Rate vs. Buffer Size - Greedy Information Gathering on the Line

Adi Rosén
CNRS - LRI
Bât. 490, Université Paris Sud
91405 Orsay, France
adiro@lri.fr

Gabriel Scalosub
Computer Science Department
Technion - Israel Institute of Technology
Haifa 32000, Israel
gabriels@cs.technion.ac.il

ABSTRACT

We consider packet networks with limited buffer space at the nodes, and are interested in the question of maximizing the number of packets that arrive to destination rather than being dropped due to full buffers.

We initiate a more refined analysis of the throughput competitive ratio of admission and scheduling policies in the Competitive Network Throughput model [2], taking into account not only the network size but also the buffer size and the injection rate of the traffic.

We specifically consider the problem of information gathering on the line, with limited buffer space, under adversarial traffic. We examine how the buffer size and the injection rate of the traffic affect the performance of the greedy protocol for this problem. We establish upper bounds on the competitive ratio of the greedy protocol in terms of the network size, the buffer size, and the adversary's rate, and present lower bounds which are tight up to constant factors. These results show, for example, that provisioning the network with sufficiently large buffers may substantially improve the performance of the greedy protocol in some cases, whereas for some high-rate adversaries, using larger buffers does not have any effect on the competitive ratio of the protocol.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Packet-Switching Network, Store and Forward Networks*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non Numerical Algorithms and Problems—*Routing and Layout, Sequencing and Scheduling*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Network Problems*

General Terms

Algorithms, Performance, Theory

Keywords

Buffer Management, Competitive Network Throughput, Information Gathering, Online Algorithms, Competitive Analysis

1. INTRODUCTION

Throughput analysis of packet networks under adversarial settings has received increasing attention in recent years. A large number of works have analyzed the competitive ratios of admission and scheduling policies, measuring the throughput of the system, when traffic is given by an adversary and buffer space is limited. Such works have addressed single buffers, e.g., [1, 15, 4, 14], switches, e.g., [8, 3, 6, 7, 16], or whole networks, e.g., [2, 12, 5, 9, 13]. The adversarial setting for this investigation is motivated by both theoretical interest as well as by practical needs, especially the increasing difficulty in obtaining tractable and accurate probabilistic models for network traffic. The setting of whole networks, which is especially relevant to the present work, has been studied in recent years in the framework of the *Competitive Network Throughput (CNT)* model, first introduced in [2]. This model aims at evaluating the throughput of online local-control packet admission and scheduling policies in networks with adversarial traffic, when buffer space at the routers is limited. In this model, packets are injected to various nodes over time, each with some prescribed destination and path to follow, and the goal is to maximize the overall number of packets delivered, rather than being dropped en-route due to limited buffer space. First results for this model have been obtained in [2], and were followed by additional results in, e.g., [5, 9, 12].

Most of the results mentioned above consider an arbitrary size for the buffers and a non-restricted adversary which can inject any sequence of packets into the network. They then give competitive ratios for various policies which are usually independent of the buffer size, and are a function of, e.g., the network size. This approach is clearly of merit in order to obtain results that would hold for all scenarios. However, some results, especially in the context of the throughput of single switches, lead to the question whether the size of the buffer influences the attainable competitive ratios for the problem at hand. For example, Azar and Litichevskey [6] consider the problem of scheduling a multi queue system, and present an algorithm whose competitive ratio depends on the size of the buffers, such that as the buffer size increases, the performance guarantee of the algorithm improves accordingly. In the context of the CNT model it is known that if the buffer size is $B = 1$ then the greedy protocol (and in fact any online deterministic protocol) on the line is $\Omega(n)$ competitive [2], while if $B > 1$ better competitive ratios (such as $O(\sqrt{n})$) can be achieved by online local-control protocols [2, 5].

In this paper we initiate a study in the framework of the CNT model of the interplay between the competitive ratio of admission and scheduling protocols, and the size of the buffers provided in the network - on the one hand, and the injection rate of the traffic into the network - on the other hand. We aim at studying the question

Range of r	Subrange of r	Result	UB	LB
$r \leq 1$	$r < \sqrt{\frac{B-1}{n}}$	Optimal	Theorem 2.6	
	$r \geq \sqrt{\frac{B-1}{n}}$	$\Theta\left(\max\left\{1, r\sqrt{\frac{n}{B}}\right\}\right)$	Theorem 2.7	Theorem 2.9
$1 < r < \min\{B, \sqrt{n}\}$	$r \leq \frac{n}{B}$	$\Theta\left(\sqrt{\frac{rn}{B}}\right)$	Theorem 3.2	Theorem 3.8
	$\frac{n}{B} < r < \min\{B, \sqrt{n}\}$	$\Theta(r)$	Theorem 3.3	Theorem 3.6
$r \geq \min\{B, \sqrt{n}\}$		$\Theta(\sqrt{n})$	[5]	Lemma 3.10

Table 1: Summary of results for $B \geq 2$, depending on the rate of the adversary. For every range, the UB column refers to the proof of the upper bound, and the LB column refers to the proof of the lower bound.

of whether providing the network with buffers whose sizes have a certain relationship with the networks size and/or the injection rate of the traffic, can influence the performance of the network, measured by the competitive ratio of the deployed protocols.

As a first test case for this approach we study the topology of the line, and the problem of information gathering (i.e., all packets are destined to a single node in the network). This question received considerable attention in the literature on its own, especially in the context of sensor networks, and wireless ad-hoc networks, e.g., [18, 11, 17], as well as in the context of the CNT model [2, 9, 5]. We give tight results, up to constant factors, for the competitive ratio of the greedy policy for information gathering on the line, as a function of the size of the line, n, the size of the buffer at each node B, and the injection rate of the adversary controlling the traffic, r. Roughly speaking, this injection rate bounds the amount of packets the adversary is allowed to inject into the network at every time step (For a formal definition of the adversary see Section 1.3).

Our results give insight into the question of whether provisioning the network with large buffers improves the performance of the system, measured by the throughput competitive ratio of the protocol. We show, for example, that for relatively small rates, increasing the buffer size available at the network's nodes indeed enables the greedy protocol to guarantee a better competitive ratio. However, this improvement is limited, in the sense that increasing the buffer size beyond a certain size, no longer helps in guaranteeing a better competitive ratio. Another consequence of our results is that when the adversary has rate $r \leq 1$, if buffers are sufficiently large, then the greedy protocol achieves optimal throughput, while if the buffer size is too small, then the greedy protocol cannot achieve optimal throughput. See Section 1.2 for a detailed description of our results.

We view our results as a first step towards a more refined analysis of throughput competitiveness, and towards providing guidelines on how should buffers be deployed in the network in adversarial settings. We believe that the results presented here give a better understanding of the role of buffer size in guaranteeing that simple protocols perform well under adversarial traffic. This may enable the use of some limited knowledge on the traffic pattern, even in an adversarial setting, which could be harnessed into providing better performance guarantees.

1.1 Related Work

Problems of maximizing throughput given limited size buffers and against adversarial traffic have been studied extensively in recent years e.g., [1, 15, 4, 8, 3, 6, 14]. See [10] for a short survey. These works consider the task of maximizing the number of packets transmitted from a single buffer, or from a switch, analyzing the performance of the algorithms using competitive analysis.

In the context of whole networks, Aiello et al. [2] introduced the Competitive Network Throughput (CNT) model to study the performance of buffer management and scheduling policies which are provided with limited buffer space and against adversarial traffic. Aiello et al. show that some protocols (e.g., Nearest-to-Go (NTG)) are competitive on all networks, and some other protocols (e.g., Furthest-to-Go (FTG)) do not have bounded competitive ratio on all networks. They further show that any greedy protocol on the line is $O(n)$ competitive, that NTG is $O(n^{2/3})$ competitive, and that no greedy policy can have a competitive ratio better than $\Omega(\sqrt{n})$. These results hold for any buffer size $B > 1$. On the other hand they show that if $B = 1$, any greedy policy has competitive ratio $\Omega(n)$. Angelov et al. [5] show that for the problem of information gathering on the line (where the destination of all packets is the same node), the greedy policy is $O(\sqrt{n})$ competitive for any $B > 1$. Two works, one by Angelov et al. [5] and the other by Azar and Zachut [9], give centralized online algorithms for the throughput maximization problem on the line, with polylogarithmic competitive ratio.

The fact that there is a connection between the competitive ratio of the system and the available buffer size is suggested in a work by Azar and Litichevskey [6]. They examine the competitive ratio of online algorithms for the problem of maximizing the throughput of a system with m input ports with buffers of size B, and a single output port, where at each time step only one buffer can send a packet. They give an online algorithm with competitive ratio $\frac{e}{e-1}\left(1 + \frac{O(\log m)}{B}\right)$, which approaches $\frac{e}{e-1}$, as we provide the input ports with larger buffers. For sufficiently large buffers, this improved upon the best known previous result of 1.89 [3].

The problem of information gathering was studied in the literature under different models. For example, Kothapalli and Scheideler in [18] study this problem for the case that an adversary controls not only the injected traffic but also the activation and deactivation of network links. They give results for the line and the cycle showing tight bounds on the excess amount of buffer space that the online algorithm needs (compared to the optimal adversary) in order to deliver all injected packets.

1.2 Our Results

We give tight bounds on the competitive ratio of the greedy protocol for information gathering on the line. We give upper bounds and lower bound on the competitive ratios, as a function of the available buffer space in every node, the rate of the adversary, and the size of the network. All our results are tight up to a constant factor.

Table 1 summarizes the results for the case where the buffer size is at least 2 (Section 4 treats the special case where $B = 1$). For

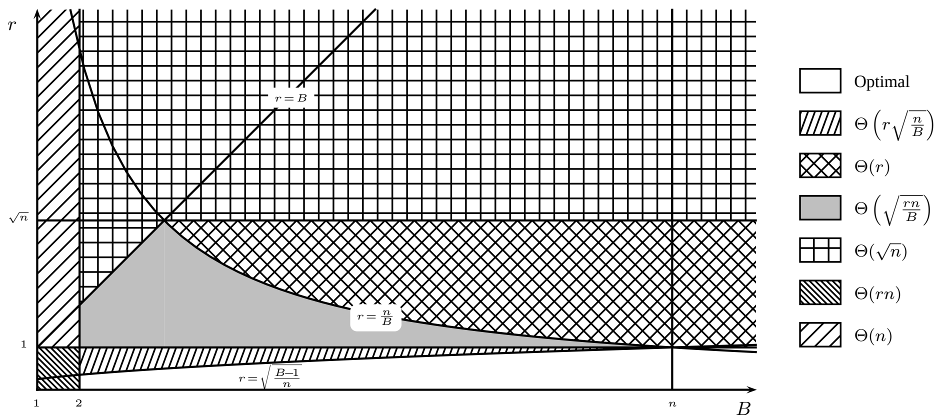

Figure 1: Graphic representation of results as a function of the buffer size B and the adversary's rate r. The X-axis represents the buffer size, and the Y-axis represents the adversary's rate. The different regions are marked according to the competitive ratio of the greedy policy, depending upon the pairing of buffer size and adversary rate values.

different ranges of the adversary's rate, r (see Section 1.3 for a formal definition of the rate), it presents the competitive ratio of the greedy policy. For a graphic representation of our results, see Figure 1.

Note specifically that these results imply that for $r \leq 1$, if the nodes are supplied with sufficiently large buffers, then the greedy policy has optimal throughput. In addition, our results imply that for $r > 1$, increasing the buffer can help guarantee a better competitive ratio, up to a point where the competitive ratio no longer depends upon the buffer size, and becomes dependant solely of the adversary's rate.

For the case where $B = 1$, we show that if the adversary has rate $1/n < r < 1$, then the competitive ratio of the greedy protocol is $\Theta(rn)$. For $r \leq 1/n$ the greedy policy is optimal, whereas by the results in [2], for $r \geq 1$ it is $\Theta(n)$ competitive.

1.3 The Model

We model the network as a digraph $G = (V, E)$, $|V| = n$, $|E| = m$. The nodes in the graph represent routers and the edges represent unidirectional communication links. The system is synchronous, and time proceeds in discrete time steps. All packets in the network have equal size, and without loss of generality we assume they are of unit size. Every link has unit capacity, and can transmit at most one packet in each time step, along the direction of the link. In the tail of every link there is a buffer of size $B \geq 1$, which can store at most B packets. Packets are injected into the network, each identified by its source node, its target node, and a predesignated path which it is has to follow from source to destination. Every packet injected into the network is injected at its source node, to be stored at the output port of the first link in its path. Each time step comprises of two substeps: the forwarding-and-injection substep followed by the switching substep. The forwarding-and-injection substep works as follows: For each link, a packet may be selected from the output buffer at the tail of the link, and this packet is forwarded to the node at the head of the link. At the same time, any number of packets can be injected into the node. In the switching sub-step packets that have arrived (or injected) to the node can be placed in the buffer of the next (or first) edge of their path. If there is not enough space in the buffer to store all packets some packets must be dropped.

A greedy protocol is a protocol that never drops a packet unless the buffer in which it has to be stored is full, and always forwards a packet from a buffer unless the buffer is empty. Protocols satisfying the latter property are sometimes referred to as *work conserving*.

We focus our attention in this paper on the directed line topology. Note that in this topology, any packet is characterized solely by its source node, and target node. Furthermore, in this topology, every node has a single outgoing link. We will therefore sometime refer to a link's output buffer, as the buffer at its tail node. We further focus on the problem of *Information Gathering* on the line, where the target node of all packets is the last node of the line. In this case, every packet is characterized solely by its source node. In this paper we consider greedy protocols for information gathering. Note that for this problem on the line all greedy protocols are equivalent, and we will therefore refer to *the greedy protocol* in this case. Since all greedy protocols are equivalent, unless stated differently, we assume for ease of analysis, without loss of generality, that when there is a packet arriving at a node from its preceding node, then this will be the packet which is forwarded on the node's outgoing link in the next time step. We call this assumption the *en-route* assumption. Also observe that at every time step there is at least space for one new packet in any buffer, since a packet is always sent from a full buffer. We can therefore assume without loss of generality that all packets that are forwarded on a link are stored in the buffer of the node at the head of the link, and never dropped. It follows that we can assume without loss of generality that any packet accepted and stored in any source node buffer, is never dropped, i.e., packets are only dropped at injection.

We are interested in maximizing the throughput of the network, i.e., maximizing the number of packets which are delivered to the last node of the line.

We assume the injections are governed by an adversary. Given any real number r, an *r-adversary* can inject any sequence of packets as long as for every time interval of length t, at most $\lceil rt \rceil$ packets are injected into the network. Note that the adversary is allowed to inject the packets to any nodes in the network, and may well inject more than one packet simultaneously, even to the same node.

We use competitive analysis to measure the performance of the greedy protocol. We say that a protocol A is c-competitive if, for every input traffic sequence σ of packets injected into the network, it holds that, $\mathrm{OPT}(\sigma) \leq c \cdot A(\sigma) + b$, where $\mathrm{OPT}(\sigma)$ is the number of packets an optimal offline algorithm delivers out of σ, $A(\sigma)$ is the number of packets A delivers out of σ, and b is a constant independent of the input traffic sequence σ. For the analysis, we assume without loss of generality that the optimal algorithm never drops a packet that it accepted at injection.

Throughout the paper, unless otherwise stated, we assume that the buffer size B is at least 2. The special case of $B = 1$ is treated in Section 4.

Organization

The rest of the paper is organized as follows. We first assume that $B > 1$. In Section 2 we consider the case of $r \leq 1$, and in Section 3 we consider the case of $r > 1$. In Section 4 we treat the special case of $B = 1$, and we conclude the paper with conclusions and open problems in Section 5. Some proofs are omitted due to space limitations.

2. LOW RATE ADVERSARIES

2.1 Large Buffers

In this section we show that for any adversary of rate $r \leq 1$, if $B \geq \max\left\{2, \lceil r^2 n \rceil + 1\right\}$ then the greedy policy does not drop packets, and is therefore optimal. To this end we analyze the system as if it has unbounded buffers and no packet is dropped, and give an upper bound of $\max\left\{2, \lceil r^2 n \rceil + 1\right\}$ on the size of the buffers.

As a first step, we prove the following lemma, which bounds the overall number of packets in the network, under the greedy policy:

LEMMA 2.1. *For any r-adversary $r \leq 1$, under the greedy policy, at any time t, the number of packets in the system is at most $\lceil rn \rceil$.*

To see this, since all greedy protocols are equivalent, it is sufficient to show that the above lemma holds for any specific greedy protocol. For analysis purposes it is convenient to consider the *Longest-in-System (LIS)* protocol.

The following lemma, whose proof is omitted, enables us to give a bound on the amount of time every packet stays in the network:

LEMMA 2.2. *Under LIS, for any adversary of rate $r \leq 1$, consider any packet p injected to node i in time t. Then for every $j \geq i$, p arrives to node j by time $t + j$, and p is sent from node j by time $t + j + 1$.*

Applying the above lemma with $j = n$ we obtain the following corollary:

COROLLARY 2.3. *Under LIS, for any adversary of rate $r \leq 1$, every packet is in the system for at most n time units.*

The following lemma gives a bound on the overall number of packets in the network at any given time under the LIS protocol.

LEMMA 2.4. *Under LIS, for any adversary of rate $r \leq 1$, and at any time t, the number of packets in the system in time t is at most $\lceil rn \rceil$.*

PROOF. Consider any time t. By Corollary 2.3, any packet injected to the system before time $t - n$ has already been delivered. Hence the system holds only packets injected during the interval $(t - n, t]$. By the definition of the adversary, the maximum number of packets injected during such an interval is at most $\lceil rn \rceil$. $\square$

Combining the above lemma with the fact that all greedy policies are equivalent, we conclude the proof of Lemma 2.1. Note that Lemma 2.1 guarantees that for any r-adversary such that $r \leq 1$, if $B \geq \lceil rn \rceil$ then any greedy policy does not drop packets, and hence any greedy policy is optimal. In what follows we show that the same result holds even for buffers of smaller size.

LEMMA 2.5. *For any adversary of rate $r \leq 1$, and any greedy policy, at any time t there are at most $\max\left\{2, \lceil r^2 n \rceil + 1\right\}$ packets in every buffer.*

PROOF. Let i be any node in the system. If at the beginning of time step t there are more than 2 packets at i's buffer, then at time $t - 1$ the node was not empty (since at most 2 packets can arrive in each time step). Let t' be the latest time prior to t where the buffer is empty. Without loss of generality assume $t' = 0$. We distinguish between two cases:

Case 1: $t \leq \lceil nr \rceil$:

In this case, at time t the number of packets in node i is at most

$$t + \lceil tr \rceil - t = \lceil tr \rceil \leq \lceil r^2 n \rceil + 1.$$

Case 2: $t > \lceil nr \rceil$:

Let $\varepsilon = \lceil tr \rceil - tr$, and note that $0 \leq \varepsilon < 1$. At time t the number of packets in node i is at most

$$\begin{aligned} \lceil nr \rceil + \lceil tr \rceil - t &= \lceil nr \rceil + t(r-1) + \varepsilon \\ &= \lceil nr \rceil - t(1-r) + \varepsilon \\ &< \lceil nr \rceil - \lceil nr \rceil (1-r) + \varepsilon \\ &\leq \lceil nr \rceil r + 1, \end{aligned}$$

where the first term follows from Lemma 2.1, and the inequality follows from the fact that $r \leq 1$. Since any integer m which satisfies $m < \lceil rn \rceil r$ also satisfies $m \leq \lceil r^2 n \rceil$, it follows that the number of packets in node i is at most $\lceil r^2 n \rceil + 1$. $\square$

The following theorem is an immediate consequence of the above lemma:

THEOREM 2.6. *For any r-adversary such that $r \leq 1$, if $B \geq \max\left\{2, \lceil r^2 n \rceil + 1\right\}$ then the greedy policy does not drop packets, and thus is optimal.*

2.2 Small Buffers

In this section we give tight bounds on the competitive ratio of the greedy policy against any r-adversary with $r \leq 1$, in a network which is supplied with relatively small buffers. Specifically, we show that if $2 \leq B \leq \Theta(r^2 n)$, then the greedy policy has competitive ratio $\Theta(r\sqrt{\frac{n}{B}})$.

2.2.1 Upper Bound

THEOREM 2.7. *If $2 \leq B \leq n$, and the packets are injected by an r-adversary with $\sqrt{\frac{B-1}{n}} \leq r \leq 1$, then the greedy policy is $O(\max\left\{1, r\sqrt{\frac{n}{B}}\right\})$ competitive.*

PROOF. For the purpose of the analysis we divide time into a sequence of intervals $P_0, K_0, P_1, K_1, P_2, K_2, \ldots$. Intervals P are defined by the number of packets that the adversary accepts. Intervals K will be fixed length intervals of length k. Formally, let $P_i = [s_i, t_i + 1)$, and $K_i = [t_i + 1, u_i)$ where

1. $s_0 = 0$ and for $i > 1$, $s_i = u_{i-1}$,

2. t_i is the earliest time after s_i where the adversary accepts $3 \cdot \lceil rn \rceil$ packets during the interval $[s_i, t_i + 1)$, and

3. $u_i = t_i + k$, for $k = \Theta(n)$.

We start by showing that we can identify $\Omega(\min\left\{rn, \sqrt{nB}\right\})$ distinct packets residing in the buffers of the greedy policy during P_i.

If the greedy policy accepts at least $\lceil rn \rceil$ of the new packets injected by the adversary during P_i then we have at least $\Omega(rn)$ packets residing in the buffers of the greedy policy during P_i.

Assume now that the greedy policy does not accept at least $\lceil rn \rceil$ of the new packets injected by the adversary during P_i. It follows that the greedy policy drops during P_i at least $2 \cdot \lceil rn \rceil$ packets.

We say that a node j is *bad in* P_i if at least one packet was dropped in j during P_i. Note that if a packet is dropped in j at time t, then the buffer at that node is full at that time, and furthermore, due to the en-route assumption, at least $B - 1 \geq \frac{B}{2}$ of the packets residing in node j at time t have been injected to j itself. Let x denote the number of bad nodes in P_i. If $x \geq \sqrt{\frac{n}{B}}$, then we can identify at least $x\frac{B}{2} = \Omega(\sqrt{nB})$ distinct packets residing in the buffers of the greedy policy during P_i. Assume now that $x < \sqrt{\frac{n}{B}}$. Recall that the greedy policy has dropped at least $2 \cdot \lceil rn \rceil$ packets during P_i, hence in at least one of the bad nodes the greedy policy has dropped at least $\frac{2 \cdot \lceil rn \rceil}{\sqrt{\frac{n}{B}}} \geq 2r\sqrt{nB}$ packets. Observe that by the assumption that $r \geq \sqrt{\frac{B-1}{n}}$ we are guaranteed to have $2r\sqrt{nB} \geq 2$. We now use the following lemma, whose proof appears later in the sequel.

LEMMA 2.8. *Any bad node j such that at least $q \geq 2$ packets were dropped at j during P_i, forwards $\Omega(q \cdot \frac{1}{r})$ packets during P_i.*

It follows that there is at least one bad node from which

$$\Omega\left(2r\sqrt{nB} \cdot \frac{1}{r}\right) = \Omega(\sqrt{nB})$$

packets have been forwarded during P_i, which means that we can identify $\Omega(\sqrt{nB})$ distinct packets residing in the buffers of greedy during P_i.

We can therefore conclude that there are $\Omega(\min\left\{rn, \sqrt{nB}\right\})$ distinct packets residing in the buffers of the greedy policy during P_i.

By the fact that $B \leq n$, we have

$$k = \Theta(n) = \Omega(n + \min\left\{rn, \sqrt{nB}\right\}).$$

Since we have shown that at least $\Omega(\min\left\{rn, \sqrt{nB}\right\})$ distinct packets resided in the buffers under the greedy policy during P_i, it follows that at least this number of packets were delivered by the greedy protocol during $P_i \cup K_i$.

As to the adversary, note that by the choice of k - the length of interval K_i - the overall number of packets accepted by the adversary during $P_i \cup K_i$ is bounded by $3 \cdot \lceil rn \rceil + r \cdot \Theta(n) = O(rn)$.

Summing the above over all i, we obtain a lower bound of

$$\Omega(\min\left\{jrn, j\sqrt{nB}\right\})$$

on the number of packets delivered by the greedy policy by the end of K_j, where on the other hand the same summation yields an upper bound of $O(jrn)$ on the number of packets accepted by the adversary by the end of interval K_j, which clearly also bounds the number of packets delivered by the adversary by the end of K_j. It therefore follows that the ratio between the number of packets delivered by the adversary and the number of packets delivered by the greedy policy is $O(\max\left\{1, r\sqrt{\frac{n}{B}}\right\})$, which completes the proof. □

PROOF OF LEMMA 2.8. By the assumption, we know that at least 2 packets were dropped at node j. Consider any two consecutive events in which a packet was dropped at node j, and assume without loss of generality that the first drop was at time 0, and the second drop was at time t. Note that for every node j' and time s, a packet is dropped at the switching substep of time s only when there has been both an injection into node j' and a forwarding to node j' in the forwarding-and-injection substep of time s, and the buffer of j' is full at the beginning of the forwarding-and-injection substep. Furthermore, since every two consecutive injections are at least $\lfloor 1/r \rfloor \geq 1$ time apart, we necessarily have $t > 0$. If the buffer at node j is full during the entire interval $[0, t]$, then clearly at least $t \geq \lfloor 1/r \rfloor = \Omega(1/r)$ packets have been forwarded from node j under the greedy policy. Otherwise, let $0 < s < t$ be the last time prior to t in which the buffer at node j was not full at the end of time slot s. By the maximality of s, and the fact that $r \leq 1$, it follows that at the end of time s there were $B - 1$ packets in the buffer of node j, and at the forwarding-and-injection substep of time $s + 1$ one packet arrived to node j on its incoming link, and one packet was injected to node j. By the fact that inter-injection time is at least $\Omega(1/r)$, it follows that the interval $(s, t]$ is of length at least $\Omega(1/r)$, and since the buffer was always full during this interval, it follows that one packet was forwarded from node j in every time step in this interval, i.e., at least $\Omega(1/r)$ packets were forwarded from node j in the interval $[0, t]$.

Since this holds for every two consecutive events of packets being dropped at j, and by the assumption on j there were at least $q \geq 2$ packets dropped at j during P_i, we conclude that at least $\Omega(q \cdot \frac{1}{r})$ packets were forwarded from node j during P_i. □

2.2.2 Lower Bound

In this section we prove that the upper bound given in Theorem 2.7 is tight up to a constant factor, for buffers smaller than $O(r^2 n)$. Note that for any constant $0 < c < 1$, and any r-adversary such that $r \leq 1$, if $cr^2 n \leq B \leq \lceil r^2 n \rceil$ then Theorem 2.7 guarantees that the greedy policy is $O(1)$-competitive. Therefore it is enough to prove our lower bound for buffers of size less than $\frac{1}{16} r^2 n$.

THEOREM 2.9. *For any $r \leq 1$, if $2 \leq B < \frac{1}{16} r^2 n$, then there exists an r-adversary A such that the ratio between the throughput of A and that of the greedy policy is $\Omega\left(r\sqrt{\frac{n}{B}}\right)$.*

PROOF. The adversary will inject packets in two epochs. We will consider the line as divided into two blocks, where the second block is divided into segments. In the first epoch the adversary injects only to the first block, whereas in the second epoch the adversary injects only to the second block. The goal of the injection sequence in the first epoch is to generate a continuous sequence of packets arriving at the second block. The second block is divided into segments, where the injection during the second epoch will cause the greedy policy to drop packets in every segment. As the analysis will show, the overall number of packets accepted by the greedy policy would be proportional to the injections made to the first block, whereas the adversary can accept all the packets injected.

Formally, let $r' = \frac{1}{\lceil 1/r \rceil}$. It follows that $r/2 \leq r' \leq r$, and $1/r'$ is integral. Let $d = \lfloor \sqrt{nB} \rfloor$. Consider the line as composed of two blocks of nodes, where the first block consists of the nodes $0, \ldots, \frac{d}{r'} - 1$, and the second block consists of nodes $\frac{d}{r'}, \ldots, n$. We divide the second block into $k = \lfloor \frac{n}{d} \rfloor - \frac{1}{r'}$ segments of length d each, $S_0, \ldots, S_{k-1}$.

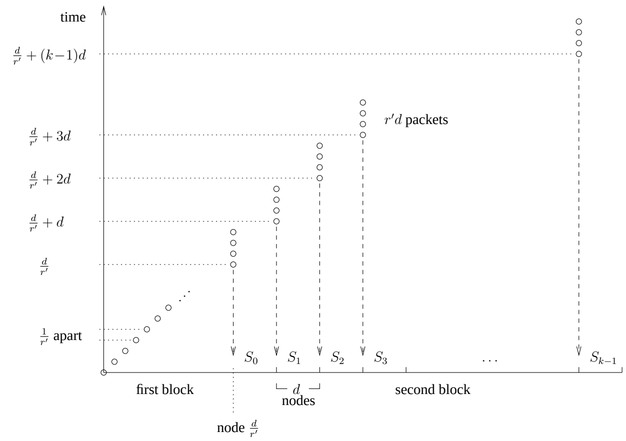

Figure 2: Outline of the injection pattern for the adversary showing the $\Omega\left(r\sqrt{\frac{n}{B}}\right)$ lower bound. The X-axis represents the line network, and each circle represents the injection of a packet. Out of the $r'd$ packets injected to every segment in the second block, only B packets would be absorbed by the greedy policy.

Note that by the assumption on B and the choice of r' and d, the number of nodes in the first block is at most

$$\frac{d}{r'} = \frac{\lfloor\sqrt{nB}\rfloor}{r'} \leq \frac{\sqrt{nB}}{r/2} < 2\frac{\sqrt{r^2n^2/16}}{r} = \frac{n}{2}.$$

Since there remain at least $\frac{n}{2}$ nodes in the second block, and the length of every segment in the second block is

$$d = \lfloor\sqrt{nB}\rfloor \leq \sqrt{nB} < \sqrt{\frac{r^2n^2}{16}} = \frac{rn}{4} \leq \frac{n}{4},$$

we are guaranteed to have at least two segments in the second block.

The injection sequence of the adversary is divided into two epochs, as follows:

Epoch 1: For every $i = 0, \ldots, d-1$, inject a packet to node $\frac{i}{r'}$ in time $\frac{i}{r'}$.

Epoch 2: For every segment $j = 0, \ldots, k-1$, inject $\lfloor r'd\rfloor$ packet to the first node of S_j, one every $1/r'$ time units, starting from time $\frac{d}{r'} + jd$. Note that by the choice of r, r' and d we have $\lfloor r'd\rfloor \geq 2$.

See Figure 2 for an outline of the injection sequence.

In addition, note that since the above injection sequence does not inject more than one packet every $1/r'$ time units, the injection rate is at most $r' \leq r$, hence it corresponds to an r-adversary.

We now turn to analyze the performance of the greedy policy given the above injection sequence. First note that the greedy policy accepts all the packets injected during epoch 1. To see this, notice that the adversary injects at most one packet to every node. It follows that there is at most one time unit where the node receives two packets simultaneously - one from its preceding node, and one injected by the adversary. Since by our assumption $B \geq 2$, the greedy policy does not drop packets during epoch 1.

The following lemma, whose proof is omitted, shows that starting from time $\frac{d}{r'}$, there is a continuous sequence of d packets arriving to the first node of S_0 from its preceding node.

LEMMA 2.10. *For every $i = 0, \ldots, d-1$, there is a continuous sequence of $i+1$ packets leaving node $\frac{i}{r'}$, starting from time $\frac{i}{r'}+1$.*

The following lemma, whose proof appears later in the sequel, bounds the number of packets which leave any of the segments in the second block, under the assumption that $2 \leq B < \frac{1}{16}r^2n$:

LEMMA 2.11. *For every $i = 0, \ldots, k-1$, there is a continuous sequence of $d+(i+1)B$ packets leaving S_i, entering segment S_{i+1} as of time $\frac{d}{r'} + (i+1)d$.*

Since the number of segments in the second block is

$$\lfloor\frac{n}{d}\rfloor - \frac{1}{r'} = \lfloor\frac{n}{\lfloor\sqrt{nB}\rfloor}\rfloor - \frac{1}{r'} = O\left(\sqrt{\frac{n}{B}}\right),$$

by Lemma 2.11, the number of packets delivered by the greedy policy is $O(d + \sqrt{\frac{n}{B}}B) = O(\sqrt{nB})$.

The adversary injects at least

$$d + \left(\lfloor\frac{n}{d}\rfloor - \frac{1}{r'}\right)r'd = \Theta(r'n)$$

packets. It can keep them all by not forwarding packets in the first block, and spreading the $r'd$ packets injected to segment S_i throughout the segment while not sending packets between different segments. Therefore after a flush-phase at the end of the injection sequence, the adversary can deliver all the packets it has accepted.

It follows that the ratio between the number of packets delivered by the adversary and the number of packets delivered by the greedy policy is at least

$$\frac{\Theta(r'n)}{O(\sqrt{nB})} = \Omega\left(r\sqrt{\frac{n}{B}}\right),$$

which completes the proof of Theorem 2.9. $\square$

PROOF OF LEMMA 2.11. The proof is by induction on i. For the base case, note that by Lemma 2.10, there is a continuous sequence of d packets entering the first node of S_0, starting from time $\frac{d}{r'}$. It follows that during d time units, there is a packet arriving to the first node of S_0 from its preceding node. In addition, during these d time units, there are $\lfloor r'd\rfloor$ packets injected by the adversary to the first node of S_0. Due to the en-route assumption, none of these packets are forwarded from this node until the entire sequence of d packets arriving on the incoming link has ended. Note that by our assumption that $2 \leq B < \frac{1}{16}r^2n$, we obtain that $r > 4\sqrt{\frac{B}{n}}$. It follows that

$$\begin{aligned} \lfloor r'd\rfloor &\geq \lfloor\frac{r\lfloor\sqrt{nB}\rfloor}{2}\rfloor \\ &\geq \lfloor 2\frac{\sqrt{B}\lfloor\sqrt{nB}\rfloor}{\sqrt{n}}\rfloor \\ &\geq \lfloor 2\frac{\sqrt{B}(\sqrt{nB}-1)}{\sqrt{n}}\rfloor \\ &= \lfloor 2\left(B - \sqrt{\frac{B}{n}}\right)\rfloor > B, \end{aligned}$$

where the last inequality follows from the fact that in our case $2 \leq B < \frac{1}{16}r^2n \leq \frac{n}{16}$. The node can store only B out of these $\lfloor r'd\rfloor$ injected packets, which are then forwarded immediately after the sequence arriving on the incoming link has terminated. This prolongs the sequence leaving the first node of S_0 by additional B packets, to a total of $d + B = d + (0+1)B$ packets, which start leaving the first node of S_0 in time $\frac{d}{r'} + 1$. Since the length of S_0 is d nodes, this sequence enters segment S_1 as of time $\frac{d}{r'} + d = \frac{d}{r'} + (0+1)d$. This completes the base case.

For the inductive step, assume the claim holds for i. It follows that there is a continuous sequence of $d+(i+1)B$ packets leaving S_i, entering segment S_{i+1} as of time $\frac{d}{r'} + (i+1)d$. Starting from this time, during a period of d time units, the adversary injects $r'd$ packets to the first node of S_{i+1}. Similar to the base case, due to the en-route assumption, none of these packets are forwarded from this node until the entire sequence of $d + (i+1)B$ packets arriving on the incoming link has ended. Since the node can only store B out of the $\lfloor r'd \rfloor$ packets injected by the adversary, these packets 'join' the sequence arriving on the incoming link, thus the continuous sequence of packets leaving the node comprises of $d + (i+1)B + B = d + (i+2)B$ packets. By the fact that the length of S_{i+1} is d, this sequence starts entering segment S_{i+2} as of time $\frac{d}{r'} + (i+1)d + d = \frac{d}{r'} + (i+2)d$, which completes the proof of the lemma. $\square$

3. HIGH RATE ADVERSARIES

In this section we treat the case of adversaries of high rates, i.e., of rates $r > 1$. We give tight bounds on the competitive ratios obtained by the greedy policy in this case. These bounds are a function of the network size n, the buffer size B, and the injection rate r. Interestingly, different functions apply for different combinations of these values.

3.1 Upper Bounds

Let $M = \max\{n, B\}$. The following lemma shows an upper bound in terms of M on the performance of the greedy policy, against any r-adversary with $r > 1$

LEMMA 3.1. *For any r-adversary such that $r > 1$ the greedy policy is $O\left(\sqrt{\frac{rM}{B}} + r\right)$ competitive.*

PROOF. The following proof is an extension of the proof appearing in [5].

For the purpose of the analysis we divide time into a sequence of intervals $P_0, K_0, P_1, K_1, P_2, K_2, \ldots$. Intervals P are defined by the number of packets that the adversary accepts. Intervals K will be fixed length intervals of length k. Formally, let $P_i = [s_i, t_i + 1)$, and $K_i = [t_i + 1, u_i)$ where

1. $s_0 = 0$ and for $i > 1$, $s_i = u_{i-1}$,
2. t_i is the earliest time after s_i where the adversary accepts $\lceil rM \rceil$ packets during the interval $[s_i, t_i + 1)$, and
3. $u_i = t_i + k$, for $k = \Theta(\sqrt{rMB} + n)$.

In what follows, we compare the throughput of the adversary and the throughput of the greedy algorithm in every interval $P_i \cup K_i$.

We start by showing that we can identify $\Omega(\sqrt{rMB})$ distinct packets residing in the buffers of the greedy policy during P_i.

Note first that if the greedy policy accepts at least $\frac{rM}{2}$ of the packets accepted by the adversary during P_i, then since $rM \geq \sqrt{rMB}$ for $r \geq 1$, we are guaranteed to have $\Omega(\sqrt{rMB})$ packets residing in the buffers of the greedy policy during P_i.

Assume next that the greedy policy does not accept at least $\frac{rM}{2}$ of the new packets accepted by the adversary. It follows that it drops at least $\frac{rM}{2}$ of the new packets accepted by the adversary during P_i. For the purpose of the proof we define a dynamic weight assignment to packets stored by the greedy protocol.

Initializing the weights: Every packet accepted by the greedy policy has its weight initialized to zero in the moment of its injection, and all packets not yet delivered have their weight reset to zero in the beginning of any interval P_i.

Increasing the weights: Any interval P_i is divided into periods for every node separately. The k'th period of a node is defined by the time interval $[x_k, x_k + B)$, where x_k is the earliest time a packet is dropped from the node after the end of the previous period. In the beginning of every period, we increase the weight of every packet in the node's buffer by 2. There is no weight increase during the intervals K_i.

Note that a packet is dropped at node i at the beginning of a period iff the buffer is full at this time, i.e., there are B packets in the buffer. By increasing the weight of each of these packets by 2, the overall weight increase is $2B$, which is an upper bound on the number of packets the adversary may accept into node i, and the greedy policy lose, during this period (of length B).

We now show that during interval P_i the greedy policy stored in its buffers at least $\Omega(\sqrt{rMB})$ distinct packets. Let $2c$ be the maximum weight a packet has at the end of interval P_i, where c is some positive integer.

By the fact that for every node j, the weight increase in every period of node j is an upper bound on the number of packets accepted by the adversary and dropped by the greedy protocol during the period at node j, and since the number of packets that were dropped by the greedy protocol but accepted by the adversary during P_i is at least $\frac{rM}{2}$, we have that the total weight of the packets of greedy is at least $\frac{rM}{2}$, and we can therefore identify at least $\frac{rM}{2c} = \Omega(\frac{rM}{c})$ distinct packets residing in the buffers of greedy during P_i. It follows that if $c = 1$, this is at least $\Omega(rM) = \Omega(\sqrt{rMB})$, and we are done.

Assume next that $c \geq 2$, and let p be any packet with weight $2c$. Note that p may have already been delivered by the algorithm. The weight of p can be divided into two categories, such that $2c = 2w + 2v$:

Weight given at p's origin node: Denote it by $2w$. It follows that p spent at least $B(w-1)$ time units at its origin node, since it was there during w periods, each lasting B time units. Since the algorithm is greedy, in every such time unit, one packet was sent from the origin node, i.e. at least $B(w-1)$ packets were sent during these time units. These are all different packets, which p will never 'beat' to the end, due to our en-route assumption.

Weight given at p's transit nodes: Denote it by $2v$. In every transit node where p had its weight increased, there are $B-1$ packets left behind (because the weight is increased only in time of overflow, where the buffer is full). Since p moves continuously, the sets of packets in two distinct such transit nodes are disjoint, because of the en-route assumption. Therefore, there are at least $v(B-1)$ different packets left 'behind' p.

The number of packets stored by greedy during P_i is at least

$$\begin{aligned} 1 + B(w-1) + v(B-1) &= cB - B - v + 1 \\ &\geq cB - B - c + 1 \\ &= (c-1)(B-1) = \Omega(cB). \end{aligned}$$

It follows that if the algorithm dropped at least $\frac{rM}{2}$ of the packets accepted by the adversary during P_i, it had stored in its buffers at least

$$\Omega\left(\max\left\{cB, \frac{rM}{c}\right\}\right) = \Omega(\sqrt{rMB})$$

packets during interval P_i.

We can therefore conclude that in any case there were at least $\Omega(\sqrt{rMB})$ packets residing in the buffers under the greedy policy during P_i.

When considering the adversary, note that by the choice of k - the length of interval K_i - the overall number of packets accepted by the adversary during the interval $P_i \cup K_i$ is upper bounded by $\lceil rM \rceil + r \cdot \Theta(\sqrt{rMB} + n) = O(rM + r\sqrt{rMB})$.

Furthermore, since we have shown that $\Omega(\sqrt{rMB})$ distinct packets resided in the buffers under the greedy policy during P_i, and since $k = \Theta(\sqrt{rMB} + n)$, it follows that at least $\Omega(\sqrt{rMB})$ packets were delivered during $P_i \cup K_i$ under the greedy policy.

By summing the above over all i, we obtain a lower bound of $\Omega(j\sqrt{rMB})$ on the number of packets delivered by the greedy policy by the end of K_j, where on the other hand the same summation yields an upper bound of $O(j(rM + r\sqrt{rMB}))$ on the number of packets accepted by the adversary by the end of interval K_j, which clearly also bounds the number of packets delivered by the adversary by the end of K_j. It therefore follows that the ratio between the number of packets delivered by the adversary and the number of packets delivered by the greedy policy is $\left(\frac{O(rM+r\sqrt{rMB})}{\Omega(\sqrt{rMB})}\right) = O\left(\sqrt{\frac{rM}{B}} + r\right)$, which completes the proof. □

The above lemma implies two upper bounds on the performance of the greedy policy, depending on the rate of the adversary. The first applies to adversaries with rates bounded by $\frac{n}{B}$:

THEOREM 3.2. *For any r-adversary such that $1 < r \leq \frac{n}{B}$, the greedy policy is $O(\sqrt{\frac{rn}{B}})$ competitive.*

PROOF. Assume $r > 1$ also satisfies $1 < r \leq \frac{n}{B}$. In particular in this case, we have $B < n$, which implies $M = n$. By the assumption that $1 < r \leq \frac{n}{B}$, we have $r \leq \sqrt{\frac{rn}{B}} = \sqrt{\frac{rM}{B}}$. It therefore follows by Lemma 3.1 that the competitive ratio is at most $O\left(\sqrt{\frac{rn}{B}}\right)$. □

The following theorem gives an upper bound for the remaining range of r.

THEOREM 3.3. *For any r-adversary such that $r > 1$ and $r > \frac{n}{B}$, the greedy policy is $O(r)$ competitive.*

PROOF. Assume $r > 1$ also satisfies $r > \frac{n}{B}$. If $B \geq n$ then $\frac{rM}{B} = r$, hence by Lemma 3.1, the competitive ratio is $O(\sqrt{r} + r) = O(r)$. If on the other hand $B < n$, then by the assumption that $r > \frac{n}{B}$, we have $\frac{rM}{B} = \frac{rn}{B} < r^2$. It therefore follows by Lemma 3.1 that the competitive ratio is at most $O(\sqrt{r^2} + r) = O(r)$. □

Angelov et al. [5] have shown that for all r, and regardless of the buffer size B, the greedy policy is $O(\sqrt{n})$ competitive. Combining their result with Theorems 3.2 and 3.3, we obtain the following two corollaries:

COROLLARY 3.4. *For any r-adversary such that $1 < r \leq \frac{n}{B}$, the greedy policy is $\min\left\{O(\sqrt{\frac{rn}{B}}), O(\sqrt{n})\right\}$ competitive.*

COROLLARY 3.5. *For any r-adversary such that $r > 1$ and $r > \frac{n}{B}$, the greedy policy is $\min\{O(r), O(\sqrt{n})\}$ competitive.*

3.2 Lower Bounds

In this section we present two lower bounds which combined with the upper bounds presented in Section 3.1, enable us to characterize the performance of the greedy policy, up to a constant factor, for any r-adversary such that $r > 1$.

THEOREM 3.6. *For any $4 < r < \sqrt{n}$, and for any buffer size B, there exists an r-adversary A such that the ratio between the throughput of A and that of the greedy policy is $\Omega(r)$.*

PROOF. We consider the line as divided into segments, and have the adversary inject at most one packet in every time step to every segment. Given any rate $4 < r < \sqrt{n}$, we show that the number of segments is at most r, hence the injection corresponds to an r-adversary. As the analysis will show, the overall number of packets accepted by the greedy policy would be proportional to the injections made to the last segment, whereas the adversary can accept all the packets injected.

Formally, Let $4 < r < \sqrt{n}$, and let $d = \lceil \frac{n}{r^2} \rceil$. Consider the line as composed of $k = \lfloor \sqrt{\frac{n}{d}} \rfloor$ segments $S_0, \ldots, S_{k-1}$, such that the length of segment S_i is $(i+1)d$.

Note that by the assumption on r we have $2 \leq d \leq \lceil \frac{n}{16} \rceil$, and the overall length of the segments is $\sum_{i=1}^{k} id = \frac{k(k+1)d}{2} \leq k^2 d \leq n$.

We now describe the sequence of injections generated by an r-adversary A. For every $i = 0, \ldots, k-1$, A injects $(i+1)dB$ packets to the first node of segment S_i, starting at time $t_i = \sum_{j=0}^{i} jd$.

See Figure 3 for an outline of the injection sequence.

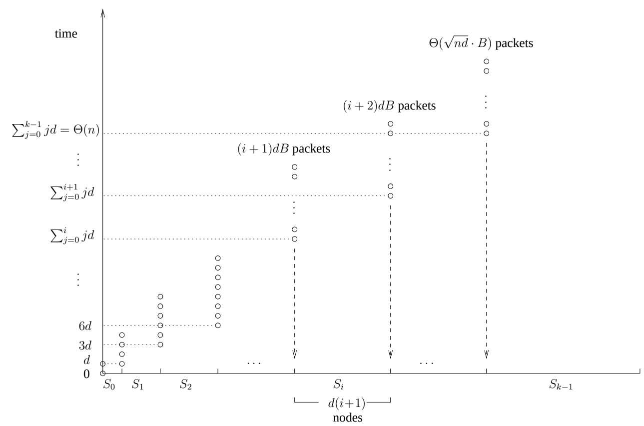

Figure 3: Outline of the injection pattern for the adversary showing the $\Omega(r)$ lower bound. The X-axis represents the line network, and each circle represents the injection of a packet. In every segment S_i except for S_0, out of the $(i+1)dB$ packets injected, only B packets would be absorbed by the greedy policy.

First note that by the choice of k, we have $k \leq \sqrt{\frac{n}{d}} \leq r$. Since the adversary injects at most one packet to every segment in every time step, we are guaranteed that the above injection sequence corresponds to an r-adversary. Furthermore, since $d \leq \lceil \frac{n}{16} \rceil$ we have that $k \geq 2$.

The following lemma enables us to bound the number of packets leaving every segment S_i under the greedy policy. The proof is by induction on the segment number, and is omitted.

LEMMA 3.7. *Under the greedy policy, for any $i \geq 1$, the packets leaving segment S_i form a continuous sequence of $(i+1)dB + B$ packets, which start arriving to S_{i+1} in time t_{i+1}.*

It follows that the greedy policy delivers $O(kdB) = O(\sqrt{dn} \cdot B) = O\left(\frac{nB}{r}\right)$ packets.

The number of packets injected to the network by the adversary is nB, and the adversary may successfully deliver them all by storing the $(i+1)dB$ packets injected to segment S_i in the buffers of that segment, and not forwarding any packet across different segments until the injection sequence has terminated.

It follows that the ratio between the number of packets delivered by the adversary and the number of packets delivered by the greedy policy is $\Omega(r)$. This concludes the proof of the theorem. □

THEOREM 3.8. *For any $\frac{16B}{n} < r \le B$ there exists an r-adversary A such that the ratio between the throughput of A and that of the greedy policy is $\Omega\left(\sqrt{\frac{rn}{B}}\right)$.*

PROOF. We consider the line as divided into equal-length segments, each of a length to be determined later. Given any rate $\frac{16B}{n} < r \le B$, we describe an adversary that inject at most one packet in every time step to every segment, and further show that the adversary does not inject to more than r segments in every time unit. This ensures that the injection sequence indeed corresponds to an r-adversary. The analysis will show that the overall number of packets accepted by the greedy policy is proportional to r times the segment length, whereas the adversary can accept all the packets injected.

Formally, Let $\frac{16B}{n} \le r \le B$, and let $d = \lfloor\sqrt{\frac{nB}{r}}\rfloor$. Consider the line as composed of $k = \lfloor\frac{n}{d}\rfloor = \Theta(\sqrt{\frac{rn}{B}})$ segments $S_0, \ldots, S_{k-1}$, each of length d. Note that by our assumption that $\frac{16B}{n} < r$, we are guaranteed to have $d < \frac{n}{4}$, and $k \ge 4$. We describe the sequence of injections generated by an r-adversary A:

For every $i = 0, \ldots, k-1$, A injects $\lfloor rd \rfloor$ packets to the first node of segment S_i, starting at time id.

Note that the above adversary injects at most one packet into every segment in every time step, and does not inject into more than r segments simultaneously. It follows that the above injection sequence corresponds to an r-adversary. The following lemma enables us to bound the number of packets leaving every segment S_i under the greedy policy. The proof is by induction on the segment number, and is omitted.

LEMMA 3.9. *Under the greedy policy, the packets leaving segment S_i form a continuous sequence of $\lfloor rd \rfloor + iB$ packets, which start arriving to S_{i+1} in time $d(i+1)$.*

It follows that the greedy policy absorbs and delivers $O(rd + kB) = O(\sqrt{rnB})$ packets.

The number of packets injected to the network by the adversary is $\lfloor rd \rfloor \cdot \lfloor\frac{n}{d}\rfloor = \Theta(rn)$, and the adversary may successfully deliver them all by storing the $\lfloor rd \rfloor$ packets injected to segment S_i in the buffers of that segment, and not forwarding any packet across different segments until the injection sequence has terminated.

It follows that the ratio between the number of packets delivered by the adversary and the number of packets absorbed by the greedy policy is $\Omega\left(\frac{rn}{\sqrt{rnB}}\right) = \Omega\left(\sqrt{\frac{rn}{B}}\right)$. This completes the proof of the theorem. □

Aiello et al. present in [2] an $\Omega(\sqrt{n})$ lower bound on the competitive ratio of the greedy policy, which is independent of B, by presenting an adversary which can deliver all the packets it injects, while any greedy policy cannot deliver more than an $O(\sqrt{n})$ fraction of the packets injected. The following lemma shows a bound on the rate of this adversary.

LEMMA 3.10. *For any buffer size B, there exists an adversary A with rate $r = \min\{B, \sqrt{n}\}$, such that the ratio between the throughput of A and that of the greedy policy is $\Omega(\sqrt{n})$.*

PROOF. The adversary used by Aiello et al. in the proof that the greedy policy cannot have competitive ratio better than $\Omega(\sqrt{n})$, is a special case of the adversary described in Section 3.2. The main difference is that their adversary uses a "stretch" factor of $d = 1$, instead of the factor $c\frac{n}{r^2}$ used in Section 3.2.

Formally, the adversary considers the line as divided into k blocks, $S_1, \ldots, S_k$, such that the length of block S_i is i, and it injects iB packets into the first node of S_i, starting from time

$$t_i = \sum_{j=1}^{i} j = \frac{i(i+1)}{2}.$$

Note that the number of segments k must satisfy $\sum_{i=1}^{k} i \le n$. We can therefore choose $k = \lfloor\sqrt{n}\rfloor$.

Clearly this adversary has rate at most $\sqrt{n}$, since the number of segments is at most $\sqrt{n}$, and it injects at most one packet to every segment in every time unit.

We now show that the rate of this adversary is bounded by B. Note that for every i we have

$$\begin{aligned} t_{i+B} - t_i &= \tfrac{(i+B)(i+B+1)}{2} - \tfrac{i(i+1)}{2} = \\ &= \tfrac{1}{2}\left((i+B)(i+B+1) - i(i+1)\right) = \\ &= \tfrac{1}{2}\left(i^2 + 2iB + B^2 + i + B - i^2 - i\right) = \\ &= \tfrac{1}{2}\left(B^2 + (2i+1)B\right) > iB. \end{aligned}$$

It follows that by the time there are packets injected to segment S_{i+B}, there are no longer packets injected to any segment S_j, for $j \le i$. Hence, the adversary has rate at most B since the number of segments to which it injects simultaneously is at most B. □

3.3 Tight Results for High Rates

In this section we conclude the results of the previous sections and derive bounds, which are tight up to constant factors, on the competitive ratio of the greedy policy for any r-adversary such that $r > 1$. We distinguish between several ranges for r. See Table 1 for a summary of the results.

For the range of $r \ge \min\{B, \sqrt{n}\}$, the upper bound appearing in [5] guarantees a competitive ratio of $O(\sqrt{n})$. By Lemma 3.10, for this range of r, there exists an r-adversary which shows that the greedy policy cannot have a competitive ratio better than $\Omega(\sqrt{n})$.

The remaining range to consider is when $1 < r < \min\{B, \sqrt{n}\}$. Assume first that $\max\{1, \frac{n}{B}\} < r < \min\{B, \sqrt{n}\}$. Theorem 3.3 gives an upper bound of $O(r)$. Theorem 3.6 gives a lower bound of $\Omega(r)$ for the case $r > 4$ (if $r \le 4$ the upper bound guaranteed by Theorem 3.3 is $O(1)$). Assume now that $1 < r \le \frac{n}{B}$. Theorem 3.2 gives an upper bound of $O\left(\sqrt{\frac{rn}{B}}\right)$. Theorem 3.8 gives a lower bound of $\Omega\left(\sqrt{\frac{rn}{B}}\right)$, for $r > \frac{16B}{n}$ (if $r \le \frac{16B}{n}$ the upper bound guaranteed by Theorem 3.2 is $O(1)$).

4. THE CASE OF $B = 1$

The case of $B = 1$ is a special case for which the competitive ratio of the greedy protocol is bad. For rates $r \ge 1$ it follows easily from Theorems 4.2 and 5.1 in [2] that the competitive ratio of the greedy protocol is $\Theta(n)$. For $r \le 1/n$ the greedy policy is optimal, since every packet is delivered before the next one can be injected. For $1/n < r < 1$ we have the following theorem:

LEMMA 4.1. *The greedy policy has competitive ratio $\Theta(rn)$ against any r-adversary such that $1/n < r < 1$.*

PROOF SKETCH. For the upper bound, note that since every packet accepted by the greedy policy is delivered by at most n time units after its injection, the number of packets accepted by the adversary, but dropped by greedy due to this packet, is at most $\lceil rn \rceil$. Hence the competitive ratio of the greedy policy against any r-adversary with $1/n < r < 1$ is $O(rn)$. For the lower bound, an adversary similar to the one used in [2] which injects a packet to the first node in the system, and then another packet every $\lceil 1/r \rceil$ time units (thus corresponding to an r-adversary), to the node where the first packet is currently stored, shows that the competitive ratio of the greedy algorithm is $\Omega(rn)$. □

5. CONCLUSIONS

In this paper we are interested in the question of how does the size of the buffers deployed in the network, and the injection rate of the traffic into the network, influence the attainable throughput-competitive ratio of scheduling and admission protocols. We initiate a study in the framework of the CNT model of a more refined analysis of the competitive ratio of the throughput that takes into account not only the size of the network but also the size of the buffers and the rate of injection of the traffic. We study the special case of the line network and the problem of information gathering (all packet are destined to the same node), and give tight bounds on the competitive ratio as a function of these parameters. Interestingly, these bounds are different for different combinations of buffer-size and adversary-rate. For example, we show that for very small rates, insufficient buffer size may be the difference between the greedy protocol achieving optimal throughput, and non-optimal throughput. Furthermore, for larger rates, we show that increasing the buffer size may help up to a certain point, whereas any further increase no longer helps the greedy protocol to achieve a better competitive ratio, and its performance depends solely on the rate of the adversary.

We believe that the questions and analysis introduced in this paper may lead to a better understanding of the interplay between the buffer size and the adversary rate, and the competitive ratio attainable by local-control protocols. Our work raises several interesting open problems. For example, can similar results be obtained for more involved topologies, and other protocols. Another interesting question is whether one can design protocols that would take advantage of the given buffer size in order to reduce the competitive ratio when possible.

Acknowledgments

We thank Zvi Lotker for many useful discussions.

6. REFERENCES

[1] W. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosén. Competitive Queue Policies for Differentiated Services. *Journal of Algorithms*, 55(2):113–141, 2005.

[2] W. Aiello, R. Ostrovsky, E. Kushilevitz, and A. Rosén. Dynamic Routing on Networks with Fixed-Sized Buffers. In *Proceedings of the 14th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 771–780, 2003.

[3] S. Albers and M. Schmidt. On the Performance of Greedy Algorithms in Packet Buffering. *SIAM Journal on Computing*, 35(2):278–304, 2005.

[4] N. Andelman and Y. Mansour. Competitive Management of Non-preemptive Queues with Multiple Values. In *Proceedings of the 17th International Symposium on Distributed Computing (DISC)*, pages 166–180, 2003.

[5] S. Angelov, S. Khanna, and K. Kunal. The Network as a Storage Device: Dynamic Routing with Bounded Buffers. In *Proceedings of the 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 1–13, 2005.

[6] Y. Azar and A. Litichevskey. Maximizing Throughput in Multi-Queue Switches. *Algorithmica*, 45(1):69–90, 2006.

[7] Y. Azar and Y. Richter. An Improved Algorithm for CIOQ Switches. In *Proceedings of the 12th Annual European Symposium on Algorithms (ESA)*, pages 65–76, 2004.

[8] Y. Azar and Y. Richter. Management of Multi-Queue Switches in QoS Networks. *Algorithmica*, 43(1-2):81–96, 2005.

[9] Y. Azar and R. Zachut. Packet Routing and Information Gathering in Lines, Rings and Trees. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, pages 484–495, 2005.

[10] L. Epstein and R. V. Stee. Buffer Management Problems. *ACM SIGACT News*, 35(3):58–66, September 2004.

[11] C. Florens, M. Franceschetti, and R. J. McEliece. Lower Bounds on Data Collection Time in Sensory Networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1110–1120, 2004.

[12] E. Gordon and A. Rosén. Competitive Weighted Throughput Analysis of Greedy Protocols on DAGs. In *Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 227–236, 2005.

[13] A. Kesselman, Z. Lotker, Y. Mansour, and B. Patt-Shamir. Buffer Overflows of Merging Streams. In *Proceedings of the 11th Annual European Symposium on Algorithms (ESA)*, pages 349–360, 2003.

[14] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko. Buffer Overflow Management in QoS Switches. *SIAM Journal on Computing*, 33(3):563–583, 2004.

[15] A. Kesselman, Y. Mansour, and R. van Stee. Improved Competitive Guarantees for QoS Buffering. *Algorithmica*, 43(1-2):63–80, 2005.

[16] A. Kesselman and A. Rosén. Scheduling policies for CIOQ switches. *Journal of Algorithms*, 60(1):60–83, 2006.

[17] K. Kothapalli, M. Onus, A. Richa, and C. Scheideler. Efficient Broadcasting and Gathering in Wireless Ad-Hoc Networks. In *Proceedings of the 8th International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pages 346–351, 2005.

[18] K. Kothapalli and C. Scheideler. Information Gathering in Adversarial Systems: Lines and Cycles. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 333–342, 2003.