsilon \geq 0$ that*

$$\Pr[X \geq (1+\epsilon)\mu] \leq \left(\frac{\mathrm{e}^{\epsilon}}{(1+\epsilon)^{1+\epsilon}} \right)^{\mu}$$

This can be simplified to

$$\Pr[X \geq (1+\epsilon)\mu] \leq \begin{cases} \mathrm{e}^{-\epsilon^2 \mu/3} & \text{if } \epsilon \in [0,1] \\ \mathrm{e}^{-\epsilon \mu/3} & \text{otherwise} \end{cases} .$$

In Section 6, we finally show how to adapt these results to the adversarial model.

2 Routing in leveled networks with bounded buffers

In this section, we consider the problem of routing packets in a leveled network with bounded buffers. In a *leveled network*, the nodes can be partitioned into levels $0, \ldots, L$ such that each link in the network leads from some node at level i to some node at level $i+1$, for $0 \leq i \leq L-1$. L is called the *depth of the network*.

The routing proceeds in discrete time steps, starting with step 0. In each step, each link can forward at most one packet. The links are assumed to be directed, that is, packets can cross them only in the direction leading to the higher level. Packet injections and arrivals are assumed to happen at the beginning of a time step, so that a packet may leave a node at the end of the time step in which it is injected or arrives at the node. The packets' routing paths may start on any level $k \geq 0$ and end on any level k' with $k < k' \leq L$. Each node has a buffer for each of its incoming edges and a buffer for newly injected packets. Each of the edge buffers has space for storing at most q packets.

Static batches of packets can be routed efficiently on leveled networks by a protocol known as Ranade's or ghost packet protocol [7, 8, 14]. The disadvantage of the static ghost packet protocol is that each node is allowed to forward only one data packet at each time step, rather than forwarding data packets along all outgoing edges in parallel. All edges that are not passed by a data packet in a step are used to exchange control packets that are called *ghost packets*. As a consequence, most of the transmitted packets are ghost packets, which shows that a simple transformation of the static ghost packet protocol into a dynamic protocol cannot yield stability for injection rates close to 1. In order to achieve stability for any injection rate $\lambda < 1$, we introduce a tuned variant of the ghost packet protocol that only uses a very limited number of ghost packets.

The tuned ghost packet protocol. The packets are assigned ranks in order to decide which packet is preferred in case of contention. For each packet p, let $\mathrm{birth}(p)$ denote the time step at which p was injected. The rank of p is set to $\mathrm{birth}(p)$ plus some small value x from the interval $[0, \kappa)$, for some $\kappa < 1$, where x is chosen such that each packet has its own, unique rank (e.g., based on the identification number of the generator that injected the packet). Packets with smaller ranks, i.e., older packets, are always preferred against packets with higher rank, i.e., younger packets. As in the static ghost packet protocol, special ghost packets help the algorithm to maintain the following invariant: A packet is routed along an edge only after all the other packets with lower ranks that must pass through the edge have done so. The nodes on level k start working in step k, for $0 \leq k \leq L$. In order to give time for initializing the network, we assume that packet injections on level k do not start before time step k. Figure 1 describes the rules for contention resolution in detail.

Ghost packets are discarded as soon as they are delayed in a step. Thus, they never block the buffer for following packets. The role of the ghost packets is to slow down too fast packets in order to avoid that a relatively old packet is blocked because younger packets occupy the slots in the next buffer. Note that each outgoing link on level k transmits one packet in each time step $t \geq k$, either a ghost or a real packet. This mechanism ensures that each link transmits packets and ghost packets in the order of strictly increasing rank. (Obviously, this property holds for the links on level 0. For higher levels the property follows by induction.) We will see that this property is crucial for the analysis. Analyzing the performance of a variant of the protocol described above that does not use ghost packets is an interesting open problem, even in the static case.

The following analysis shows that the tuned ghost packet protocol is stable for any injection rate $\lambda \leq 1$ in the adversarial model, and $\lambda < 1$ in the stochastic model, provided that the edge buffer size is sufficiently large.

Theorem 2.1 *Let L denote the depth of the network, Δ the maximum node degree, and q the size of the edge buffers.*

- *Suppose the packets are injected according to the adversarial model with injection rate (w, λ), for any $w \leq (q-2)/(2\lambda)$ and $0 \leq \lambda \leq 1$. Then the tuned ghost packet protocol is stable, and each packet reaches its destination in at most $L + \lambda \cdot w \cdot L$ time steps.*

- *Suppose the packets are injected according to the stochastic model with injection rate $\lambda \leq 1 - \epsilon$, for a suitably chosen $\epsilon = \Theta(\log(q\Delta)/q)$. Then the tuned ghost packet protocol is stable, and the routing time for each individual packet is $O(L \cdot \log(\Delta/\epsilon)/\epsilon + \log(1/\epsilon)/\epsilon^2)$, expected, and $O(L \cdot \log(\Delta/\epsilon)/\epsilon + (\log N)/\epsilon^2)$, w.h.p., where the probability is with respect to the stochastic packet injections.*

The following algorithm is executed for each outgoing link e of a node v on level k in each time step $t \geq k$. Each edge buffer can hold up to q packets.

- Let r denote the minimum rank of a packet that is stored in one of v's buffers and aims to pass edge e. If there is no such packet then $r = \infty$.
- Let g denote the minimum over all ranks of packets or ghost packets that arrived on v at the beginning of step t. If there is no such packet (as v is a node without incoming edges, e.g., on level 0) then g is set to $t + \kappa$.
- if $r < g$ then

  if the buffer of e contains less than q packets at the beginning of step t then

  forward the (unique) packet with rank r along e

  else

  send a ghost packet with rank r along e

  else

  send a ghost packet with rank g along e.

Figure 1: Contention resolution in the tuned ghost packet protocol.

Proof. We use a "delay sequence argument" to analyze the tuned ghost packet protocol. Our analysis is similar to the one for the static ghost packet protocol given in [8]. A delay sequence witnesses that a packet needs many time steps to reach its destination. For the adversarial model, we will show that a delay sequence witnessing a long routing time does not exist, so that every packet reaches its destination within the time bound given in the theorem. For the stochastic model, we will show that "large" delay sequences are very unlikely so that each packet needs only limited time, with high probability, to reach its destination.

The ghost packet protocol uses fractional ranks. The only reason for the fractional additive is to define a total order among all packets such that a packet p or a ghost packet corresponding to p (i.e., a ghost packet that has the same rank as p) that delays a packet p' in a step cannot be delayed by packet p' or a ghost packet corresponding to p' in another time step. In the following, however, we mainly use *integral ranks*, i.e., the integral values of the fractional ranks, which, in the case of the ghost packet protocol, are equal to the birth date of the corresponding packet.

Definition 2.2 ((p, s, ℓ, r)-delay sequence) *Let p denote a packet, and let $s, \ell, r \geq 1$ denote integers. Then a (p, s, ℓ, r)-delay sequence consists of*

- *a path of length ℓ starting at the destination node of packet p. This path is called the* delay path. *Let $v_0, \ldots, v_\ell$ denote the nodes on the delay path. The delay path may include edges in both directions and, hence, follows a course going up and down the levels of the network.*

- ℓ delay edges $e_1, \ldots, e_\ell$ *such that* e_i *is incident to* v_i*, for* $1 \leq i \leq \ell$*. These edges are not necessarily included in the delay path.*
- ℓ *non-empty* intervals of integral ranks $r_1, \ldots, r_\ell$ *such that* $\sum_{i=1}^{\ell} |r_i| = r$*, where* $|r_i|$ *denotes the length of interval* r_i*. The maximum integral rank in* r_1 *is equal to the birth date of packet* p*, and, for* $2 \leq i \leq \ell$*, the maximum integral rank in* r_i *is equal to the minimum in* r_{i-1}.

A (p, s, ℓ, r)*-delay sequence is called* active *if the adversary or the stochastic generators inject* s *packets* $p_1, \ldots, p_s$ *(different from* p*) such that, for every* $1 \leq i \leq s$*, packet* p_i *has an integral rank in* r_j *and its routing path includes edge* e_j*, for some* $1 \leq j \leq \ell$*. These packets are called* delay packets*.*

The following lemma shows that a long routing time of a packet p_0 is always accompanied by an active (p_0, s, ℓ, r)-delay sequence with relatively large s and small ℓ and r.

Lemma 2.3 *Suppose a packet* p_0 *takes* $L + T$ *or more steps to reach its destination, for any* $T > 0$*. Then there is an active* (p_0, s, ℓ, r)*-delay sequence with* $s \geq T + r + (\frac{q}{2} - 2) \cdot \ell - (\frac{q}{2} - 1) \cdot L'$*, where* L' *denotes the difference between the level of the first node,* v_0*, and the level of the last node,* v_s*, of the delay path.*

Proof. We will construct a sequence of packets or ghost packets $p_0, \ldots, p_{s'}$ and nodes $v'_0, v'_1, \ldots, v'_{s'}$ such that v'_0 denotes the destination of p_0, and packet p_i or a ghost packet corresponding to this packet delays packet p_{i-1} at node v'_i, for $1 \leq i \leq s'$. (Ultimately, s will be set either to s' or to $s' - 1$.) There are two reasons why a packet may be delayed: It is delayed either by a packet or ghost packet with lower rank that wants to traverse the same link, or it is delayed by q other packets with lower ranks occupying the next edge buffer on its path. The first kind of delay is called m*-delay*, the second kind of delay is called f*-delay*.

The active delay sequence is constructed incrementally. Suppose we have already fixed the packets $p_0, \ldots, p_{j-1}$ and nodes $v'_1, \ldots, v'_{j-1}$. Starting from the time step in which p_{j-1} delayed p_{j-2} on node v'_{j-1} or, if $j = 1$, the time step in which $p_{j-1} = p_0$ reached its destination node v'_0, we follow the course of the packet p_{j-1} backwards in time. If p_{j-1} is a ghost packet whose generation was caused by the arrival of another packet, then we identify p_{j-1} with that packet and continue the trace. We stop following either when we reach a node on which p_{j-1} was delayed, on which p_{j-1} was injected as a non-ghost packet, or on which p_{j-1} was injected as a ghost packet on a node without incoming edges. The node on which this event happened is called v'_j. If we stop because of an m-delay, then the packet that caused the delay is called p_j. If we stop because of an f-delay, then the q packets that occupy the corresponding buffer are called $p_j, \ldots, p_{j+q-1}$, in decreasing order of ranks. Moreover we set $v'_{j+1}, \ldots, v'_{j+q-1} = v'_j$. In both of these cases we can continue our construction. In the other cases, however, our construction ends with a packet p_j that was injected at node v'_j, and we define $s' = j$.

The path from the destination of p_0 to the source of $p_{s'}$ recorded by this process is called the *delay path*. The nodes on this path are defined to be $v_0, \ldots, v_\ell$. Note that these nodes are not necessarily identical to $v'_0, \ldots, v'_{s'}$, but $\{v'_0, \ldots, v'_{s'}\} \subseteq \{v_0, \ldots, v_\ell\}$. The edges on which the recorded delays of the packets $p_0, \ldots, p_{s'-1}$ take place are defined to be the delay edges. We have to show that the number of these edges is at most ℓ. (Note that the delay edges are not necessarily included in the delay path. Consider, for example, the following scenario. Suppose packet p_j is delayed in step t on level k by an f-delay caused by the packets $p_{j+1}, \ldots, p_{j+q}$ stored in a buffer on level $k + 1$. Suppose the next event recorded by the delay sequence is an m-delay of packet p_{j+q} caused by packet p_{j+q+1} moving along an edge e to level $k + 2$ in step $t - 1$, and suppose this packet arrived on level $k + 1$ coming from level k in step $t - 2$. Then the delay path goes from level k to level $k + 1$ and then back to level k, and hence, skips the delay edge e.)

Although not every delay edge is included in the delay path, each of these edges is incident to a node of the delay path, and the number of different delay edges is at most ℓ, which can be shown as follows. All delay events in a sequence of consecutive m-delays recorded by the construction above in consecutive time steps, i.e., not separated by packet movements or f-delays, take place at the same edge. Further, an f-delay following

immediately after a sequence of m-delays takes place at the same edge, too. (Note that these properties are not given for the original ghost packet protocol.) Hence, considering the incremental construction of the delay sequence, the delay edge changes only in those incremental steps in which the delay path is increased by at least one node. Consequently, every node v_i of the delay path can be assigned one delay edge, which is called e_i, for $1 \leq i \leq \ell$.

The ghost packet protocol ensures that the ranks in the delay sequence are decreasing. In particular, the ranks of the packets traversing edge e_i are larger than the ranks of the packets traversing edge e_{i+1}, for $1 \leq i \leq \ell - 1$. Hence, we can define consecutive intervals of integral ranks $r_1, \ldots, r_\ell$ in such a way that every neighboring pair of intervals has one integral rank in common (i.e., overlap by one) and the packets that are delayed at edge e_i have an integral rank in r_i, for $1 \leq i \leq \ell$. We define $r' = \sum |r_i| = \mathrm{birth}(p_0) - \mathrm{birth}(p_{s'}) + \ell$. (Ultimately, r will be set either to r' or to $r' - 1$.)

Next we investigate the relationship between the parameters of the delay sequence. Define t to be the number of time steps covered by the construction, i.e., the time from the birth of the packet $p_{s'}$ to the arrival of the packet p_0 at its destination. Then

$$\begin{aligned} t &\geq \mathrm{birth}(p_0) + L + T - \mathrm{birth}(p_{s'}) \\ &= L + T + r' - \ell \ . \end{aligned} \tag{1}$$

The delay path is enlarged by one node in each of those time steps that do not represent an m-delay. Let m denote the number of m-delays. Then we can conclude that $t = m + \ell$, or $m = t - \ell$. Applying the bound in 1 to this equation yields

$$m \geq L + T + r' - 2\ell \ . \tag{2}$$

Let f be the number of f-delays. Since each f-delay enlarges the delay path by two edges, we have $\ell = L' + 2 \cdot f$, where L' denotes the difference between the level of v_0 and the level of v_s. Hence,

$$f = \frac{\ell - L'}{2} \ . \tag{3}$$

Further, each m-delay adds one packet to the active delay sequence, and each f-delay adds q packets. As a consequence,

$$\begin{aligned} s' &= m + q \cdot f \\ &\overset{(2,3)}{\geq} L + T + r' - 2\ell + \tfrac{q}{2} \cdot \ell - \tfrac{q}{2} \cdot L' \\ &\geq T + r' + \left(\tfrac{q}{2} - 2\right) \cdot \ell - \left(\tfrac{q}{2} - 1\right) \cdot L' \ , \end{aligned}$$

where the last estimation holds because $L' \leq L$.

Finally, we fix the parameters s and r. For $0 \leq i \leq s' - 1$, every packet p_i is delayed by packet p_{i+1}. As ghost packets are never delayed, $p_1, \ldots, p_{s'-1}$ must be non-ghost packets. The definition of s and r depends on whether or not $p_{s'}$ is a ghost packet. If it is not, then we set $s = s'$ and $r = r'$ so that $p_1, \ldots, p_{s'}$ are the packets of the active (p_0, s, ℓ, r)-delay sequence. Otherwise, if $p_{s'}$ is a ghost packet then we set $s = s' - 1$. In this case, the packets $p_1, \ldots, p_{s'-1}$ are the delay packets. Since $p_{s'}$ is a ghost packet with rank $\mathrm{birth}(p_{s'}) + \kappa$ and $p_{s'}$ is preferred against $p_s = p_{s'-1}$, we have $\mathrm{birth}(p_s) \geq \mathrm{birth}(p_{s'}) + 1$. (Recall that the increment in the rank for ghost packets, i.e., κ, is larger than the increment in the rank for other packets, i.e., some $x \in [0, \kappa)$.) As a consequence, we can set $r = r' - 1$, and the constructed delay sequence fulfills all desired requirements. Hence, Lemma 2.3 is proven. □

Analysis for the adversarial model. We assume an adversary that injects packets at rate (w, λ), for any $w \leq (q-2)/(2\lambda)$ and $0 \leq \lambda \leq 1$, that is, the adversary injects no more than $\lambda \cdot w$ packets for the same edge during every time interval of length w.

Suppose the routing time of a packet p_0 is $L + T$ or more. Then we can construct a (p_0, s, ℓ, r)-delay sequence with parameters as described in Lemma 2.3. The delay sequence specifies edges $e_1, \ldots, e_\ell$ such that e_i is traversed by packets with integral ranks from the interval r_i. On the one hand, the adversary is allowed to inject at most $\lambda \cdot (|r_i| + w - 1)$ packets with an integral rank in r_i that traverse e_i, for $1 \leq i \leq \ell$. (Recall that the integral rank of a packet corresponds to its birth date.) Hence, the number of packets that traverse one of the delay edges and have the corresponding integral rank is at most

$$\sum_{i=1}^{\ell} \lambda \cdot (|r_i| + w - 1) \quad = \quad \lambda \cdot (r + \ell \cdot (w-1)) \ .$$

On the other hand, we conclude from Lemma 2.3 that the total number of packets needed for an active delay sequence corresponding to a routing time of $L+T$ steps is at least $s \geq T + r + \left(\frac{q}{2} - 2\right) \cdot \ell - \left(\frac{q}{2} - 1\right) \cdot L'$. This yields the constraint

$$\lambda \cdot (r + \ell \cdot (w-1)) \quad \geq \quad T + r + \left(\tfrac{q}{2} - 2\right) \cdot \ell - \left(\tfrac{q}{2} - 1\right) \cdot L'$$

and, therefore,

$$\begin{aligned} T \quad &\leq \quad \lambda \cdot (r + \ell \cdot (w-1)) - \left(r + \left(\tfrac{q}{2} - 2\right) \cdot \ell - \left(\tfrac{q}{2} - 1\right) \cdot L'\right) \\ &= \quad (\lambda - 1) \cdot (r - \ell) + \left(\lambda \cdot w - (\tfrac{q}{2} - 1)\right) \cdot \ell + \left(\tfrac{q}{2} - 1\right) \cdot L' \ . \end{aligned}$$

This upper bound on T can be simplified as follows. First, $(\lambda - 1) \cdot (r - \ell) \leq 0$ because $\lambda \leq 1$ and $r \geq \ell$. Second, $\left(\lambda \cdot w - (\frac{q}{2} - 1)\right) \cdot \ell \leq \left(\lambda \cdot w - (\frac{q}{2} - 1)\right) \cdot L'$ because $\ell \geq L'$ and $w \leq (q-2)/(2\lambda)$, so that the factor in front of the ℓ is negative or 0. Applying these two equations to the above bound on T yields

$$\begin{aligned} T \quad &\leq \quad \left(\lambda \cdot w - (\tfrac{q}{2} - 1)\right) \cdot L' + \left(\tfrac{q}{2} - 1\right) \cdot L' \\ &\leq \quad \lambda \cdot w \cdot L' \quad \leq \quad \lambda \cdot w \cdot L \ . \end{aligned}$$

Consequently, each packet takes at most $L + \lambda \cdot w \cdot L$ time steps to reach its destination and the tuned ghost packet protocol is stable for any injection rate $\lambda \leq 1$.

Analysis for the stochastic model. Now we assume that the packets are injected at random by independent generators at a rate of $\lambda = 1 - \epsilon$ with $\epsilon > 0$. Again we use a delay sequence argument. The major difference between our analysis and the analysis for the static ghost packet protocol given in [8] is that we have to deal with an arbitrarily long history of packet delays. Further, we have to use Chernoff bounds (see Lemma 1.1) instead of simple counting methods in order to prove stability results for injection rates arbitrarily close to 1.

Let L denote the depth of the network, Δ the maximum node degree, and q the size of the edge buffers. We will show that each individual packet reaches its destination within $L + T$ time steps with probability $1 - 2^{-\Omega(\epsilon^2 \cdot T)}/\epsilon^2$, provided that q and T are chosen sufficiently large, i.e., $q \geq 2(k+1)$ and $T \geq k \cdot L$ for some $k = \Theta(\log(\Delta/\epsilon)/\epsilon)$ which will be specified during the proof.

Fix an arbitrary packet p_0. Suppose that p_0 needs more than $L + T$ time steps to reach its destination. Then Lemma 2.3 yields that a (p_0, s, ℓ, r)-delay sequence is active, with

$$\begin{aligned} s \quad &\geq \quad T + r + \left(\tfrac{q}{2} - 2\right) \cdot \ell - \left(\tfrac{q}{2} - 1\right) \cdot L' \\ &\geq \quad k \cdot L + r + (k-1) \cdot \ell - k \cdot L' \\ &\geq \quad r + (k-1) \cdot \ell \ . \end{aligned} \tag{4}$$

We assume $k \geq 1$. Then at least r packets are needed for an active (p_0, s, ℓ, r)-delay sequence. However, we will show that the expected number of packets that pass the delay edges and have the corresponding integral rank is at most $(1-\epsilon) \cdot r$. Hence, this event is very unlikely.

Suppose that the delay edges and the range of integral ranks for each delay edge are fixed. Define $R = r_1 \cup \ldots \cup r_\ell$. For each integral rank $j \in R$, let the binary random variable X_j^g be one if and only if generator g generates a packet that has integral rank $j \in r_i$ and traverses delay edge e_i. (The integral rank j may fall in two or more rank intervals corresponding to different delay edges.) Let $X = \sum_{j,g} X_j^g$. Every active (p_0, s, ℓ, r)-delay sequence of length s with fixed delay edges and a fixed rank assignment has at least one choice for the X_j^g's such that $X \geq s$, because each of the packets $p_1, \ldots, p_s$ traverses one of the delay edges e_i and has an integral rank in r_i. (Note that s distinct packets are needed to have an active sequence, regardless of whether a packet traverses more than one delay edge responsible for its integral rank.) Hence, if a delay sequence of length s can be constructed, then $X \geq s$. Consequently, the probability that a fixed delay sequence becomes active is bounded above by $\Pr[X \geq s]$.

Because the integral rank of a packet corresponds to its birth date and the expected number of packets injected in a time step that include delay edge e_i in their path is at most $\lambda = 1 - \epsilon$,

$$\begin{aligned} \mathrm{E}[X] &\leq \sum_{i=1}^{\ell} (1-\epsilon) \cdot |r_i| \\ &\leq (1-\epsilon) \cdot r \qquad (5) \\ &\overset{(4)}{\leq} (1-\epsilon) \cdot s \ , \qquad (6) \end{aligned}$$

for $k \geq 1$. The binary random variables in the sum of X are stochastically independent, because each generator is operating independently from other generators and previous time steps. Therefore, the probability for a deviation from the expectation can be estimated by using a Chernoff bound (see Lemma 1.1). We define $\delta = s/((1-\epsilon) \cdot r) - 1$. Then

$$\begin{aligned} \Pr[X \geq s] &= \Pr\left[X \geq (1+\delta) \cdot (1-\epsilon) \cdot r\right] \\ &\overset{(5)}{\leq} \mathrm{e}^{-\min\{\delta, \delta^2\} \cdot (1-\epsilon) \cdot r/3} \ . \end{aligned}$$

A further bound, solely depending on s and ϵ, can be derived as follows.

$$\begin{aligned} \Pr[X \geq s] &= \Pr[X \geq \left(1 + \frac{\epsilon}{1-\epsilon}\right) \cdot (1-\epsilon) \cdot s] \\ &\overset{(6)}{\leq} \mathrm{e}^{-\min\left\{\frac{\epsilon}{1-\epsilon}, \left(\frac{\epsilon}{1-\epsilon}\right)^2\right\} \cdot (1-\epsilon) \cdot s/3} \\ &\leq \mathrm{e}^{-\epsilon^2 \cdot s/3} \ . \end{aligned}$$

Up to now we have only calculated the probability that a fixed delay sequence becomes active. It remains to sum over all delay sequences that possibly caused the delay of packet p_0. The number of different (p_0, s, ℓ, r)-delay sequences can be bounded as follows. The maximum node degree in the network is Δ. Hence, the number of possibilities to determine the delay path starting at the destination of p_0 is at most Δ^ℓ. Once the path is fixed, the number of possibilities to choose the delay edges is at most Δ^ℓ, too, because edge e_i is incident to node v_i, for $1 \leq i \leq \ell$. Further, the number of different ways for specifying the rank intervals is equivalent to the number of binary strings of length r with exactly $\ell - 1$ ones. This number is $\binom{r}{\ell-1}$. As a consequence, the probability that there exists an active (p_0, s, r, ℓ)-delay sequence for a fixed set of paramaters s, ℓ, and r fulfilling the equation

in Lemma 2.3 is at most

$$\binom{r}{\ell-1} \cdot \Delta^{2\ell} \cdot \Pr[X \geq s] \;\leq\; \underbrace{\left(\frac{\mathrm{e}\Delta^2 \cdot r}{\ell-1}\right)^{\ell} \cdot \sqrt{\mathrm{e}^{-\min\{\delta,\delta^2\}\cdot(1-\epsilon)\cdot r/3}} \cdot \sqrt{\mathrm{e}^{-\epsilon^2\cdot s/3}}}_{=:\,Y}$$
$$\leq\; \sqrt{\mathrm{e}^{-\epsilon^2\cdot s/3}}\ . \tag{7}$$

The last inequality assumes $Y \leq 1$, which holds if k is chosen sufficiently large in $\Theta(\log(\Delta/\epsilon)/\epsilon)$. We defer the corresponding calculations to the end of this proof.

Let $t(p_0)$ denote the routing time of packet p_0. If $t(p_0) > L + T$, for any $T > 0$, then we can construct an active delay sequence as described in Lemma 2.3. Consequently, the probability that $t(p_0) > L + T$ is bounded above by the probability for the existence of an active (p_0, s, ℓ, r)-delay sequence whose parameters fulfill the following constraints. Equation 4 yields $r \leq s$ and $\ell \leq s/(k-1) \leq s$, assuming $k \geq 2$. Lemma 2.3 yields $s \geq T$. Now applying the bound in equation 7, which holds for $T \geq k \cdot L$, we obtain

$$\Pr[t(p_0) > L + T] \;\leq\; \sum_{s=T}^{\infty} \sum_{\ell=1}^{s} \sum_{r=1}^{s} \sqrt{\mathrm{e}^{-\epsilon^2\cdot s/3}}$$
$$\leq\; \frac{2^{-\beta\cdot\epsilon^2\cdot T}}{\epsilon^2}\ , \tag{8}$$

for some suitable constant β. This term is at most $N^{-\alpha}$, for any constant $\alpha > 0$, if $T \geq \alpha' \log N/\epsilon^2$, for a suitably large constant α'. (Recall that $N \geq 1/\epsilon$.) Therefore,

$$t(p_0) \;\leq\; \max\left\{L \cdot (k+1), \frac{\alpha' \log N}{\epsilon^2}\right\} \;=\; O\left(\frac{L \cdot \log(\Delta/\epsilon)}{\epsilon} + \frac{\log N}{\epsilon^2}\right)\ ,$$

w.h.p. It remains to prove the bound on the expected routing time of p_0. In general, for any integer $Z \geq 0$,

$$\mathrm{E}[t(p_0)] \;=\; \sum_{i=1}^{\infty} \Pr[t(p_0) \geq i] \;\leq\; Z + \sum_{i=Z+1}^{\infty} \Pr[t(p_0) \geq i]\ .$$

Applying equation 8, we obtain that $\sum_{i=Z+1}^{\infty} \Pr[t(p_0) \geq i] \leq 2 \cdot Z$, for $Z \geq \max\{L\cdot(k+1), L + 4\cdot\log(1/\epsilon)/\epsilon^2\}$. In this case,

$$\mathrm{E}[t(p_0)] \;\leq\; 3 \cdot Z \;=\; O\left(\frac{L \cdot \log(\Delta/\epsilon)}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon^2}\right)\ ,$$

which corresponds to the bound on the expected routing time given Theorem 2.1.

Deferred calculations: Finally, we show that $Y \leq 1$ if k is chosen sufficiently large in $\Theta(\log(\Delta/\epsilon)/\epsilon)$. First, we estimate δ.

$$\delta \;=\; \frac{s}{(1-\epsilon)\cdot r} - 1 \;\overset{(4)}{\geq}\; \frac{r + (k-1)\cdot\ell}{(1-\epsilon)\cdot r} - 1 \;\geq\; \epsilon + \frac{(k-1)\cdot\ell}{r}\ .$$

Depending on the value of δ, we distinguish two cases. Consider the case $\delta \leq 1$. Assume $Y > 1$. Then

$$\left(\frac{\mathrm{e}\Delta^2 \cdot r}{\ell-1}\right)^{\ell} \cdot \mathrm{e}^{-\delta^2(1-\epsilon)\cdot r/6} \;>\; 1$$

Applying $\delta \geq \epsilon$, that is, substituting ϵ for δ, and solving the resulting equation for r/ℓ yields $r/\ell = O\left(\log(\Delta/\epsilon)/\epsilon^2\right)$. We assume that k is chosen in such a way that

$$k \;\geq\; \sqrt{\frac{r}{\ell} \cdot \frac{6}{(1-\epsilon)} \cdot \log\left(\frac{\mathrm{e}\Delta^2 \cdot r}{\ell - 1}\right)} + 1 \;=\; O\left(\frac{\log(\Delta/\epsilon)}{\epsilon}\right) \;,$$

for all possible choices of r and ℓ. Now applying $\delta \geq (k-1) \cdot \ell/r$ yields

$$Y \;\leq\; \left(\frac{\mathrm{e}\Delta^2 \cdot r}{\ell - 1}\right)^{\ell} \cdot \mathrm{e}^{-\left(\frac{(k-1)\cdot\ell}{r}\right)^2 \cdot (1-\epsilon)\cdot r/6} \;\leq\; 1 \;.$$

Similarly, we get $Y \leq 1$ for the case $\delta > 1$, too, already for $k = O(\log(\Delta/\epsilon))$. This completes the proof of Theorem 2.1. □

3 Dynamic routing in arbitrary networks with bounded buffers

The tuned ghost protocol can be used to construct an efficient routing algorithm for arbitrary paths in an arbitrary (non-leveled) network G with bounded buffers. In this section, we only consider the stochastic model. In Section 6, we will show how the results obtained in this model can be adapted to the adversarial model. In the stochastic model, our approach yields a dynamic routing protocol that is stable for any injection rate $\lambda < 1$ and requires only small edge buffers. The dynamic protocol uses the following simulation technique.

Suppose the maximum length of a routing path is D. Define $L = \lceil D \cdot (1+1/\epsilon) \rceil$ with $\epsilon = 1 - \lambda$. G simulates the tuned ghost protocol on a leveled network G' of depth L under a maximum injection rate of $\lambda' \leq 1 - \epsilon^2$. G' is defined as follows. On each level, it contains a node for every node in G. A node u from level i and a node v from level $i+1$, for $0 \leq i \leq L-1$, are connected by an edge if and only if the corresponding nodes in G are connected by an edge.

Each edge of the leveled network is simulated by its respective counterpart in G. Hence, every edge in G has to simulate L edges of G' and, therefore, the buffer size in G has to be L times the buffer size of the simulated network G'. The simulation works in a round robin fashion, that is, in each time step t, the edges of G simulate the edges of G' between the nodes of level i and level $i+1$ with $i = t \bmod L$. For each injected packet p, the generator chooses an *offset* κ_p uniformly at random from the range 0 to $\lceil D/\epsilon \rceil - 1$. The routing path in the leveled network starts from level κ_p and simply follows the course prescribed by the original path until it reaches the packet's destination on level $\kappa_p + d_p \leq L$, where d_p denotes the length of the routing path.

Next we calculate the virtual injection rate λ' in the simulated network G'. On the one hand, the virtual rate at which each generator injects packets into G' is L times larger than the actual rate in G, because each edge in G is activated only every Lth step. On the other hand, the probability that an injected packet that traverses an edge e in G also traverses a fixed edge e' in G' that corresponds to e is at most $1/\lceil D/\epsilon \rceil$ because of the randomly selected offset. Therefore,

$$\lambda' \;\leq\; \frac{\lambda \cdot L}{\lceil D/\epsilon \rceil} \;=\; \frac{(1-\epsilon) \cdot \lceil D \cdot (1+1/\epsilon) \rceil}{\lceil D/\epsilon \rceil} \;\leq\; (1-\epsilon) \cdot (1+\epsilon) \;=\; 1 - \epsilon^2 \;.$$

Substituting this injection rate into Theorem 2.1 and applying $L = \Theta(D/\epsilon)$ yields the following result.

Corollary 3.1 *The simulation of the tuned ghost packet protocol yields a dynamic routing algorithm that is stable for any injection rate $\lambda \leq 1 - \epsilon$, for any $\epsilon > 0$, provided that buffers of sufficiently large size $O(D \cdot \log(\Delta/\epsilon)/\epsilon^3)$ are used. Furthermore, each packet is delivered in time $O(D^2 \cdot \log(\Delta/\epsilon)/\epsilon^4 + D \cdot \log(1/\epsilon)/\epsilon^5)$, expected, and $O(D^2 \cdot \log(\Delta/\epsilon)/\epsilon^4 + D \cdot (\log N)/\epsilon^5)$, w.h.p.*

4 Routing along shortest paths in arbitrary networks

In this section, we assume that each buffer has space for an unlimited number of packets. The goal is to achieve a better routing time than in the previous section in which we assumed buffers of limited size. We assume that packets do not make detours, that is, all routing paths are shortest paths. Initialy, we investigate the stochastic model. In Section 6, we will show how the results obtained in this model can be adapted to the adversarial model.

We investigate a dynamic variant of the growing rank protocol that was introduced for static routing in [10, 11]. We have introduced this dynamic variant before in [16], but the analysis we give there only holds for injection rates $\lambda < 1/\mathrm{e}$. In the following, we show that the results can be extended to hold for any constant injection rate $\lambda < 1$.

The dynamic growing rank protocol works as follows. Define the initial rank of a packet to be the time step in which the packet is injected. Whenever the packet traverses a link, its rank is increased by some fixed integer $m \geq 1$, which will be specified later on. If several packets want to traverse the same link at the same time, then the packet with minimal rank is chosen. (In order to break ties, if there are several packets with the same rank, the packet with minimum generator id is taken.)

The following theorem summarizes the results of our analysis of the dynamic growing rank protocol. Note that neither the maximum injection rate for which the protocol is stable nor the routing time depend on the degree of the network.

Theorem 4.1 *Suppose all routing paths are shortest paths. Then the growing rank protocol is stable for any injection rate λ up to some $1-\epsilon$ with $\epsilon = \Theta((\log m)/\sqrt{m})$. Furthermore, the routing time for each individual packet p that has to travel along a routing path of length d_p is $O(m \cdot d_p)$, expected, and $O(m \cdot (d_p + \log N))$, w.h.p.*

We point out that similar results can be obtained by applying the black box transformation scheme presented later in Section 5. The direct transformation of the growing rank protocol, however, is more natural and more elegant as it does not require to partition the time into fixed size blocks in which the static protocol is executed. Further, the expected routing time guaranteed by the specific transformation is slightly better, i.e., $O(d_p)$ rather than $O(d_p + \log N)$.

Proof. We use a delay sequence argument that is similar to the one for the ghost packet protocol. A (p, s, ℓ, r)-delay sequence is defined as in Definition 2.2 except for the following changes. The delay edges are the edges on the delay path rather than edges that are only incident on that path. The intervals of ranks do not overlap. Instead, the smallest rank in r_i is equal to the maximum rank in r_{i+1} plus m, that is, neighboring intervals are separated by a gap of $m-1$ (integral) ranks. As an additional component, the delay sequence includes ℓ integers $s_1, \ldots, s_\ell$ such that $s = (\sum_{i=1}^{\ell} s_i) - \ell + 1$. In an active delay sequence, each delay edge e_i must be traversed by s_i packets with a rank from r_i.

Lemma 4.2 *Suppose a packet p_0 that has a routing path of length d_{p_0} takes $m \cdot d_{p_0} + T$ or more steps to reach its destination. Then there is an active (p_0, s, ℓ, r)-delay sequence with $s \geq T + r + (m-2) \cdot \ell$.*

Proof. The construction of the active delay sequence is analogous to the one used in Section 2 for the ghost packet protocol. In fact, the construction becomes slightly simpler as we neither have to consider delays because of blocked edges nor delays due to ghost packets. The shortest path restriction ensures that all recorded delay packets are distinct. (For a proof see [11], Lemma 2.3.) Neighboring intervals of ranks are separated by a gap of $m-1$ ranks because the rank of a packet increases by m whenever the packet moves along an edge.

It remains to prove the bound on the parameters of the delay sequence. Recall that $r = \sum_{i=1}^{\ell} |r_i|$ with $|r_i|$ denoting the length of the interval of ranks assigned to delay edge e_i. The highest rank in the recorded sequence

is $\mathrm{birth}(p_0) + (d_{p_0} - 1) \cdot m$, which is the rank of p_0 on the last edge of its routing path, and the lowest rank is $\mathrm{birth}(p_s)$, which is the rank of p_s on the first edge of its routing path. As neighboring intervals of ranks are separated by a gap of $m - 1$ ranks we get

$$\begin{aligned} r &= \mathrm{birth}(p_0) - \mathrm{birth}(p_s) + (d_{p_0} - 1) \cdot m - (\ell - 1) \cdot (m - 1) + 1 \\ &= \mathrm{birth}(p_0) - \mathrm{birth}(p_s) + d_{p_0} \cdot m - \ell \cdot (m - 1) \ . \end{aligned} \tag{9}$$

Define t to be the number of time steps covered by the construction, i.e., the time from the birth of the packet p_s to the arrival of the packet p_0 at its destination. Then

$$\begin{aligned} t &\geq \mathrm{birth}(p_0) - \mathrm{birth}(p_s) + d_{p_0} \cdot m + T \\ &\overset{(9)}{=} r + (m - 1) \cdot \ell + T \ . \end{aligned} \tag{10}$$

Each of the t time steps recorded in the delay sequence is either one of s delays or one of ℓ packet movements. Therefore, $t = s + \ell$, and we can conclude

$$\begin{aligned} s &= t - \ell \\ &\overset{(10)}{\geq} r + (m - 2) \cdot \ell + T \ , \end{aligned}$$

which completes the proof of Lemma 4.2. □

Now fix an arbitrary packet p_0, and suppose that p_0 needs more than $d_{p_0} \cdot m + T$ time steps to reach its destination. Then Lemma 4.2 yields that a (p_0, s, ℓ, r)-delay sequence is active with

$$s \ \geq \ r + (m - 2) \cdot \ell \ \geq \ r + \ell \ , \tag{11}$$

for $m \geq 3$. Hence, at least $r + \ell$ packets are needed for an active (p_0, s, ℓ, r)-delay sequence. However, we will show that the expected number of packets that pass the delay edges and have the corresponding rank is at most $(1 - \epsilon) \cdot r$. Hence, this event is very unlikely.

Suppose that the delay edges and the range of ranks for each delay edge are fixed. For $j \in r_1 \cup \ldots \cup r_\ell$ and $1 \leq i \leq \ell$, let the binary random variable $X^g_{j,i}$ be one if and only if generator g generates a packet that has rank $j \in r_i$ at delay edge e_i. Let $X = \sum_{j,i,g} X^g_{j,i}$. Then the probability that a fixed delay sequence becomes active is bounded by $\Pr[X \geq s]$.

The expected number of packets traversing a fixed edge e with some fixed rank r is at most $\lambda = 1 - \epsilon$ because the rank of a packet p at edge e corresponds to its injection time plus an offset that depends on the distance between e and the source node of p. Therefore,

$$\begin{aligned} \mathrm{E}[X] &\leq \sum_{i=1}^{\ell} (1 - \epsilon) \cdot |r_i| \\ &\leq (1 - \epsilon) \cdot r \\ &\overset{(11)}{\leq} (1 - \epsilon) \cdot s \ . \end{aligned} \tag{12}$$

Thus, applying a Chernoff bound (see Lemma 1.1) yields

$$\begin{aligned} \Pr[X \geq s] &= \Pr\left[X \geq \left(1 + \frac{\epsilon}{1 - \epsilon}\right) \cdot (1 - \epsilon) \cdot s\right] \\ &\overset{(12)}{\leq} \mathrm{e}^{-\min\left\{\frac{\epsilon}{1-\epsilon}, \left(\frac{\epsilon}{1-\epsilon}\right)^2\right\} \cdot (1-\epsilon) \cdot s/3} \\ &\leq \mathrm{e}^{-\epsilon^2 \cdot s/3} \ . \end{aligned} \tag{13}$$

So far, we have only calculated the probability that a fixed delay sequence becomes active. It remains to multiply this probability with the number of possible delay sequences. The number of possibilities to choose the r_i's is $\binom{r}{\ell-1}$. The number of possibilities to choose the s_i's is $\binom{s+\ell-1}{\ell-1}$. Enumerating explicitely all possible delay paths, as we have done for the ghost packet protocol, would give us another factor of Δ^ℓ. However, we can avoid this factor in case of the growing rank protocol because the delay path is fixed when the delay packets and the s_i's are specified.

The technical problem with the last assumption is that we supposed before, when estimating the probability for $X \geq s$, that the delay path is fixed whereas we assume here that the delay packets and, hence, the random variables $X^g_{j,i}$ are fixed. This dilemma can be solved by constructing the delay path iteratively. Suppose we have specified the delay path up to the dth edge, for some $d \geq 0$, starting from the destination of packet p_0. At this point only some of the $X^g_{j,i}$ random variables above are well defined, namely those variables with $i \leq d$. The specification of the outcome of these variables, however, gives us the delay packets $p_0, \ldots, p_k$, for $k = \sum_{i=1}^{d} s_i$, and following the path of packet p_k backwards for one edge gives us the next edge on the delay path. In this way, the delay path and the outcome of the $X^g_{j,i}$ variables can be determined alternately. Note that the fact that the definition of some random variables depends on the outcome of other variables does not effect the applicability of the Chernoff bounds because their outcome remains independent.

Combining all these results, the probability that a (p_0, s, ℓ, r)-delay sequence, for a fixed set of parameters p_0, s, ℓ, and r, is active is bounded above by

$$\binom{r}{\ell-1} \cdot \binom{s+\ell-1}{\ell-1} \cdot \Pr[X \geq s] \overset{(11)(13)}{\leq} \left(\frac{\mathrm{e}s}{\ell}\right)^{2\ell} \cdot \mathrm{e}^{-\epsilon^2 \cdot s/3} \ .$$

Now we can bound the probability that packet p_0 has a long routing time. If this packet takes $d_{p_0} \cdot m + T$ steps then a (p_0, s, ℓ, r)-delay sequence is active with $s \geq T + r + (m-2) \cdot \ell$. This constraint yields $s \geq T$, $\ell \leq s/(m-2)$, and $r \leq s$. Hence, the probability for this event is bounded above by

$$\begin{aligned} \sum_{s=T}^{\infty} \sum_{\ell=1}^{s/(m-2)} \sum_{r=1}^{s} \left(\frac{\mathrm{e}s}{\ell}\right)^{2\ell} \cdot \mathrm{e}^{-\epsilon^2 \cdot s/3} &\overset{(11)}{\leq} \sum_{s=T}^{\infty} s^2 \cdot \left(\mathrm{e}(m-2)\right)^{2s/(m-2)} \cdot \mathrm{e}^{-\epsilon^2 \cdot s/3} \\ &= \frac{2^{-\Omega(\epsilon^2 \cdot T)}}{\epsilon^2} \ , \end{aligned}$$

if m is chosen appropriately, that is, if we set $m \geq k \cdot \log(1/\epsilon)/\epsilon^2 + 2$, for a sufficiently large constant k. From this result the bounds on the routing time given in Theorem 4.1 can be derived analogously to the calculations for the ghost packet protocol in Section 2. □

5 A black box transformation scheme for universal routing protocols

In this section, we present a black box transformation scheme for arbitrary static, oblivious routing protocols, using the stochastic injection model. In Section 6, we will show that the results for the stochastic model can be adapted to the adversarial model, too.

In the adversarial model, the bounds on the routing time obtained by our transformation are almost equivalent to the results achieved by Rabani and Tardos [13]. However, in addition we prove the stability of the dynamic protocols. Furthermore, in the stochastic model, we show an interesting extra feature: the dynamic protocol delivers each packet in a time corresponding to its individual path length even if the transformed static protocol does not provide that property. Moreover, we show that our black box transformation ensures a fast recovery from any worst case scenario.

Our transformation scheme is especially simple and efficient if the given static protocol is pure. A protocol is called *pure* if it does not use any control packets and does not duplicate any of its packets, that is, every packet

crosses one edge after the other on its routing path, and no other messages are sent. The main results of this section are listed in the following theorem.

Recall that the congestion C of a static routing problem is the maximum number of paths in a path collection crossing the same edge, the dilation D is the maximum length of a path in a path collection, and the complexity M of a static routing problem is the maximum of the number of edges in the network and paths in the path collection. Further, the complexity N of a dynamic routing problem is defined to be the maximum of the number of generators, the number of edges, and $1/(1-\lambda)$.

Theorem 5.1 *Let $\mathcal{S}$ denote any set of paths in an arbitrary network. Consider any static routing protocol $\mathcal{P}$ that sends packets along every collection of paths or subpaths from $\mathcal{S}$ in at most $\gamma C + \delta D + O(\log^{\alpha} M)$ time steps, w.h.p. Then $\mathcal{P}$ can be transformed into a dynamic routing protocol $\mathcal{P}'$ possessing the following properties in the stochastic model. Suppose packets are injected at rate λ and are to be routed solely along paths or subpaths from $\mathcal{S}$. Then the expected routing time of a packet following a path of length d is at most*

1) $O\left(\epsilon^{-2}\left(\delta d + \log^{\alpha} N\right)\right)$, *if* $\gamma = \Theta(1)$, $\delta = \Theta(\log^{\beta} N)$, *for some constant* $0 \leq \beta \leq 1$, *and* $\lambda = (1-\epsilon)/\gamma$ *for some* $\epsilon > 0$;

2) $O\left((\epsilon/2)^{-\frac{2}{1-\beta}}\left(d\log^{\alpha\beta} N + \log^{\alpha} N\right)\right)$, *if* $\gamma = \Theta(1)$, $\delta = C^{\beta}$, *for some constant* $0 < \beta < 1$, *and* $\lambda = (1-\epsilon)/\gamma$ *for some* $\epsilon > 0$;

3) $O\left(\epsilon^{-2}\left(d\log N + \log^2 N\right)\right)$ *if the bound on the runtime of $\mathcal{P}$ is $C \cdot D$ and* $\lambda = (1-\epsilon)/\log N$ *for some* $\epsilon > 0$.

All time bounds also hold w.h.p. Furthermore, the recovery time in all three cases is equal to the respective time bound with d replaced by $D(\mathcal{S})$, the length of the longest path in $\mathcal{S}$. If $\mathcal{P}$ is pure then $\mathcal{P}'$ is stable for any $\lambda < 1$.

Notice that the bounds on the expected routing time of a packet imply stability up to the specified injection rate, depending on the parameters of the static protocol. The stability of pure protocols, however, is independent of these parameters, although the delay of a packet might become exponential in its path length if the injection rate is too high.

Certainly, more than the three cases stated in the theorem can be solved with the techniques below, but we believe that these cases are the most natural ones. Case 1 covers the case in which γ, δ, and α are constants, which we believe is the most important case. Let us apply Theorem 5.1 to some of the static protocols mentioned in Section 1.1.1.

- **Routing along simple paths in arbitrary networks:** The static protocol of Ostrovsky and Rabani [12] with runtime $O(C + D + \log^{1+\kappa} M)$, w.h.p., for any constant $\kappa > 0$, can be transformed into a dynamic protocol that guarantees a routing time of $O(d + \log^{1+\kappa} N)$, w.h.p., for any λ up to some constant < 1.
- **Routing along arbitrary paths in arbitrary networks:** The simple static protocol with runtime $(1+\kappa)\cdot C + O(D\log M)$, w.h.p., for any constant $\kappa > 0$, presented by Leighton, Maggs and Rao in [9], can be used to obtain a dynamic protocol that guarantees a routing time of $O(d\log N)$, w.h.p., for any constant $\lambda < 1$.
- **Greedy routing along simple paths in arbitrary networks:** Any greedy routing protocol can be transformed into a dynamic protocol that delivers each packet in time $O\left(d\log N + \log^2 N\right)$, w.h.p., for any constant $\lambda < 1/\log N$. We point out that the dynamic protocol is greedy, too.

The first and the second example follow from case 1 of the theorem whereas the last example follows from case 3. Since all of these protocols are pure, their dynamic counterparts are stable for any $\lambda < 1$.

In the following we show how to transform $\mathcal{P}$ into a dynamic protocol $\mathcal{P}'$ so that Theorem 5.1 holds.

5.1 Description of $\mathcal{P}'$

Let $\mathcal{P}$ be any static, oblivious routing protocol, and let the injection rate λ be $1-\epsilon$ for some $\epsilon > 0$. Consider the time to be partitioned into consecutive intervals of length T. Let c_T and d_T denote suitable integers. (These parameters will be specified later.) Every newly injected packet waits until the beginning of the next T-interval. Afterwards it tries to traverse d_T edges of its path in each T-interval until it reaches its destination. Whenever it manages to traverse d_T edges within a T-interval, it waits for the next T-interval. If it fails to traverse d_T edges in some T-interval, it is declared a *failed* packet for the rest of the routing.

$\mathcal{P}'$ now works as follows: At the beginning of each T-interval, $\mathcal{P}$ is started with parameters c_T, used as a bound for the congestion, and d_T, used as a bound for the dilation. All packets that have not failed yet are allowed to participate in $\mathcal{P}$. After $(1-\epsilon^2/2)T$ time steps of the T-interval, $\mathcal{P}$ is halted. (All packets that did not manage to traverse d_T edges up to this point fail.) The remaining $T \cdot \epsilon^2/2$ time steps are reserved for the failed packets. To the failed packets, a contention resolution rule called shortest-in-system (or SIS) is applied. SIS always gives precedence to the packet most recently injected (i.e., of youngest age) in case that several packets contend for the same edge.

If $\mathcal{P}$ is pure, then we can improve $\mathcal{P}'$ so that it becomes greedy, i.e., a packet only has to wait because the next edge on its path is used by another packet. In the pure case, we use SIS as a primary contention resolution rule but we manipulate the age of the non-failed packets as follows. (As before, a packet that has not traversed $i \cdot d_T$ edges by the end of its ith T-interval is declared a *failed* packet for the rest of its life.) Every newly injected packet gets an initial age of ∞ until the beginning of the next T-interval. Afterwards, as long as it has not failed or reached its destination yet, its age is set to 0 at the beginning of each T-interval. Whenever it traverses $i \cdot d_T$ edges before the end of the ith T-interval in which it participates, its age is set back to ∞ for the rest of that interval. The age of a failed packet is defined by its injection time. In each time interval, the packets with age 0 are scheduled according to protocol $\mathcal{P}$. Notice that packets that participate in $\mathcal{P}$ always have a lower age than the other packets. Hence, if we use SIS as primary contention resolution rule, we can allow the failed packets to be routed together with the other packets without disturbing the schedule of $\mathcal{P}$.

In order to analyse the performance of $\mathcal{P}'$, we need to study the behavior of SIS.

Bounding the routing time for shortest-in-system.

In this section we bound the routing time for shortest-in-system, given the following model:

Suppose that packets are injected with rate λ, using our standard stochastic model. A packet is said to be *old* if the difference between the actual time step and its injection time is more than K. Otherwise it is called *young*. As long as a packet is young, it is allowed to set its age to an arbitrary value in each step. The age of an old packet is determined by its injection time. The probability that a packet becomes old before it reaches its destination is assumed to be at most p_f (which denotes its *failure probability*). The event that a packet is old may influence the probability that another packet may also be old. We model these dependencies via a *dependency graph* $G = (V, E)$. Each node V represents a (generator,time step)-pair. Edges are chosen in G such that for any independent set $\{(g_1, t_1), \ldots, (g_k, t_k)\} \in V$ with $k \in \mathbb{N}$, under the assumption that g_i injects a packet P_i at step t_i, the probability that P_i becomes old for all $i \in \{1, \ldots, k\}$ is at most p_f^k. That is, for a proof of an upper bound on the number of old packets, any independent set of (g,t)-pairs can be viewed as having independent failure probabilities.

For this model we show the following lemma.

Lemma 5.2 *For any $0 < \epsilon < 1$ and $K \geq 0$ the SIS protocol ensures that, under the above model with injection rate $\lambda = 1-\epsilon$, failure probability p_f and a dependency graph G of maximum degree b, the expected routing time of a packet following a path of length d is at most*

1. *$O((d/\epsilon) \cdot (d(\epsilon^{-1} + b^2) + K))$ if $p_f \leq \epsilon/(2(4d+1))$, and*

2. $O((K+d)\cdot d/\epsilon^{2d})$ *otherwise.*

Proof. We start with proving item 2. Suppose that there is a packet P that has a routing time of more than $\sum_{i=1}^{d}\tau/\epsilon^{2i}$ steps, for some $\tau \geq 2K$. In this case there must exist an $i \in \{1,\ldots,d\}$ for which P was delayed for at least τ/ϵ^{2i} steps at the ith edge of its path. Consider the minimum i for which this holds. Let e be the ith edge on P's path. Then the time interval I from the injection of P till the time when P waited at e for the τ/ϵ^{2i}th time consists of at most

$$\begin{aligned} t &= \sum_{j=1}^{i}\tau/\epsilon^{2j} = \tau\cdot\frac{1}{\epsilon^2}\cdot\frac{(1/\epsilon)^{2i}-1}{(1/\epsilon)^2-1} \\ &= \tau\cdot\frac{(1/\epsilon)^{2i}-1}{1-\epsilon^2} \end{aligned}$$

time steps. Let the random variable X denote the number of (young and old) packets generated during I with paths containing e. Since any packet is allowed to choose its age in an arbitrary way only during the first K steps of its life it holds: If P had to wait for at least τ/ϵ^{2i} steps at e, then $X \geq \tau/\epsilon^{2i} - K$. Clearly,

$$\begin{aligned} \mathrm{E}[X] &\leq (1-\epsilon)\cdot t = \frac{1}{1+\epsilon}\cdot\tau((1/\epsilon)^{2i}-1) \\ &\leq \left(1-\frac{\epsilon}{1+\epsilon}\right)(\tau/\epsilon^{2i}-K)\,. \end{aligned}$$

This is less than the number of packets generated in I that have to delay P at e. Because of the independence assumptions in our stochastic injection model we can apply Chernoff bounds (see Lemma 1.1) to show that for any $0<\epsilon\leq 1$ the probability that p is delayed by at least τ/ϵ^{2i} packets at e is at most

$$\begin{aligned} \Pr\left[X \geq (1+\epsilon)\left(1-\tfrac{\epsilon}{1+\epsilon}\right)(\tau/\epsilon^{2i}-K)\right] &\leq \mathrm{e}^{-\epsilon^2\cdot\left(1-\frac{\epsilon}{1+\epsilon}\right)\cdot(\tau/\epsilon^{2i}-K)/3} \\ &\leq \mathrm{e}^{-\epsilon^2\cdot(1/2)\cdot(\tau-K)(1/\epsilon^{2i})/3} \leq \mathrm{e}^{-\tau\cdot(1/\epsilon)^{2(i-1)}/12}\,. \end{aligned}$$

Hence the probability that the routing time of P exceeds $\sum_{i=1}^{d}\tau/\epsilon^{2i}$ is at most

$$\sum_{i=1}^{d}\mathrm{e}^{-\tau\cdot(1/\epsilon)^{2(i-1)}/12} \leq d\cdot\mathrm{e}^{-\tau/12}\,.$$

Thus the expected routing time of P is at most

$$\begin{aligned} &\sum_{i=1}^{d}2K/\epsilon^{2i} + \sum_{\tau\geq 2K}\left(\sum_{i=1}^{d}(\tau+1)/\epsilon^{2i} - \sum_{i=1}^{d}\tau/\epsilon^{2i}\right)\cdot d\cdot\mathrm{e}^{-\tau/12} \\ &\leq d\cdot 2K/\epsilon^{2d} + \sum_{\tau\geq 2K}(d/\epsilon^{2d})\cdot d\cdot\mathrm{e}^{-\tau/12} \\ &= O\left((K+d)\cdot d/\epsilon^{2d}\right)\,. \end{aligned}$$

Next we prove item 1. Suppose that there is a packet P that has a routing time of more than $K + d\cdot 4\tau/\epsilon$ steps, for some $\tau \geq K$. In this case there must exist an $i \in \{1,\ldots,d\}$ for which P was delayed for at least $4\tau/\epsilon$ time steps at the ith edge of its path while it was already old. Let I be a time interval covering $4\tau/\epsilon$ of these steps. Furthermore, let the random variable X denote the number of young packets that delayed P in I, and let the random variable Y denote the number of old packets that delayed P in I. It clearly holds that

$$\Pr[X+Y = 4\tau/\epsilon] \leq \Pr[X \geq \tau(4/\epsilon-1)] + \Pr[Y\geq\tau]\,.$$

We first bound the probability that $X \geq \tau(4/\epsilon - 1)$. Since any young packet contributing to X must have been injected in a time interval of size $|I| + K$,

$$\mathrm{E}[X] \leq \lambda(|I| + K) = (1 - \epsilon)(4\tau/\epsilon + K) \; .$$

Because of the independence assumptions in our stochastic injection model it follows that

$$\begin{aligned} \Pr[X \geq (1 + \tfrac{\epsilon}{2(1-\epsilon)})(1-\epsilon)(4\tau/\epsilon + K)] &\leq \mathrm{e}^{-\min\{(\frac{\epsilon}{2(1-\epsilon)})^2, \frac{\epsilon}{2(1-\epsilon)}\}(1-\epsilon)(4\tau/\epsilon + K)/3} \\ &\leq \mathrm{e}^{-\min\{\frac{\epsilon}{1-\epsilon},\, 2\}\cdot\tau/3} \\ &\leq \mathrm{e}^{-\epsilon\cdot\tau/3} \; . \end{aligned}$$

Since $(1 + \frac{\epsilon}{2(1-\epsilon)})(1-\epsilon) = 1 - \epsilon/2$ and

$$\frac{4\tau}{\epsilon} - (1 - \tfrac{\epsilon}{2}) \left(\frac{4\tau}{\epsilon} + K \right) \geq 2\tau - K \geq \tau \; ,$$

we also have

$$\Pr[X \geq \tau(4/\epsilon - 1)] \leq \mathrm{e}^{-\epsilon\tau/3} \; .$$

Next we bound the probability that $Y \geq \tau$. Since P can only be delayed in I by old packets that are younger than P and P has an age at the end of I of at most $K + d \cdot 4\tau/3$ steps, we get with $p_f \leq \epsilon/(2(4d+1))$ that

$$\mathrm{E}[Y] \leq \left(K + \frac{d \cdot 4\tau}{\epsilon} \right) \cdot p_f \leq \frac{\tau}{2} \; .$$

In order to prove a tail estimate for Y, we need the following claim, which is a generalization of a result by Rabani and Tardos [13] to non-identically distributed random variables.

Claim 5.3 *If $X_1, \ldots, X_n$ are binary random variables with a dependency graph of degree at most d and $\mathrm{E}[X_i] \leq p$ for all i, then for any $\delta > 0$,*

$$\Pr[S \geq (1 + 2\delta)\mu + 4\delta pd] \leq 4d \cdot \mathrm{e}^{-\min\{\delta,\, \delta^2\}\mu/(6d)} \; .$$

Proof. We will use the following vital fact.

Fact 5.4 *For any sequence of non-negative reals $a_1, \ldots, a_n$ with $A = \sum_{i=1}^{n} a_i$ and $a_i \leq m$ for all $i \in [n]$ and any $k \in \mathbb{N}$, there are integers $1 \leq i_1 < i_2 < \ldots < i_{k-1} \leq n$ such that, with $i_0 = 1$ and $i_k = n + 1$, $\sum_{j=i_\ell}^{i_{\ell+1}-1} a_j \leq A/k + m$ for all $0 \leq \ell \leq k - 1$.*

Obviously, any graph of degree d can be partitioned into at most $d + 1$ independent sets. Let $I_1, \ldots, I_{d+1}$ be the corresponding sets in the dependency graph of $X_1, \ldots, X_n$. Fact 5.4 implies that for any $k \in \mathbb{N}$, $I_1, \ldots, I_{d+1}$ can be split into $m \leq k \cdot d + (d+1)$ independent sets $I'_1, \ldots, I'_m$ with the property that

$$\mu_j = \mathrm{E}\left[\sum_{i \in I'_j} X_i \right] \leq \frac{\mu}{k \cdot d} + p$$

for all $1 \leq j \leq m$. Let us choose $k = 2$, and let $S_j = \sum_{i \in I'_j} X_i$ for all j. Since I'_j is an independent set, its random variables are independent and therefore, according to the Chernoff bounds,

$$\begin{aligned} \Pr[S_j \geq \mu_j + \delta(\mu/(2d) + p)] &= \Pr[S_j \geq (1 + \delta')\mu_j] \quad \text{with } \delta' = \delta(\mu/(2d) + p)/\mu_j \\ &\leq \mathrm{e}^{-\min\{\delta',\, (\delta')^2\}\mu_j/3} \\ &\leq \mathrm{e}^{-\min\{\delta,\, \delta^2\}\mu/(6d)} \; . \end{aligned}$$

Thus we obtain

$$\Pr[S \geq (1+2\delta)\mu + 4\delta pd] \leq \Pr\left[\bigvee_{j\in[m]} (S_j \geq \mu_j + \delta(\mu/(2d)+p))\right] \leq 4d \cdot \mathrm{e}^{-\min\{\delta,\,\delta^2\}\mu/(6d)} \ .$$

□

Using this claim with $p = p_f$, $d = b$ and $\delta = 1/3$ we get

$$\begin{aligned}\Pr[Y \geq \tau] &\leq 4b \cdot \mathrm{e}^{-\delta^2(\tau/2)/(6b)} \\ &= 4b \cdot \mathrm{e}^{-\tau/(108b)} \ .\end{aligned}$$

The probability bounds for X and Y and the fact that there are d possibilities to select an edge where P experiences a high delay imply that the probability that the routing time of P exceeds $K + d \cdot 4\tau/\epsilon$ is at most

$$d\left(\mathrm{e}^{-\epsilon\tau/3} + 4b \cdot \mathrm{e}^{-\tau/(108b)}\right) \ .$$

Thus the expected routing time of P is at most

$$\begin{aligned}&\left(K + \sum_{i=1}^{d} 4K/\epsilon\right) + \sum_{\tau\geq K} (K + d\cdot 4(\tau+1)/\epsilon)\cdot d\left(\mathrm{e}^{-\epsilon\tau/3} + 4b\cdot\mathrm{e}^{-\tau/(108b)}\right) \\ &\leq\ d\cdot 5K/\epsilon + \sum_{\tau\geq K} d\cdot 5(\tau+1)/\epsilon \cdot d\cdot \mathrm{e}^{-\epsilon\tau/3} + \sum_{\tau\geq K} d\cdot 5(\tau+1)/\epsilon\cdot d\cdot 4b\cdot\mathrm{e}^{-\tau/(108b)} \\ &=\ O\left(\frac{K\cdot d}{\epsilon} + \frac{d^2}{\epsilon^2} + \frac{d^2\cdot b^2}{\epsilon}\right) = O\left(\tfrac{d}{\epsilon}(d(\tfrac{1}{\epsilon}+b^2)+K)\right) \ .\end{aligned}$$

□

5.2 Analysis of $\mathcal{P}'$ for non-pure $\mathcal{P}$

First let us introduce some notation. Given a time interval I, $|I|$ denotes its *size*, i.e. the time range it includes. A packet p is called a *$\mathcal{P}$-packet* in a T-interval I if $\mathcal{P}$ is applied to p in I, i.e., p was generated before I and has not failed nor reached its destination yet.

Bounding the routing time of successful packets.

To prove bounds on the routing time of successful packets, we show the following lemma.

Lemma 5.5 *Let $\mathcal{S}$ denote any set of paths, and let ϕ be any constant greater than 0. Consider any static routing protocol $\mathcal{P}$ that sends packets along any path collection (consisting of paths from some specified set) with congestion C and dilation D in at most $\gamma C + \delta D + O(\log^\alpha N)$ time steps, with probability at least $1 - N^{-\phi}/2$, where $\alpha, \gamma, \delta \geq 1$. Then it holds for each T-interval I:*

1) *If $\gamma = \Theta(1)$, $\delta = \Theta(\log^\beta N)$ for some constant $0 \leq \beta \leq 1$, $\lambda = (1-\epsilon)/\gamma$ for some $\epsilon > 0$, $d_T = (\log N)^{\alpha-\beta}$ and $T = \Theta(\epsilon^{-2}(\phi \log N + \log^\alpha N))$ is sufficiently large, then with probability at least $1 - N^{-\phi}$, $\mathcal{P}'$ requires at most $(1-\epsilon^2/2)T$ time steps in I to send any fixed $\mathcal{P}$-packet along d_T edges.*

2) *If $\gamma = \Theta(1)$, $\delta = C^\beta$ for some constant $0 < \beta < 1$, $\lambda = (1-\epsilon)/\gamma$ for some $\epsilon > 0$, $d_T = (\log N)^{\alpha(1-\beta)}$ and $T = \Theta((\epsilon/2)^{-\frac{2}{1-\beta}}(\phi \log N + \log^\alpha N))$ is sufficiently large, then with probability at least $1 - N^{-\phi}$, $\mathcal{P}'$ requires at most $(1-\epsilon^2/2)T$ time steps in I to send any fixed $\mathcal{P}$-packet along d_T edges.*

3) If the runtime of $\mathcal{P}$ is at most $C \cdot D$, $\lambda = (1-\epsilon)/\log N$ for some $\epsilon > 0$, $d_T = \log N$ and $T = \Theta(\epsilon^{-2}\phi \log^2 N)$ is sufficiently large, then with probability at least $1 - N^{-\phi}$, $\mathcal{P}'$ requires at most $(1 - \epsilon^2/2)T$ time steps in I to send any fixed $\mathcal{P}$-packet along d_T edges.

Proof. First, we bound the expected number of $\mathcal{P}$-packets that participate in a T-interval. Consider any fixed edge e. Let the random variable X_t^e denote the number of packets generated at time step t that intend to cross e. Clearly, $\mathrm{E}[X_t^e] \leq \lambda$. Furthermore, let the binary random variable $X_{t,k}^g$ be 1 if and only if generator g generates at time step t a packet that has to cross e as kth edge. Then it holds that $X_t^e = \sum_{g,k} X_{t,k}^g$. Now, let us consider some fixed T-interval I that starts at time t_0. Since we require each packet to traverse $i \cdot d_T$ edges till the end of the ith T-interval in which it participates, the expected number of $\mathcal{P}$-packets that intend to cross e in I is given by

$$\begin{aligned}
\sum_g \sum_{i \geq 0} \sum_{j=0}^{T-1} \sum_{k=0}^{d_T-1} \mathrm{E}[X^g_{t_0-(i+1)T+j, i\cdot d_T+k}] &= \sum_g \sum_{i \geq 0} \sum_{k=0}^{d_T-1} T \cdot \mathrm{E}[X^g_{t_0, i\cdot d_T+k}] \\
&= \sum_g \sum_d T \cdot \mathrm{E}[X^g_{t_0,d}] \\
&\leq T \cdot \lambda \ ,
\end{aligned}$$

because $\Pr[X^g_{t_1,d} = 1] = \Pr[X^g_{t_2,d} = 1]$ for any t_1, t_2, since the injection of packets is independent of the time step.

Now we bound T so that with probability at least $1 - N^{-\phi}$ the $\mathcal{P}$-packets participating in some T-interval I are successful in I. Let us assume that $\lambda = (1-\epsilon)/\gamma$ for some $\epsilon > 0$ (resp. $\lambda = (1-\epsilon)/\log N$ in case 3). We now consider the three cases for the choice of γ and δ given in Lemma 5.5.

Case 1: $\gamma = \Theta(1)$ and $\delta = \Theta(\log^\beta N)$ for some constant $0 \leq \beta \leq 1$.
From above we know that the expected number of $\mathcal{P}$-packets participating in a T-interval that cross some fixed edge is at most λT.

Suppose that $\epsilon \leq 7/8$. Since the injection of a packet is independent of the injection of other packets, we can use a Chernoff bound to show that the probability that the congestion caused by $\mathcal{P}$-packets exceeds $(1+\epsilon)\lambda T$ at some fixed edge is at most $\mathrm{e}^{-\epsilon^2 \lambda T/3}$. Hence, for every $\phi > 0$ there is a $t(\epsilon, \phi) = \Theta(\frac{\phi}{\epsilon^2} \log N)$ such that for all $T \geq t(\epsilon, \phi)$, the probability is at least $1 - N^{-\phi}/2$ that the congestion caused by $\mathcal{P}$-packets in I is at most $(1+\epsilon)\lambda T$. Assume in the following that $T \geq t(\epsilon, \phi)$. Set $c_T = (1+\epsilon)\lambda T$ and $d_T = \log^\psi N$. According to the assumptions of Lemma 5.5, the runtime of $\mathcal{P}$, given a path collection with congestion c_T and dilation d_T, is at most $\gamma c_T + \delta d_T + O(\log^\alpha N)$ with probablity at least $1 - N^{-\phi}/2$. Hence, the time required by any fixed $\mathcal{P}$-packet participating in I to be successful is at most

$$\gamma \cdot (1+\epsilon)\lambda T + \delta \log^\psi N + O(\log^\alpha N) \quad = \quad (1-\epsilon^2)T + \delta \log^\psi N + O(\log^\alpha N) \ ,$$

with probability at least $1 - N^{-\phi}$. This time bound is at most $T(1-\epsilon^2/2)$ if $T \geq \frac{2}{\epsilon^2}(\delta \log^\psi N + O(\log^\alpha N))$. (The remaining $T \cdot \epsilon^2/2$ time steps will be important when considering the failed packets.)

Now suppose that $\epsilon > 7/8$. In this case, $\lambda < 1/(8\gamma)$. Thus, according to the Chernoff bounds the probability that the congestion caused by $\mathcal{P}$-packets exceeds $T/(4\gamma)$ can be made as small as $N^{-\phi}$ for any constant $\phi > 0$ if $T = \Theta(\phi \log N)$ is chosen large enough. Set $c_T = T/(4\gamma)$ and $d_T = \log^\psi N$. Analogous to above, the routing time required by the $\mathcal{P}$-packets participating in I to be successful can be made as small as $T(1 - \epsilon^2/2)$ with probability of at least $1 - N^{-\phi}$ if $T = \Theta(\delta \log^\psi N + \log^\alpha N)$ is chosen large enough.

In both cases for ϵ, the bound for T is (asymptotically) minimal for $\beta + \psi = \alpha$. So altogether $T = \Theta(\frac{1}{\epsilon^2}(\phi \log N + \log^\alpha N))$ suffices to guarantee that any fixed $\mathcal{P}$-packet successfully manages a T-interval with probability at least $1 - N^{-\phi}$.

Case 2: $\gamma = \Theta(1)$ and $\delta = C^\beta$ for some constant $0 < \beta < 1$.
If $\epsilon \leq 15/16$, we set $c_T = (1+\epsilon)\lambda T$ and $d_T = \log^\psi N$. As in case 1, the probability is at most $N^{-\phi}/2$ that the congestion in I exceeds c_T if $T = \Theta(\frac{\phi}{\epsilon^2}\log N)$ is sufficiently large. In this case, the time required by any fixed $\mathcal{P}$-packet participating in I to be successful is at most

$$\gamma \cdot (1+\epsilon)\lambda T + \delta \log^\psi N + \log^\alpha N \leq (1-\epsilon^2)T + T^\beta \log^\psi N + \log^\alpha N \ ,$$

with probability at least $1 - N^{-\phi}$. It holds that

$$\log^\alpha N \leq \frac{\epsilon^2}{4} \cdot T \quad \text{if} \quad T \geq \frac{4}{\epsilon^2} \cdot \log^\alpha N$$

and

$$T^\beta \log^\psi N \leq \frac{\epsilon^2}{4} \cdot T \quad \text{if} \quad \psi = \alpha(1-\beta) \text{ and } T \geq (\epsilon/2)^{-2/(1-\beta)} \log^\alpha N \ .$$

Thus, if $T = \Theta((\epsilon/2)^{-2/(1-\beta)} \log^\alpha N)$ is sufficiently large, then

$$(1-\epsilon^2)T + T^\beta \log^\psi N + \log^\alpha N \leq (1-\epsilon^2/2)T \ .$$

Hence $T = \Theta((\epsilon/2)^{-2/(1-\beta)}(\phi \log N + \log^\alpha N))$ suffices to ensure that any fixed $\mathcal{P}$-packet successfully manages a T-interval with probability at least $1 - N^{-\phi}$.

If $\epsilon > 15/16$, we set $c_T = T/(8\gamma)$ and $d_T = \log^\psi N$. Similar to case 1, the probability is at most $N^{-\phi}/2$ that the congestion in I exceeds c_T if $T = \Theta(\phi \log N)$ is sufficiently large. In this case, the time required by any fixed $\mathcal{P}$-packet participating in I to be successful is at most

$$\gamma \cdot T/(8\gamma) + \delta \log^\psi N + \log^\alpha N \leq T/8 + T^\beta \log^\psi N + \log^\alpha N \ ,$$

with probability at least $1 - N^{-\phi}$. This is at most $(1-\epsilon^2/2)T$ if

$$T^\beta \log^\psi N \leq T/4 \quad \text{and} \quad \log^\alpha N \leq T/8 \ ,$$

which is true if $\psi = \alpha(1-\beta)$ and $T = \Theta((\epsilon/2)^{-2/(1-\beta)} \log^\alpha N)$ is sufficiently large. Hence, if $T = \Theta((\epsilon/2)^{-2/(1-\beta)}(\phi \log N + \log^\alpha N))$, then also in this case any fixed $\mathcal{P}$-packet successfully manages a T-interval with probability at least $1 - N^{-\phi}$.

Case 3: The runtime of $\mathcal{P}$ is at most $C \cdot D$.
Assume that $\lambda = \frac{1-\epsilon}{\log N}$. If $\epsilon \leq 7/8$, we set $c_T = (1+\epsilon)\lambda T$ and $d_T = \log N$. As above, it can be shown that the probability is at most $N^{-\phi}$ that the congestion in I exceeds c_T if $T = \Theta(\frac{\phi}{\epsilon^2}\log^2 N)$ is sufficiently large. In this case, the time required by any fixed $\mathcal{P}$-packet participating in I to be successful is at most

$$(1+\epsilon)\lambda T \cdot \log N = (1-\epsilon^2)T$$

with probability at least $1 - N^{-\phi}$.

If $\epsilon > 7/8$, then we set $c_T = T/(2\log N)$ and $d_T = \log N$. If $T = \Theta(\phi \log^2 N)$ is sufficiently large, the probability is at most $N^{-\phi}$ that the congestion in I exceeds c_T. In this case, the time required by any fixed $\mathcal{P}$-packet participating in I to be successful is at most

$$\frac{T}{2\log N} \cdot \log N \leq (1-\epsilon^2/2)T$$

with probability at least $1 - N^{-\phi}$. □

It follows for case 1 of Lemma 5.5 that, if a packet following a path of length d successfully manages all T-intervals, its runtime is bounded by

$$O\left(\left(\frac{d}{\log^{\psi} N}+1\right)\frac{1}{\epsilon^2}\log^{\alpha} N\right)=O\left(\frac{1}{\epsilon^2}\left(\delta d+\log^{\alpha} N\right)\right) \quad .$$

Similarly, for case 2, the runtime of a packet following a path of length d that successfully manages all T-intervals is bounded by

$$O\left(\left(\frac{d}{\log^{\psi} N}+1\right)(\epsilon/2)^{-\frac{2}{1-\beta}}\log^{\alpha} N\right)=O\left((\epsilon/2)^{-\frac{2}{1-\beta}}\left(d\log^{\alpha\beta} N+\log^{\alpha} N\right)\right) \quad ,$$

and for case 3 the runtime of the packet is bounded by

$$O\left(\left(\frac{d}{\log N}+1\right)\frac{1}{\epsilon^2}\log^2 N\right)=O\left(\frac{1}{\epsilon^2}\left(d\log N+\log^2 N\right)\right) \quad .$$

These time bounds match the time bounds given in Theorem 5.1 if a packet never fails. According to Lemma 5.5, the probability that a packet fails in some T-interval can be made polynomially small in N if λ is sufficiently small. Hence, in this case all time bounds hold w.h.p. It remains to bound the runtimes of failed packets to be able to compute the expected time a packet needs to reach its destination.

Bounding the routing time of failed packets

Next we bound the runtime of failed packets under the assumption that the injection rate is sufficiently small, that is, each individual packet is successful in every T-interval, w.h.p. For this, we will show how to apply Lemma 5.2 to bound the expected routing time of a failed packet by N^c for some constant c. Combined with the upper bound of $N^{-\phi}$ on the probability that a packet fails, this will result in an expected routing time not much larger than the time that the packet requires if it does not fail in any of its T-intervals, provided that $\phi \geq c$. This yields the bounds for the expected routing time in Theorem 5.1 and implies that $\mathcal{P}$ is stable.

Now we show how to apply Lemma 5.2. First, we change the situation that only every $\frac{2}{\epsilon^2}$th step can be used by a failed packet to a situation in which every time step can be used by a failed packet. For this, we replace each generator g by $2/\epsilon^2$ generators $g_1, \ldots, g_{2/\epsilon^2}$, where g_i is responsible for the simulation of the behavior of g at time steps t with $(t \bmod 2/\epsilon^2)+1=i$. Furthermore, we assume each time step to represent now $2/\epsilon^2$ time steps in the original situation. Let us consider some fixed edge e. For any generator g and time step t in the new situation, let the binary random variable Y_t^g be 1 if and only if g generates a packet at step t that intends to cross e and that fails in some T-interval of the original situation. From Lemma 5.5 we know that $\Pr[Y_t^g=1]$ can be made as small as $N^{-\phi}$ for any constant $\phi > 0$.

If the probabilities $\Pr[Y_t^g=1]$ were independent for different Y_t^g, we could model the injection of failed packets as a simple stochastic injection model with injection rate $\lambda = \Pr[Y_t^g=1]\cdot 2/\epsilon^2$. For $\Pr[Y_t^g=1]\cdot 2/\epsilon^2 \leq 1/D(\mathcal{S})$, it would follow directly from Lemma 5.2 (choose $K=0$ and $\lambda = 1/D(\mathcal{S})$) that the routing time of a failed packet is less than N^{ϕ} if ϕ is sufficiently large.

Coping with the dependencies

Unfortunately, there can be high dependencies among failures of packets. In order to incorporate these dependencies in the model of Lemma 5.2, we construct a dependency graph $G=(V,E)$ that has a node (g,t) for each random variable Y_t^g, and in which two nodes (g,t) and (g',t') are connected in G if and only if $t-2(T+1)\cdot D(\mathcal{S})\cdot(\epsilon^2/2) \leq t' \leq t+2(T+1)\cdot D(\mathcal{S})\cdot(\epsilon^2/2)$. Since there are $2N/\epsilon^2$ generators, the maximum degree of G is at most $(4(T+1)\cdot D(\mathcal{S})+2/\epsilon^2)\cdot N$ which is polynomial in N. In order to apply Lemma 5.2,

we need to show that for any independent set $S \subseteq V$, the probability that $Y_t^g = 1$ for all $(g,t) \in S$ is at most $N^{-\phi|S|}$ for any constant $\phi > 0$ (depending on T).

Recall that the proof of Lemma 5.5 bounds the failure probability of a packet within a T-interval I solely by considering the injection events of packets that could have still been successful at the beginning of I and the behavior of $\mathcal{P}$ within I. Further recall that the proof of Lemma 5.5 only uses the congestion (which is upper bounded by the injection events) in I to obtain a probability bound for the success of $\mathcal{P}$. Hence the space $\Omega_{g,t}$ of relevant outcomes that need to be considered to obtain a probability of at most $N^{-\phi}$ for the random variable Y_t^g to be 1 can be limited to contain solely injection events of packets that can participate in $\mathcal{P}$ in some T-interval together with the packet generated by g at step t. Since a packet can be alive without a failure for at most $(T+1) \cdot D(\mathcal{S}) \cdot (\epsilon^2/2)$ time steps, the outcome spaces of any two random variables Y_t^g and $Y_{t'}^{g'}$ with either $t' < t - 2(T+1) \cdot D(\mathcal{S}) \cdot (\epsilon^2/2)$ or $t' > t + 2(T+1) \cdot D(\mathcal{S}) \cdot (\epsilon^2/2)$ must be disjoint (that is, they do not contain a common injection event (g'', t'')). Thus, using the proof of Lemma 5.5, the probability that both of these random variables are 1 can be shown to be at most $N^{-2\phi}$. The same argument extends to any set of random variables that form an independent set in G.

In order to set the remaining parameters in the model of Lemma 5.2, we set λ equal to the given injection rate and $K = 0$ (successful packets cannot interfere with failed packets). According to Lemma 5.2, for these parameters the expected routing time of any failed packet is at most N^{ϕ} if ϕ is large enough. This completes the proof that for a small enough λ, even when considering failures, the expected routing time of any fixed packet is within the time bounds given in Theorem 5.1.

Recovery

We show that $\mathcal{P}'$ recovers very quickly from any worst case scenario. Clearly, the shortest-in-system (SIS) rule instantly recovers from any worst case scenario because younger packets are always preferred. Similar, we show for $\mathcal{P}'$ that after a certain amount of time any configuration of the network has *no* influence on the runtime of the newly generated packets any more, concerning our probability bounds.

Consider any worst case scenario for the injection of packets that ends at time step t_0. Since each packet has to traverse d_T edges in each T-interval in order to remain successful, $R = \lceil \frac{D(\mathcal{S})}{d_T} \rceil \cdot T$ time steps after t_0 there can be no successful packet any more that was generated at time step t_0 or earlier. This implies that afterwards the congestion in a T-interval is only based on packets injected after the worst case scenario, which according to our stochastic injection model is independent of whatever happened during or before a worst case scenario. Since in the proof of Lemma 5.5 we used the worst case assumption that all packets have been successful so far to upper bound the congestion in a T-interval, all probability bounds in Lemma 5.5 are again valid R steps after the worst case scenario. That is, the probability of a packet to fail in a T-interval is again polynomially small in N. This ensures that sufficiently few packets fail. Once a packet fails, its age is determined by its injection time. Since SIS is used for the failed packets, a failed packet generated at a time step $t > t_0$ cannot be blocked by the packets generated during the worst case scenario. Hence $\mathcal{P}'$ recovers after $\lceil \frac{D(\mathcal{S})}{d_T} \rceil \cdot T$ time steps. Substituting the right T for each of the three cases (see Lemma 5.5) yields the recovery time given in Theorem 5.1.

5.3 Analysis of $\mathcal{P}'$ for pure $\mathcal{P}$

The analysis for the successful packets is the same as above. However, since we do not have reserved time slots for the failed packets, we need a different analysis for them if $\mathcal{P}$ is pure.

Bounding the routing time of failed packets

Suppose that the injection rate is sufficiently small, that is, each individual packet is successful in every T-interval, w.h.p. For this, we will show how to apply Lemma 5.2 to bound the routing time of a failed packet by

N^c for some constant c. Combined with the bound of $N^{-\phi}$ on the probability that a packet fails, this will result in an expected routing time not much larger than the time that the packet requires if it does not fail in any of its T-intervals, provided that $\phi \geq c$. This would yield the bounds for the expected routing time in Theorem 5.1 and would imply that $\mathcal{P}'$ is stable.

Now we show how to apply Lemma 5.2. Let us consider some fixed edge e. For any generator g and time step t, let the binary random variable Y_t^g be 1 if and only if g generates a packet at step t that intends to cross e and that fails in some T-interval. From Lemma 5.5 we know that $\Pr[Y_t^g = 1]$ can be made as small as $N^{-\phi}$ for any constant ϕ. Furthermore, similar to the case that $\mathcal{P}$ is non-pure, the event $Y_t^g = 1$ causes at most $(4(T+1) \cdot D(\mathcal{S}) + 1) \cdot N$ other random variables $Y_{t'}^{g'}$ to have a probability of $\Pr[Y_{t'}^{g'} = 1] > N^{-\phi}$, namely those with $t - 2(T+1) \cdot D(\mathcal{S}) \leq t' \leq t + 2(T+1) \cdot D(\mathcal{S})$. In order to incorporate these dependencies in the model of Lemma 5.2, we again construct a dependency graph $G = (V, E)$ that has a node (g, t) for each random variable Y_t^g, and in which two nodes (g, t) and (g', t') are connected in G if and only if $t - 2(T+1) \cdot D(\mathcal{S}) \leq t' \leq t + 2(T+1) \cdot D(\mathcal{S})$. Thus the maximum degree of G is $(4(T+1) \cdot D(\mathcal{S}) + 1) \cdot N$. Furthermore, in the model of Lemma 5.2 we set λ equal to the given injection rate and $K = D(\mathcal{S}) \cdot T$ (successful packets live for at most $D(\mathcal{S}) \cdot T$ time steps). According to Lemma 5.2, in this case the expected routing time of any failed packet is at most N^{ϕ} if ϕ is large enough. This completes the proof that, also for a pure $\mathcal{P}$, for a small enough λ the expected routing time of any fixed packet is within the time bounds given in Theorem 5.1.

Stability for any $\lambda < 1$

Next we consider the case that λ is arbitrarily close to 1. In this situation $\mathcal{P}$ might not be good enough to ensure that a packet manages its T-interval with high probability. However, from Lemma 5.2 (set $K = D(\mathcal{S}) \cdot T$) it directly follows that nevertheless $\mathcal{P}'$ remains stable for any $\lambda < 1$, although the runtime of a packet might get exponential in the length of its path.

6 Adapting our results to the adversarial model

Consider any bounded adversary of rate (w, λ). Set $\epsilon = 1 - \lambda$. For each injected packet p we choose uniformly and independently at random an initial delay of δ_p from the set $\{0, \ldots K-1\}$, where $K = (k \cdot D + w)/\epsilon^2$ and D is the length of a longest possible path. k will be specified later as it depends on the routing protocol for which we want to adapt the results. After waiting δ_p time steps in its injection buffer, p chooses the actual time step as its *new injection time* and participates in whatever routing protocol chosen. We say that every packet with new injection time t *touches* an edge e at step t' if e is the ith edge on the routing path of p and $t' = t + k \cdot i$. Let λ' denote the maximum, over all edges and time steps, of the expected number of packets that touch an edge in a time step. Then the following lemma holds.

Lemma 6.1 $\lambda' \leq 1 - \epsilon/2$.

Proof. Consider some fixed edge e and time step t. The maximum number of packets that touch edge e at step t is at most

$$(1-\epsilon)w \cdot \left\lceil \frac{k \cdot D + K}{w} \right\rceil \quad \leq \quad (1-\epsilon) \cdot (k \cdot D + w) \cdot (1 + 1/\epsilon^2) \ .$$

Since each of these packets chooses a random initial delay out of a range of $[(k \cdot D + w)/\epsilon^2]$, it holds that

$$\begin{aligned} \lambda' &\leq \frac{(1-\epsilon) \cdot (k \cdot D + w) \cdot (1 + 1/\epsilon^2)}{(k \cdot D + w)/\epsilon^2} \\ &\leq (1-\epsilon) \cdot (1 + \epsilon^2) \leq 1 - \epsilon/2 \end{aligned}$$

for any $\epsilon \in [0,1]$. □

As is not difficult to verify, this lemma can be used to transfer all results presented in this paper to the adversarial model. For the ghost packet protocol on general networks, we choose $k = 0$. Then it follows from Lemma 6.1 that the expected number of packets with a fixed rank r that traverse an edge e is at most $1 - \epsilon/2$. Hence, the analysis for the ghost packet protocol on general networks holds without any further change also for the adversarial model. We only have to add w/ϵ^2 to the delay bound of the packets to include the random initial delay.

For the growing rank protocol we choose $k = m$. Then it follows from Lemma 6.1 that, for any edge e and rank r, the expected number of packets with rank r that traverse e is at most $1 - \epsilon/2$. This allows us to use the same analysis as in the proof of Theorem 4.1 to show that the delay of any packet is bounded by $O(\frac{1}{\epsilon^2}(m \cdot D + w) + m \cdot \log N) = O(m^2 \cdot D + m \cdot (w + \log N))$, w.h.p.

For the black box transformation it suffices to choose $k = T(1/d_T + 1/D)$. Then we get that, for any edge e and T-interval I, the expected number of packets that intend to cross e in I is at most $T(1 - \epsilon/2)$. In this case, the same analysis as in Section 5 can be used to show that, for instance, for $\gamma = \Theta(1)$ and $\delta = \Theta(\log^\beta N)$ the delay of any packet is bounded by $O(\frac{1}{\epsilon^2}(\frac{1}{\epsilon^2}\delta D + w + \log^\alpha N))$, w.h.p., if $\lambda \leq (1 - \epsilon)/\gamma$.

7 Open Problems

In this paper, we presented transformations of well-known static routing algorithms (such as the ghost packet protocol and the growing rank protocol) into efficient dynamic routing algorithms. We, for instance, obtained a dynamic routing algorithm for arbitrary leveled networks that only requires buffers of size depending on the injection rate and the maximum degree of the network to be stable up to a maximum possible injection rate. This algorithm, however, uses ghost or control packets. Although our analysis implicitly shows that these packets are sent very rarely, the question arises whether or not control packets can be avoided completely.

Furthermore, we presented a black box transformation scheme applicable to every static, oblivious routing algorithm. Our results show that it might be important for static routing protocols to know the exact constant in front of the C in the runtime bound, since this determines up to which injection rate the resulting dynamic protocol ensures a small delay for the packets. It is an interesting question how large this constant is for the known static routing protocols, and whether there exist static routing protocols with runtime $C + O(D + \log N)$.

References

[1] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention-resolution protocols. In *Proc. of the 37th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 380–389, 1996.

[2] M. Andrews, A. Fernández, M. Harchol-Balter, T. Leighton, and L. Zhang. General dynamic routing with per-packet delay guarantees of O(distance + 1 / session rate). In *Proc. of the 38th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 294–302, 1997.

[3] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson. Adversarial queueing theory. In *Proc. of the 28th ACM Symp. on Theory of Computing (STOC)*, pages 376–385, 1996.

[4] A. Z. Broder, A. M. Frieze, and E. Upfal. A general approach to dynamic packet routing with bounded buffers. In *Proc. of the 37th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 390–399, 1996.

[5] E. G. Coffman, N. Kahale, and F. T. Leighton Processor-Ring Communication: A Tight Asymptotic Bound on Packet Waiting Times. *SIAM Journal on Computing*, 27(5):1221–1236, 1998.

[6] R. Cypher, F. Meyer auf der Heide, C. Scheideler, and B. Vöcking. Universal algorithms for store-and-forward and wormhole routing. In *Proc. of the 28th ACM Symp. on Theory of Computing (STOC)*, pages 356–365, 1996.

[7] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays • Trees • Hypercubes*. Morgan Kaufmann, San Mateo, CA, 1992.

[8] F. T. Leighton, B. M. Maggs, A. G. Ranade, and S. B. Rao. Randomized routing and sorting on fixed-connection networks. *Journal of Algorithms*, 17:157–205, 1994.

[9] F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, 1994.

[10] F. Meyer auf der Heide and B. Vöcking. A packet routing protocol for arbitrary networks. In *Proc. of the 12th Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 291–302, 1995.

[11] F. Meyer auf der Heide and B. Vöcking. Shortest paths routing in arbitrary networks. *Journal of Algorithms, to appear*, 1999.

[12] R. Ostrovsky and Y. Rabani. Universal $O(\text{congestion} + \text{dilation} + \log^{1+\epsilon} n)$ local control packet switching algorithms. In *Proc. of the 29th ACM Symp. on Theory of Computing (STOC)*, pages 644–653, 1997.

[13] Y. Rabani and E. Tardos. Distributed packet switching in arbitrary networks. In *Proc. of the 28th ACM Symp. on Theory of Computing (STOC)*, pages 366–375, 1996.

[14] A. G. Ranade. How to emulate shared memory. *Journal of Computer and System Science*, 42:307–326, 1991.

[15] G.D. Stamoulis and J.N. Tsitsiklis. The Efficiency of Greedy Routing in Hypercubes and Butterflies. In *Proc. of the 3rd Annual ACM Symposium on Parallel Algorithms and Architectures*, pp. 248-259, 1991.

[16] C. Scheideler and B. Vöcking. Universal continuous routing strategies. In *Proc. of the 8th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 142–151, 1996.

[17] C. Scheideler and B. Vöcking. Universal continuous routing strategies. *Theory of Computing Systems*, 31:425–449, 1998.

[18] C. Scheideler and B. Vöcking. From static to dynamic routing: Efficient transformations of store-and-forward protocols. In *Proc. of the 31th ACM Symp. on Theory of Computing (STOC)*, 1999.