Throughput-Optimal Routing in Unreliable Networks

Paul Bunn Rafail Ostrovsky

Abstract

We demonstrate the feasibility of throughput-efficient routing in a highly unreliable network. Modeling a network as a graph with vertices representing nodes and edges representing the links between them, we consider two forms of unreliability: unpredictable edge-failures, and deliberate deviation from protocol specifications by corrupt nodes. The first form of unpredictability represents networks with dynamic topology, whose links may be constantly going up and down; while the second form represents malicious insiders attempting to disrupt communication by deliberately disobeying routing rules, by e.g. introducing junk messages or deleting or altering messages. We present a robust routing protocol for end-to-end communication that is simultaneously resilient to both forms of unreliability, achieving provably optimal throughput performance. Our proof proceeds in three steps: 1) We use competitive-analysis to find a lower-bound on the optimal throughput-rate of a routing protocol in networks susceptible to only edge-failures (i.e. networks with no malicious nodes); 2) We prove a matching upper bound by presenting a routing protocol that achieves this throughput rate (again in networks with no malicious nodes); and 3) We modify the protocol to provide additional protection against malicious nodes, and prove the modified protocol performs (asymptotically) as well as the original.

Keywords. Network Routing; Fault Localization; Multi-Party Computation in Presence of Dishonest Majority; Communication Complexity; End-to-End Communication; Competitive Analysis; Asynchronous Protocols

1 Introduction

With the immense range of applications and the multitude of networks encountered in practice, there has been an enormous effort to study routing in various settings. For the purpose of developing network models in which routing protocols can be developed and formally analyzed, networks are typically modelled as a graph with vertices representing nodes (processors, routers, etc.) and edges representing the connections between them. Beyond this basic structure, additional assumptions and restrictions are then made in attempt to capture various features that real-world networks may display. In deciding which network model is best-suited to a particular application, developers must make a choice with respect to each of the following considerations: 1) Synchronous or Asynchronous; 2) Static or Dynamic Topology; 3) Global Control or Distributed/Local Control; 4) Connectivity/Liveness Assumptions; 5) Existence of Faulty/Malicious Nodes.

Notice that in each option above there is an inherent trade-off between generality/applicability of the model verses optimal performance within the model. For instance, a protocol that assumes a fixed network topology will likely out-perform a protocol designed for a dynamic topology setting, but the former protocol may not work in networks subject to edge-failures. Similarly, a protocol that protects against the existence of faulty or deliberately malicious nodes will likely be out-performed in networks with no faulty behavior by a protocol that assumes all nodes act honestly.

From both a theoretical and a practical standpoint, it is important to understand how each (combination) of the above listed factors affects routing performance. In this paper, we explore the feasibility of end-to-end routing in highly unreliable networks, i.e. networks that simultaneously consider *all* of the more general features: Asynchronous, Dynamic Topology, Local Control, no Connectivity Assumptions, and the existence of deliberately Malicious Nodes. Admittedly, in this "worst-case" model it is unlikely that any protocol will perform well, and one (or more) stronger assumption(s) must be made to achieve a reasonable level of performance. However, understanding behavior in the worst case, even with respect to the most basic task of end-to-end communication, is important to determine how much (if any) the addition of each assumption improves optimal protocol performance.

1.1 Previous Work

As mentioned above, development and analysis of routing protocols relies heavily on the choices made for the network model. To date, all network models have guaranteed at least one (and more commonly multiple) "reliability" assumption(s) with respect to the above list of five network characteristics. In this section, we explore various combinations of assumptions that have been made in recent work, highlighting positive and negative results with respect to each network model, emphasizing clearly which assumptions are employed in each case. Since our work focuses on theoretical results, for space considerations we do not discuss below the vast amount of research and analysis of routing issues for specific network systems encountered in practice, e.g. the Internet. Even still, the amount of research regarding network routing and analysis of routing protocols is extensive, and as such we include only a sketch of the most related work, indicating how their models differ from ours and providing references that offer more detailed descriptions.

End-to-End Communication: One of the most relevant research directions to our paper is the notion of End-to-End communication in distributed networks, where two nodes (sender and receiver) wish to communicate through a network. While there is a multitude of problems that involve end-to-end communication (e.g. End-to-End Congestion Control, Path-Measurement, and Admission Control), we discuss here work that consider networks whose only task is to facilitate communication between sender and receiver. Some of these include a line of work developing the *Slide* protocol (the starting point of our protocol): Afek and Gafni [2], Awerbuch et al. [12], Afek et al. [1], and Kushilevitz et al. [18]. The Slide protocol (and its variants) have been studied in a variety of network settings, including multi-commodity flow (Awerbuch and Leighton [11]), networks controlled by an online bursty adversary (Aiello et al. [4]), and networks that allow corruption of nodes (Amir et al. [7]). However, prior to our work there was no version of the Slide protocol that considered routing in the "worst case" network setting: only [7] considers networks in which nodes are corruptible, but their network model assumes synchronous communication and demands minimal connectivity guarantees.

Fault Detection and Localization Protocols: There have been a number of papers that explore the possibility of corrupt nodes that deliberately disobey protocol specifications in order to disrupt communication. In particular, there is a recent line of work that considers a network consisting of a *single path* from the sender to the receiver, culminating in the recent work of Barak et al. [13] (for further background on fault localization see references therein). In this model, the adversary can corrupt any node (except the sender and receiver) in a dynamic and malicious manner.

Since corrupting any node on the path will sever the honest connection between sender and receiver, the goal of a protocol in this model is *not* to guarantee that all messages sent are received. Instead, the goal is to *detect* faults when they occur and to *localize* the fault to a single edge.

Goldberg et al. [17] show that a protocol's ability to detect faults relies on the assumption that One-Way Functions (OWF) exist, and Barak et al. [13] show that the (constant factor) overhead (in terms of communication cost) incurred for utilizing cryptographic tools (such as MACs or Signature Schemes) is mandatory for any fault-localization protocol. Awerbuch et al. [10] also explore routing in the Byzantine setting, although they do not present a formal treatment of security, and indeed a counter-example that challenges their protocol's security is discussed in the appendix of [13].

Fault Detection and Localization protocols focus on very restrictive network models (typically synchronous networks with fixed topology and some connectivity assumptions), and throughput-performance is usually not considered when analyzing fault detection/localization protocols.

Competitive Analysis: Competitive Analysis was first introduced by Sleator and Tarjan [21] as a mechanism for measuring the *worst-case* performance of a protocol, in terms of how badly the given protocol may be out-performed by an *off-line* protocol that has access to perfect information. Recall that a given protocol has *competitive ratio* $1/\lambda$ (or is λ-*competitive*) if an ideal off-line protocol has advantage over the given protocol by at most a factor of λ.

One place competitive analysis has been used to evaluate performance is the setting of distributed algorithms in asynchronous shared memory computation, including the work of Ajtai et al. [6]. This line of work has a different flavor than the problem considered in the present paper due to the nature of the algorithm being analyzed (computation algorithm verses network routing protocol). In particular, network topology is not a consideration in this line of work (and malicious deviation of processors is not considered).

Competitive analysis is a useful tool for evaluating protocols in unreliable networks (e.g. asynchronous networks and/or networks with no connectivity guarantees), as it provides best-possible standards (since absolute performance guarantees may be impossible due to the lack of network assumptions). For a thorough description of competitive analysis, see [14].

Max-Flow and Multi-Commodity Flow: The Max-flow and multi-commodity flow models assume networks that are synchronous with connectivity/liveness guarantees and have incorruptible nodes (max-flow networks also typically have fixed topology and are global-control). There has been a tremendous amount of work in these areas, see e.g. Leighton et al. [19] for a discussion of the two models and a list of results, as well as Awerbuch and Leighton [11] who show optimal throughput-competitive ratio for the network model in question.

Admission Control and Route Selection: The admission control/route selection model differs from the multi-commodity flow model in that the goal of a protocol is not to meet the demand of all ordered pairs of nodes (s,t), but rather the protocol must decide which requests it can/should honor, and then designate a path for honored requests. There are numerous models that are concerned with questions of admission control and route selection: The Asynchronous[1] Transfer Model (see e.g. Awerbuch et al. [9]), Queuing Theory (see e.g. Borodin and Kleinberg [15] and Andrews et al. [8]), Adversarial Queuing Theory (see e.g. Broder et al. [16] and Aiello et al. [5]). For an extensive discussion about these research areas, see [20] and references therein.

[1]We emphasize that the definition of *asynchronicity* in ATM is different than the one considered in this paper. In particular, "asynchronicity" in ATM literature is meant to emphasize the fact that the requests are *not* known ahead of time, and thus protocols face the added challenge of handling new requests adaptively.

The admission control/route selection model assumes synchronous communication and incorruptible nodes and makes connectivity/liveness guarantees. Among the other options (fixed or dynamic topology, global or local control), each combination has been considered by various authors, see the above reference for further details and results within each specific model.

1.2 Our Results

In this paper, we consider the feasibility of end-to-end routing in unreliable networks. We begin by exploring optimal throughput performance in networks whose nodes are trustworthy, but otherwise the network represents a "worst-case" network model. In particular, we use competitive analysis to prove matching upper and lower bounds on throughput performance for end-to-end communication in networks that are asynchronous, local-control, and have dynamic topology with no connectivity guarantees.

Theorem 1 (Informal) *The best competitive-ratio that any protocol can achieve in a distributed asynchronous network with dynamic topology (and no connectivity assumptions) is* $1/n$ *(where* n *is the number of nodes). In particular, given any protocol* $\mathcal{P}$*, there exists an alternative protocol* $\mathcal{P}'$*, such that* $\mathcal{P}'$ *will out-perform* $\mathcal{P}$ *by a factor of at least* n*.*

Theorem 2 (Informal) *There exists a protocol that achieves a competitive ratio of* $1/n$ *in a distributed asynchronous network with dynamic topology (and no connectivity assumptions).*

Next, we move to networks where the nodes are susceptible to corruption and may deviate from the specified protocol in any desired manner to disrupt communication as much as possible. Somewhat surprisingly, we show that this increased level of unreliability does not affect optimal throughput performance; indeed, we demonstrate a protocol that achieves $1/n$ competitive ratio, which matches the lower-bound of Theorem 1.

Theorem 3 (Informal) *Assuming one-way functions exist and Public-Key Infrastructure, there exists a protocol with competitive ratio* $1/n$ *in a distributed asynchronous network with dynamic topology (and no connectivity assumptions), even if an arbitrary subset of malicious nodes deliberately disobey the protocol specifications in order to disrupt communication as much as possible.*

In Section 2 we define formally the network model(s) and our mechanism for analyzing throughput performance, then in Sections 3-5 we go through the ideas for Theorems 1-3 (respectively). Rigorous proofs of all theorems can be found in the Appendix.

2 The Model

In this section, we describe formally the model in which we will be analyzing routing protocols. We begin by modeling the network as a graph G with n vertices (or *nodes*). Two of these nodes are designated as the *sender* S and *receiver* R, and the sender has a stream of messages $\{m_1, m_2, \dots\}$ that it wishes to transmit through the network to the receiver.

Asynchronous communication networks vary from synchronous networks in that the transmission time across an edge in the network is not fixed (even along the same edge, from one message transmission to the next). Since there is no common global clock or mechanism to synchronize events, an asynchronous network is often said to be "message driven," in that the actions of the nodes in the network occurs exactly (and only) when they have just sent/received a message.

Asynchronous networks are commonly modelled by introducing a scheduling adversary that controls the edges of the network as follows. Informally, we focus on a single edge $E(u, v)$, and then a "round" consists of allowing the edge to deliver a message in both directions.[2] To model unpredictable delivery times across each edge, we have each node u decide on the next message to send to v immediately after receiving a message from v, and this message is then sent to the adversary who stores the message until the next time the adversary activates edge $E(u, v)$.

Formally, we define a round to consist of a single edge $E(u, v)$ in the network chosen by the adversary in which two sequential events occur: 1a) Among the packets from u to v that the adversary is storing, it will choose one (in any manner it likes) and deliver it to v; 1b) Similarly, the adversary chooses one of the packets it is storing from v to u and delivers it to u; 2a) After seeing the delivered packet, u sends requests of the form $(u, v, m) =$ (sending node, target node, message) to the adversary, which will be stored by the adversary and may be delivered the *next* time $E(u, v)$ is made a round; 2b) Similarly for v. If e.g. u does not have a packet he wishes to send v in step (2a), then u can choose to send nothing here. Similarly, the adversary does not send anything to v in step (1a) if he is not storing a message from u to v during round $E(u, v)$.

Modelling asynchronicity in this manner captures the intuition that a node has no idea how long a message "sent" to an adjacent node will take to arrive, and this definition also captures the "worst-case" asynchronicity, in that a (potentially deliberately malicious) adversary controls the scheduling of rounds/edges.

For ease of discussion, we assume that all edges in the network have a fixed bandwidth/capacity, and that this quantity is the same for all edges in the network. We emphasize that this assumption does not restrict the validity of our claims in a more general model allowing varying bandwidths, but is only made for ease of exposition.

Aside from obeying the above specified rules, we place no restriction on the scheduling adversary. In other words, it may honor whatever edges it likes (this models the fact our network makes no connectivity assumptions), wait indefinitely long between honoring the same edge twice (modeling both the dynamic and asynchronous features of our network), and do anything else it likes (so long as it respects steps 1) and 2) above each time it honors an edge) in attempt to hinder the performance of a routing protocol.

In Section 5, our model will also allow a polynomially bounded node-controlling adversary to corrupt the *nodes* in the network. The node-controlling adversary is malicious, meaning that he can take complete control over the nodes he corrupts, and can therefore force them to deviate from any protocol in whatever manner he likes. We further assume that the adversary is dynamic, which means that he can corrupt nodes at any stage of the protocol, deciding which nodes to corrupt based on what he has observed thus far. We do not impose any "access-structure" limitations on the adversary. That is, the adversary may corrupt any nodes it likes (although if the sender and/or receiver is corrupt, secure routing between them is impossible). Because integrity of the messages received by the receiver is now a concern (as corrupt nodes can delete and/or modify messages), we will say a routing protocol is secure if the receiver eventually gets all of the messages sent by the sender, in order and without duplication or modification.

[2]The demand that the adversary deliver messages in *both* directions when honoring an edge $E(u, v)$ does not restrict the power of the adversary. To generalize to the case where the adversary can deliver messages in only one direction, one could simply define an edge to be "down" until at least one message has been able to travel in each direction. Since competitive analysis can be used to show that acknowledgements of some kind are requisite to achieve finite competitive-ratio, it is natural to define a round in such a way so as to allow communication in both directions.

The separation of the adversaries into two distinct entities is solely for conceptual reasons. Note that the scheduling adversary cannot be controlled or eliminated: edges themselves are not inherently "good" or "bad," so identifying an unresponsive edge does not allow us to forever refuse the protocol to utilize this edge. By contrast, our protocol will limit the amount of influence the node-controlling adversary has in the network. Specifically, we will show that if a node deviates from the protocol in a sufficiently destructive manner (in a well-defined sense), then our protocol will be able to identify it as corrupted in a timely fashion. Once a corrupt node has been identified, it will be eliminated from the network by excluding it from all future communication.

Note that our network model is on-line and distributed, in that we do not assume that the nodes have access to any information (including future knowledge of the adversary's schedule) aside from the packets they receive during a round they are a part of. Also, we insist that nodes have bounded memory which is at least $\Omega(n^2)$.[3]

The goal of this paper is to analyze the performance of routing protocols in a network model that is: on-line, distributed, asynchronous, dynamic with no connectivity assumptions, and susceptible to misbehaving nodes. Our mechanism for evaluating protocols will be to measure their *throughput*, a notion we can now define formally in the context of rounds and the scheduling adversary. In particular, let $f_{\mathcal{P}}^{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$ be a function that measures, for a given protocol $\mathcal{P}$ and adversary $\mathcal{A}$, the number of packets that the receiver has received as a function of the number of rounds that have passed. Note that in this paper, we will consider only deterministic protocols, so $f_{\mathcal{P}}^{\mathcal{A}}$ is well-defined. The function $f_{\mathcal{P}}^{\mathcal{A}}$ formalizes our notion of throughput.

As mentioned in the Introduction, we utilize competitive analysis to gauge the performance (with respect to throughput) of a given protocol against all possible competing protocols. In particular, for any fixed adversary $\mathcal{A}$, we may consider the ideal "off-line" protocol $\mathcal{P}'$ which has perfect information: knowledge of all future decisions of the scheduling adversary, as well as knowledge of which nodes are/will become corrupt. That is, for any fixed round x, there exists an ideal off-line protocol $\mathcal{P}'(\mathcal{A}, x)$ such that $f_{\mathcal{P}'}^{\mathcal{A}}(x)$ is maximal. We demand that the ideal protocol $\mathcal{P}'$ never utilizes corrupt nodes, once they have been corrupted (this restriction is not only reasonable, it is necessary, as it can easily be shown that allowing $\mathcal{P}'$ to utilize corrupt nodes will result in *every* on-line protocol having competitive ratio $\frac{1}{\infty}$).

Definition 2.1. We say that a protocol $\mathcal{P}$ has competitive ratio $1/\lambda$ (respectively is λ-competitive) if there exists a constant k and function *g(n,C)* (C is the memory bound per node) such that for all possible adversaries $\mathcal{A}$ and for all $x \in \mathbb{N}$:[4]

$$f_{\mathcal{P}'}^{\mathcal{A}}(x) \ \leq \ (k \cdot \ \lambda) \ \cdot f_{\mathcal{P}}^{\mathcal{A}}(x) \ + \ g(n, C) \tag{1}$$

We assume that there is a Public-Key Infrastructure (PKI) that allows digital signatures. In particular, before the protocol begins we choose a security parameter l sufficiently large and run a key generation algorithm for a digital signature scheme, producing $n = |G|$ (secret key, verification key) pairs (sk_u, vk_u). As output to the key generation, each processor $u \in G$ is given its own private signing key sk_u and a list of all n signature verification keys vk_v for all nodes $v \in G$. In particular, this allows the sender and receiver to sign messages to each other that cannot be forged (except with negligible probability in the security parameter) by any other node in the system.

[3]For simplicity, we assume that all nodes have the same memory bound, although our argument can be readily extended to handle the more general case.

[4]Typically, λ is a function of the number of nodes in the network n, and Definition 2.1 implicity assumes the minimal value of λ for which (1) holds.

3 Optimal Competitive Ratio in Unrestricted Networks

Due to space constraints and the complexity of the argument, we will only be able to sketch the proof of Theorem 1 in this section. At a high level, the idea is to describe an adversary that schedules edges based on the given protocol's actions such that the packets of the protocol get "spread out" among the nodes of the network. Meanwhile, with knowledge of the adversary's schedule, an offline protocol can choose to only move packets along edges leading to the receiver. A short description is below; the full proof can be found in Appendix A.

The network model assumes that nodes have bounded memory, so let C denote the maximal number of packets that any node can store at any time. We will show that for any deterministic protocol $\mathcal{P}$, there exists an adversary $\mathcal{A}$, a protocol $\mathcal{P}'$, and a sequence of strictly positive integers $\{m_1, m_2, \dots\}$ such that for any $\alpha > 0$, by round $x = \sum_{i=1}^{\alpha} m_i C$:

$$f^{\mathcal{A}}_{\mathcal{P}'}(x) = \alpha C \quad \text{and} \quad f^{\mathcal{A}}_{\mathcal{P}}(x) \le \frac{\alpha C}{(n-2)} \approx \frac{\alpha C}{n}, \tag{2}$$

from which we conclude that the competitive ratio of $\mathcal{P}$ is at best $1/n$.

We begin by describing the adversary, i.e. a schedule (or order) of edges that will be honored. The schedule will proceed in **cycles**, with the i^{th} cycle lasting $m_i C$ rounds. Let the **height** of a node refer to the number of packets currently stored by that node. For the first C rounds, the adversary finds the *internal* node A_1 with the largest height (ties are broken arbitrarily), and honors edge $E(S, A_1)$ for C rounds (here S denotes the Sender). The protocol then proceeds inductively, starting with $j = 2$ and $\widehat{A}_1 = A_1$:

1. The adversary finds node A_j, where A_j is the node in the network closest in height (but *smaller*) to $\widehat{A}_{j-1}$. If there is no such node, set A_j to the Receiver R.
2. The adversary honors edge $E(\widehat{A}_{j-1}, A_j)$ for C rounds
3. The adversary sets $\widehat{A}_j$ to be whichever node ($\widehat{A}_{j-1}$ or A_j) has *fewer* packets after the C rounds of edge $E(\widehat{A}_{j-1}, A_j)$ has just passed.

The above three steps are continued until the end of the C rounds for which $A_j = R$.

Notice a few features of the adversarial strategy: 1) The Sender's ability to insert packets is hindered by the fact the adversary is choosing to honor edge $E(S, N)$ for the node N with the smallest capacity to store *more* packets; 2) By selecting in Step 2 the node storing fewer packets, the adversary is attempting to minimize the number of packets that make progress towards the Receiver; indeed 3) Among all nodes in the network, the node N that is currently storing the fewest packets will be the one connected to the Receiver in the final C rounds of the cycle. Also, it is clear that an off-line protocol $\mathcal{P}'$ with knowledge of all future rounds will be able to deliver C packets every cycle. Since a cycle consists of $C * m$ rounds for some positive integer m, we can generate a sequence of positive integers $\{m_i\}$ coming from the i^{th} cycle, yielding the first equality of (2), so it remains to prove the second bound in (2).

Fix any on-line protocol $\mathcal{P}$ we wish to analyze. If we could demonstrate that $\mathcal{P}$ delivers at most $C/(n-2)$ packets per cycle, then (2) would be immediate. Unfortunately, one can imagine e.g. the state of the network at the beginning of some cycle being such that *all* internal nodes are storing the maximum C allowed packets. In this case, $\mathcal{P}$ will be able to deliver C packets this cycle. Therefore, we instead need to argue that if $\mathcal{P}$ ever reaches a state where it is able to deliver more than $C/(n-2)$ packets in some cycle (e.g. all nodes are full), then it must be that $\mathcal{P}$ has delivered *fewer* than an average of $C/(n-2)$ packets per cycle in the past.

With this counter-example in mind, we define a potential function Ψ^α, which intuitively measures the ability of $\mathcal{P}$ to deliver packets in the α^{th} cycle. We will show that whenever $\mathcal{P}$ delivers more than $C/(n-2)$ packets, the difference $\Psi^\alpha - \Psi^{\alpha+1}$ will be positive and "sufficiently large." Conversely, any time $\Psi^{\alpha+1} > \Psi^\alpha$, we will show that necessarily $\mathcal{P}$ delivered "significantly fewer" than $C/(n-2)$ packets in the α^{th} cycle.

Formally, at the start of any cycle α, label the internal nodes as $\{N_1, \dots, N_{n-2}\}$ in descending order in terms of how full their buffers are at the start of α. Let H_i^α denote the number of packets that node N_i^α is storing at the outset of α, and then define:

$$\Psi^\alpha = \sum_{i=1}^{n-2} \left(\frac{1}{2}\right)^{n\text{-}i\text{-}2} \cdot \max\left(0, H_i^\alpha - (n-i-2)\frac{C}{n-2}\right) \tag{3}$$

Let Z^α denote the number of packets the Receiver receives in the α^{th} cycle. Our main technical result for this section is then:

Lemma 3.1. *For all $\alpha \in \mathbb{N}$:*

$$Z^\alpha + (\Psi^{\alpha+1} - \Psi^\alpha) \leq \frac{7C}{n-2} \tag{4}$$

Proof. See the proof of Lemma A.12 in the Appendix.

With Lemma 3.1 in hand, we obtain the second inequality of (2) as an immediate corollary:

Lemma 3.2. *For any $\alpha \in \mathbb{N}$ and $x = (n-2)\alpha C$:*

$$f_{\mathcal{P}}^{\mathcal{A}}(x) \leq \frac{7\alpha C}{n-2} \tag{5}$$

Proof. Consider the string of inequalities:

$$f_{\mathcal{P}}^{\mathcal{A}}(x) = \sum_{\beta \leq \alpha} Z^\beta \leq \sum_{\beta \leq \alpha} \left(\frac{7C}{(n-2)} - (\Psi^{\beta+1} - \Psi^\beta)\right) = \frac{7\alpha C}{n-2} + \Psi^1 - \Psi^{\alpha+1} \leq \frac{7\alpha C}{n-2}, \tag{6}$$

where the last inequality follows from the fact that $\Psi^{\alpha+1} \geq 0$ and $\Psi^1 = 0$ (the latter is true since at the outset of the protocol, all nodes are not storing any packets).

4 Optimal On-line Local Control Protocol

In this section we present an on-line protocol that enjoys competitive ratio $1/n$. The protocol is a basic implementation of the "Slide" protocol (or *gravitational-flow*), which was first introduced by Afek, Gafni, and Rosén [3], and further developed in a series of work [1] and [18]. We chose to analyze the performance of this protocol in our "unrestricted" network model because its inherent message-driven protocol is well-suited for the asynchronous network, and it has also been shown to out-perform more naive candidates for asynchronous routing protocols (e.g. broadcast) when stronger network assumptions are made [7].

Because the Slide protocol has nodes make routing decisions based on their current height (how many packets they are currently storing), it will be easier to work in a simplified model for asynchronicity over the one presented in Section 2. In particular, for the remainder of this section, we assume a semi-asynchronous model, defined as follows:

$1'$ The adversary does not maintain a buffer of requests of packets from nodes and must instead satisfy them immediately as specified in $3'$ below

$2'$ The adversary proceeds in the same manner as before, by selecting an edge $E(u, v)$ to honor according to the same guidelines as in Section 2

$3'$ During a round $E(u, v)$, the adversary first "awakens" u and v to alert them they are a part of the current round. Nodes u and v may now submit their request, consisting only of a packet plus control information, to the adversary who must directly deliver the packet p to v during this round (similarly the packet p' that v submitted is delivered to u).

Comparing this to the fully asynchronous model defined in Section 2, the difference is that here the packets that u and v deliver to each other, with their height information included, are *current*; in the model of Section 2, the packets and height information delivered in some round $E(u, v)$ were actually set the *previous* time $E(u, v)$ was honored. This slightly complicates things for routing protocols in the fully asynchronous model, as the nodes are forced to make routing decisions based on outdated information.

It turns out that proving our protocol enjoys a certain competitive-ratio in the semi-asynchronous setting is the hard part, and it is not difficult to extend the proof to work in the fully asynchronous setting. Indeed, all of the major ideas come from considering only the semi-asynchronous setting. In the next subsection we describe our protocol in the semi-asynchronous setting, and then sketch a proof that it enjoys competitive-ratio $1/n$. The formal details of the proof are presented in Appendix B, and a description of the protocol extended to the fully asynchronous setting, together with formal proofs that it has the same competitive ratio, are provided in Appendix C.

4.1 Description of the Protocol

There are numerous instantiations of the Slide protocol that vary slightly between one another, but the basic principle is always the same. Due to space constraints, we will not provide a detailed description of the protocol, but refer the reader to [3] for the original protocol, and [1], [18], and [7] for various modifications. Below, we present a basic implementation of the Slide protocol, and then go on to prove that the basic Slide protocol achieves competitive ratio $1/n$ in the restricted *semi-asynchronous* model of $1' - 3'$ described above. Somewhat surprisingly, even though the Slide protocol has been in existence for over a decade, no throughput competitive analysis for the asynchronous (or even semi-asynchronous) model has ever been performed.

The network model assumes that nodes have bounded memory, so let C denote the maximal number of packets that any node can store at any time. Also, we will assume $C/n \in \mathbb{N}$ and in particular that $C/n \geq 2$ (the former assumption is not necessary but will make the exposition easier; the latter is necessary for the Slide protocol to work). Within the context of the semi-asynchronous network model ($1' - 3'$ above), we describe the request that a node u will make to the adversary when it is "awakened," and also how this node u will respond to the packet it receives from v:

1. If u is the Sender, then u finds the next packet $p_i \in \{p_1, p_2, \dots\}$ that has not yet been deleted (see 1a below), and forms the packet to send to the adversary: $p := (p_i, C + \frac{C}{n} - 1)$. Meanwhile, when u receives (in the same round) the packet (p_j, h):
   (a) If $h < C$, then u deletes packet p_i from his input stream $\{p_1, p_2, \dots\}$ (and ignores the received packet p_j)
   (b) If $h \geq C$, then u keeps p_i (and ignores the received packet p_j)

2. If u is the Receiver, then u forms the packet to send $p := (\perp, \frac{-C}{n})$. Meanwhile, when u receives a packet of form (p_j, h), if $p_j \neq \perp$, u stores/outputs p_j as a packet successfully received.
3. If u is an internal node (not Sender or Receiver) and u currently has height H, then u finds the last[5] packet p_i that it has received, and sets the packet to send to the adversary: $p := (p_i, H)$ (if $H = 0$, then set $p_i = \perp$). Meanwhile, when u receives (in the same round) a packet of form (p_j, h):
   (a) If $H \geq h + C/n$, then u will delete p_i (and ignore the packet p_j)
   (b) If $H \leq h - C/n$, then u will keep p_i, and also store p_j (as the most recent packet received)
   (c) If $|H - h| < C/n$, then u will keep p_i and ignore packet p_j

Notice that rules 1-3 essentially state that internal nodes will always accept packets from the Sender (if they have room), always send packets to the Receiver (if they have any to send), and will transfer a packet to a neighboring internal node if and only if they are currently storing at least C/n *more* packets than that neighbor.

4.2 Competitive Analysis of Slide in the Semi-Asynchronous Model

Due to space constraints, we provide here only a very brief sketch of the proof that the above described Slide protocol enjoys competitive ratio $1/n$. The full proof can be found in Appendix B.

Recall that we wish to show that there exists a constant k and function $g(n, C)$ such that for any round x and against any adversary $\mathcal{A}$ (see (1)):

$$f^{\mathcal{A}}_{\mathcal{P}'}(x) \ \leq \ (kn) \ \cdot f^{\mathcal{A}}_{\mathcal{P}}(x) \ + \ g(n, C) \tag{7}$$

Above (and through the remainder of this section), $\mathcal{P}$ will denote the Slide protocol, and for fixed choice of adversary $\mathcal{A}$ and round x, $\mathcal{P}'(\mathcal{A}, x)$ will denote the ideal off-line protocol (since we will be fixing x and $\mathcal{A}$, we will usually write simply $\mathcal{P}'$). We will show that (7) will be true for all rounds x and all adversaries $\mathcal{A}$ for $k = 4$ and $g(n, C) = 4n^2C$. We proceed by fixing an arbitrary adversary $\mathcal{A}$ and round $x \in \mathbb{N}$, and showing that for these (arbitrary) choices, (7) will be satisfied. Let $Y^{\mathcal{P}'}$ (resp. $Z^{\mathcal{P}'}$) denote the packets that have been inserted (resp. received) by the Sender (resp. the Receiver) for protocol $\mathcal{P}'$ as of round x (define $Y^{\mathcal{P}}$ and $Z^{\mathcal{P}}$ analogously). Notice that $f^{\mathcal{A}}_{\mathcal{P}'}(x)$, the left-hand-side of (7), is equal to $|Z^{\mathcal{P}'}|$ (we will occasionally write $Z^{\mathcal{P}'}$ when we really mean $|Z^{\mathcal{P}'}|$; the meaning will be clear from context). We split $Z^{\mathcal{P}'}$ into two disjoint subsets $Z^{\mathcal{P}'} = Z^{\mathcal{P}'}_1 \cup Z^{\mathcal{P}'}_2$, which we now describe.

We can view the adversary $\mathcal{A}$ as simply a schedule (or order) of edges that the adversary will honor. We will imagine a virtual world, in which the two protocols (Slide and the ideal off-line protocol) are run simultaneously in the same network. Define $Z^{\mathcal{P}'}_1$ to be the subset of $Z^{\mathcal{P}'}$ consisting of packets p' for which there exists at least one round $E(u, v)$ such that both p' *and* some packet $p \in Y^{\mathcal{P}}$ were both transferred this round.[6] Set $Z^{\mathcal{P}'}_2 = Z^{\mathcal{P}'} \setminus Z^{\mathcal{P}'}_1$.

Lemma 4.1. $|Z^{\mathcal{P}'}_1| \ \leq \ n|Z^{\mathcal{P}}| + n^2C$

Proof Sketch. Since every packet in $Z^{\mathcal{P}'}_1$ travelled at the same time as a packet transfer in $\mathcal{P}$, we can bound $|Z^{\mathcal{P}'}_1|$ by the number of packet transfers in $\mathcal{P}$. Since any fixed packet drops in height at least C/n each time it is transferred, the total number of packet transfers is at most $n|Y^{\mathcal{P}}|$. Finally, since the maximal number of packets that can be stored in all internal buffers is nC, we have $|Y^{\mathcal{P}}| \leq |Z^{\mathcal{P}}| + nC$. ∎

[5]The Slide protocol typically utilizes FILO storage buffers, and then uses error-correcting codes to compensate the packets that get "stuck" in a node's storage.

[6]Note that we make no condition that the two packets traveled in the same direction.

Lemma 4.2. $|Z_2^{\mathcal{P}'}| \leq 2n|Y^{\mathcal{P}}| \leq 2n|Z^{\mathcal{P}}| \ + \ 2n^2C$

Proof Sketch. Consider a fixed packet $p' \in Z_2^{\mathcal{P}'}$. When this packet was first inserted by $\mathcal{P}'$, say into some node u's buffer, since $\mathcal{P}$ did *not* insert a packet in this round (by definition of $Z_2^{\mathcal{P}'}$), we have that u's buffer must have been full (rule 1(a)). Meanwhile, when the receiver received p' from some node v, since $\mathcal{P}$ did *not* transfer a packet this round, it must have been that v had an empty buffer during this round. Thus, p' travelled from a node with a completely full buffer to one with a completely empty buffer. In Appendix B we show how to use this fact to bound $|Z_2^{\mathcal{P}'}|$ by the number of packet transfers in $\mathcal{P}$, which can then be bounded by $2n|Y^{\mathcal{P}}|$ as in Lemma 4.1. ■

5 Protocol Secure Against Malicious Adversary

We now move to the network setting that allows both unreliable edges controlled by the scheduling adversary and unreliable nodes corrupted by the node-controlling adversary (see Section 2 for a formal discussion of the network model and these two adversaries). Below is a high-level description of the protocol and a statement of the main result. Pseudo-code of the protocol, as well as rigorous proofs of security and throughput performance, can be found in Appendix D.

5.1 High Level Description

Our strategy in developing a protocol that routes effectively in this highly unreliable network setting will be to start with the Slide+ protocol, which has optimal competitive ratio in terms of throughput, and add elements from cryptography to provide extra security against the node-controlling adversary. Specifically, we will modify the Slide+ protocol by using digital signatures in the following two ways:

1. The sender signs every packet, so that honest nodes do not waste resources on modified or junk packets, and so that packets the receiver gets are unmolested
2. Communication between nodes will be signed by each node. This information will then be used later by the sender (if there has been malicious activity) to hold nodes accountable for their actions, and ultimately eliminate corrupt nodes

The routing rules for each internal node are the same as in the Slide+ protocol, except that whenever a node u sends a packet to a neighbor v, there will be four parts to this communication:

(a) The packet itself, i.e. one of the packets from the sender intended for the receiver
(b) The current *height* of u, i.e. how many packets u is currently storing
(c) A signature on the communication that u has had so far with v, to be described shortly
(d) Signatures from other nodes that the sender has requested, to be described shortly

The first two parts of each communication are identical to the Slide+ protocol, so it remains to discuss the second two items, which are used for the identification of corrupt nodes. Note that the second two items each consist of a signature on some quantity; for this reason we will require that the bandwidth of each edge is large enough to allow for simultaneous transmission of two signatures (plus the packet itself).[7] The signature that u includes on his communications with v for Item (c) above pertains to the following four items:

[7]This assumption on bandwidth is not unreasonable: for a signature scheme with security parameter k, each signature requires only $O(k)$ bits. Also, the requirement that bandwidth is large enough to allow two signatures is made for convenience of exposition; our protocol can be modified to handle the case of smaller bandwidth, although this is not pursued here.

Sig. 1. The total number of packets u has sent to v so far

Sig. 2. The total number of times the previous packet p that was exchanged between them has crossed the edge $E(u, v)$ (in general, the same packet may cross the same edge multiple times)

Sig. 3. The cumulative difference in u and v's heights, measured from each time u and v exchanged a packet

Sig. 4. An index representing how many times $E(u, v)$ has been honored, to serve as a time-stamp on the above three items

It remains to explain Item (d) from above, for which it will be useful to first describe from a high-level how our protocol handles malicious activity by corrupt nodes. We first note that if either the sender or receiver is corrupted by the node-controlling adversary, then secure routing is impossible (indeed it is not clear what is even meant by "secure routing" in this case). We will therefore assume that the sender and receiver are incorruptible, and they will be responsible for regulation of the network (e.g. identifying and eliminating corrupt nodes). Also, because our definition of security (see Section 2) requires that the receiver gets all of the packets sent by the sender, it is no longer enough to simply measure throughput in terms of number of packets received (as was done for the Slide and Slide+ protocols above). Instead, we will use error-correction and first expand the messages into codewords so that the receiver can reconstruct each message if he has a constant fraction of the codeword packets. See e.g. [7] for a specific description of how this can be done. We note that because the definition of throughput only cares about asymptotic performance (i.e. constants are absorbed in the k that appears in Definition 1), the use of error-correction will not affect the throughput of our protocol.

From a high-level, the protocol attempts to transfer one message (codeword), consisting of $O(nC)$ bits, at a time. The sender will continue inserting packets corresponding to the same codeword until one of the following occurs:

- S1 The sender gets a message from the receiver indicating he could decode the current codeword
- F2 The sender gets a message from the receiver indicating inconsistencies in height differences
- F3 The sender has inserted all packets corresponding to the current codeword
- F4 The sender gets a message from the receiver indicating the receiver got the same packet twice
- F5 The sender is able to identify a corrupt node

In the case of S1, the message/codeword was delivered successfully, and the sender will begin inserting packets corresponding to the next message/codeword. In the case of F5, the sender will eliminate the identified node (i.e. alert all nodes in the network to never trust or utilize the corrupt node again), and begin anew transmitting packets corresponding to the current codeword. The other three cases all correspond to failed attempts to transfer the current message/codeword due to corrupt nodes disobeying protocol rules, and in each case the sender will use the signed information from Item (c) above to identify a corrupt node.

In cases F2-F4, the sender will begin anew transmitting packets corresponding to the current codeword. Before nodes are allowed to participate in transferring the codeword packets, they must first learn that the last transmission failed, the reason for failure (F2-F4), and the sender must receive all of the signatures the node was storing from its neighbors (i.e. all signed information from Item (c) above). Note that the network itself is the only medium of communication available for relaying the signatures a node is storing to the sender, and hence part of the bandwidth of each edge (and part of the storage capacity of each node) is devoted to returning these pieces of signed information to the sender (this is Item (d) from the above list). The specific rules regarding storing and transferring other nodes' signatures back to the sender can be found in the pseudo-code in Appendix D.

Until the sender has received all of a node's information corresponding to a failed transmission, that node will remain on the **blacklist**. That is, no honest node u will transfer any codeword packets to another node v until u obtains verification from the sender that the sender has received all signatures from v. In Appendix D, we prove rigorously our main theorem:

Theorem 3. *If at any time $\mathcal{P}'$ has received $\Theta(xn)$ messages, then $\mathcal{P}$ has received $\Omega((x - n^2))$ messages. Thus, if the number of messages $x \in \Omega(n^2)$, then our protocol has competitive ratio $1/n$.*

References

[1] Y. Afek, B. Awerbuch, E. Gafni, Y. Mansour, A. Rosen, N. Shavit. "*Slide*– The Key to Poly. End-to-End Communication." *J. of Algorithms 22, pp. 158-186.* 1997.

[2] Y. Afek, E. Gafni "End-to-End Communication in Unreliable Networks." *PODC, pp.* 1988.

[3] Y. Afek, E. Gafni, A. Rosén. "The Slide Mechanism with Applications in Dynamic Networks." *Proc. 11th ACM Symp. on Principles of Dist. Comp., pp. 35-46.* 1992.

[4] W. Aiello, E. Kushilevitz, R. Ostrovsky, and A. Rosén. "Adaptive Packet Routing For Bursty Adversarial Traffic." *J. Comput. Syst. Sci. 60(3): 482-509.* 2000.

[5] W. Aiello, R. Ostrovsky, E. Kushilevitz, and A. Rosén. "Dynamic Routing on Networks with Fixed-Size Buffers." *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms, pp. 771-780.* 2003.

[6] M. Ajtai, J. Aspnes, C. Dwork, and O. Waarts. "A Theory of Competitive Analysis for Distributed Algorithms." *Proc. 35th IEEE Symp. on Foundations of Computer Science, pp. 32-40.* 1994.

[7] Y. Amir, P. Bunn, and R. Ostrovsky. "Authenticated Adversarial Routing." *6th Theory of Crypt. Conf., pp. 163-182.* 2009.

[8] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. "Universal Stability Results for Greedy Contention-Resolution Protocols." *Proc. 37th IEEE Symp. on Foundations of Computer Science, pp. 380-389.* 1996.

[9] B. Awerbuch, Y. Azar, and S. Plotkin. "Throughput-Competitive On-Line Routing." *Proc. 34th IEEE Symp. on Foundations of Computer Science, pp. 401-411.* 1993.

[10] B. Awerbuch, D. Holmer, C. Nina-Rotaru, and H. Rubens. "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures." *Proc. of 2002 Workshop on Wireless Security, pp. 21-30.* 2002.

[11] B. Awerbuch and T. Leighton. "Improved Approximation Algorithms for the Multi-Commodity Flow Problem and Local Competitive Routing in Dynamic Networks." *Proc. 26th ACM Symp. on Theory of Computing, pp. 487-496.* 1994.

[12] B. Awerbuch, Y Mansour, N Shavit "End-to-End Communication With Polynomial Overhead." *Proc. of the 30th IEEE Symp. on Foundations of Computer Science, FOCS.* 1989.

[13] B. Barak, S. Goldberg, and D. Xiao. "Protocols and Lower Bounds for Failure Localization in the Internet." *Proc. of Advances in Crypt., 27^{th} EUROCRYPT, Springer LNCS 4965, pp. 341-360.* 2008.

[14] A. Borodin and R. El-Yaniv. "Online Computation and Competitive Analysis." *Camb. Univ Press.* 1998.

[15] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. "Adversarial Queuing Theory." *Proc. 28th ACM Symp. on Theory of Computing, pp. 376-385.* 1996.

[16] A. Broder, A. Frieze, and E. Upfal. "A General Approach to Dynamic Packet Routing with Bounded Buffers." *Proc. 37th IEEE Symp. on Foundations of Computer Science, pp. 390-399.* 1996.

[17] S. Goldberg, D. Xiao, E. Tromer, B. Barak, and J. Rexford. "Path-Quality Monitoring in the Presence of Adversaries." *ACM SIGMETRICS Vol. 36, pp. 193-204.* June 2008.

[18] E. Kushilevitz, R. Ostrovsky, and A. Rosén. "Log-Space Polynomial End-to-End Communication." *SIAM Journal of Computing 27(6): 1531-1549.* 1998.

[19] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas. "Fast Approximation Algorithms for Multicommodity Flow Problem." *Proc. 23rd ACM STOC, pp. 101-111.* 1991.

[20] S. Plotkin. "Competitive Routing of Virtual Circuits in ATM Networks." *IEEE J. on Selected Areas in Communications, Vol. 13, No. 6, pp. 1128-1136.* 1995.

[21] D. Sleator and R. Tarjan. "Amortized Efficiency of List Update and Paging Rules." *Commun. ACM, Vol. 28, No. 2, pp. 202-208.* 1985.

Appendix

A Formal Proof of Throughput Bound

In this section, we go through the rigorous details of the proof of Theorem 1, which was sketched in Section 3. We will use the same notation introduced there for the remainder of this section. In particular, recall that there is some fixed protocol $\mathcal{P}$ that we wish to analyze, and we are considering a scheduling adversary $\mathcal{A}$ that proceeds in cycles.

We begin with a reduction of the given protocol $\mathcal{P}$ to a *virtual* protocol $\mathcal{P}'$, which will be operating with respect to a *different* scheduling adversary $\mathcal{A}'$ than $\mathcal{P}$. The schedule of edges honored by $\mathcal{A}'$ will be (in general) different than those honored by $\mathcal{A}$, but $\mathcal{A}'$ will also proceed in *cycles*. For any cycle α in $\mathcal{P}'$'s world, define Ψ'^{α} and Z'^{α} analogous to Ψ^{α} and Z^{α} that were defined for $\mathcal{P}$ in Section 3. We emphasize that the two worlds of $\mathcal{P}$ and $\mathcal{P}'$ are different, and we are *not* attempting to apply competitive analysis to these two protocols. Rather, the property that $\mathcal{P}'$ will satisfy is:

$$\forall \alpha \in \mathbb{N}: \quad \Psi^{\alpha} = \Psi'^{\alpha} \quad \text{and} \quad Z^{\alpha} = Z'^{\alpha} \tag{8}$$

Then given that (8) holds for all cycles α, if we can show for all α (subject to $\mathcal{A}'$'s schedule):

$$Z'^{\alpha} + (\Psi'^{\alpha+1} - \Psi'^{\alpha}) \leq \frac{7C}{n-2}, \tag{9}$$

then the equivalent statement will be true for $\mathcal{P}$, which is Lemma 3.1 in Section 3, and thus the proof will be complete.

We now explain the alternate scheduling adversary $\mathcal{A}'$, which will be defined in terms of any arbitrary protocol attempting to route in a network controlled by $\mathcal{A}'$. As mentioned above, the schedule of $\mathcal{A}'$ will proceed in cycles, each of which will last $(n-1)C$ rounds. At the beginning of any cycle α, $\mathcal{A}'$ labels the internal nodes by $\{N_1^\alpha, N_2^\alpha, \ldots, N_{n-2}^\alpha\}$, so that for all $1 \le i \le n-3$, node N_i^α is storing *more* packets than N_{i+1}^α at the outset of cycle α (note that the labels/indices of the internal nodes will change every cycle). For the first C rounds of the cycle, the adversary will honor edge $E(S, N_1)$ (here S denotes the Sender). We describe the remaining rounds in this cycle inductively (starting below for $i = 1$, and $\widetilde{N}_1^\alpha = N_1^\alpha$):

1. The adversary honors edge $E(\widetilde{N}_i^\alpha, N_{i+1}^\alpha)$ for C rounds
2. After the first $(i+1)C$ rounds of cycle α have passed (i.e. edge $E(\widetilde{N}_i^\alpha, N_{i+1}^\alpha)$ has just been honored C times), let $\widetilde{N}_{i+1}^\alpha \in \{\widetilde{N}_i^\alpha, N_{i+1}^\alpha\}$ denote the node storing *fewer* packets than the other.

Steps 1-2 are repeated through $i = n-3$, so that $E(\widetilde{N}_{n-3}^\alpha, N_{n-2}^\alpha)$ has just completed, and $\widetilde{N}_{n-2}^\alpha$ has been defined. Then for the last C rounds of cycle α, the adversary honors edge $E(\widetilde{N}_{n-2}^\alpha, R)$.

Lemma A.1. *Given protocol $\mathcal{P}$ routing in a network controlled by $\mathcal{A}$ (whose schedule was described in Section 3), there exists a protocol $\mathcal{P}'$ competing against $\mathcal{A}'$, such that with respect to each protocol's own cycle,* (8) *is valid.*

Proof. Since we are considering only deterministic protocols, we can define what $\mathcal{P}'$ will do in any round based on what $\mathcal{P}$ is doing. We will actually demonstrate something slightly stronger than (8), that is:

> **Induction Hypothesis.** Up to permutation of the internal nodes, the heights of each of the internal nodes in both worlds is the same at the start/end of any cycle, as is the number of packets delivered in any cycle.

We proceed by induction on the cycle. In particular, fix some cycle α, and assume that the induction hypothesis is true for all cycles $\beta < \alpha$. In the first C rounds of α in $\mathcal{P}$'s world, $\mathcal{A}$ opens edge $E(S, A_1)$, where A_1 is the internal node currently storing the most packets. Similarly, in the first C rounds, $\mathcal{A}'$ opens edge $E(S, A_1')$, where A_1' is the internal node currently storing the most packets in $\mathcal{P}'$'s world. By the induction hypothesis, although the labels of node A_1 verses A_1' may be different, the node that label represents will have the same height in the two worlds, and we define $\mathcal{P}'$ to do the same thing that $\mathcal{P}$ does in these first C rounds.

Let A_2 denote the node for which the adversary $\mathcal{A}$ will honor edge $E(A_1, A_2)$ for the next C rounds, and similarly for A_2' with respect to $\mathcal{A}'$. Note that by the induction hypothesis together with the definition of $\mathcal{P}'$ (so far) for the first C rounds of cycle α, we have that the height of A_1 equals the height of A_1', and similarly the heights of A_2 and A_2' match. Now define $\mathcal{P}'$ to do in the C rounds $E(A_1', A_2')$ whatever $\mathcal{P}$ does in the C rounds $E(A_1, A_2)$.[8] Thus, after $2C$ rounds have passed, the two networks are still identical (up to permutation of the nodes).

Let $\widetilde{A}_2$ denote the node among $\{A_1, A_2\}$ that is storing fewer packets after the C rounds of $E(A_1, A_2)$. Now in $\mathcal{P}$'s world, the adversary will search for the node A_3 with height closest to (but *smaller* than) $\widetilde{A}_2$, and the adversary $\mathcal{A}$ will next honor edge $E(\widetilde{A}_2, A_3)$ for C rounds. Notice that, if e.g. $\mathcal{P}$ had A_2 transfer all its packets to A_1 during the C rounds of $E(A_1, A_2)$, it is possible that A_3

[8] In order to preserve Fact 1 below, we demand that after the C rounds of $E(A_1', A_2')$, A_2' is storing *fewer* packets than A_1'. Therefore, if this is *not* the case for $E(A_1, A_2)$, then define $\mathcal{P}'$ to end in a symmetric state as $\mathcal{P}$, i.e. so that the pair of nodes (A_1, A_2) have the same height as the pair of nodes (A_1', A_2'), but in the latter pair, necessarily A_1' is storing at least as many packets as A_2' after the C rounds of $E(A_1', A_2')$.

is *not* the node that had the third highest height at the start of cycle α (indeed, its even possible that $A_3 = R$).

By the induction hypothesis, there is some node A'_i ($i \geq 3$) in $\mathcal{P}'$'s world such that at the start of α, the height of A_3 equals the height of A'_i (if $A_3 = R$, then $i = n-1$, i.e. set $A'_i = R$). Notice that in contrast to $\mathcal{P}$'s world, the schedule of $\mathcal{A}'$ will necessarily go through *every* internal node at least once. Indeed, for any $2 \leq m \leq n-2$, the node in $\mathcal{P}'$'s world that started cycle α as the m^{th} fullest node will necessarily be a part of rounds mC through $(m+1)C - 1$. Therefore, for each $3 \leq m \leq i$, dictate that during rounds mC through $(m+1)C-1$, protocol $\mathcal{P}'$ will have the two nodes swap final states. In particular, for any $3 \leq m \leq i$, if H'_m denotes the height of A'_m at the start of cycle α, then we dictate that $\mathcal{P}'$ transfers enough packets from A'_m to A'_{m-1} during the C rounds of $E(A'_{m-1}, A'_m)$ such that the height of A'_{m-1} at the end of the C rounds is equal to H'_m. In this manner, it is clear that by the time the virtual world of $\mathcal{P}'$ reaches the end of iC cycles (recall that i is defined so that the height of A_3 equals the height of A'_i), the state of the networks in the two worlds will be identical (up to permutation of the nodes). Furthermore, during the next C rounds of each cycle, the adversaries $\mathcal{A}$ and $\mathcal{A}'$ will honor an edge between two nodes ($E(A_2, A_3)$ verses $E(A'_{i-1}, A'_i)$) such that at the moment the C rounds start, the height of A_2 equals A'_{i-1}, and the height of A_3 equals A'_i. Therefore, this process may be repeated iteratively through the end of the cycle in each respective world, and it is clear that the induction hypothesis will remain valid by the end of cycle α. ∎

For the remainder of the section, we will seek to prove (9) for the protocol $\mathcal{P}'$. To simplify notation, it will be convenient to define $m = n-2$. At the outset of every cycle α, we label the internal (i.e. excluding the Sender and Receiver) nodes $\{N_1^\alpha, N_2^\alpha, \ldots, N_m^\alpha\}$, such that if $i < j$, then node N_i^α is storing *more* (or an equal number of) packets at the start of cycle α than N_j^α. For all α, let $N_0^\alpha = S$ and $N_{n-1}^\alpha = R$. For any $1 \leq i \leq n-2$, let H_i^α denote the height the node had *at the outset of* α. We emphasize that while the heights of nodes may change through the course of cycle α, the labeling $\{N_i^\alpha\}$ and the quantities $\{H_i^\alpha\}$ will remain fixed throughout the cycle. Indeed, the following fact implies that the labeling of nodes is independent of α (and in fact is fixed for all time):

Fact 1. For all $\alpha \in \mathbb{N}$ and all $1 \leq i \leq m$: $N_i^\alpha = N_i^{\alpha+1}$

Fact 2. For any cycle α, node N_i is a part of $2C$ rounds of the cycle: first for C rounds with $E(N_{i-1}, N_i)$, and then for C rounds with $E(N_i, N_{i+1})$

These facts, along with the following observations, all follow from the definition/construction of $\mathcal{P}'$ in the proof of Lemma A.1 above. To fix notation, for each $0 \leq i \leq m$ let A_i^α denote the number of packets sent from A_i to A_{i+1} during the C rounds $E(N_i, N_{i+1})$ of cycle α. Note that A_i^α may be negative if the net packet flow during $E(N_i, N_{i+1})$ was towards N_i.

Lemma A.2. *For any cycle α and for any $1 \leq i \leq m$:*

$$1)\ A_i^\alpha \leq \frac{A_{i-1}^\alpha + H_i^\alpha - H_{i+1}^\alpha}{2} \tag{10}$$

$$2)\ A_i^\alpha \leq H_i^{\alpha+1} - H_{i+1}^\alpha \tag{11}$$

Proof. Statement 1 follows from the two facts above as follows. Note that after the C rounds $E(N_{i-1}, N_i)$ but before the next C rounds, node N_i will have height $A^\alpha_{i-1} + H^\alpha_i$. Now by definition of protocol $\mathcal{P}'$, at the end of the C rounds of $E(N_i, N_{i+1})$, N^α_i will have a greater (or equal) number of packets than N^α_{i+1}. In particular, since there are $A^\alpha_{i-1} + H^\alpha_i + H^\alpha_{i+1}$ total packets between the two nodes at the start of the C rounds $E(N^\alpha_i, N^\alpha_{i+1})$, it must be that at the end of these C rounds, N^α_i is storing at least half of these. Since the number of packets stored by N^α_i after the C rounds of $E(N^\alpha_i, N^\alpha_{i+1})$ is given by $A^\alpha_{i-1} + H^\alpha_i - A^\alpha_i$, Statement 1 follows.

Also, again since protocol $\mathcal{P}'$ specifies that N^α_i must have more (or an equal number of) packets as N^α_{i+1} immediately after the C rounds of $E(N^\alpha_i, N^\alpha_{i+1})$, and by Fact 2 the height of N^α_i will not change through the remainder of cycle α, Statement 2 follows. ■

Statement 1 above immediately implies the following, which we state separately for later use:

Corollary A.3. *For any cycle α and for any $1 \le i \le m$:*

$$A^\alpha_i \le \frac{A^\alpha_{i-1} + H^\alpha_i - \min\left(H^\alpha_{i+1}, \frac{C}{m}(m-i-1)\right)}{2}$$

We are interested in the potential function:

$$\Psi'^\alpha = \sum_{i=1}^{m} \left(\frac{1}{2}\right)^{m-i} \cdot \max\left(0, H^\alpha_i - (m-i)\frac{C}{m}\right) \tag{12}$$

For each $1 \le i \le m$, define:

$$\delta^\alpha_i = \begin{cases} 1 & \text{if the } 2^{nd} \text{ term of the max statement in (12) dominates} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Also, for any pair of indices $1 \le i < j \le m$, define:

$$(\Psi'^{\alpha+1} - \Psi'^\alpha)_{i,j} = \sum_{k=i}^{j} \left(\frac{1}{2}\right)^{m-k} \cdot \left[\max\left(0, H^{\alpha+1}_k - (m-k)\frac{C}{m}\right) - \max\left(0, H^\alpha_k - (m-k)\frac{C}{m}\right)\right] \tag{14}$$

Claim A.4. *For any index $1 \le i \le m$ and any cycle α:*

$$H^{\alpha+1}_i = H^\alpha_i + A^\alpha_{i-1} - A^\alpha_i \tag{15}$$

Proof. Notice $N^{\alpha+1}_i = N^\alpha_i$ (Fact 1) and N_i is a part of exactly $2C$ rounds for the α^{th} cycle (Fact 2). In the first C rounds, H_i changes by A^α_{i-1}, and in the second C rounds it changes by $-A^\alpha_i$. Since N_i began the cycle with height H^α_i, we have that its height at the start of the $(\alpha+1)^{th}$ cycle will be $H^\alpha_i + A^\alpha_{i-1} - A^\alpha_i$. ■

It will be convenient to introduce the following notation:

Definition A.5. For any $1 \le i \le m$ and any cycle α, define:

$$\nu^\alpha_i := \max\left(0,\ H^\alpha_i - (m-i)\frac{C}{m}\right) \quad \text{and} \quad \omega^\alpha_i := \min\left(0,\ H^\alpha_i - (m-i)\frac{C}{m}\right) \tag{16}$$

Claim A.6. *For any index $1 \leq i \leq m$ and any cycle α:*

$$1) \text{ If } \delta_i^{\alpha+1} = 1, \text{ then: } \quad (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i,i} = \frac{1}{2^{m-i}}(A_{i-1}^{\alpha} - A_i^{\alpha} + \omega_i^{\alpha})$$

$$2) \text{ If } \delta_i^{\alpha+1} = 0, \text{ then: } \quad (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i,i} = \frac{1}{2^{m-i}}\nu_i^{\alpha} \tag{17}$$

Proof. If $\delta^{\alpha+1} = 1$, then consider the equalities:

$$\begin{aligned}
(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i,i} &= \frac{1}{2^{m-i}}\left[\max\left(0, H_i^{\alpha+1} - (m-i)\frac{C}{m}\right) - \max\left(0, H_i^{\alpha} - (m-i)\frac{C}{m}\right)\right] \\
&= \frac{1}{2^{m-i}}\left[(A_{i-1}^{\alpha} - A_i^{\alpha} + H_i^{\alpha}) - (m-i)\frac{C}{m} - \max\left(0, H_i^{\alpha} - (m-i)\frac{C}{m}\right)\right] \\
&= \frac{1}{2^{m-i}}(A_{i-1}^{\alpha} - A_i^{\alpha}) + \begin{cases} 0 & \text{if } H_i^{\alpha} \geq (m-i)\frac{C}{m} \\ \frac{1}{2^{m-i}}\left(H_i^{\alpha} - \frac{(m-i)C}{m}\right) & \text{if } H_i^{\alpha} < (m-i)\frac{C}{m} \end{cases} \\
&= \frac{1}{2^{m-i}}(A_{i-1}^{\alpha} - A_i^{\alpha} + \omega_i^{\alpha})
\end{aligned}$$

where the second equality is from Claim A.4 together with the assumption that $\delta^{\alpha+1} = 1$. Otherwise, if $\delta^{\alpha+1} = 0$, then Statement 2 is immediate. ∎

Lemma A.7. *For any pair of indices $1 \leq i < j < m$ for which $\delta_k^{\alpha+1} = 1$ for every $i \leq k \leq j$:*[9]

$$(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i,j} + \frac{A_j}{2^{m-j-1}} - \sum_{k=i}^{j} \frac{\omega_k}{2^{m-k}} \leq \frac{A_{i-1}}{2^{m-i}} + \frac{(j\text{-}i\text{+}1)}{2^{m-i+1}}(A_{i-1}+H_i) - \frac{H_{j+1}}{2^{m-j+1}} + \sum_{k=i+1}^{j-1} \frac{(j-k)}{2^{m-k+2}} H_k$$

Proof. This follows via an inductive argument on $j - i$ together with Lemma A.2 and Claim A.6:

<u>BASE CASE: $j = i + 1$</u>: First consider the right-hand-side of the inequality of Lemma A.7 with $j = i + 1$:

$$\begin{aligned}
\text{RHS A.7} &= \frac{A_{i-1}}{2^{m-i}} + \frac{2}{2^{m-i+1}}(A_{i-1} + H_i) - \frac{H_{i+2}}{2^{m-i}} \\
&= \frac{A_{i-1}}{2^{m-i}} + \frac{1}{2^{m-i}}(A_{i-1} + H_i) - \frac{H_{i+2}}{2^{m-i}} \\
&= \frac{A_{i-1}}{2^{m-i-1}} + \frac{1}{2^{m-i}}(H_i - H_{i+2})
\end{aligned} \tag{18}$$

[9] Unless explicity written otherwise, assume all superscripts are α, which we have suppressed for notational convenience.

Meanwhile, for $j = i+1$, the left-hand-side of the inequality of Lemma A.7 is:

$$\begin{aligned}
\text{LHS A.7} &= (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i,i+1} + \frac{A_{i+1}}{2^{m-i-2}} - \sum_{k=i}^{i+1} \frac{\omega_k}{2^{m-k}} \\
&= (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i,i} + (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,i+1} + \frac{A_{i+1}}{2^{m-i-2}} - \sum_{k=i}^{i+1} \frac{\omega_k}{2^{m-k}} \\
&= \frac{1}{2^{m-i}}(A_{i-1} - A_i + \omega_i) + \frac{1}{2^{m-i-1}}(A_i - A_{i+1} + \omega_{i+1}) + \frac{A_{i+1}}{2^{m-i-2}} - \sum_{k=i}^{i+1} \frac{\omega_k}{2^{m-k}} \\
&= \frac{1}{2^{m-i-1}} A_{i+1} + \frac{1}{2^{m-i}}(A_i + A_{i-1}) \\
&\leq \frac{1}{2^{m-i}}((A_i + H_{i+1} - H_{i+2}) + (A_i + A_{i-1})) \\
&= \frac{1}{2^{m-i}}(A_{i-1} + H_{i+1} - H_{i+2}) + \frac{1}{2^{m-i-1}} A_i \\
&\leq \frac{1}{2^{m-i}}(A_{i-1} + H_{i+1} - H_{i+2}) + \frac{1}{2^{m-i}}(A_{i-1} + H_i - H_{i+1}) \\
&= \frac{A_{i-1}}{2^{m-i-1}} + \frac{1}{2^{m-i}}(H_i - H_{i+2})
\end{aligned} \tag{19}$$

where the third equality is due to Claim A.6, the first inequality is Statement 1 of Lemma A.2 (applied to A_{i+1}), and the second inequality is Statement 1 of Lemma A.2 (applied to A_i). Notice (18) matches (19), as required.

Induction Step: Consider the string of inequalities:

$$
\begin{aligned}
(\Psi'^{\alpha+1}-\Psi'^{\alpha})_{i,j} + \frac{A_j}{2^{m-j-1}} - \sum_{k=i}^{j} \frac{\omega_k}{2^{m-k}} &= (\Psi'^{\alpha+1}-\Psi'^{\alpha})_{i,i} + (\Psi'^{\alpha+1}-\Psi'^{\alpha})_{i+1,j} + \frac{A_j}{2^{m-j-1}} - \sum_{k=i}^{j} \frac{\omega_k}{2^{m-k}} \\
&\leq \frac{A_{i-1}-A_i}{2^{m-i}} + \frac{A_i}{2^{m-i-1}} + \frac{j-i}{2^{m-i}}(A_i + H_{i+1}) \\
&\quad - \frac{H_{j+1}}{2^{m-j+1}} + \sum_{k=i+2}^{j-1} \frac{(j-k)}{2^{m-k+2}} H_k \\
&= \frac{A_{i-1}-A_i}{2^{m-i}} + \frac{A_i}{2^{m-i-1}} + \frac{j-i+1}{2^{m-i}}(A_i + H_{i+1}) \\
&\quad - \frac{A_i + H_{i+1}}{2^{m-i}} - \frac{H_{j+1}}{2^{m-j+1}} + \sum_{k=i+2}^{j-1} \frac{(j-k)}{2^{m-k+2}} H_k \\
&\leq \frac{A_{i-1}-A_i}{2^{m-i}} + \frac{A_i}{2^{m-i-1}} + \frac{j-i+1}{2^{m-i+1}}(A_{i-1} + H_i + H_{i+1}) \\
&\quad - \frac{A_i + H_{i+1}}{2^{m-i}} - \frac{H_{j+1}}{2^{m-j+1}} + \sum_{k=i+2}^{j-1} \frac{(j-k)}{2^{m-k+2}} H_k \\
&= \frac{A_{i-1}}{2^{m-i}} + \frac{j-i+1}{2^{m-i+1}}(A_{i-1} + H_i) + \frac{j-i-1}{2^{m-i+1}}(H_{i+1}) + \\
&\quad \frac{2}{2^{m-i+1}}(H_{i+1}) - \frac{H_{i+1}}{2^{m-i}} - \frac{H_{j+1}}{2^{m-j+1}} + \sum_{k=i+2}^{j-1} \frac{(j-k)}{2^{m-k+2}} H_k \\
&= \frac{A_{i-1}}{2^{m-i}} + \frac{j-i+1}{2^{m-i+1}}(A_{i-1} + H_i) - \frac{H_{j+1}}{2^{m-j+1}} + \sum_{k=i+i}^{j-1} \frac{(j-k)}{2^{m-k+2}} H_k
\end{aligned}
$$

where the first inequality is by the induction hypothesis together with Claim A.6 and the second inequality is by Statement 1 of Lemma A.2. ■

Lemma A.8. *For any pair of indices $1 \leq i < i+1 < j \leq m$ for which $\delta_j^{\alpha+1} = 1$ but $\delta_k^{\alpha+1} = 0$ for every $i < k < j$:*[10]

$$
(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,j-1} + \frac{A_{j-1}}{2^{m-j}} - \sum_{k=i+1}^{j-1} \frac{\omega_k}{2^{m-k}} \quad \leq \quad \frac{A_i}{2^{m-i-1}} + \frac{H_{i+1}}{2^{m-i}} - \frac{H_j}{2^{m-j+1}} + \sum_{k=i+1}^{j-1} \frac{H_k}{2^{m-k+1}}
$$

Proof. This follows via an inductive argument on $j-i$ together with Lemma A.2:

Base Case: $j-i=2$: Looking at the right-hand-side of the inequality of Lemma A.8 for $j = i+2$:

$$
\begin{aligned}
\text{RHS A.8} &= \frac{A_i}{2^{m-i-1}} + \frac{H_{i+1}}{2^{m-i}} - \frac{H_{i+2}}{2^{m-i-1}} + \frac{H_{i+1}}{2^{m-i}} \\
&= \frac{A_i + H_{i+1} - H_{i+2}}{2^{m-i-1}} \qquad (20)
\end{aligned}
$$

[10]On the right-hand side of the inequality of Lemma A.7, all superscripts are α, which we have suppressed for notational convenience.

Meanwhile, looking at the left-hand-side of the inequality of Lemma A.8 for $j = i+2$:

$$\begin{aligned}
\text{LHS A.8} &= (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,i+1} + \frac{A_{i+1}}{2^{m-i-2}} - \frac{\omega_{i+1}}{2^{m-i-1}} \\
&= \frac{A_{i+1}}{2^{m-i-2}} \\
&\leq \frac{A_i + H_{i+1} - H_{i+2}}{2^{m-i-1}},
\end{aligned} \tag{21}$$

where the second equality is from Claim A.6 (since $\delta_{i+1}^{\alpha+1} = 0$) and the inequality is Statement 1 of Lemma A.2. Notice (20) matches (21), as required.

<u>Induction Step</u>: Consider the string of inequalities:

$$\begin{aligned}
(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,j-1} + \frac{A_{j-1}}{2^{m-j}} - \sum_{k=i+1}^{j-1} \frac{\omega_k}{2^{m-k}} &= (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,i+1} + (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+2,j-1} + \frac{A_{j-1}}{2^{m-j}} - \sum_{k=i+1}^{j-1} \frac{\omega_k}{2^{m-k}} \\
&\leq \frac{A_{i+1}}{2^{m-i-2}} + \frac{H_{i+2}}{2^{m-i-1}} - \frac{H_j}{2^{m-j+1}} + \sum_{k=i+2}^{j-1} \frac{H_k}{2^{m-k+1}} \\
&\leq \frac{A_i}{2^{m-i-1}} + \frac{H_{i+1}}{2^{m-i}} - \frac{H_j}{2^{m-j+1}} + \sum_{k=i+1}^{j-1} \frac{H_k}{2^{m-k+1}}
\end{aligned}$$

where the first inequality is by the induction hypothesis together with Claim A.6 and the last inequality is by Statement 1 of Lemma A.2. ■

Lemma A.9. *For any cycle α and any index $1 \leq i < m-1$, if $\delta_i^{\alpha+1} = 1$, $\delta_{i+1}^{\alpha+1} = 0$, and $\delta_{i+2}^{\alpha+1} = 1$, then:*

$$(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,i+1} + \frac{A_{i+1}}{2^{m-i-2}} - \sum_{k=i+1}^{i+1} \frac{\omega_k}{2^{m-k}} \quad \leq \quad \frac{A_i}{2^{m-i-1}} + \frac{1}{2^{m-i-1}} \frac{C}{m} \tag{22}$$

Proof. Consider:

$$\begin{aligned}
(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,i+1} + \frac{A_{i+1}}{2^{m-i-2}} - \sum_{k=i+1}^{i+1} \frac{\omega_k}{2^{m-k}} \quad &= \quad \frac{A_{i+1}}{2^{m-i-2}} \\
&\leq \frac{A_i + H_{i+1} - H_{i+2}}{2^{m-i-1}} \\
&\leq \frac{A_i}{2^{m-i-1}} + \frac{1}{2^{m-i-1}} \frac{C}{m}
\end{aligned}$$

where the first equality is Statement 2 of Lemma A.6, the first inequality is Statement 1 of A.2, and the last inequality follows from the fact that $\delta_{i+1}^{\alpha+1}{=}0$, and $\delta_{i+2}^{\alpha+1}{=}1$ implies that $H_{i+1} - H_{i+2} \leq \frac{C}{m}$. ■

Lemma A.10. *For any cycle α and any index $1 \leq i < m-2$, if $\delta_i^{\alpha+1} = 0$, $\delta_{i+1}^{\alpha+1} = 1$, and $\delta_{i+2}^{\alpha+1} = 0$, then:*

$$(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,i+1} + \frac{A_{i+1}}{2^{m-i-2}} - \sum_{k=i+1}^{i+1} \frac{\omega_k}{2^{m-k}} \quad \leq \quad \frac{A_i}{2^{m-i-1}} + \frac{1}{2^{m-i-1}} \frac{C}{m} \tag{23}$$

Proof. Consider:

$$\begin{aligned}
(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{i+1,i+1} + \frac{A_{i+1}}{2^{m-i-2}} - \sum_{k=i+1}^{i+1} \frac{\omega_k}{2^{m-k}} &= \frac{A_i}{2^{m-i-1}} + \frac{A_{i+1}}{2^{m-i-1}} \\
&\leq \frac{A_i + H_{i+1}^{\alpha+1} - H_{i+2}^{\alpha}}{2^{m-i-1}} \\
&\leq \frac{A_i}{2^{m-i-1}} + \frac{1}{2^{m-i-1}} \frac{C}{m}
\end{aligned}$$

where the first equality is Statement 1 of Lemma A.6, the first inequality is Statement 2 of A.2, and the last inequality follows from the fact that $\delta_i^{\alpha+1} = 0$, $\delta_{i+1}^{\alpha+1} = 1$, and $\delta_{i+2}^{\alpha+1} = 0$ implies that $\frac{H_{i+1}^{\alpha+1} - H_{i+2}^{\alpha}}{2^{m-i-1}} \leq \frac{1}{2^{m-i-1}} \frac{C}{m}$. ∎

Claim A.11. *For any cycle α, we have:*

$$Z^{\alpha} + (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{m,m} \leq A_{m-1}^{\alpha} \tag{24}$$

Proof. Since $(H_m^{\alpha+1} - (m-m)\frac{C}{m}) = H_m^{\alpha+1} \geq 0$, we have that the second term of $\min(0, H_m^{\alpha+1} - (m-m)\frac{C}{m})$ always dominates, and hence for all cycles, $\delta_m^{\alpha+1} = 1$. Therefore, applying Claim A.6 (for $i = m$):

$$\begin{aligned}
(\Psi'^{\alpha+1} - \Psi'^{\alpha})_{m,m} &= A_{m-1}^{\alpha} - A_m^{\alpha} + \omega_m^{\alpha} \\
&\leq A_{m-1}^{\alpha} - A_m^{\alpha} \\
&= A_{m-1}^{\alpha} - Z^{\alpha}
\end{aligned} \tag{25}$$

where the inequality follows since $\omega_i^{\alpha} \leq 0$ for all cycles α and nodes i, and the last equality is because N_m is the node that will be connected to the Receiver in the last C rounds of α, so by definition $A_m^{\alpha} = Z^{\alpha}$. ∎

We are now ready to prove the main result of this section, namely that (9) is satisfied for all cycles α:

Lemma A.12. *For all cycles α, the following is always true:*

$$Z'^{\alpha} + (\Psi'^{\alpha+1} - \Psi'^{\alpha}) \leq 7\frac{C}{m},$$

Proof. Fix cycle α, and consider the string of bits $\{\delta_i^{\alpha+1}\}_{i=1}^{m}$:

$$(\delta_1^{\alpha+1}, \delta_2^{\alpha+1}, \ldots, \delta_{m-1}^{\alpha+1}, \delta_m^{\alpha+1}) \tag{26}$$

By Claim A.11, we have:

$$Z^{\alpha} + \Psi'^{\alpha+1} - \Psi'^{\alpha} = Z^{\alpha} + (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{1,m} \leq (\Psi'^{\alpha+1} - \Psi'^{\alpha})_{1,m-1} + A_{m-1}^{\alpha} \tag{27}$$

We now use Lemmas A.7, A.8, A.9, and A.10 on the appropriate indices (based on the form of $\{\delta_i^{\alpha+1}\}$), which yields:[11]

[11] We combine these lemmas by starting at the far right index $i = m - 1$, and working our way down through smaller indices by using the appropriate lemma. Notice that the first term on the RHS of the inequality of each lemma is exactly the term needed on the LHS of the next lemma.

1. For the smallest index i such that $\delta_i^{\alpha+1} = 1$, we have leading term:

$$\frac{A_{i-1}}{2^{m-i}} \tag{28}$$

2. For any indices (i,j) falling under Lemma A.7, we have contributions:

$$\frac{j-i+1}{2^{m-i+1}}(A_{i-1}+H_i) + \sum_{k=i+1}^{j-1} \frac{(j-k+1)(m-i)}{2^{m-k+2}} \tag{29}$$

3. For any indices (i,j) falling under Lemma A.8, we have contribution:

$$\sum_{k=i}^{j} \frac{m-i}{2^{m-k+1}} \tag{30}$$

4. For any indices (i,j) falling under Lemma A.9 or A.10, we have contribution:

$$\frac{1}{2^{m-i-1}} \frac{C}{m} \tag{31}$$

Notice that in terms of the contributions from (29), $(A_{i-1}+H_i) \leq \frac{(m-i-1)C}{m}$ by Statement 2 of Lemma A.2 together with the fact that $\delta_{i-1}^{\alpha+1} = 0$ implies $H_{i-1}^{\alpha+1} < \frac{(m-i+1)C}{m}$. The theorem now follows immediately from the facts:

1. For any $1 \leq i < j < \infty$, $\quad \sum_{k=i}^{j} \frac{1}{2^k} \leq \sum_{k=1}^{\infty} \frac{1}{2^k} = 1$
2. For any $1 \leq i < j < \infty$, $\quad \sum_{k=i}^{j} \frac{k}{2^k} \leq \sum_{k=1}^{\infty} \frac{k}{2^k} = 2$
3. For any $1 \leq i < j < \infty$, $\quad \sum_{k=i}^{j} \frac{k(k-1)}{2^k} \leq \sum_{k=1}^{\infty} \frac{k(k-1)}{2^k} = 4$ ■

The remainder of the proof that the optimal competitive ratio is $1/n$ was presented in Section 3.

B Rigorous Proof of Competitive Ratio of Slide

The high-level ideas of the proof of Theorem 2 were sketched in Section 4.2, and we encourage the reader to re-read that section before proceeding here. In this Section, we begin by providing in Section B.1 a deeper explanation of the proof than was provided in Section 4.2, but still does not go into the details of the proofs. Then in Sections B.2-B.5 we rigorously prove all the lemmas and theorems.

B.1 Motivation and Definitions

In what follows, unless stated otherwise, all notation is as defined in Section 4.2. Recall from Section 4.2 that we wish to construct two potential functions. The first one, denoted by $\varphi_{p'}$, will be associated to every packet $p' \in Z_2^{\mathcal{P}'}$. However, $\varphi_{p'}$ will not be exactly as defined in Section 4.2, so we provide now the motivation to explain how $\varphi_{p'}$ is actually defined, and why we need to slightly change what it represents.

Our first attempt employed in Section 4.2 was to define $\varphi_{p'}$ to be the height, *with respect to* $\mathcal{P}$, of the node in which p' was currently being stored. **We state once-and-for-all that when referencing the *height* of a node, we will mean its height with respect to the Slide protocol** $\mathcal{P}$. As noted in Section 4.2, if we define $\varphi_{p'}$ this way, then for every $p' \in Z_2^{\mathcal{P}'}$, $\varphi_{p'}$ will be initially set to C (when $\mathcal{P}'$ first inserts p'), and $\varphi_{p'}$ will be zero when p' is delivered to the Receiver. Thus, there is a net change of $-C$ to $\varphi_{p'}$ from the time of insertion by the Sender to the time of reception by the Receiver. The goal was then to define a second overall network potential function Φ, which increases by C every time $\mathcal{P}$ transfers a packet, and such that any time $\varphi_{p'}$ changes for any $p' \in Z_2^{\mathcal{P}'}$, the cumulative changes of $\sum_{p' \in Z_2^{\mathcal{P}'}} \varphi_{p'}$ will be mimicked by Φ. Since Φ increases by C when there is a packet transfer in $\mathcal{P}$, one (good) way to think of this approach is that for each drop in $\varphi_{p'}$, we would like to find a packet transfer in $\mathcal{P}$ that can be "charged," i.e. this packet transfer "allowed" $\varphi_{p'}$ to decrease.

Unfortunately, with the simplistic definition of $\varphi_{p'}$ equal to the height of the node it is currently stored in, we encounter a problem. To clarify the problem, as well as to set notation, at the very beginning of each round x, we will label the *internal* nodes (i.e. not the Sender or Receiver) as: $\{N_1^x, N_2^x, \dots, N_{n-2}^x\}$, where the labeling respects heights, so that at the start of the round x, N_{i+1}^x is storing at least as many packets as N_1^x (ties are broken arbitrarily). Letting H_i^x denote the height of N_i^x at the start of x (i.e. the number of packets N_i^x is storing with respect to $\mathcal{P}$), we may restate the criterion for labeling nodes at the start of each round by writing: $H_1^x \le H_2^x \le \dots \le H_{n-2}^x$. Note that nodes may change labels from one round to the next, i.e. we may have $N_i^x \ne N_i^{x+1}$. When the round is unimportant, we will suppress the superscript x. Let S denote the Sender and R denote the Receiver.

We may now explain why the simplistic definition of $\varphi_{p'}$ above will not be adequate. Define $Q := \frac{C-n}{n}$, and consider the following two scenarios that may be present at the start of some round x:

Scenario 1:	$H_{n-2} = C$	$H_{n-3} = C$	$\dots$	$H_3 = C$	$H_2 = C$	$H_1 = (n-3)Q$
Scenario 2:	$H_{n-2} = (n-3)Q$	$H_{n-3} = (n-4)Q$	$\dots$	$H_3 = 2Q$	$H_2 = Q$	$H_1 = 0$

In Scenario 1, consider a packet $p' \in Z_2^{\mathcal{P}'}$ that begins round x in node N_1, so that $\varphi_{p'} = (n-3)Q$. Notice that if the adversary honors the edge $E(N_1, R)$, the Slide protocol will transfer a packet to the Receiver (Rules 2 and 3a of Section 4.1). Now by definition of being in the set $Z_2^{\mathcal{P}'}$, in order for p' to be delivered to the Receiver via node[12] N_1, node N_1 must have height zero when the adversary honors edge $E(N_1, R)$. Therefore, there must be exactly $(n-3)Q$ transfers in $\mathcal{P}$ (to drain N_1) before p' can be delivered to R via N_1. Thus, loosely speaking, we can "charge" the resulting drop in $\varphi_{p'}$ from $(n-3)Q$ to 0 to these $(n-3)Q$ transfers in $\mathcal{P}$.

[12]Of course there is no reason to assume that p' must be transferred to R via N_1, but for the sake of the example, we imagine this is the case.

Now instead imagine we are in Scenario 2, and again fix a packet $p' \in Z_2^{\mathcal{P}'}$ such that $\varphi_{p'} = (n-3)Q$ at the start of round x, so $p' \in N_{n-2}$. In this case, notice that p' has a way to reach R without *any* packets being transferred in $\mathcal{P}$. In particular, the adversary could honor edge $E(N_{n-2}, N_{n-3})$ in round x, and then $E(N_{n-3}, N_{n-4})$ in round $x+1$, and so forth. Since the difference in heights between adjacent nodes is less than C/n, the Slide protocol will not transfer any packets during these rounds. Meanwhile, protocol $\mathcal{P}'$ may dictate that p' is transferred each of these rounds, all the way to the Receiver. Thus, in this scenario, $\varphi_{p'}$ was able to decrease from $(n-3)Q$ to zero *without any packets being transferred in* $\mathcal{P}$. Because we are trying to associate drops in $\varphi_{p'}$ to packet transfers in $\mathcal{P}$, this is problematic.

Notice that the problem in Scenario 2 is that there exists a "bridge" between N_{n-2} and R. That is, even though N_{n-2} has a relatively large height, there is still a way for packets $p' \in Z_2^{\mathcal{P}'}$ that are in N_{n-2} to reach R without $\mathcal{P}$ being able to transfer any packets. In contrast, in Scenario 1, $p' \in N_1$ will also have $\varphi_{p'} = (n-3)Q$, but now there must be $(n-3)Q$ transfers in $\mathcal{P}$ before p' can reach R (again, since $p' \in Z_2^{\mathcal{P}'}$ requires that p' is never transferred at the same time as a packet in $\mathcal{P}$). In summary, one might say that even though node N_1 in Scenario 1 has the same height as node N_{n-2} from Scenario 2, these two nodes have different "effectual" heights.

Considering the above two Scenarios, we were encouraged to modify our definition of $\varphi_{p'}$ as follows:

- For node N_i, define the node's effectual height:[13] $\widetilde{H}_i := \max(0, H_i - (i-1)\frac{C}{n})$
- For any $p' \in Z_2^{\mathcal{P}'}$ that is currently in N_i, define its potential: $\varphi_{p'} := \widetilde{H}_i$

This is *almost* the actual definition we eventually make for φ, but we will need to first "smooth-out" this definition. To motivate the need to smooth the definition, consider the following events, which represent the only ways that $\varphi_{p'}$ can change (based on the new definition of $\varphi_{p'}$):

Case 1. p' is transferred from N_i to N_j in some round $E(N_i, N_j)$

Case 2. $p' \in N_i$ when N_i changes height due to a packet transfer in $\mathcal{P}$, but this packet transfer does *not* cause a re-indexing of nodes

Case 3. p' is in some node N_i when a packet transfer in $\mathcal{P}$ causes N_i to change index to N_j (i.e. this node moves from the i^{th} fullest node to the j^{th} fullest node)

Since we are only concerned with $p' \in Z_2^{\mathcal{P}'}$, we note that whenever $\varphi_{p'}$ changes as by 1) above, necessarily $\mathcal{P}$ did not transfer a packet this round. In particular, this means that $|H_i - H_j| < C/n$. In order to control changes to $\varphi_{p'}$ that are a result of Case 1, we would therefore like for $\widetilde{H}_i \approx \widetilde{H}_j$ whenever $H_i \approx H_j$. Although the definition of effectual height $\widetilde{H}_i$ above almost captures this, there is necessarily a "jump" of C/n between the values $\widetilde{H}_i$ and $\widetilde{H}_j$. This is one of the reasons we will want to "smooth-out" the definition of $\varphi_{p'}$.

Changes to $\varphi_{p'}$ that come from Case 2 above are okay, since in such cases $\varphi_{p'}$ will change by one, and this can be "charged" to the fact that there has been a packet transfer in $\mathcal{P}$. Lastly, notice that $\varphi_{p'}$ can only change as in Case 3 above if there are two nodes at the outset of some round x, N_i and N_{i+1}, such that a packet transfer during round x causes them to switch places (e.g. before the transfer, $H_i = H_{i+1}$, and then N_i receives a packet in round x). Because there has been a packet

[13] The "maximum" is added to prevent the effectual height of a node from being negative.

transfer in $\mathcal{P}$, we can "charge" some of the changes in $\varphi_{p'}$ to this packet transfer, but again the fact that there will be a "jump" of C/n to changes in φ will encourage a "smoothing" of the definition of φ.

This leads to the notion of a family of nodes. In particular, we will partition the internal nodes into families. Intuitively, two nodes will be in the same family if they are relatively close to each other in height (or more generally, if there is a "bridge" connecting them, as in Scenario 2 above). Then within each family, we will distribute the cumulative effectual height of the nodes in that family evenly among all nodes in the family. Formally, for a family of nodes[14] $\mathcal{F} = \{N_i, N_{i+1}, \dots, N_j\}$, define the cumulative effectual height $H_{\mathcal{F}}$ of the family $\mathcal{F}$ by:

$$\widetilde{H}_{\mathcal{F}} := \sum_{k=i}^{j} \widetilde{H}_k = \sum_{k=i}^{j} \max\left(0, H_k - (k-1)\frac{C}{n}\right)$$

For any $p' \in Z_2^{\mathcal{P}'}$ such that p' is currently in some node of family $\mathcal{F}$, we will define $\varphi_{p'}$ to be the *average* effectual height of the family, i.e.:

$$\varphi_{p'} := \frac{\widetilde{H}_{\mathcal{F}}}{|\mathcal{F}|}$$

Of course, $\widetilde{H}_{\mathcal{F}}$ may not divide evenly among the nodes in the family $\mathcal{F}$, and then to force $\varphi_{p'} \in \mathbb{N}$, we will distribute the excess weight (the remainder) to the nodes with higher indices. Based on this definition of $\varphi_{p'}$, note that if p' transfers between two nodes of the same family, $\varphi_{p'}$ can change by at most one.

We re-visit the three ways $\varphi_{p'}$ may change, explaining in each case how we can find a packet transfer in $\mathcal{P}$ to "charge" for the change in $\varphi_{p'}$. In terms of changes to $\varphi_{p'}$ resulting from Case 1 above, we recall that necessarily $|H_i - H_j| < C/n$. We show in Lemma B.12 that anytime $|H_i - H_j| < C/n$, N_i and N_j are necessarily in the same family, in which case our definition of φ now guarantees that $\varphi_{p'}$ can change by at most one when p' is transferred between nodes. Changes to $\varphi_{p'}$ due to Case 2 will be at most one (since the cumulative effectual height of the family will change by at most one, and this change will be distributed among nodes in the family), and we can "charge" such changes to the packet transfer in $\mathcal{P}$ that caused Case 2 to occur. Finally, for Case 3, if $p' \in N_i$ when N_i's index changes but N_i remains in the same family, then since φ is distributed evenly among nodes in the family, the change in index will be irrelevant (i.e. this will not cause $\varphi_{p'}$ to change). On the other hand, we will show that whenever a node N_i switches *families* as a result of a packet transfer in $\mathcal{P}$, the average effectual height of its new family will differ by at most one from the average effectual height of its old family. Thus, in this case the change in $\varphi_{p'}$ is also bounded by one, and we can "charge" this change to the packet transfer that caused families to re-align.

Defining how to partition nodes into families so that the families behave the way we want (e.g. so that: 1) nodes with height within C/n of each other are in the same family; 2) Families can only re-align during a round in which $\mathcal{P}$ transfers a packet; and 3) When families re-align, the average effectual height of any node before and after the re-alignment differs by at most one) requires a little thought, and it is done precisely in the following section. Once we have the formal definition of a family, we would like to formalize the notion of "charging a change in $\varphi_{p'}$ to a packet transfer in

[14] We will show in the next section that nodes within the same family will always have adjacent indices.

$\mathcal{P}$." Namely, as mentioned in Section 4.2, we define a second network potential Φ that will increase by C every time there is a packet transfer in $\mathcal{P}$, and that will also mirror the cumulative changes of $\varphi_{p'}$ for each $p' \in Z_2^{\mathcal{P}'}$. In order to prove Φ is always positive, we will distribute the total network potential between the families:

$$\Phi = \Phi_{\mathcal{F}_1} + \cdots + \Phi_{\mathcal{F}_l} \tag{32}$$

and then show in Lemma B.17 that within each family $\mathcal{F}$:

$$\Phi_{\mathcal{F}} \ \geq \ 0. \tag{33}$$

The careful definition of families and the precise definition of the potential φ and the network potential Φ is presented below in Section B.2. The main lemma and proof of the fact that at all times $\Phi \geq 0$ can be found in Section B.5.

B.2 Formal Definition of "Family" and Potential of a Packet ($\varphi_{p'}$)

We begin by defining formally the notion of a family introduced in the previous section. Note that families will in general re-align during a round when there is a packet transfer in $\mathcal{P}$, so we use the notation $\mathcal{F}^x$ to denote some family $\mathcal{F}$ that was in existence at the start of round x. Recall that at the start of each round x, the internal nodes are indexed according to their heights with respect to $\mathcal{P}$: $\{N_1, N_2, \dots, N_{n-2}\}$, so that $H_i \leq H_j$ if $i < j$ (ties are broken arbitrarily). Also recall from the previous section the definition of the effectual height $\widetilde{H}_i$ of node N_i:

$$\widetilde{H}_i := \max\left(0, H_i - (i-1)\frac{C}{n}\right) \tag{34}$$

At the start of each round, we will partition the internal nodes into families inductively (starting from the emptiest nodes), so that the average effectual height of each family is minimized. In particular:

Definition B.1. At the start of round x, internal nodes will be partitioned into families $\{\mathcal{F}_i^x\}$ as follows. Starting at $i = 1$ and $k_0 = 0$:

F1 *Find index $k_{i-1} < k_i \leq n-2$ such that the following quantity is **minimal**:*

$$\frac{\displaystyle\sum_{j=k_{(i-1)}+1}^{k_i} \widetilde{H}_j}{(k_i - k_{i-1})} \tag{35}$$

*In case there are multiple values for k_i that achieve the same minimum, define k_i to be the **largest** of all possibilities. Then define*[15] *family $\mathcal{F}_i^x := \{N^x_{k_{(i-1)}+1}, \dots, N^x_{k_i}\}$.*

F2 *Set $i = i+1$ and repeat Step F1 until all internal nodes are in some family.*

F3 *The Sender and Receiver will form their own, separate, families. Denote the Sender's family by $\mathcal{F}_n$ and the Receiver's family by $\mathcal{F}_0$.*[16]

[15]When the round x is unimportant, we will suppress the superscript in our notation.

[16]The only reason we place the Sender and Receiver in a family at all is to make the terminology easier in the lemmas that follow. In particular, the notation we use for the Sender's family ensures that it will have a higher index than all other nodes (there will be a gap between the index of the largest indexed family of internal nodes and the Sender's family, which is unimportant), and conversely the Receiver's family will have a smaller index than all other nodes.

Definition B.2. The cumulative effectual height $\widetilde{H}_{\mathcal{F}}$ of a family $\mathcal{F}$ is the sum of the effectual heights of each of the nodes in the family. The average effectual height $\langle\widetilde{H}_{\mathcal{F}}\rangle$ of a family is the cumulative effectual height divided by the size of the family. Succinctly, if $\mathcal{F} := \{N_i, N_{i+1}, \dots, N_j\}$:

$$\widetilde{H}_{\mathcal{F}} := \sum_{k=i}^{j} \widetilde{H}_k \qquad \text{and} \qquad \langle\widetilde{H}_{\mathcal{F}}\rangle := \frac{\widetilde{H}_{\mathcal{F}}}{|\mathcal{F}|} = \frac{\sum_{k=i}^{j} \widetilde{H}_k}{j-i+1} \tag{36}$$

Notice that by construction (see Rules F1 and F2), families are created so that the average effectual height of (the lowest indexed) families is minimized.

With the formal definition of families in hand, we are ready to formally define the first kind of potential, φ. Recall that this potential will be associated to packets $p' \in Z_2^{\mathcal{P}'}$, and if $p' \in N_i \in \mathcal{F}$ at the start of some round, then $\varphi_{p'}$ will (roughly) represent the average effectual height $\langle\widetilde{H}_{\mathcal{F}}\rangle$. More precisely, we will ascribe to each node $N_i \in \mathcal{F}$ a potential φ_i equal to the average effectual height, except that the potential for some nodes in the family will be one bigger to account for the case that $\frac{\widetilde{H}_{\mathcal{F}}}{|\mathcal{F}|} \notin \mathbb{Z}$. Formally:

Definition B.3. Let $\mathcal{F} = \{N_i, N_{i+1}, \dots, N_j\}$. Then the potential φ_k of a node $N_k \in \mathcal{F}$ will be either $\langle\widetilde{H}_{\mathcal{F}}\rangle$ or $\langle\widetilde{H}_{\mathcal{F}}\rangle + 1$. More precisely, writing:

$$\widetilde{H}_{\mathcal{F}} = \lfloor\langle\widetilde{H}_{\mathcal{F}}\rangle\rfloor * |\mathcal{F}| \; + \; r \tag{37}$$

Then define subsets of $\mathcal{F}$:

$$\mathcal{F}^- := \{N_i, N_{i+1}, \dots, N_{j-r}\} \quad \text{and} \quad \mathcal{F}^+ := \{N_{j-r+1}, \dots, N_j\} \tag{38}$$

Then for nodes $N_k \in \mathcal{F}^+$, define $\varphi_k = \lfloor\langle\widetilde{H}_{\mathcal{F}}\rangle\rfloor + 1$. For nodes $N_k \in \mathcal{F}^-$, define $\varphi_k = \lfloor\langle\widetilde{H}_{\mathcal{F}}\rangle\rfloor$. Finally, if $p' \in Z_2^{\mathcal{P}'}$ and p' is currently being stored in N_k, then define the potential $\varphi_{p'}$ to be the potential of N_k, i.e. $\varphi_{p'} := \varphi_k$.

One immediate consequence of the above definition that we will need later is:

Lemma B.4. *At the beginning of any round x and for any family $\mathcal{F}^x$, the sum of the potentials for the nodes in $\mathcal{F}$ equals the cumulative effectual height of the family:*

$$\sum_{N \in \mathcal{F}} \varphi_N = \widetilde{H}_{\mathcal{F}} \tag{39}$$

Definition B.5. The network potential Φ is an integer satisfying the following properties:

1. Φ begins the protocol equal to zero.
2. Φ increases by $4C$ every time a packet is transferred in protocol $\mathcal{P}$
3. For any packet $p' \in Z_2^{\mathcal{P}'}$, any time $\varphi_{p'}$ changes, Φ changes by the same amount.

B.3 Preliminary Lemmas

In this section, we state and prove the basic properties that follow from the definitions of the previous section.

Lemma B.6. *At all times, all families consist of nodes with adjacent indices. In particular, if at the start of any round x there are l families, then there exist indices $k_1 < k_2 < \cdots < k_{l-1}$ such that:*

$$\mathcal{F}_1 = \{N_1, \ldots, N_{k_1}\}, \quad \mathcal{F}_2 = \{N_{k_1+1}, \ldots, N_{k_2}\}, \; \ldots, \; \mathcal{F}_l = \{N_{k_{l-1}+1}, \ldots, N_{n-2}\} \tag{40}$$

Proof. This follows immediately from the rules regarding the construction of families (see F1 and F2 in the previous section). ■

Lemma B.7. *Fix some round x and some pair of nodes N_i^x and N_j^x for $i < j$. Then:*

1. *If $H_i^x \geq H_j^x - C/n$, then $\widetilde{H}_i^x \geq \widetilde{H}_j^x$.*
2. *If $H_i^x < H_j^x - (j-i)C/n$ and $\widetilde{H}_j > 0$, then $\widetilde{H}_i^x < \widetilde{H}_j^x$.*

Proof. Consider the following string of inequalities:

$$\begin{aligned}
\widetilde{H}_i - \widetilde{H}_j &= \max(0, H_i - (i-1)C/n) - \max(0, H_j - (j-1)C/n) \\
&\geq \max(0, H_i - (i-1)C/n) - \max(0, (H_i + C/n) - (j-1)C/n) \\
&\geq \max(0, H_i - (i-1)C/n) - \max(0, (H_i + C/n) - ((i+1)-1)C/n) \\
&= \max(0, H_i - (i-1)C/n) - \max(0, (H_i - (i-1)C/n) \\
&= 0
\end{aligned}$$

This proves Statement 1. For Statement 2, if $\widetilde{H}_i = 0$, then it is immediate. Otherwise, consider the inequalities:

$$\begin{aligned}
\widetilde{H}_j - \widetilde{H}_i &= H_j - (j-1)C/n - (H_i - (i-1)C/n) \\
&= H_j - H_i + ((i-1) - (j-1))C/n \\
&> (j-i)C/n + (i-j)C/n \\
&= 0
\end{aligned}$$

■

We state a trivial observation regarding fractions of positive numbers that will be useful in proving the lemmas below.

Observation 1. *For any positive numbers $a, b, c, d \in \mathbb{N}$:*

1. $\frac{a}{b} < \frac{c}{d} \Rightarrow \frac{a}{b} < \frac{a+c}{b+d} < \frac{c}{d}$
2. $\frac{a}{b} = \frac{c}{d} \Rightarrow \frac{a}{b} = \frac{a+c}{b+d} = \frac{c}{d}$

Lemma B.8. *Let x be any round, and suppose that at the outset of the round there is some family $\mathcal{F}_\alpha^x = \{N_i, N_{i+1}, \ldots, N_j\}$. Then the following statements are all true at the outset of round x:*

$$1) \textit{ For any } i \leq k < j: \quad \frac{\sum_{m=i}^{k} \widetilde{H}_m}{k-i+1} \;\geq\; \langle \widetilde{H}_{\mathcal{F}_\alpha} \rangle \geq \; \frac{\sum_{m=k+1}^{j} \widetilde{H}_m}{j-k}$$

$$2) \textit{ For any } j < k \leq n-2: \quad \langle \widetilde{H}_{\mathcal{F}_\alpha} \rangle \;<\; \frac{\sum_{m=j+1}^{k} \widetilde{H}_m}{k-j}$$

$$3) \; \langle \widetilde{H}_{\mathcal{F}_\alpha} \rangle < \langle \widetilde{H}_{\mathcal{F}_{\alpha+1}} \rangle$$

Proof. The fact that $\frac{\sum_{m=i}^{k}\widetilde{H}_m}{k-i+1} \geq \frac{\sum_{m=k+1}^{j}\widetilde{H}_m}{j-k}$ follows immediately from Observation 1 together with the rules regarding the construction of families (see Rule F1 from the previous section), and in particular the fact that indices are found by *minimizing* (35). Statement 1 now follows from Observation 1. Statement 2 also follows immediately from Rule F1 and Observation 1, and Statement 3 follows immediately from Statement 2. ■

Statement 3 of Lemma B.8 can be immediately extended:

Corollary B.9. *Let x be any round, and suppose that at the outset of the round there are l families. Then:*

$$\langle \widetilde{H}_{\mathcal{F}_1} \rangle < \langle \widetilde{H}_{\mathcal{F}_2} \rangle < \cdots < \langle \widetilde{H}_{\mathcal{F}_l} \rangle$$

Lemma B.10. *Let x be any round, and suppose that at the outset of the round there is some family $\mathcal{F}_\alpha^x = \{N_i, N_{i+1}, \ldots, N_j\}$. Then:*

$$\text{For any } 1 \leq k < i: \quad \frac{\sum_{m=k}^{i-1}\widetilde{H}_m}{i-k} \; < \; \langle \widetilde{H}_{\mathcal{F}_\alpha} \rangle \tag{41}$$

Proof. Since $k < i$, necessarily N_k is in some family $\mathcal{F}_\beta$ with index $\beta < \alpha$. Then:

$$\frac{\sum_{m=k}^{i-1}\widetilde{H}_m}{i-k} \; \leq \; \langle \widetilde{H}_{\mathcal{F}_\beta} \rangle \; < \; \langle \widetilde{H}_{\mathcal{F}_{\beta+1}} \rangle \; < \; \ldots \; < \; \langle \widetilde{H}_{\mathcal{F}_{\alpha-1}} \rangle \; < \; \langle \widetilde{H}_{\mathcal{F}_\alpha} \rangle, \tag{42}$$

where the first inequality is from Statement 1 of Lemma B.8 and the other inequalities are from Corollary B.9. ■

Lemma B.11. *If at the start of some round x we have that $\widetilde{H}_{j+1}^x \leq \widetilde{H}_j^x$, then N_j and N_{j+1} are in the same family at the start of round x.*

Proof. Suppose for the sake of contradiction that they are not in the same family at the start of round x. Let $\mathcal{F}^x$ denote N_j's family at the start of the round. By Lemma B.6 and the fact that j and $j+1$ are adjacent indices, we must have that $\mathcal{F}^x = \{N_i, N_{i+1}, \ldots, N_j\}$ for some $i \leq j$. The key observation is that:

$$\frac{\widetilde{H}_{j+1}}{1} \leq \frac{\widetilde{H}_j}{1} \quad \Rightarrow \quad \frac{\widetilde{H}_{j+1}}{1} \leq \frac{\widetilde{H}_{j+1} + \widetilde{H}_j}{2} \leq \frac{\widetilde{H}_j}{1} \tag{43}$$

If $i = j$, then (43) contradicts Statement 2 of Lemma B.8 (set $k = j+1$). If $i < j$, then define:

$$A := \sum_{l=i}^{j-1} \widetilde{H}_l \qquad \text{and} \qquad B := j - i \tag{44}$$

Then by Lemma B.8:

$$\frac{\widetilde{H}_{j+1}}{1} \leq \frac{\widetilde{H}_j}{1} \leq \frac{A}{B} \quad \Rightarrow \quad \frac{\widetilde{H}_{j+1} + \widetilde{H}_j + A}{B+2} \leq \frac{\widetilde{H}_j + A}{B+1} = \langle \widetilde{H}_{\mathcal{F}} \rangle, \tag{45}$$

which contradicts Statement 1 of Lemma B.8. ■

Lemma B.12. *If at the outset of any round x, we have that $|H_i^x - H_j^x| \leq C/n$ for any pair of nodes N_i^x and N_j^x, then necessarily the nodes are in the same family at the start of round x.*

Proof. Suppose for the sake of contradiction that there exists some round x and some pair of nodes N_i^x and N_j^x for which $|H_i^x - H_j^x| \leq C/n$, but these nodes are in different families. Since families consist of adjacent indices (Lemma B.6) and nodes are indexed according to their heights at the start of the round, we may assume without loss of generality that i and j are adjacent (i.e. that $j = i+1$). By definition of indexing, we must have $H_i \leq H_{i+1}$, which combined with the hypothesis of the lemma implies that $H_{i+1} - C/n \leq H_i$. But then $\widetilde{H}_i \geq \widetilde{H}_{i+1}$ by Lemma B.7, and then N_i^x and N_{i+1}^x in different families contradicts Lemma B.11. ■

B.4 Lemmas Regarding the Re-structuring of Families

In this section, we discuss all possible changes between how families are arranged at the beginning of one round and the next.

Lemma B.13. *Families can only re-align during rounds $E(N_a, N_b)$ during which there is a packet transfer in $\mathcal{P}$ from N_a to N_b.*

Proof. This is immediate from the rules regarding constructing families, since the values of $\{\widetilde{H}_i\}$ (34) can only change if there is a packet transfer in $\mathcal{P}$, and thus the analysis in Rule F1 (35) will not change if there has been no packet transfer in $\mathcal{P}$. ■

Lemma B.14. *Suppose that in some round $x = E(N_a, N_b)$, the Slide protocol transfers a packet from N_a to N_b. Let $\mathcal{F}_\alpha := \{N_e, \ldots, N_a, \ldots, N_f\}$ denote N_a's family at the start of round x ($e \leq a \leq f$), and $\mathcal{F}_\beta := \{N_c, \ldots, N_b, \ldots, N_d\}$ denote N_b's family*[17] *at the start of x ($c \leq b \leq d$). The following describes all possible changes to the way families are organized between the start of round x and the next round:*

Case 1: $\widetilde{H}_a$ and $\widetilde{H}_b$ do not change. *Then the families at the start of round $x+1$ are identical the arrangement of families at the start of x.*

Case 2: $\widetilde{H}_a$ does not change, and $\widetilde{H}_b$ increases by one. *Then:*

- *(a) Families $\mathcal{F}_\delta$ to the left of $\mathcal{F}_\beta$ (i.e. $\delta < \beta$) do not change*
- *(b) For any node N_m with $b \leq m \leq d$, N_m will be in the same family as N_b at the start of round $x+1$*
- *(c) For any node N_m with $d < m$, letting $\mathcal{F}_\mu^x$ denote N_m's family at the start of round x, one of the following happens:*
  - *i. $\mathcal{F}_\mu^x$ does not change*
  - *ii. Every node in $\mathcal{F}_\mu^x$ is in the same family as N_b at the start of $x+1$*

Case 3: $\widetilde{H}_a$ decreases by one, and $\widetilde{H}_b$ does not change. *Then:*

- *(a) Families $\mathcal{F}_\delta$ to the right of $\mathcal{F}_\alpha$ (i.e. $\delta > \alpha$) do not change*

[17] Note that necessarily $\beta \leq \alpha$, as if both N_a and N_b are internal nodes, then Rule 3 of the Slide protocol (together with the definition of how nodes are indexed) guarantees that $b < a$, and then $\beta \leq \alpha$ by Lemma B.6. If N_a is the Sender and/or N_b is the Receiver, then $\beta \leq \alpha$ comes from our choice to denote the Sender's family by $\mathcal{F}_n$ and the Receiver's family by $\mathcal{F}_0$ (see Rule F3 regarding the formation of families).

(b) For any node N_m with $e \leq m \leq a$, N_m will be in the same family as N_a at the start of round $x+1$

(c) For any node N_m with $m < e$, letting $\mathcal{F}^x_\mu$ denote N_m's family at the start of round x, one of the following happens:

i. $\mathcal{F}^x_\mu$ does not change

ii. Every node in $\mathcal{F}^x_\mu$ is in the same family as N_a at the start of $x+1$

<u>CASE 4: $\widetilde{H}_a$ DECREASES BY ONE, AND $\widetilde{H}_b$ INCREASES BY ONE.</u> *Then:*

(a) Families $\mathcal{F}_\delta$ to the right of $\mathcal{F}_\alpha$ (i.e. $\delta > \alpha$) and to the left of $\mathcal{F}_\beta$ (i.e. $\delta < \beta$) do not change

(b) For any node N_m with $e \leq m \leq a$, N_m will be in the same family as N_a at the start of round $x+1$

(c) For any node N_m with $b \leq m \leq d$, N_m will be in the same family as N_b at the start of round $x+1$

(d) For any node N_m with $d < m < e$, letting $\mathcal{F}^x_\mu$ denote N_m's family at the start of round x, one of the following happens:

i. $\mathcal{F}^x_\mu$ does not change

ii. Every node in $\mathcal{F}^x_\mu$ is in the same family as N_a at the start of $x+1$

iii. Every node in $\mathcal{F}^x_\mu$ is in the same family as N_b at the start of $x+1$

iv. Every node in $\mathcal{F}^x_\mu$ is in the same family as N_a AND N_b at the start of $x+1$

Proof. That the four cases stated in the lemma cover all possibilities is immediate from the definition of effective height $\widetilde{H}$ (see Definition (34)). Case 1 follows immediately from the rules F1-F2 for forming families (see Definition B.1) since the effective heights have not changed. We go through each of the other cases, and prove each Statement.

Suppose that we are in Case 2, so that $\widetilde{H}_a$ does not change, and $\widetilde{H}_b$ increases by one. For $\delta < \beta$, consider a family $\mathcal{F}_\delta := \{N_i, \ldots, N_j\}$, and for the sake of contradiction, suppose that $\mathcal{F}_\delta$ changes in some way from the start of round x to the start of round $x+1$. Without loss of generality, we will suppose that $\delta < \beta$ is the *minimal* index for which $\mathcal{F}_\delta$ changes.

Case A: $\mathcal{F}_\delta$ *Splits.* In other words, N_i and N_j are *not* in the same family at the start of round $x+1$. Let $\mathcal{F}^{x+1}_\iota := \{N_i, \ldots, N_k\}$ denote N_i's new family at the start of $x+1$, where $k < j$ by assumption.[18] Notice that for all $i \leq m \leq j$, the effective height $\widetilde{H}_m$ will not change between the start of x and $x+1$ (since $j < b < a$). Therefore:

$$\frac{\sum_{l=k+1}^{j} \widetilde{H}_l}{j-k} \leq \frac{\sum_{l=i}^{k} \widetilde{H}_l}{k-i+1} = \langle \widetilde{H}_{\mathcal{F}^{x+1}_\iota} \rangle < \frac{\sum_{l=k+1}^{j} \widetilde{H}_l}{j-k}, \tag{46}$$

where the first inequality is Statement 1 of Lemma B.8 and the last inequality is Statement 2 of Lemma B.8. Clearly (46) is impossible, yielding the desired contradiction.

[18]Necessarily N_i is the smallest-indexed node in $\mathcal{F}_\iota$ by our choice of minimality for δ.

Case B: $\mathcal{F}_\delta$ *Grows.* In other words, at the start of round $x+1$ there is some family $\mathcal{F}_\iota^{x+1} := \{N_i, \dots, N_k\}$ for $k > j$. If $k < b$, then for all $i \le m \le k$, the effective height $\widetilde{H}_m$ will not change between the start of x and $x+1$, so:

$$\frac{\sum_{l=i}^{j} \widetilde{H}_l}{j-i+1} < \frac{\sum_{l=j+1}^{k} \widetilde{H}_l}{k-j} \le \frac{\sum_{l=i}^{j} \widetilde{H}_l}{j-i+1}, \tag{47}$$

where the first inequality is Statement 2 of Lemma B.8 and the second inequality is Statement 1 of Lemma B.8. Clearly (47) is impossible, yielding the desired contradiction. On the other hand, if $k \ge b$, then for all $i \le m \le k$ and $m \ne b$, the effective height $\widetilde{H}_m$ will not change between the start of x and $x+1$, but the effective height $\widetilde{H}_b$ increases by one from the start of x and $x+1$. Therefore (using superscripts only when necessary to specify the round):

$$\frac{\sum_{l=i}^{j} \widetilde{H}_l}{j-i+1} < \frac{\sum_{l=j+1}^{k} \widetilde{H}_l^x}{k-j} < \frac{\sum_{l=j+1}^{k} \widetilde{H}_l^{x+1}}{k-j} \le \frac{\sum_{l=i}^{j} \widetilde{H}_l}{j-i+1}, \tag{48}$$

where the first inequality is Statement 2 of Lemma B.8 and the last inequality is Statement 1 of Lemma B.8. Clearly (48) is impossible, yielding the desired contradiction.

This proves Statement (a) of Case 2. For Statement (b), fix index $m \in [b, d]$ (Statement (b) is trivially true for $m = b$, so assume $b < m \le d$). For the sake of contradiction, suppose that N_m is *not* in the same family as N_b at the start of $x+1$. Let $\mathcal{F}_\beta^{x+1} := \{N_i, \dots, N_b, \dots, N_j\}$ denote N_b's new family at the start of $x+1$, so by assumption $j < m \le d$, and also $c \le i$ by Statement (a) of Case 2. Notice that $\widetilde{H}_b^x + 1 = \widetilde{H}_b^{x+1}$, but that for all other $i \le l \le m$, $\widetilde{H}_l$ does not change from the start of x and $x+1$. If $i = c$ (using superscripts only when necessary to specify the round):

$$\frac{\sum_{l=j+1}^{d} \widetilde{H}_l}{d-j} \le \frac{\sum_{l=c}^{j} \widetilde{H}_l^x}{j-c+1} < \frac{\sum_{l=c}^{j} \widetilde{H}_l^{x+1}}{j-c+1} < \frac{\sum_{l=j+1}^{d} \widetilde{H}_l}{d-j}, \tag{49}$$

where the first inequality is Statement 1 of Lemma B.8 and the last inequality is Statement 2 of Lemma B.8. Clearly (49) is impossible, yielding the desired contradiction. If on the other hand $c < i$, then (using superscripts only when necessary to specify the round):

$$\begin{aligned}
\frac{\sum_{l=j+1}^{d} \widetilde{H}_l}{d-j} &\le \frac{\sum_{l=c}^{j} \widetilde{H}_l^x}{j-c+1} = \langle \widetilde{H}_{\mathcal{F}_\beta^x} \rangle \\
&\le \frac{\sum_{l=c}^{i-1} \widetilde{H}_l^x}{i-c} \\
&< \frac{\sum_{l=c}^{i-1} \widetilde{H}_l^{x+1}}{i-c} \\
&< \langle \widetilde{H}_{\mathcal{F}_\iota^{x+1}} \rangle = \frac{\sum_{l=i}^{j} \widetilde{H}_l^{x+1}}{j-i+1} \\
&< \frac{\sum_{l=j+1}^{d} \widetilde{H}_l}{d-j},
\end{aligned} \tag{50}$$

where the first and second inequalities are both Statement 1 of Lemma B.8, the fourth inequality is Lemma B.10, and the last inequality is Statement 2 of Lemma B.8. Clearly (50) is impossible, yielding the desired contradiction.

This proves Statement (b) of Case 2. It remains to prove Statement (c). Fix some $m > d$, and let $\mathcal{F}^w_\mu = \{N_w, \ldots, N_m, \ldots, N_y\}$ denote N_m's family at the start of x. We prove Statement (c) via the following two subclaims:

Subclaim 1. *$\mathcal{F}_\mu$ does not* Split. *In other words, N_w and N_y will be in the same family at the start of round $x+1$.*

Proof. Suppose not. Let $\mathcal{F}^{x+1}_\omega = \{N_i, \ldots, N_w, \ldots, N_j\}$ denote N_w's family at the start of round $x+1$, so $c \leq i \leq w \leq j < y$ (where the first inequality is due to Statement (a)). Notice that for every $i \leq l \leq y$, the only possible effective height $\widetilde{H}_l$ that can possibly change in round x is for $l = b$, in which case $\widetilde{H}^x_b + 1 = \widetilde{H}^{x+1}_b$. If $i = w$, then (using superscripts only when necessary to specify the round):

$$\frac{\sum_{l=w}^{j} \widetilde{H}_l}{j-w+1} < \frac{\sum_{l=j+1}^{y} \widetilde{H}_l}{y-j} \leq \frac{\sum_{l=w}^{j} \widetilde{H}_l}{j-w+1}, \tag{51}$$

where the first inequality is Statement 2 of Lemma B.8 and the second is Statement 1 of Lemma B.8. Clearly, (51) is impossible, yielding the desired contradiction. If on the other hand $i < w$, then (using superscripts only when necessary to specify the round):

$$\frac{\sum_{l=w}^{j} \widetilde{H}_l}{j-w+1} \leq \frac{\sum_{l=i}^{w-1} \widetilde{H}^{x+1}_l + \sum_{l=w}^{j} \widetilde{H}^x_l}{j-i+1} \leq \frac{\sum_{l=j+1}^{y} \widetilde{H}_l}{y-j} \leq \frac{\sum_{l=w}^{j} \widetilde{H}_l}{j-w+1}, \tag{52}$$

where the second inequality is Statement 2 of Lemma B.8, the third is Statement 1 of Lemma B.8, and the first comes from:

$$\frac{\sum_{l=w}^{j} \widetilde{H}_l}{j-w+1} \leq \frac{\sum_{l=i}^{w-1} \widetilde{H}^{x+1}_l}{w-i} \quad \Rightarrow \quad \frac{\sum_{l=w}^{j} \widetilde{H}_l}{j-w+1} \leq \frac{\sum_{l=i}^{w-1} \widetilde{H}^{x+1}_l + \sum_{l=w}^{j} \widetilde{H}^x_l}{j-i+1}, \tag{53}$$

where the first inequality is Statement 1 of Lemma B.8. Clearly, (52) is impossible, yielding the desired contradiction.

Subclaim 2. *If $\mathcal{F}_\mu$ gets larger, then necessarily N_b will be in the same family as N_w and N_y at the start of round $x+1$.*

Proof. Suppose not. Let $\mathcal{F}^{x+1}_\omega = \{N_i, \ldots, N_w, \ldots, N_j\}$ denote N_w's family at the start of round $x+1$, so $b < i \leq w \leq y \leq j$. Notice that for every $i \leq l \leq y$, since $b < i$, the effective height $\widetilde{H}_l$ does not change. If $i = w$, then since we are assuming $\mathcal{F}_\mu$ grows, we have $j > y$, and:

$$\frac{\sum_{l=w}^{y} \widetilde{H}_l}{y-w+1} < \frac{\sum_{l=y+1}^{j} \widetilde{H}_l}{j-y} \leq \frac{\sum_{l=w}^{y} \widetilde{H}_l}{y-w+1}, \tag{54}$$

where the first inequality is Statement 2 of Lemma B.8 and the second is Statement 1 of Lemma B.8. Clearly, (54) is impossible, yielding the desired contradiction. If on the other hand $i < w$ and $j > y$, then:

$$\frac{\sum_{l=i}^{w-1} \widetilde{H}_l}{w-i} < \frac{\sum_{l=w}^{y} \widetilde{H}_l}{y-w+1} < \frac{\sum_{l=y+1}^{j} \widetilde{H}_l}{j-y}, \tag{55}$$

where the first inequality is from Lemma B.10, and the second is from Statement 1 of Lemma B.8. But then (55) implies:

$$\frac{\sum_{l=i}^{w-1} \widetilde{H}_l + \sum_{l=w}^{y} \widetilde{H}_l^x}{y-i+1} < \frac{\sum_{l=y+1}^{j} \widetilde{H}_l}{j-y}, \tag{56}$$

which contradicts Statement 1 of Lemma B.8. Finally, if $i < w$ and $j = y$, then:

$$\frac{\sum_{l=i}^{w-1} \widetilde{H}_l}{w-i} < \frac{\sum_{l=w}^{y} \widetilde{H}_l}{y-w+1}, \tag{57}$$

which contradicts Statement 1 of Lemma B.8.

Cases 3 and 4 follow analogous arguments. ∎

B.5 Statement and Proof of Fact that Slide has Competitive Ratio $1/n$

Lemma B.15. *Suppose at the start of round x, there exists nodes $\{N_i^x, N_{i+1}^x, \dots, N_j^x\}$ such that $H_i^x = \dots = H_j^x$. Then under any permutation of the indices $\sigma : \{i, i+1, \dots, j\} \to \{i, i+1, \dots, j\}$, we have that:*

$$\sum_{k=i}^{j} \widetilde{H}_k^x = \sum_{k=i}^{j} \max(0, H_k^x - (k-1)C/n) = \sum_{k=i}^{j} \max(0, H_{\sigma(k)}^x - (k-1)C/n) \tag{58}$$

In particular, the value for $\sum_{k=i}^{j} \widetilde{H}_k^x$ will not change if we re-index the nodes $\{N_i, \dots, N_j\}$ in any arbitrary manner.

Proof. This is immediate from the hypothesis that $H_i^x = H_{i+1}^x = \dots = H_j^x$. ∎

Lemma B.16. *Suppose that in some round x, N_a transfers a packet to N_b in the Slide protocol. Let $\mathcal{F}_\beta$ denote N_b's family and $\mathcal{F}_\alpha$ denote N_a's family. Then either there is exactly one node $N_{b'} \in \mathcal{F}_\beta$ such that $\varphi_{b'}$ increases by one, or φ_N does not change for every $N \in \mathcal{F}_\beta$. Similarly, either there is exactly one node $N_{a'} \in \mathcal{F}_\alpha$ such that $\varphi_{a'}$ decreases by one, or φ_N does not change for every $N \in \mathcal{F}_\alpha$. No other node $N \in G$ will have φ_N change as a result of this packet transfer.*

Proof. If N_b's effectual height $\widetilde{H}_b$ does *not* increase as a result of the packet transfer (e.g. the '0' in the maximum statement of (34) dominates), then $\mathcal{F}_\beta$'s cumulative effectual height does not change, and as a result, the potential φ of all nodes in $\mathcal{F}_\beta$ remains unchanged. If on the other hand B's effectual height does increase, then this will raise the cumulative effectual height $\widetilde{H}_{\mathcal{F}_\beta}$ by one, and this will be absorbed by some node in $\mathcal{F}^-$. A similar argument works with respect to N_a in $\mathcal{F}_\alpha$. The last statement of the lemma follows from Lemma B.4. ∎

We are now ready to prove the main lemma that will allow us to argue that the Slide protocol has competitive ratio $1/n$. To fix notation, for any internal node N, let $H_N^{\mathcal{P}'}$ denote the number of packets $p' \in Z_2^{\mathcal{P}'}$ that N is currently storing. Recall the definition of Φ (see Definition B.5); we will distribute the overall potential Φ between all the families, and show that with the rules regarding changes in Φ, the potential of a family is always positive. Namely:

Lemma B.17. *For every round x and for all families $\mathcal{F}$ that are present at the start of x:*

$$\Phi \ \geq \ \sum_{\mathcal{F}} \max\left(\sum_{N \in \mathcal{F}^-} C - H_N^{\mathcal{P}'}, \quad \sum_{N \in \mathcal{F}^+} H_N^{\mathcal{P}'} \right) \ \geq \ 0 \tag{59}$$

Proof. We prove this based on induction on the round x. The lemma is clearly true at the outset of the protocol, when $\Phi = \Phi_{\mathcal{F}} = 0$, and all nodes are in the same family, since all nodes have height zero. Suppose that at the start of round $x = E(N_a, N_b)$, (59) is satisfied. We show that no matter what happens in round x, (59) will remain satisfied at the start of round $x+1$.

Case 1: Neither $\mathcal{P}$ nor $\mathcal{P}'$ transfer a packet. In this case, families will not change (Lemma B.13), and no packets in $Z_2^{\mathcal{P}'}$ move, so there will be no changes to either side of (59).

Case 2: $\mathcal{P}'$ transfers a packet during x, but $\mathcal{P}$ does not. If the packet p' transferred by $\mathcal{P}'$ is in $Z_1^{\mathcal{P}'}$, then neither side of (59) will change. So suppose $p' \in Z_2^{\mathcal{P}'}$. Note that in Case 1, N_a and N_b are in the same family, call it $\mathcal{F}$ (Since Slide does not transfer a packet, we have $|H_a - H_b| < C/n$, and see Lemma B.12).

- If N_a and N_b are in $\mathcal{F}^+$, then $\varphi_a = \varphi_b$, so $\varphi_{p'}$ does not change. In particular, neither side of (59) changes in this case. The same is true if N_a and N_b are both in $\mathcal{F}^-$
- If $N_a \in \mathcal{F}^+$ and $N_b \in \mathcal{F}^-$, then the change on the left-hand side of (59) is -1 (since $\Delta\varphi_{p'} = -1$), which matches the change on the right-hand side of (59) (since $H_b^{\mathcal{P}'}$ increases by one, and $H_a^{\mathcal{P}'}$ decreases by one). If instead $N_a \in \mathcal{F}^-$ and $N_b \in \mathcal{F}^+$, then similar reasoning shows that the change of both sides of (59) is $+1$.

Case 3: $\mathcal{P}$ transfers a packet from N_a to N_b in round x. Notice that this case is not concerned with whether or not $\mathcal{P}'$ also transfers a packet, as such a packet would necessarily be in $Z_1^{\mathcal{P}'}$ (by definition), and hence this packet movement in $\mathcal{P}'$ will not affect either side of (59). Also, without loss of generality N_a is the sending node and N_b is the receiving node. By Lemma B.14, there are 4 cases we must consider:

Case 3A: $\widetilde{H}_b$ and $\widetilde{H}_a$ do not change. Then by Lemma B.14, there will be no re-structuring of families between rounds x and $x+1$. Consequently, if $\mathcal{F}_\beta$ denotes N_b's family and $\mathcal{F}_\alpha$ denotes N_a's family (possible $\alpha = \beta$), then for all other families, (59) will remain valid. Also, φ_N does not change for any $N \in \mathcal{F}_\beta$ (similarly for $N \in \mathcal{F}_\alpha$) since $\widetilde{H}_b$ and $\widetilde{H}_a$ do not change. Therefore, the right-hand side of (59) also will not change for $\mathcal{F}_\beta$ and $\mathcal{F}_\alpha$, and the only change in the left-hand side comes from the increase of $4C$ to Φ (see Rule 2 of Definition B.5), which can be divided arbitrarily among the families $\{\mathcal{F}\}$, and this will only help (59).

Case 3B: $\widetilde{H}_b$ increases by one, but $\widetilde{H}_a$ does not change. Let $\mathcal{F}_\beta = \{N_c, \ldots, N_b, \ldots, N_d\}$ for some $c \leq b \leq d$. By Lemma B.14, there exist integers $r, s \geq 0$ and indices $\{k_1, \ldots, k_r\}$ and $\{l_1, \ldots, l_s\}$ such that $c \le k_1 < \cdots < k_r \le b \le d < l_1 < \cdots < l_s$ and:

Families at the start of x	Families at the start of x
$\mathcal{F}_\beta = \{N_c, \ldots, N_b, \ldots, N_d\}$	$\widehat{\mathcal{F}}_\beta = \{N_c, \ldots, N_{k_1-1}\}$
$\mathcal{F}_{\beta+1} = \{N_{d+1}, \ldots, N_{l_1-1}\}$	$\widehat{\mathcal{F}}_{\beta+1} = \{N_{k_1}, \ldots, N_{k_2-1}\}$
$\mathcal{F}_{\beta+2} = \{N_{l_1}, \ldots, N_{l_2-1}\}$	$\widehat{\mathcal{F}}_{\beta+2} = \{N_{k_2}, \ldots, N_{k_3-1}\}$
$\vdots$	$\vdots$
$\mathcal{F}_{\beta+s} = \{N_{l_{s-1}}, \ldots, N_{l_s-1}\}$	$\widehat{\mathcal{F}}_{\beta+r-1} = \{N_{k_{r-1}}, \ldots, N_{k_r-1}\}$
	$\widehat{\mathcal{F}}_{\beta+r} = \{N_{k_r}, \ldots, N_{l_s-1}\}$

and no other families change.

By Lemma B.16, there is only one node $N \in \mathcal{F}_\beta^-$ for which φ_N increases by one as a result of the packet transfer. Although $\mathcal{F}_\beta$ will change in the manner described by the table above, by Lemma B.4, the *number* of nodes $N \in G$ with $\varphi_N = \lfloor \langle \widetilde{H}_{\mathcal{F}_\beta} \rangle \rfloor$ (respectively $\varphi_N = \lfloor \langle \widetilde{H}_{\mathcal{F}_\beta} \rangle \rfloor$) will not change (aside from the single node N' for which $\varphi_{N'}$ increases by one, as guaranteed by Lemma B.16), although the specific nodes in $\mathcal{F}^+$ and $\mathcal{F}^-$ may vary. A simple computation ensures that the right-hand side of (59) changes in the exact same way as the left-hand side of (59) whenever any two nodes in $\mathcal{F}$ swap places (in $\mathcal{F}^+$ and $\mathcal{F}^-$). Therefore, we may assume without loss of generality that there is exactly one node $N' \in \mathcal{F}_\beta^-$ for which $\varphi_{N'}$ increases by one as a result of the packet transfer, and for all other nodes $N \in G$, φ_N does not change between the start of x and $x+1$.

For each $0 \le i \le r$ and $0 \le j \le s$, define the following quantities:

$$\begin{array}{l|l}
\underline{\text{Families at the start of } x} & \underline{\text{Families at the start of } x} \\
X_i = \sum_{N \in \widehat{\mathcal{F}}^-_{\beta+i}} (C - H_N^{\mathcal{P}'}) & \mathcal{X}_j = \sum_{N \in \mathcal{F}^-_{\beta+j}} (C - H_N^{\mathcal{P}'}) \\
Y_i = \sum_{N \in \widehat{\mathcal{F}}^+_{\beta+i}} H_N^{\mathcal{P}'} & \mathcal{Y}_j = \sum_{N \in \mathcal{F}^+_{\beta+j}} H_N^{\mathcal{P}'} \\
A_i = |\widehat{\mathcal{F}}^+_{\beta+i}| & \mathcal{A}_i = |\mathcal{F}^+_{\beta+i}| \\
B_i = |\widehat{\mathcal{F}}^-_{\beta+i}| & \mathcal{B}_i = |\mathcal{F}^-_{\beta+i}|
\end{array} \tag{60}$$

Also define $\mathcal{F}_* = \widehat{\mathcal{F}}_{\beta+r} \cup \mathcal{F}_\beta$, and:

$$\mu = \sum_{N \in \widehat{\mathcal{F}}_*^-} (C - H_N^{\mathcal{P}'}) \qquad \nu = \sum_{N \in \mathcal{F}_*^+} H_N^{\mathcal{P}'} \qquad \alpha = |\widehat{\mathcal{F}}_*^+| \quad \text{and} \quad \beta = |\mathcal{F}_*^-| \tag{61}$$

By the induction hypothesis, we have that at the start of round x:

$$\sum_{j=0}^{s} \Phi_{\mathcal{F}_{\beta+j}} \ge \sum_{j=0}^{s} \left(\frac{\mathcal{A}_j \mathcal{X}_j + \mathcal{B}_j \mathcal{Y}_j}{\mathcal{A}_j + \mathcal{B}_j} \right) \tag{62}$$

In addition to the above potential, we also have that Φ increases by $4C$ as a result of the packet transfer in Slide. Meanwhile, the goal is to show that at the start of round $x+1$:

$$\sum_{i=0}^{r} \Phi_{\widehat{\mathcal{F}}_{\beta+i}} \ge \sum_{i=0}^{r} \left(\frac{A_i X_i + B_i Y_i}{A_i + B_i} \right) \tag{63}$$

Putting all these facts together, we want to show that:

$$4C + \sum_{j=0}^{s} \left(\frac{\mathcal{A}_j \mathcal{X}_j + \mathcal{B}_j \mathcal{Y}_j}{\mathcal{A}_j + \mathcal{B}_j} \right) \ge \sum_{i=0}^{r} \left(\frac{A_i X_i + B_i Y_i}{A_i + B_i} \right) \tag{64}$$

We demonstrate in the remainder of the proof how to show (64) is satisfied.

First look at the term $i = r$ for the right-hand side of (64):

$$
\begin{aligned}
\frac{A_r X_r + B_r Y_r}{A_r + B_r} &= \frac{(\alpha + 1 + \sum_{j=1}^s \mathcal{A}_j)(\mu + \sum_{j=1}^s \mathcal{X}_j - (C - H_{N'}^{\mathcal{P}'}))}{A_r + B_r} \\
&\quad + \frac{(\beta - 1 + \sum_{j=1}^s \mathcal{B}_j)(\nu + H_{N'}^{\mathcal{P}'} + \sum_{j=1}^s \mathcal{Y}_j)}{A_r + B_r} \\
&= \frac{\alpha+1}{\alpha+\beta}(\mu - (C - H_{N'}^{\mathcal{P}'})) + \sum_{j=1}^s \mathcal{X}_j \frac{\mathcal{A}_j}{\mathcal{A}_j + \mathcal{B}_j} + \frac{\beta - 1}{\alpha+\beta}(\nu + H_{N'}^{\mathcal{P}'}) + \sum_{j=1}^s \mathcal{Y}_j \frac{\mathcal{B}_j}{\mathcal{A}_j + \mathcal{B}_j} \\
&\quad + (\mathcal{Y}_1 - \mathcal{X}_1)\left(\frac{\alpha \sum \mathcal{B}_j - \beta \sum \mathcal{A}_j}{(\alpha+\beta)(A_r + B_r)}\right) \\
&\quad + \cdots + (\mathcal{Y}_s - \mathcal{X}_s)\left(\frac{\mathcal{A}_s(\beta + \sum \mathcal{B}_j) - \mathcal{B}_s(\alpha + \sum \mathcal{A}_j)}{(\mathcal{A}_s + \mathcal{B}_s)(A_r + B_r)}\right) \\
&< C + \frac{\alpha+1}{\alpha+\beta}(\mu - (C - H_{N'}^{\mathcal{P}'})) + \sum_{j=1}^s \mathcal{X}_j \frac{\mathcal{A}_j}{\mathcal{A}_j + \mathcal{B}_j} + \frac{\beta - 1}{\alpha+\beta}(\nu + H_{N'}^{\mathcal{P}'}) + \sum_{j=1}^s \mathcal{Y}_j \frac{\mathcal{B}_j}{\mathcal{A}_j + \mathcal{B}_j}
\end{aligned}
$$

We have used above that (by Lemmas B.8 and Corollary B.9):

$$
\frac{\alpha}{\alpha+\beta} < \frac{\mathcal{A}_1}{\mathcal{A}_1 + \mathcal{B}_1} < \cdots < \frac{\mathcal{A}_s}{\mathcal{A}_s + \mathcal{B}_s} < \frac{1 + \alpha + \mathcal{A}_1 + \cdots + \mathcal{A}_s}{\alpha + \beta + \sum_{j=1}^s (\mathcal{A}_j + \mathcal{B}_j)} \tag{65}
$$

Meanwhile, we look at the left-hand side of (64) for the $j = 0$ term:

$$
\begin{aligned}
\frac{\mathcal{A}_0 \mathcal{X}_0 + \mathcal{B}_0 \mathcal{Y}_0}{\mathcal{A}_0 + \mathcal{B}_0} &= \frac{(\alpha + \sum_{i=0}^{r-1} A_i)(\mu + \sum_{i=0}^{r-1} X_i}{\mathcal{A}_0 + \mathcal{B}_0} \\
&\quad + \frac{(\beta + \sum_{i=0}^{r-1} B_i)(\nu + \sum_{i=0}^{r-1} Y_i)}{\mathcal{A}_0 + \mathcal{B}_0} \\
&\geq \mu\left(\frac{\alpha}{\alpha+\beta}\right) + \nu\left(\frac{\beta}{\alpha+\beta}\right) - \frac{\mu + \sum_{i=0}^{r-1} X_i}{\mathcal{A}_0 + \mathcal{B}_0} \\
&\quad + \sum_{i=0}^{r-1} \frac{A_i X_i + B_i Y_i}{A_i + B_i},
\end{aligned} \tag{66}
$$

where we have used for the inequality above:

$$
\frac{\mathcal{A}_0}{\mathcal{A}_0 + \mathcal{B}_0} < \frac{A_0}{A_0 + B_0} < \frac{A_1}{A_1 + B_1} < \cdots < \frac{A_{r-1}}{A_{r-1} + B_{r-1}} < \frac{1 + \alpha + \sum_{i=0}^{r-1} A_i}{\alpha + \beta + \sum_{i=0}^{r-1}(A_i + B_i)}, \tag{67}
$$

with the inequalities following from Lemma B.8 and Corollary B.9. Putting this all together, we have that:

$$
4C + \sum_{j=0}^s \left(\frac{\mathcal{A}_j \mathcal{X}_j + \mathcal{B}_j \mathcal{Y}_j}{\mathcal{A}_j + \mathcal{B}_j}\right) \geq \sum_{i=0}^r \left(\frac{A_i X_i + B_i Y_i}{A_i + B_i}\right)
$$

which is (64).

The other cases are proven similarly. ∎

We state as an immediate consequence the lemma we needed in the discussion of Section 4:

Lemma B.18. *At all times:*

$$|Z_2^{\mathcal{P}'}| \leq 2nY^{\mathcal{P}} \leq 2n|Z^{\mathcal{P}}| \; + \; 2n^2C \tag{68}$$

C Competitive Analysis of the Slide+ Protocol

C.1 Description of Slide+

Recall that we model an asynchronous network via a scheduling adversary that maintains a buffer of requests of the form (u, v, p), which is a request from node u to send packet p to node v. The scheduling adversary proceeds in a sequence of honored edges (called rounds), whereby we will mean the following when we talk about an edge $E(u, v)$ being honored by the adversary:

STEP 1. From its buffer of requests, the adversary selects one request of form (u, v, p) and delivers p to v, and also selects one request of form (v, u, p') and delivers p' to u. If there are no requests (u, v, p) (resp. (v, u, p')), then the adversary sets p (resp. p') to $\perp$.

STEP 2. Node u (resp. v) sends new requests to the adversary of form (u, v, p) (resp. (v, u, p')).

Note that the two above-mentioned actions take place *sequentially*, so that the requests queued to the adversary in Step 2 can depend on the packets received in Step 1, but requests formulated during Step 2 of some round $E(u, v)$ will not be delivered until edge $E(u, v)$ is honored again (at the earliest). Since nodes in the network only send/receive packets when they are at one end of an edge currently being honored, nodes will not do anything except when they are a part of an honored edge. Thus, in describing Slide+, we need only describe what a node u will do when it is part of an honored edge $E(u, v)$. Recall that C denotes the size of each node's memory[19], and for simplicity we will assume that $C/n \in \mathbb{N}$, and also for Slide+, we will require $C \geq 8n^2$.

Slide+ Protocol Description.
During honored edge $E(u, v)$, let $(v, u, (p', h'))$ denote the message that u receives from v in Step 1 of the round (via the scheduling adversary). Also, u has recorded the request $(u, v, (p, h))$ that it made during Step 2 of the previous round in which $E(u, v)$ was honored; note that v will be receiving this message during Step 1 of the current round.

1. If u is the Sender, then:
   (a) If $h < C$, then u deletes packet p from his input stream $\{p_1, p_2, \dots\}$ (and ignores the received packet p'), and then proceeds to Step (c).
   (b) If $h' \geq C$, then u keeps p (and ignores the received packet p'), and proceeds to Step (c).
   (c) The Sender finds the next packet $p_i \in \{p_1, p_2, \dots\}$ that has not been deleted and is not currently an outstanding request already sent to the adversary, and sends the request $(u, v, (p_i, C + \frac{C}{n} + n))$ to the adversary. Also, u will update the fact that the current message request sent to v is $(u, v, (p_i, C + \frac{C}{n} + n))$.

[19] For simplicity, we assume that all nodes have the same memory bound, although our argument can be readily extended to handle the more general case.

2. If u is the Receiver, then u sends the request $(u, v, (\perp, \frac{-C}{n} - 2n + 1))$ to the adversary. Meanwhile, if $p' \neq \perp$, then u stores/outputs p' as a packet successfully received.
3. If u is any internal node, then:
   (a) If $h \geq h' + (C/n + 2n)$, then u will ignore p', delete p and the "ghost packet associated to p" (see Step 3d below), and slide down any packets/ghost packets to fill any gaps created. Also, u will update his height $h = h - 1$, and proceed to Step 3d below.
   (b) If $h \leq h' - (C/n + 2n)$, then u will keep p, and also store p' in the stack location that u had been storing the "ghost packet" for p (see Step 3d below), deleting the ghost packet in the process. Also, u will update his height $h = h + 1$, and proceed to Step 3d below.
   (c) If $|h - h'| < C/n + 2n$, then u will ignore packet p' and keep p, but delete the "ghost packet" associated to p, and then proceed to Step 3d.
   (d) Node u will search its stack for the highest packet p'' (not including ghost packets) that it has not already committed in an outstanding request to the adversary. It then sends the request $(u, v, (p'', h))$ to the adversary. Additionally, u will create a "ghost packet associated to the packet/request p''" that it has just sent the adversary. This "ghost packet" will assume the first un-filled spot in u's memory stack. Finally, u will update the fact that the current message request sent to v is $(u, v, (p'', h))$.

In the following section, we will prove that the above routing rules are compatible with memory requirements (e.g. that Steps 3b and 3d do not require a node to store more than C (ghost) packets), as well as prove that Slide+ enjoys competitive ratio $1/n$.

C.2 Analysis of Slide+

Before providing the full details of the proof that Slide+ enjoys competitive ratio $1/n$, we will provide a brief high-level description of how the proof works. First, notice that the main technical challenge in moving from the semi-asynchronous model of Section 4 to the fully asynchronous model is that nodes can no longer make routing decisions based on *current* information. Indeed, the current state of a node may change drastically from the time it makes a request in Step 2 of some round $E(u, v)$ and the time the request is finally sent by the adversary in Step 1 of the next round in which $E(u, v)$ is honored. Since the Slide protocol uses the current height of a node to make routing decisions, the fact that the height of a node may change substantially between the time a packet request is made and the time the receiving node receives the packet is an issue that must be resolved.

The above described protocol handles this issue by allotting "ghost packets" in Step 3d (this will ensure there is always room to store a packet sent from an honest neighbor), as well as having nodes make routing decisions based on *old* height considerations. In particular, Steps 1-3 above dictate what u should do based on the height that u and v had during *the last time* $E(u, v)$ was honored. Therefore, although this information may have become outdated since the last time u and v communicated with each other, at least the decisions will be made *consistently*, both in the sense that the heights being compared are *synchronized* (i.e. they are from the same time as each other, although possible now out-dated), and in the sense that the nodes will know what the other will do in terms of whether or not it will keep the packet just sent/received. This last fact is crucial to prevent packet deletion and duplication from occurring.

The proof will follow the main structure of the proof provided for the semi-asynchronous Slide protocol, with one additional category to account for packet transferring decisions that were based on significantly outdated height information.

Theorem C.1. *The Slide+ protocol achieves competitive ratio $1/n$ in any distributed, asynchronous, bounded memory network with dynamic topology (and no minimal connectivity assumptions). More specifically, for any adversary/off-line protocol pair $(\mathcal{A}, \mathcal{P}')$, if $\mathcal{P}$ denotes the Slide+ protocol, C denotes the capacity (memory bound) of each node, and $Z_x^{\mathcal{P}}$ (resp. $Z_x^{\mathcal{P}'}$) denotes the number of packets received by protocol $\mathcal{P}$ (resp. $\mathcal{P}'$) as of round x, then for all rounds x:*

$$Z_x^{\mathcal{P}'} \leq 8nZ^{\mathcal{P}} + 8n^2C \tag{69}$$

Proof. Fix any adversary/off-line protocol pair $(\mathcal{A}, \mathcal{P}')$, and let $\mathcal{P}$ denote the Slide+ protocol and $Z_x^{\mathcal{P}}$ and $Z_x^{\mathcal{P}'}$ as in the statement of the theorem. Motivated by the proof in the semi-asynchronous setting, we imagine a virtual world in which the two protocols are run simultaneously in the same network. We split $Z_x^{\mathcal{P}'}$ into the following three subsets (we will henceforth suppress the index referencing the round x):

1. $Z_1^{\mathcal{P}'}$ consists of packets $p' \in Z^{\mathcal{P}'}$ for which there exists at least one round $E(u, v)$ such that both p' was transferred by $\mathcal{P}'$ *and* some packet p was transferred by $\mathcal{P}$.[20]

2. $Z_2^{\mathcal{P}'}$ consists of packets $p' \in Z^{\mathcal{P}'}$ that were *never* transferred alongside a packet in $\mathcal{P}$ as in 1 above, and such that *every time* p' was transferred between two nodes u and v during a round $E(u, v)$, the heights H and h that were used by u and v in determining whether to store/delete the packets delivered by the adversary during Step 1 of $E(u, v)$ (see protocol description above) were each within n of the current heights of u and v.

3. $Z_3^{\mathcal{P}'} = Z^{\mathcal{P}'} \setminus (Z_1^{\mathcal{P}'} \cup Z_2^{\mathcal{P}'})$.

Clearly, $|Z^{\mathcal{P}'}| = |Z_1^{\mathcal{P}'}| + |Z_2^{\mathcal{P}'}| + |Z_3^{\mathcal{P}'}|$, and hence the theorem follows from Lemmas C.3, C.4, and C.5 below. ■

We will need the following trivial observation, which follows immediately from the description of the Slide+ protocol in Section C.1.

Observation 2. *At all times, an internal node u has at most n ghost packets and at most n outstanding requests (one for each of its edges v).*

Proof. Rules 1(c) and 3(d) only allow a node to submit a single request for each round the node is part of an honored edge, and this request is then delivered by the adversary in Step 1 of the next round in which the edge is honored. Also, Rules 3(a-c) guarantee that the ghost packet corresponding to the current honored edge will be deleted before another one is created in Rule 3(d). ■

In order to bound $|Z_1^{\mathcal{P}'}|$, we will need to bound the number of times any packet p can be transferred by the Slide+ protocol. In the asynchronous Slide protocol of Section 4, we showed that any packet p could be transferred at most $2n$ times, as during every packet transfer in Slide, the packet must drop in height by at least $C/n - 1$. At first glance, it might seem that we cannot make the same argument in the fully asynchronous setting since the Slide+ protocol is making routing

[20] Note that we make no condition that the two packets traveled in the same direction.

decisions based on (potentially) outdated height information. However, the introduction of "ghost packets" will allow us to retain this quality. Indeed, the purpose of utilizing ghost packets is to anticipate future packet transfers and reserve spots in a node's memory stack at the appropriate height, allowing us to argue that even if nodes nodes are using out-dated height information, packets will still "flow downhill" from Sender to Receiver. This is captured in the following lemma.

Lemma C.2. *Let $Y_x^{\mathcal{P}}$ denote the the set of packets inserted by $\mathcal{P}$ as of round x. Also let $T_x^{\mathcal{P}}$ denote the set of packet transfers that have occurred in $\mathcal{P}$ as of round x. Then any packet in the Slide+ protocol is transferred at most $2n$ times.*[21] *In particular, $|T_x^{\mathcal{P}}| \leq 2n|Y_x^{\mathcal{P}}| \leq 2n(|Z_x^{\mathcal{P}}| + nC)$.*

Proof. We show that anytime a packet is transferred in the Slide+ protocol, the packet's height in the new buffer is necessarily at least $C/n - 4n$ *lower* than its height in the old buffer. Since packets only move within buffers when they are received or sent (or when they slide *down* as in 3(a)), and since[22] $2n(C/n - 4n) > C$, the lemma will follow. Fix a packet p, and consider a round $x = E(u, v)$ in which p is transferred from u to v. In particular, it must have been that the *previous* round $x' < x$ in which $E(u, v)$ was honored, u sent some request of form $(u, v, (p, h))$ to the adversary in Step 2. Notice that when u selected p to form a part of its request as in 3(d), since u had height h and u has at most $n - 1$ packets already committed as an outstanding request (Observation 2), p must have height at least $h - n$ in u's buffer. Meanwhile, let $(v, u, (p', h'))$ denote the request that v sent to the adversary in Step 2 of round x'. Notice that in 3(d), v reserved a position in its buffer (the "ghost packet"), into which p will be inserted when it is received in round x. Since the ghost packet is assigned the topmost unoccupied (by packet or ghost packet) position in v's buffer, we have that p will have height no bigger than $h' + n$. Therefore, p will drop in height by at least $(h - n) - (h' + n) = h - h' - 2n$ when it is transferred from u to v. Since the criterion for accepting a new packet (see 3(d)) demands that $h - h' \geq C/n - 2n$, we have that p will necessarily drop in height by at least $C/n - 4n$ when it is transferred. ∎

Notice that Lemma C.2 is valid *regardless* of how long a request $(u, v, (p, h))$ has been queued in the adversary's buffer, and also of how u and v's stacks may have changed in the meantime. We are now ready to state and prove the first requisite bound:

Lemma C.3. $|Z_1^{\mathcal{P}'}| \leq 2n|Z^{\mathcal{P}}| + 2n^2C$

Proof. By definition, $|Z_1^{\mathcal{P}'}| \leq |T^{\mathcal{P}}|$, and the latter is bounded by $2n|Z^{\mathcal{P}}| + 2n^2C$ by Lemma C.2. ∎

Lemma C.4. $|Z_2^{\mathcal{P}'}| \leq 2n|Z^{\mathcal{P}}| + 2n^2C$

Proof. This bound follows the same reasoning as the proof of Lemma B.18. Suppose that packet $p' \in Z_2^{\mathcal{P}'}$ is transferred by $\mathcal{P}'$ from u to v in round x. By definition of $Z_2^{\mathcal{P}'}$, Slide+ did not transfer a packet, and thus (with the notation as in Rule 3(d) for Slide+) $|h - h'| < C/n - 2n$. Also by definition of $Z_2^{\mathcal{P}'}$, we have that v's height in round x is within n of h', and u's height in round x is within n of h. Consequently, u's height in round x must be within C/n of v's height. Then if we define families the same way as in the proof for the semi-synchronous Slide protocol (see Section B), by Lemma B.12, u and v must be in the same family at the start of x. Indeed, all the lemmas and proofs of Section B will remain valid[23], and hence Lemma B.18, which states that $|Z_2^{\mathcal{P}'}| \leq 2n|Z^{\mathcal{P}}| + 2n^2C$, remains valid. ∎

[21] This matches the bound for the semi-asynchronous Slide protocol of Section 4.

[22] For Slide+, we have demanded that $C > 8n^2$.

[23] The only necessary modification is to consider the present definition of $Z_2^{\mathcal{P}'}$ instead of the one used in Section B

Lemma C.5. $|Z_3^{\mathcal{P}'}| \leq 4n|Z^{\mathcal{P}}| + 4n^2C$

Proof. Fix a packet $p' \in Z_3^{\mathcal{P}'}$. By definition of $Z_3^{\mathcal{P}'}$, there exists some round $x_{p'} = E(u,v)$ in which p' was transferred from u to v, where either u's height or v's height has changed by at least n since the previous round $x'_{p'} < x$ in which $E(u,v)$ was honored. Let $\mathcal{S}_{p'} \subseteq T^{\mathcal{P}}$ denote n of these packet transfers, where each packet transfer in $\mathcal{S}_{p'}$ corresponds to a packet sent (or received) by u (or v), and took place between $x'_{p'}$ and $x_{p'}$.

> **Observation.** For any packet transfer in Slide+, there are at most $2n$ packets $p' \in Z_3^{\mathcal{P}'}$ for which the packet transfer appears in $\mathcal{S}_{p'}$.
>
> *Proof.* Consider any round $x' = E(u,v)$ in which a packet is transferred from u to v by Slide+, and refer to this specific packet transfer as $t_{x'}$. Then for each edge of u and each edge of v and for any $p' \in Z_3^{\mathcal{P}'}$, there can be at most one round $x_{p'} > x'$ for which $t_{x'} \in \mathcal{S}_{p'}$. After all, once a given edge of u or v, say for example $E(u,w)$, transfers a packet $p' \in Z_3^{\mathcal{P}'}$ in round $x_{p'} > x'$, the heights of both u and w are updated, and there can never be another $p'' \in Z_3^{\mathcal{P}'}$ and later round $x_{p''} > x_{p'}$ such that $x_{p''} = E(u,w)$ and $t_{x'} \in \mathcal{S}_{p''}$. Therefore, $t_{x'}$ can appear in at most $2n$ sets of form $\mathcal{S}_{p'}$.

Since $|\mathcal{S}_{p'}| = n$ for each $p' \in Z_3^{\mathcal{P}'}$, we have that:

$$\sum_{p' \in Z_3^{\mathcal{P}'}} |\mathcal{S}_{p'}| = n|Z_3^{\mathcal{P}'}| \tag{70}$$

Now since for any given packet transfer $t_x \in T^{\mathcal{P}}$ there can be at most $2n$ different values of $p' \in Z_3^{\mathcal{P}'}$ such that $t_x \in \mathcal{S}_{p'}$, we have that:

$$\left| \bigcup_{p' \in Z_3^{\mathcal{P}'}} \mathcal{S}_{p'} \right| \geq \frac{n|Z_3^{\mathcal{P}'}|}{2n} \tag{71}$$

But $\cup_{p' \in Z_3^{\mathcal{P}'}} \mathcal{S}_{p'} \subseteq T^{\mathcal{P}}$, so:

$$|T^{\mathcal{P}}| \geq |\cup_{p' \in Z_3^{\mathcal{P}'}} \mathcal{S}_{p'}| \geq \frac{|Z_3^{\mathcal{P}'}|}{2} \tag{72}$$

In particular, $|Z_3^{\mathcal{P}'}| \leq 2|T^{\mathcal{P}}| \leq 4nZ^{\mathcal{P}} + 4n^2C$, where the second inequality is Lemma C.2. ∎

D Pseudo-Code and Proofs for Protocol Secure Against Malicious Adversary

D.1 Pseudo-Code

In this section we present pseudo-code for implementing our protocol that is secure against a coordinated attack of the edge-scheduling and node-controlling adversaries. Formal proofs of security, referring to line numbers of the pseudo-code of the following four figures, are in the next section.

Variable and Notation Definitions ## Each of the below variables are transmission dependent

$C =$ Capacity of each (internal) node's buffer (i.e. number of codeword packets a node can store)
$B =$ Capacity of each node to hold extraneous (broadcast) information
$D = \frac{4nC}{\lambda} =$ Number of packets per codeword
$EN =$ List of Eliminated nodes
$Y =$ Set of packets inserted by sender
$Z =$ Set of packets received by receiver
$P_{u,v} =$ Net no. of p's to cross $E(u,v)$
$[p]_{u,v} =$ Net no. p crossed $E(u,v)$
$\Phi_{u,v} =$ Net decrease in potential as a result of packet transfers from u to v
$\Phi_u =$ Total potential drop caused by packet transfers across *all* edges adjacent to u
$G_p =$ Ghost packet associated to packet p (See Figure Internal Node Create Next Request)
$H_u =$ Height of u's buffer; i.e. the number of codeword packets u is currently storing
$BB_u = u$'s Broadcast Buffer
$BL_u = u$'s version of the Blacklist
$DB_s =$ Sender's Data Buffer, used to store status report parcels that will help eliminate corrupt nodes

Figure 1: Definition of Variables

Routing Rules for Node $u \in G$

Input:
  $(v, u, (p', H'), (q'_1, q'_2), (\alpha', \sigma(\alpha')))$ ## Received From v (via $\mathcal{A}$)
  $(u, v, (p, H), (q_1, q_2), (\alpha, \sigma(\alpha)))$ ## *Previous* request sent to v (via $\mathcal{A}$)
DO:
  Process the parcel q'_1 as in Process Parcel below
  If $\alpha = \alpha'$, $\sigma(\alpha')$ is valid, and $v \notin (EN_u \cup BL_u)$
    If $u = s$ and Ready(v) is TRUE and $H' < C$: ## Insert Packet
      Delete p from input stream $\{p_1, p_2, \dots\}$
      Increase $\Phi_{s,v}$ by the amount indicated by α
      Increase $P_{s,v}$, $[p]_{s,v}$, and $|Y|$ by one
    Else If $u = r$ and Ready(v) is TRUE and $p' \neq \perp$: ## Receive Packet
      Store/output p' as a packet successfully received
      Increase $\Phi_{r,v}$ by the amount indicated by α
      Decrease $P_{r,v}$ and $[p]_{r,v}$ by one and increase $|Z|$ by one
    Else If $u \neq r, s$ and Ready(v) is TRUE and $H \geq H' + (C/n - 2n)$: ## Send Packet
      Delete p and G_p and *Slide* ## Slide down (ghost) packets to fill gaps
      Increase $\Phi_{u,v}$ and Φ_u by the amount indicated by α
      Increase $P_{u,v}$ and $[p]_{r,v}$ by one, and set $H_u = H_u - 1$
    Else If $u \neq r, s$ and Ready(v) is TRUE and $H \leq H' - (C/n - 2n)$: ## Receive Packet
      Store p' in location occupied by G_p
      Increase $\Phi_{u,v}$ and Φ_u by the amount indicated by α
      Decrease $P_{u,v}$ and $[p]_{u,v}$ by one and set $H_u = H_u + 1$
  Send to $\mathcal{A}$ the returned value of Create Next Request

Figure 2: Routing Rules

Process Parcel for Internal Nodes and Receiver u

Input:
  (q'_1, q'_2) ## Received From v (via $\mathcal{A}$)
DO:
  Store q'_2 in BB_u ## $q'_2 = \Phi_w$ for some w. Replace old value, provided new value is larger
  Add q'_1 to BB_u ## Also mark edge $E(u, v)$ as having transmitted this information
  If $q'_1 = \Omega_{\mathtt{T}}$
    Clear outgoing, incoming, BL_u, and BB_u (except status report parcels)
  Else If $q'_1 = w \notin EN_u$ denotes a node to eliminate
    Add q'_1 to EN_u
  Else If $q'_1 = w$ denotes a node to blacklist
    Add q'_1 to BL_u
    If $w = u$, Sign and Add $n - 1$ status report parcels to BB_u
      ## Find reason u was blacklisted from SoT. For each $v \in G$:
      ## if case F2, add $\Phi_{u,v}$, if case F3, add $P_{u,v}$, if case $(F4, p')$, add $[p']_{u,v}$
  If $u = r$ and q'_1 indicates $\mathtt{T} - 1$ failed due to F2:
    For each $v \in G$, add Φ_w to BB_r

Process Parcel for Sender

Input:
  q'_1 ## Received From v (via $\mathcal{A}$)
DO:
  Add q'_1 to DB_s
  If q'_1 is the last missing status report parcel for some $w \in BL_s$
    Remove w from BL_s, and add fact $w \notin BL$ to BB_s

Figure 3: Rules For Processing Broadcast Information

D.2 High-Level Proofs Ideas for Competitive Analysis of Throughput

In this section, we sketch the proof that our protocol is n-competitive, leaving the rigorous details to the next subsection. As was done for analysis of Slide and Slide+, we use competitive analysis to evaluate the throughput performance of our routing protocol. To this end, let $(\mathcal{A}, \mathcal{P}')$ denote an adversary/off-line protocol pair for which we compare our routing protocol $\mathcal{P}$.

Theorem D.1. *If at any time $\mathcal{P}'$ has received $\Theta(xn)$ messages, then $\mathcal{P}$ has received $\Omega((x - n^2))$ messages. Thus, if the number of messages $x \in \Omega(n^2)$, then our protocol has competitive ratio $1/n$, which is optimal.*

Proof. This follows as an immediate corollary to Lemmas D.3 and D.4 below. ∎

Lemma D.2. *If a transmission fails as in F2-F4, as soon as the sender receives all of the signed communications between all nodes, he will necessarily be able to identify a corrupt node.*

Proof. Intuitively, a transmission fails as in case F2 when a corrupt node is transferring packets against transfer rules (e.g. from *smaller* heights to *larger* heights, or when a corrupt node is duplicating packets). Both of these can be detected by looking at the node's communication with each of its (honest) neighbors, who have recorded the height differences caused by each packet transfer. If a transmission ends as in case F2, the sender will look for a node whose cumulative height drop is negative; this information is available through the **Sig. 3** signed communications (see above section).

When a transmission fails as in case F3, this means that there is a corrupt node that is deleting packets. The sender can identify such a node u when he has received each of the signed communications (**Sig. 1**) from each of u's (honest) neighbors. Finally, transmission failure as in case F4 means there is a corrupt node that has duplicated some packet p. The sender can identify such a node u when he has received each of the signed communications (**Sig. 2**, corresponding to the packet p) from each of u's (honest) neighbors.

This lemma is proved rigorously in Appendix D.5. ■

Lemma D.3. *After a corrupt node has been eliminated (or at the outset of the protocol) and before the next corrupt node is eliminated, there can be at most $n-1$ failed transmissions before the next node can be eliminated. In particular, there are at most n^2 failed transmissions.*

Proof. The intuition for the proof is that the **blacklist** forces corrupt nodes to return their signed communication to the sender if they want to further disrupt future transmissions. Then use Lemma D.2 above to show that with the signed communication, the sender can identify a corrupt node. A rigorous proof is provided in Appendix D.4. ■

Lemma D.4. *For every message/codeword transmission, by the time the transmission ends as a result of S1 or F2-F5, we have that the ideal offline protocol $\mathcal{P}'$ has received at most $O(n^2C)$ packets.*

We will need the following definition for the proof:

Definition D.5. A round $\mathtt{t} = E(u, v)$ of a transmission is wasted if u and v are honest nodes, and they were not allowed to transfer a packet because one (or both) of them was on the blacklist.

Proof Sketch of Lemma D.4. Let C' denote the number of packets per codeword.[24] The structure of the proof will be to show that if $\mathcal{P}'$ has received $3nC'$ packets as of some round $\mathtt{t}$, then necessarily S1 or F2-F5 has occurred. To do this, we follow the proof of the competitive ratio for Slide and Slide+ and imagine a virtual world in which $\mathcal{P}$ and $\mathcal{P}'$ are run simultaneously. Let $Z^{\mathcal{P}'}$ denote the packets delivered to the receiver by $\mathcal{P}'$, and let $Z_3^{\mathcal{P}'}$ denote the subset of packets that travelled between two nodes during a wasted round. Define $Z_1^{\mathcal{P}'}$ to be the subset of $Z^{\mathcal{P}'} \setminus Z_3^{\mathcal{P}'}$ consisting of packets p' for which there exists at least one round $E(u, v)$ such that both p' *and* some packet $p \in Y^{\mathcal{P}}$ were both transferred this round.[25] Set $Z_2^{\mathcal{P}'} = Z^{\mathcal{P}'} \setminus (Z_1^{\mathcal{P}'} \cup Z_3^{\mathcal{P}'})$. Also, let $T_{\mathtt{t}}^{\mathcal{P}}$ denote the number of packet transfers in $\mathcal{P}$ between two ***honest*** nodes as of round $\mathtt{t}$. We begin with the following observation, which is analogous to the corresponding statements for Slide and Slide+ (see e.g. Lemmas 4.1 and 4.2), and is proved in Appendix D.4:

Observation. $|Z_1^{\mathcal{P}'}| \le T_{\mathtt{t}}^{\mathcal{P}}, \qquad |Z_2^{\mathcal{P}'}| \le T_{\mathtt{t}}^{\mathcal{P}}, \quad$ and $\quad |Z_3^{\mathcal{P}'}| \le n^4 + 2n^3$

Notice that since $T_{\mathtt{t}}^{\mathcal{P}}$ only takes into account packet transfers between honest nodes, we have that $T_{\mathtt{t}}^{\mathcal{P}} \le Y^{\mathcal{P}} * C/(C/n) = nY^{\mathcal{P}}$, since every packet starts at height at most C and drops in height by at least $\approx C/n$ every time it is transferred. Therefore, the above observation together with the assumption that $3nC'$ packets have been received by $\mathcal{P}'$ say:

$$3nC' = |Z^{\mathcal{P}'}| = |Z_1^{\mathcal{P}'}| + |Z_2^{\mathcal{P}'}| + |Z_3^{\mathcal{P}'}| \le 2T_{\mathtt{t}}^{\mathcal{P}} + n^4 + 2n^3 \quad \Rightarrow \quad T_{\mathtt{t}}^{\mathcal{P}} \ge \lambda n^2 C^2 \tag{73}$$

[24] $C' = \lambda nC$ is a constant multiple of n times the buffer-size C (the constant λ depends on the error-correction rate).

[25] Note that we make no condition that the two packets traveled in the same direction.

where in the last inequality we have used $C' \geq n^3 + 2n^2$ and $C' = \lambda nC$. Since each packet transfer corresponds to a height difference of at least C/n between the honest nodes exchanging the packet, (73) implies that honest nodes will have recorded a cumulative height difference of $\lambda n^2 C^2$, which is precisely the condition for a transmission ending as in case F2. See Appendix D.3 for details. ∎

D.3 Proof of Lemma D.4

In this section, we prove the following lemma (which is a formal restatement of Lemma D.4). Before stating and proving this lemma, it will be convenient to introduce new terminology and fix notation:

Definition D.6. We will say a node $N \in G$ *participated* in transmission $\mathtt{T}$ if there was at least one round in the transmission for which w was not on the (sender's) blacklist. The sender's variable that keeps track of nodes participating in transmission $\mathtt{T}$ will be called the *participating list* for transmission $\mathtt{T}$, denoted by $\rho_{\mathtt{T}}$ (updated at the end of failed transmissions on line 30 of Figure 4).

Also, we will refer to specific line numbers for the pseudo-code via (**X.YY**), where **X** refers to the Figure number, and **YY** refers to the line number. Finally, let D denote the number of packets per codeword, and note that:

$$D = \frac{nC}{\lambda}, \tag{74}$$

where λ is the error-rate of the error-correcting code.

Lemma D.7. *In any transmission $\mathtt{T}$, $|Z_{\mathtt{T}}^{\mathcal{P}'}| \leq 3nD$. If the transmission was* successful *(i.e. r sent EoT parcel "S1" on* **4.14-15** *and* **4.20***), then $|Z_{\mathtt{T}}^{\mathcal{P}}| \geq (1-\lambda)D = O(nC)$.*

We will prove Lemma D.7 via a sequence of Lemmas. First, recall from Section 5 the reasons a transmission may fail:

- S1, F2, F4 Sender receives End of Transmission (EoT) parcel from the receiver (**4.25**, **4.28**)
- F3 Sender has inserted D packets since the end of $\mathtt{T}_{i-1}$ (**4.28**)
- F5 Sender receives enough information to eliminate a new corrupt node (**4.22**)

In order to prove Lemma D.7, we will show that if there is a transmission in which the ideal off-line protocol $\mathcal{P}'$ has received at least $3nD$ packets, then necessarily the sender had received the EoT parcel from R indicating "F2," a contradiction (the transmission should have ended). In other words, we show that if a transmission does not end as on (**4.22**) or (**4.28**), then necessarily the transmission will end as on (**4.25**) before $\mathcal{P}'$ is able to receive more than $3nD$ packets.

Lemma D.8. *If the receiver forms any EoT parcel in round $\mathtt{t}$ of some transmission and $\mathcal{P}'$ has inserted $Z = Z_{\mathtt{t}}^{\mathcal{P}'}$ packets at this point, then the sender will necessarily receive EoT before $\mathcal{P}'$ is able to receive $n^2C + nC$ more packets.*

Proof. We will show that there can be at most n^2C packet insertions by $\mathcal{P}'$ before the EoT parcel necessarily has reached the sender, from which the lemma follows since there can be at most nC packets in the buffers of the honest nodes at round $\mathtt{t}$. Thus, the lemma follows immediately from Lemma D.15 in Appendix D.4. ∎

By the above lemma, it remains to show that if at any time $\mathtt{t}$ we have that $|Z_{\mathtt{t}}^{\mathcal{P}'}| \geq 3nD - n^2C - nC$, then necessarily R will enter lines 16-17 of Figure 4. First, we will split $Z^{\mathcal{P}'}$ into three disjoint subsets $Z^{\mathcal{P}'} = Z_1^{\mathcal{P}'} \cup Z_2^{\mathcal{P}'} \cup Z_3^{\mathcal{P}'}$, which were described in Section 5, but are now restated in terms of the pseudo-code.

Definition D.9. We will say a round $\mathtt{t} = E(u, v)$ of a transmission is wasted if u and v are honest nodes, and $\mathsf{Ready}(u)$ returned false for v or $\mathsf{Ready}(v)$ returned false for u (see lines **2.15**, **2.17**, and **4.41-43**).

Intuitively, a round is wasted if two honest nodes would have transferred a packet (based on their relative heights), but they were not allowed to because they had not yet transmitted requisite broadcast information across $E(u, v)$, or because one was on the other's blacklist.

We can view the scheduling adversary $\mathcal{A}$ as simply a schedule (or order) of edges that the adversary will honor. We will imagine a virtual world, in which $\mathcal{P}$ and $\mathcal{P}'$ are run simultaneously. Let $Z_3^{\mathcal{P}'}$ denote the set of packets in $Z^{\mathcal{P}'}$ that travelled between two nodes during a wasted round. Define[26] $Z_1^{\mathcal{P}'}$ to be the subset of $Z^{\mathcal{P}'} \setminus Z_3^{\mathcal{P}'}$ consisting of packets p' for which there exists at least one round $E(u, v)$ such that both p' *and* some packet $p \in Y^{\mathcal{P}}$ were both transferred this round.[27] Set $Z_2^{\mathcal{P}'} = Z^{\mathcal{P}'} \setminus (Z_1^{\mathcal{P}'} \cup Z_3^{\mathcal{P}'})$. Also, let $T_{\mathtt{t}}^{\mathcal{P}}$ denote the number of packet transfers in $\mathcal{P}$ between two ***honest*** nodes (as on lines 15-22 of Figure 2) as of round $\mathtt{t}$.

Lemma D.10. *For any round* $\mathtt{t}$*:* $\quad |Z_{1,\mathtt{t}}^{\mathcal{P}'}| \leq T_{\mathtt{t}}^{\mathcal{P}}$ *and* $|Z_{2,\mathtt{t}}^{\mathcal{P}'}| \leq T_{\mathtt{t}}^{\mathcal{P}}$

Proof. These are Lemmas D.16 and D.17 in Appendix D.4. ∎

For each packet $p' \in Z_3^{\mathcal{P}'}$, we can find the *first* wasted round $\mathtt{t}_{p'}$ in which p' was transferred between two nodes. Define $\mathcal{W} := \{\mathtt{t}_{p'} | p' \in Z_3^{\mathcal{P}'}\}$. Clearly, we have:

$$|Z_3^{\mathcal{P}'}| = |\mathcal{W}| \tag{75}$$

Lemma D.11. *For any transmission:* $\quad |\mathcal{W}| \leq n^4 + 2n^3$

Proof. This is re-stated in Appendix D.4. ∎

Lemma D.12. $\quad |Z^{\mathcal{P}'}| \leq 2T^{\mathcal{P}} + n^4 + 2n^3$

Proof. Follows immediately from Lemmas D.16, D.17, and D.11, and (75). ∎

Notice that although every packet transfer in $\mathcal{P}$ will cause a drop in potential, it may take some time before a node's cumulative potential drop for the current transmission reaches the receiver, since only one node's potential is transferred across an edge during a given round (**4.08**). In order to account for this, we will utilize the following notation. For any honest node u, let $\mathcal{U}_u \subseteq Z^{\mathcal{P}'}$ denote the set of packets that have reached R (in $\mathcal{P}'$) and travelled through u at some point en route to R. Let $\mathcal{U}_{u,2} \subseteq \mathcal{U}_u$ denote the subset consisting of the (at most) n^3 packets that left u (for the last time) latest (chronologically), and let $\mathcal{U}_{u,1} = \mathcal{U}_u \setminus \mathcal{U}_{u,2}$. If $\mathcal{U}_{u,1} \neq \emptyset$, let $\mathtt{t}_u$ denote the latest round such that some $p' \in \mathcal{U}_{u,1}$ last left u (otherwise set $\mathtt{t}_u = 0$).

Lemma D.13. *For any honest node u, R's stored value for Φ_u is at least as current as $\mathtt{t}_u$.*

[26] If we wish to emphasize the round, we will write $Z_{1,\mathtt{t}}^{\mathcal{P}'}$.

[27] Note that we make no condition that the two packets traveled in the same direction.

Proof. This is re-stated and proved in Appendix D.4. ∎

We are finally ready to put all the pieces together to prove Lemma D.7.

Proof of Lemma D.7. Suppose for the sake of contradiction that there is some transmission for which $|Z^{\mathcal{P}'}| = 3nD$ and the transmission has not yet ended. By Lemma D.17, we have that if $\mathtt{t}$ denotes the round when $|Z^{\mathcal{P}'}| = 3nD - (n^2C + nC)$, then as of round $\mathtt{t}$:

$$\sum_{u \in G} \Phi_u < CD \tag{76}$$

where Φ_u denotes the value of this variable stored by R as of round $\mathtt{t}$. Meanwhile, by Lemma D.12, we have that:

$$T^{\mathcal{P}}_{\mathtt{t}} \geq (1/2)(|Z^{\mathcal{P}'}_{\mathtt{t}}| - n^4 - 2n^3) \tag{77}$$

Since packet transfers in $\mathcal{P}$ correspond to a potential drop of at least C/n, even if we ignore contributions to potential drop from the transfers of each of the (up to) n^3 packets in $\mathcal{U}_{u,2}$ for each u, by Lemma D.13 the receiver has recorded as of round $\mathtt{t}$:

$$\begin{aligned}
\sum_{u \in G} \Phi_u &\geq (C/n)(T^{\mathcal{P}}_{\mathtt{t}}) \\
&\geq (C/n)(1/2)((|Z^{\mathcal{P}'}_{\mathtt{t}}| - n^4) - n^4 - 2n^3) \\
&\geq (C/n)(1/2)((3nD - n^2C - nC) - 2n^4 - 2n^3) \\
&\geq (C/n)(1/2)(3nD - nD) \\
&= CD
\end{aligned} \tag{78}$$

where on the second line from $Z^{\mathcal{P}'}_{\mathtt{t}}$ we have subtracted out the up to n^4 packets in $\mathcal{U}_{u,2}$ for each u, and for the third time we used that $nD \geq n(n+1)(2n^2 + C)$ (since $C \geq 8n^2$, $\lambda \leq 1/2$, and $D = \frac{nC}{\lambda}$). This contradicts (76), completing the proof. ∎

D.4 Miscellaneous Lemmas and Proofs

We restate and prove the lemmas used in the previous subsections. The first is a formal re-statement of Lemma D.3.

Lemma D.14. *After a corrupt node has been eliminated (or at the outset of the protocol) and before the next corrupt node is eliminated, there can be at most $n-1$ failed transmissions $\{\mathtt{T}_1, \ldots, \mathtt{T}_n\}$ before there is necessarily some index $1 \leq i \leq n$ such that the sender has the complete status report from every node on $\rho_{\mathtt{T}_i}$.*

Proof. We first state a simple observation:

> **Observation.** If $w \in \rho_{\mathtt{T}}$, then the sender is not missing any status report parcel for w for any transmission prior to transmission $\mathtt{T}$. In other words, there is no transmission $\mathtt{T}' < \mathtt{T}$ such that w was blacklisted at the end of $\mathtt{T}'$ (as in Sender Create Next Request), and the sender is still missing status report information from w at the end of $\mathtt{T}$.

Proof. Nodes are added to the blacklist whenever they were participating in a transmission that failed (see as in Sender Create Next Request). Nodes are removed from the blacklist whenever the sender receives all of the status report information he requested of them (**3.21-22**), or when a node is eliminated (**4.22-24**), in which case the sender no longer needs status reports from nodes for old failed transmissions[28] (and in particular, this case falls outside the hypotheses of the Lemma). Since $\rho_{\mathtt{T}}$ is defined as non-blacklisted nodes, the fact that $w \in \rho_{\mathtt{T}}$ implies that w was not on the sender's blacklist at the end of T (but before $BL_{\mathtt{T}}$ is created on **4.30**). Also, notice that (**4.30**) guarantees that *all* nodes not already on the sender's blacklist will be put on the blacklist if the transmission fails. Therefore, in the case that w has not been blacklisted since the last node was eliminated, then there have not been any failed transmissions, and hence the sender is not missing any status reports. Otherwise, let $\mathtt{T}' < \mathtt{T}$ denote the last time w was put on the blacklist, as on (**4.30**). In order for w to be put on $\rho_{\mathtt{T}}$ on line (**4.30**) of transmission T, it must have been removed from the blacklist at some point between $\mathtt{T}'$ and the end of T. In this case, the remarks at the start of the proof of this observation indicate the sender is not missing any status reports from w. □

Suppose now for the sake of contradiction that we have reached the end of transmission $\mathtt{T}_n$, which marks the n^{th} transmission $\{\mathtt{T}_1, \ldots, \mathtt{T}_n\}$ such that for each of these n failed transmissions, the sender does not have the complete status report from at least one of the nodes that participated in the transmission. Define the set $\mathcal{S}$ to be the set of nodes that were necessarily *not* on $\rho_{\mathtt{T}_n}$, and initialize this set to be empty.

Since the sender is missing some node's complete status report that participated in $\mathtt{T}_1$, there is some node $w_1 \in \rho_{\mathtt{T}_1}$ from which the sender is still missing a status report parcel corresponding to $\mathtt{T}_1$ by the end of transmission $\mathtt{T}_{n-1}$. Notice by the observation above that w_1 will not be on $\rho_{\mathtt{T}'}$ for any $\mathtt{T}_2 \leq \mathtt{T}' \leq \mathtt{T}_{n-1}$, so put w_1 into the set $\mathcal{S}$. Now looking at $\mathtt{T}_2$, there must be some node $w_2 \in \rho_{\mathtt{T}_2}$ from which the sender is still missing a status report parcel from $\mathtt{T}_2$ by the end of transmission $\mathtt{T}_{n-1}$. Notice that $w_2 \neq w_1$ since $w_1 \notin \rho_{\mathtt{T}_2}$, and also that $w_2 \notin \rho_{\mathtt{T}_{n-1}}$ (both facts follow from the above observation), so put w_2 into $\mathcal{S}$. Continue in this manner, until we have found the $(n-1)^{st}$ distinct node that was put into $\mathcal{S}$ due to information the sender was still missing by the end of $\mathtt{T}_{n-1}$. But then $|\mathcal{S}| = n - 1$, which implies that all nodes, except for the sender, are not on $\rho_{\mathtt{T}_n}$.

We reach a contradiction by showing that transmission T can not be a failed transmission (unless a corrupt node can be immediately identified). Recall that there are 3 ways a transmission can fail: 1) F2, i.e. R has stored value $\sum_{u \in G} \Phi_u > CD$; 2) F3, sender has inserted D packets; 3) F4, R has received a duplicated packet p. However, each of these cases is impossible, since no node is on the participating list $\rho_{\mathtt{T}_n}$, and hence no (honest) node should have transferred a packet ($\rho_{\mathtt{T}_n} = \emptyset$ implies that all nodes except S are on the blacklist), as line 41f of Figure 4 will fail for all honest nodes. Therefore, no honest nodes will transfer any codeword packets during T, so the sender has not inserted any packets and the receiver has not received any packets, and any node u that reports a non-zero value for Φ_u is necessarily corrupt. ■

We are now ready to prove Theorem D.1, reserving the proof of Lemma D.19 to the next section.

[28] The sender already received enough information to eliminate a node. Even though it is possible that other nodes acted maliciously and caused one of the failed transmissions, it is also possible that the node just eliminated caused all of the failed transmissions. Therefore, the protocol does not spend further resources attempting to detect another corrupt node, but rather starts anew with a reduced network (the eliminated node no longer legally participates), and will address future failed transmissions as they arise.

Proof of Theorem D.1. By Lemma D.7, for every successful transmission we have $\frac{1}{n}|Z_{\mathtt{T}}^{\mathcal{P}'}| \leq 8nC \sim (1-\lambda)D = |Z_{\mathtt{T}}^{\mathcal{P}}|$, so it remains to show that there are at most n^2 failed transmissions. By Lemma D.14, by the end of at most $n-1$ failed transmissions, there will be at least one failed transmission $\mathtt{T}$ such that the sender will have all status report parcels from every node on $\rho_{\mathtt{T}}$. Then by Lemma D.19, the sender can eliminate a corrupt node. At this point, lines (**4.22-24**) essentially call for the protocol to start over, wiping clear all buffers except for the eliminated nodes buffer, which will now contain the identity of a newly eliminated node. The transmission of the latest codeword not yet transmitted then resumes, and the argument can be applied to the new network, consisting of $n-1$ nodes. Since the node-controlling adversary can corrupt at most $n-2$ nodes (the sender and receiver are incorruptible), this can happen at most $n-2$ times, yielding the bound of n^2 for the maximum number of failed transmissions. ∎

Lemma D.15. *$\forall 1 \leq i \leq n$, if $\mathcal{P}'$ has inserted $(i \cdot nC)$ packets since round $\mathtt{t}$, then either the sender has received the EoT parcel, or there are at least i distinct (honest) nodes that have received EoT.*

Proof. (Induction on i). The subclaim is clearly true for $i = 1$, since R knows EoT as soon as it creates it in round $\mathtt{t}$. Assume the subclaim is true for $i-1$, and we aim to show it will then be true for i. If the sender has received EoT after $\mathcal{P}'$ inserts inC packets (after $\mathtt{t}$), then done. Otherwise, let $\mathcal{S}_{i-1} = \{u_1, \ldots, u_{i-1}\}$ denote the set of (honest) nodes that had EoT as of the $(i-1)nC^{th}$ packet inserted after $\mathtt{t}$ by $\mathcal{P}'$. Now during the next nC insertions by $\mathcal{P}'$, since nC exceeds the capacity of the honest nodes, one of the last nC packets (say p') just inserted necessarily reached the receiver. Let u_j denote the first (with respect to time, not with respect to the index ordering within $\mathcal{S}_{i-1}$) node in $\mathcal{S}_{i-1}$ travelled to en route from S to R (that such a node exists is immediate since $s \notin \mathcal{S}_{i-1}$ but $r \in \mathcal{S}_{i-1}$). Let v denote the node that passed p' to u_j. Then in the round when p' was passed from v to u_j, u_j necessarily[29] sent v EoT (see lines 02-03 of Figure 4), i nodes will know EoT, as required. ∎

Lemma D.16. *For any round* $\mathtt{t}$*:*

$$|Z_{1,\mathtt{t}}^{\mathcal{P}'}| \leq T_{\mathtt{t}}^{\mathcal{P}} \tag{79}$$

Proof. This follows immediately from the definition of $Z_{1,\mathtt{t}}^{\mathcal{P}'}$ together with the fact that $\mathcal{P}'$ is restricted to transferring packets between honest nodes. ∎

The following lemma follows directly from Lemma 4.2:

Lemma D.17. *For any round* $\mathtt{t}$*:*

$$|Z_{2,\mathtt{t}}^{\mathcal{P}'}| \leq T_{\mathtt{t}}^{\mathcal{P}} \tag{80}$$

Proof. This is Lemma 4.2 of [BO] together with Lemma B.17 of [BO]. Note that even though the network setting of [BO] assumes no malicious activity, the proof remains valid because $\mathcal{P}'$ is restricted to the honest nodes of G. In particular, we may restrict our graph G (which consists of honest and corrupt nodes) to G' (consisting of only honest nodes), and follow the lemmas and proofs leading to Lemma B.17 on the subgraph G'. Since $Z_2^{\mathcal{P}'}$ excludes $Z_3^{\mathcal{P}'}$ (the packets of $Z^{\mathcal{P}'}$ that travelled during a *wasted* round), the analysis leading to Lemma B.17 remains valid. ∎

[29] $\mathcal{P}'$ is restricted to the sub-graph of G consisting of *honest* nodes, so there is no danger that v or u_j will disobey protocol rules.

Lemma D.11. For any transmission:

$$|\mathcal{W}| \leq n^4 + 2n^3 \tag{81}$$

Proof. By investigating line 41 of Figure 4, there are 5 reasons a round may be wasted. By Lemma D.18 below, we need only consider lines 41c, 41d, 41e, and 41f. We bound the number of wasted rounds for each of these, noting that each edge will only transmit a broadcast parcel across it once:

1. Since there are only $2n$ parcels total comprising the SoT broadcast and EoT parcel and less than $n^2/2$ edges, lines 41c-d can cause at most n^3 wasted rounds.

2. A node can only be removed from the blacklist once per transmission. Since there are n nodes that may need to be removed from the blacklist, and less than $n^2/2$ edges, line 41e can cause at most n^3 wasted rounds.

3. We will split wasted rounds caused by 41f into two categories. In the first category, the node that is blacklisted has not yet passed all of its status report parcels across the relevant edge. Since each node's status report consists of $n-1$ parcels, and each edge will only transmit a status report parcel once, this first category can cause up to $(n-1)n(n^2/2) < n^4/2$ rounds. In the second category, the blacklisted node has already passed all of its status report parcels across the relevant edge. To bound the number of wasted rounds caused by this second category, we focus on a single such wasted round $\mathtt{t} = E(u, v)$ caused by packet $p' \in Z_3^{\mathcal{P}'}$. Without loss of generality we may assume that the round was wasted because v was on u's blacklist, and since we are in the second category, u already has all of v's status report parcels.

    Subclaim. v was on BL_s when p' was inserted.

    Proof. If $v \notin BL_s$ when p' was inserted, then S must have received all of v's status report parcels and removed v from BL_s (**3.22**). Therefore, the broadcast parcel that indicates that v should be removed from the blacklist is put into the sender's broadcast buffer when it removes v from BL_s (**2.38-39**). Let w denote the first node that p' travels to en route from S to u such that w does not know that v should be removed from the blacklist, and let $\mathtt{t}'$ denote the round that w received p'. Note that $\mathtt{t}' < \mathtt{t}$. Also, since w received p' from a node that knew v should be removed from the blacklist, round $\mathtt{t}'$ must have been wasted (**2.41e**), which contradicts minimality of $\mathtt{t}$.

    Thus, for fixed $p' \in Z_3^{\mathcal{P}'}$ corresponding to wasted round $E(u_{p'}, v_{p'})$, we have that $v_{p'}$ was on BL_s when p' was inserted (subclaim above) and $u_{p'}$ had all of $v_{p'}$'s status report parcels before the start of round $E(u_{p'}, v_{p'})$. Therefore, for each $p' \in Z_3^{\mathcal{P}'}$, let $w_{p'}$ denote the first node that p' travelled to that had $v_{p'}$'s complete status report when it received p'. Since $w_{p'} \neq s$ (otherwise $v_{p'} \notin BL_s$ when p' is inserted), we have that the node that sent p' to $w_{p'}$ (in say round $\mathtt{t}_{p'}$) must not have known $v_{p'}$'s complete status report. Since $\mathtt{t}_{p'}$ was not a wasted round, $w_{p'}$ must have sent a status report parcel (not necessarily corresponding to $v_{p'}$) during round $\mathtt{t}_{p'}$. Therefore, for every $p' \in Z_3^{\mathcal{P}'}$, we can associate a round in which a status report parcel was sent across an edge. Since there are less than n^2 total status reports and $n^2/2$ edges, this category of 41f can cause at most $n^4/2$ wasted rounds.

Adding contributions from 41c-41f, we obtain the lemma. ■

Lemma D.18. *For any $p' \in Z_3^{\mathcal{P}'}$, the corresponding first wasted round $\mathtt{t}_{p'} \in \mathcal{W}$ was wasted as a result of line 41c, 41d, 41e, or 41f (see Figure 4).*

Proof. Fix any $\mathtt{t}_{p'} \in \mathcal{W}$, and for notation, let $\mathtt{t}_{p'} = \mathtt{t} = E(u, v)$, and without loss of generality, assume p' passed from u to v in this round. We will show that necessarily u has the full SoT broadcast at the start of $\mathtt{t}$, from which the lemma follows. Suppose for the sake of contradiction that u did not have the full SoT broadcast at the start of $\mathtt{t}$. Let $\mathtt{t}_0$ denote the round in which p' was inserted by the sender (in protocol $\mathcal{P}'$). Let w denote the first node that p' visited en route from S to u such that w did not have the complete SoT broadcast, and let w' denote the node that sent p' to w in round $\mathtt{t}'$. By choice of w, we have that w' knew the complete SoT broadcast when it received p', and hence it had the complete broadcast by $\mathtt{t}'$ (when p' was sent to w). But then line 41c should have been true, so round $\mathtt{t}'$ must have been wasted. Since clearly $\mathtt{t}' < \mathtt{t}$, we have the required contradiction. ■

Lemma D.13. *For any honest node u, R's stored value for Φ_u is at least as current as $\mathtt{t}_u$.*

Proof. We prove the following statement, from which the lemma follows immediately:

> For any node u and for any $1 \leq i \leq n$, if in^2 of the n^3 packets in $\mathcal{U}_{u,2}$ have reached R, then either R has stored a value for Φ_u that is at least as recent as $\mathtt{t}_u$, or at least i distinct (honest) nodes have stored values for Φ_u that are at least as recent as $\mathtt{t}_u$.

We prove the statement via induction on i. For $i = 1$, there is nothing to show, as clearly u itself has a current value stored for Φ_u. Let $\mathtt{t}_{i-1}$ denote the round in which the $(i-1)n^2$ packet in $\mathcal{U}_{u,2}$ last left u, and let $\mathtt{t}_i$ denote the round in which the in^2 packet of $\mathcal{U}_{u,2}$ last left u, so $\mathtt{t}_u < \mathtt{t}_{i-1} < \mathtt{t}_i$. If as of $\mathtt{t}_i$ the receiver has a stored value for Φ_u that is at least as recent as $\mathtt{t}_u$, then done. Otherwise, the induction hypothesis guarantees that there exists some set $\mathcal{F}_{i-1} = \{v_1, \ldots, v_{i-1}\} \subseteq G$ of honest nodes that, as of round $\mathtt{t}_{i-1}$, have a stored value of Φ_u that is at least as recent as $\mathtt{t}_u$. Let $\mathcal{S}_u$ denote the n^2 packets in $\mathcal{U}_{u,2}$ that left u between $\mathtt{t}_{i-1}$ and $\mathtt{t}_i$.

> **Claim.** There exists (at least) one pair of honest nodes $(v_j, v_k) \in \mathcal{F}_{i-1} \times G \setminus \mathcal{F}_{i-1}$ such that at least n packets in $\mathcal{S}_u$ were transferred across $E(v_j, v_k)$ at some point after they left u and before they reached R.
>
> *Proof.* Notice that each of the n^2 packets in $\mathcal{S}_u$ had not left u for the last time as of round $\mathtt{t}_{i-1}$. For each $p' \in \mathcal{S}_u$, we may therefore find the first node v'_p such that $v_{p'} \in \mathcal{F}_{i-1}$ had a value for Φ_u at least as current as round $\mathtt{t}_u$, but the node that $v_{p'}$ passed p' to did not (since $\mathcal{P}'$ is restricted to honest nodes, necessarily $v_{p'}$ is honest). Finding $v_{p'}$ for each $p' \in \mathcal{S}$ and using an averaging argument, there is (at least) one honest node $v \in \mathcal{F}_{i-1}$ such that n packets in $\mathcal{S}$ left from v to a node not in $\mathcal{F}_{i-1}$. Since the assignment of values Φ_w to the parcel q_2 are made in a round-robin fashion (see line 08 of Figure 4), v sent his value for Φ_u to some node $w \notin \mathcal{F}_{i-1}$ during one of these n transfers, thus growing the family of nodes who have a stored value for Φ_u (at least as current as $\mathtt{t}_u$) by one. ■

D.5 Proof of Lemma D.2

In this section, we aim to prove the following lemma, which is a restatement of Lemma D.2, and which states that the sender will be able to eliminate a corrupt node if he has the complete status reports from every node that participated in some failed transmission $\mathtt{T}$.

Lemma D.19. *Suppose transmission* $\mathtt{T}$ *failed and at some later time (after transmission* $\mathtt{T}$ *but before any additional nodes have been eliminated) the sender has received all of the status report parcels from all nodes on* $\rho_{\mathtt{T}}$*. Then the sender can eliminate a corrupt node.*

Recall that there are three ways a transmission can fail:

F2. The sender receives EoT parcel indicating "F2"
F3. The sender inserted D packets
F4. The sender receives EoT parcel indicating "$(F4, p')$"

We will see that case F2 roughly corresponds to *packet duplication*, since the nodes are reporting a cumulative potential drop greater than is possible based on the packet insertions by the sender. Case F3 roughly corresponds to *packet deletion*, since the D packets the sender inserted do not reach the receiver (otherwise the receiver could have decoded by Fact 1), and case F4 corresponds to a mixed adversarial strategy of *packet deletions and duplications*. We treat each case separately in Lemmas D.20, D.21 and D.22 below, thus proving Lemma D.19:

Proof of Lemma D.19. The theorem is proven for each case below in Lemmas D.20, D.21 and D.22. ■

We declare once-and-for-all that at any time, G will refer to nodes still a part of the network, i.e. nodes that have not been eliminated by the sender.

Handling Failures as in F2: Packet Duplication

The goal of this section will be to prove the following theorem.

Lemma D.20. *Suppose transmission* $\mathtt{T}$ *failed and falls under case F2, and at some later time (after transmission* $\mathtt{T}$ *but before any additional nodes have been eliminated) the sender has received all of the status report parcels from all nodes on* $\rho_{\mathtt{T}}$*. Then the sender can eliminate a corrupt node.*

Proof. The idea of the proof is as follows. Case F2 of transmission failure roughly corresponds to *packet duplication*: there is a node $w \in G$ who is jamming the network either by outputting duplicate packets or disobeying transfer rules (e.g. by transferring a packet from a node with small height to a node with large height). This means that w will be responsible for illegal increases in potential. Using the status reports for case F2, which include nodes' signatures on changes of potential due to packet transfers, we will catch w by looking for a node who caused a greater increase in potential than is possible if it had been acting honestly.

More specifically, Case F2 means that R had stored potential values such that: $\sum_{u \in G} \Phi_u > CD$. Since we are not in Case F3, the sender did not insert D packets. Since each packet insertion can cause an *increase* in potential of at most C, the total (valid) *increase* of potential for the transmission is at most CD, which is less than the claimed potential drop $\sum_{u \in G} \Phi_u$ of the internal nodes. In particular, there is an extra potential drop in the network that cannot be accounted for by packet insertions; i.e. there is a node creating duplicated packets or lying about height information when transferring packets. The formal details of how the signed status reports $\{\Phi_{u,v}\}$ can be used by the sender to identify a corrupt node can be found in the proof of Theorem 10.6 of [7]. ■

Handling Failures as in F3: Packet Deletion

The goal of this section will be to prove the following theorem.

Lemma D.21. *Suppose transmission* T *failed and falls under case F3, and at some later time (after transmission* T *but before any additional nodes have been eliminated) the sender has received all of the status report parcels from all nodes on* $\rho_{\mathtt{T}}$*. Then the sender can eliminate a corrupt node.*

Proof. Case F3 of transmission failure roughly corresponds to *packet deletion*: the sender has inserted D packets, and yet the receiver has gotten less than $D - nC$ of them (otherwise, R could decode by Fact 1, and the transmission would not have failed). Since the total capacity of the network is only nC, there is (at least) one node $w \in G$ who is deleting packets (or storing more than C packets, which an honest node would not do). Using the status reports for case F3, which include nodes' signatures on $P_{u,v}$ (the net number of packets that have passed across each adjacent edge), we will catch w by looking for a node who input more packets than it output, and this difference is greater than the buffer capacity of the node. The formal details of how the signed status reports $\{P_{u,v}\}$ can be used by the sender to identify a corrupt node can be found in the proof of Theorem 10.11 of [7]. ■

Handling Failures as in F4: Packet Duplication + Deletion

The goal of this section will be to prove the following theorem.

Lemma D.22. *Suppose transmission* T *failed and falls under case F4, and at some later time (after transmission* T *but before any additional nodes have been eliminated) the sender has received all of the status report parcels from all nodes on* $\rho_{\mathtt{T}}$*. Then the sender can eliminate a corrupt node.*

Proof. Case F4 of transmission failure roughly corresponds to packet duplication *and* packet deletion: clearly packet duplication has occurred since R has received a duplicated packet p (which would not happen if all nodes were acting honestly), but the transmission did not fail due to Case F2, and so likely the adversary is deleting packets as he duplicates them so that signatures on potential cannot catch him. We will use the status reports for case F4, which include nodes' signatures on $[p]_{u,v}$ (the net number of times p has crossed each adjacent edge), to find a corrupt node w by looking for a node who output p more times than it input p. The formal details of how the signed status reports $[p]_{u,v}\}$ can be used by the sender to identify a corrupt node can be found in the proof of Theorem 10.12 of [7]. ■

Internal Node Create Next Request for $E(u,v)$
01 DO:
02 Set q_1 to be a parcel from BB_u not yet transferred across $E(u,v)$, chosen according to priority:
03 1) EoT parcel; 2) SoT parcels; 3) Node to *remove* from BL; 4) Status report parcel of a node on BL_u
04 If $q_1 \neq$ EoT or SoT parcel and $v \notin (EN_u \cup BL_u)$ ## Okay to send/receive p's with v
05 Set new p ## Look in stack to find highest p not already sent as a request to $\mathcal{A}$
06 Set new G_p ## Reserve the highest non-committed spot of stack
07 Else set $p = \perp$
08 Set new q_2 ## Chosen from u's (current) values of Φ_w in round-robin fashion
09 Set $\alpha = (P_{u,v}, [p']_{u,v}, \Phi_{u,v})$ ## p' is packet transferred across $E(u,v)$ the *previous* round $E(u,v)$ was honored
10 *Return* $(u, v, (p, H), (q_1, q_2), (\alpha, \sigma(\alpha)))$ ## Also remember this request for next time $E(u,v)$ is honored

Receiver Create Next Request for $E(r,v)$
11 DO:
12 If *rec'd duplicate* ## The packet p' just received had already been received by R
13 Form EoT: $q_1 =$ ("F4", p')
14 Else If $|Z| = (1-\lambda)D$ ## R now has enough packets to decode codeword
15 Form EoT: $q_1 =$ "S1"
16 Else If $\sum_{w \in G} \Phi_w \geq CD$ ## Too much potential drop: packet duplication has occurred
17 Form EoT: $q_1 =$ "F2"
18 Else set q_1 as for Internal Nodes
19 Set $p, q_2 = \perp$, and set α as for Internal Nodes
20 *Return* $(r, v, (\perp, \frac{-C}{n}), (q_1, \perp), (\alpha, \sigma(\alpha)))$ ## Also remember this request for next time $E(u,v)$ is honored

Sender Create Next Request for $E(s,v)$
21 DO:
22 If S can eliminate a node w ## Status report parcel just rec'd allows S to identify corrupt node
23 Add w to EN_s, clear BB_s and DB_s (including BL_s but *not* EN), refill Outgoing buffer
24 Set $\Omega_{\mathtt{T}+1} = (|EN|, 0, 0, 0)$
25 Else If S received EoT = "S1" ## R was able to decode codeword
26 Refill Outgoing Buffer
27 Set $\Omega_{\mathtt{T}+1} = (|EN|, |\mathcal{B}_{\mathtt{T}}|, F, 0)$ ## F denotes no. failed trans's since prev. node eliminated
28 Else If $|Y| = D$ or S received EoT = "F2" or ("F4", p') ## Failed Transmission due to mal. activity
29 Refill Outgoing Buffer
30 $\forall w \notin (BL_s \cup EN_s)$: Add w to $\rho_{\mathtt{T}}$ and then add w to BL_s
31 If EoT = ("F4", p'), set $\Omega_{\mathtt{T}+1} = (|EN|, |\mathcal{B}_{\mathtt{T}}|, F, p')$
32 Else If $|Y| = D$, set $\Omega_{\mathtt{T}+1} = (|EN|, |\mathcal{B}_{\mathtt{T}}|, F, 1)$
33 Else If EoT = "F2", set $\Omega_{\mathtt{T}+1} = (|EN|, |\mathcal{B}_{\mathtt{T}}|, F, 2)$
34 If transmission just ended ## I.e. line 22, 25, or 28 was true
35 Set SoT to be the following $2n$ parcels, and add to BB_s:
36 1) $\Omega_{\mathtt{T}+1}$; 2) EN_s; 3) BL_s; 4) Reason the prev. $n-1$ trans's failed: ("F2", "F3", or ("F4",p'))
37 Set new p ## Look in stack to find highest p not already sent as a request to $\mathcal{A}$
38 Set new q_1: Choose parcel not yet transferred across $E(s,v)$ by priority:
39 1) SoT parcel; 2) a node w to remove from BL; 3) $\perp$
40 *Return* $(s, v, (p, C + \frac{C}{n}\text{-}1), (q_1, \perp), (\alpha, \sigma(\alpha)))$ ## Also remember this request for next time $E(u,v)$ is honored

Ready(v) ## Called from node u
41 If
- u does not have $(\Omega_{\mathtt{T}}, \mathtt{T})$ in BB_u OR
- u has $(\Omega_{\mathtt{T}}, \mathtt{T})$ with $\Omega_{\mathtt{T}} = (|EN|, |\mathcal{B}_{\mathtt{T}}|, F, *)$, but has not yet rec'd $|EN|$ parcels as in line 200b, F parcels as in line 200c, or $|\mathcal{B}_{\mathtt{T}}|$ parcels as in line 200d OR
- u has rec'd the complete SoT broadcast, but every parcel hasn't yet passed across $E(u,v)$ OR
- u has EoT $\in BB_u$, but this has not passed across $E(u,v)$ yet OR
- u knows some node w to remove from BL, but hasn't yet passed this fact across $E(u,v)$ OR
- u or $v \in BL_u$

42 Return False
43 Else: Return True

Figure 4: Rules For Finding Codeword Packet and Broadcast Parcel to Send