Analysis of Binary Adjustment Algorithms in Fair Heterogeneous Networks

Sergey Gorinsky Harrick Vin

Technical Report TR2000-32
Department of Computer Sciences, University of Texas at Austin
Taylor Hall 2.124, Austin, Texas 78712-1188, USA
{*gorinsky, vin*}*@cs.utexas.edu*

November 29, 2000

Abstract

Many congestion control schemes rely on binary notifications of congestion from the network: on detecting network congestion, they reduce transmission rates; and on receiving a signal indicating no congestion, they increase transmission rates. For conventional networks with First-In First-Out (FIFO) scheduling of packets, the effectiveness of such algorithms has been evaluated with respect to their responsiveness, smoothness, and fairness properties. Recently, it has been argued that it is possible to design high-speed network routers that can guarantee fair allocation of link capacities and buffers. In networks that employ such routers, fairness is ensured by the routers, thereby making responsiveness and smoothness the two main criteria for evaluating and selecting a binary adjustment algorithm.

In this paper, we consider binary adjustment algorithms with four increase policies proposed in the literature: multiplicative increase (MI), additive increase (AI), inverse-square-root increase (ISI), and inverse increase (II). We analyze these algorithms in fair heterogeneous networks. We find that the multiplicative increase policy, which is considered inappropriate for conventional networks due to its fairness property, provides superior performance over the other policies in fair networks.

1 Introduction

This paper studies congestion control schemes based on *binary adjustment algorithms* that adjust load on the network in response to a binary feedback about the congestion status of the network. Numerous congestion control schemes use binary adjustment algorithms. For instance, DECbit relies on the additive-increase multiplicative-decrease (AIMD) algorithm to adjust the load in response to an explicit binary feedback: if the feedback indicates congestion, the load is reduced to its fraction; otherwise, the load is raised by a constant [22]. Binary adjustment algorithms also enjoy wide deployment in the Internet where most of the traffic is subject to congestion control by Transmission Control Protocol (TCP) [3]: in the slow start mode, the congestion window of a TCP session is approximately doubled during each round-trip time when congestion is not detected; in the congestion avoidance mode of TCP, the adjustments of the congestion window are similar on the round-trip timescale to the behavior of AIMD [1, 10].

The design of binary adjustment algorithms for conventional networks – with First-In First-Out (FIFO) link scheduling – has been motivated by three requirements: *responsiveness* to congestion notifications; *smoothness* of rate adjustments; and *fairness* of resource allocation across flows [11, 12, 16, 29]. For instance, the large load oscillations characteristic of TCP has led to the development of several adjustment algorithms that provide smoother congestion control for streaming media applications [27]. To achieve the goal of smoothness, some solutions offer new settings for the parameters of the TCP adjustment algorithms [6, 30], while other proposals suggest replacing the TCP adjustment algorithm for the congestion avoidance mode by new algorithms such as the IIAD (inverse-increase additive-decrease) and SQRT algorithms [2].

The aim of this paper is to analyze the performance of binary adjustment algorithms in *fair networks* – networks in which routers instantiate fair resource allocation mechanisms such as fair link scheduling [5] and fair buffer management [26]. Fair networks have a number of advantages over traditional networks. For example, while the performance of a traditional network can be disrupted by the flows that do not exercise congestion control [18], fair networks offer protection against these nonadaptive flows. Furthermore, as long as flows employ some form of congestion control in a fair network (i.e., each flow decreases or increases its load depending on the congestion status), the network converges towards the fair allocation of its capacity [9]. The argument against fair networks has traditionally been the complexity, and hence the perceived lack of scalability, of the mechanisms for ensuring fairness in routers. However, recent studies suggest that fair resource allocation can be implemented in high-speed networks [26]; in fact, a number of manufacturers are currently designing routers with support for fair link scheduling [20]. Besides, there exist promising approaches to build simpler fair networks where core routers do not perform per-flow management [25]. We would like to point out that this paper does not argue for ubiquitous deployment of fair link scheduling or fair buffer management. It aims to establish which adjustment algorithms would be preferable in fair networks if such networks were to be deployed.

It is important to note that fair networks are characterized by inherent fairness; hence the design of adjustment algorithms in a fair network is driven solely by considerations of efficiency of resource utilization (i.e., the responsiveness and smoothness requirements). In conventional routers with FIFO link scheduling and Drop-Tail buffer management, on the other hand, the goal of achieving fairness restricts the choice of adjustment algorithms. For instance, the objective of TCP-friendliness in traditional networks [15, 19] couples the increase and decrease policies of an adjustment algorithm (and thus imposes an undesirable coupling between the speed of capacity acquisition and responsiveness to congestion): in GAIMD, the parameter setting of the decrease policy is determined by the parameter setting selected for its increase policy [30]; similarly, the choice of the increase policy for a binomial algorithm, such as IIAD and SQRT, dictates the decrease policy of the algorithm [2]. Congestion control schemes in fair networks do not need to coordinate their load adjustments policies in order to support fairness; rate adjustment algorithms can select the increase and decrease policies independently.

In this paper, we evaluate rate adjustment algorithms with respect to their efficiency in a fair network. Our analysis methodology has two unique features.

1. Due to the intrinsic fairness of resource allocation in fair networks, we conduct the evaluation of the algorithms in a new way. We examine the impact of a binary adjustment algorithm on the performance of a particular flow without making many unrealistic assumptions common for the analysis of traditional networks. Our methodology allows cross traffic to: (1) have different round-trip times, (2) be bottlenecked at different links, (3) use different adjustment algorithms, and (4) transmit less data than suggested by congestion control mechanisms.

2. We analyze binary adjustment algorithms in heterogeneous environments, where the capacity available to a flow changes over time. It is known that the efficiency of a binary adjustment algorithm is subject to a fundamental tradeoff between the smoothness and responsiveness of the algorithm: an algorithm with smoother oscillations of load at a steady state is less responsive to changing network conditions [4]. Earlier studies of the tradeoff between smoothness and responsiveness were conducted for relatively static network conditions [4]. Such an approach seems inappropriate since tuning the parameters of an algorithm for a particular network setting does not ensure good performance of the selected algorithm in diverse scenarios. For instance, consider the following additive algorithm A and multiplicative algorithm M: algorithm A adjusts the current load by 2 units; algorithm M adjusts the current load by 10%. When the fair share of load is 100 units, algorithm A is smoother at the fair state than algorithm M. If the fair share of load equals 10 units, algorithm M is smoother at the fair state than algorithm M. What is required in reality is an assurance that the examined algorithm provides acceptable performance for all possible (or important in practice) configurations resulting from the mix of network technologies as well as from the dynamic nature of network traffic. Our methodology establishes whether the evaluated algorithm provides an appropriate tradeoff between smoothness and responsiveness in fair heterogeneous networks.

Using this methodology, we analyze binary adjustment algorithms with four increase policies proposed in the literature: multiplicative increase (MI), additive increase (AI), inverse-square-root increase (ISI), and inverse increase (II). We find that the multiplicative increase policy, which is considered inappropriate for conventional networks due to its fairness property, provides superior performance than the other policies in fair networks.

Before proceeding to the main part of the paper, we would like to point out that adjustments of load in response to a binary congestion signal are not the only means of congestion control. Even though binary adjustment algorithms are routinely adopted by congestion control schemes for unicast [23, 30] and multicast [17, 24], adjustment algorithms can be more effective in congestion control designs with more sophisticated feedback. Examples of such schemes include the equation-based congestion control for traditional networks [7] or packet-pair protocols for fair networks [13, 14]. Our paper considers only binary adjustment algorithms. Assessment of non-binary adjustment algorithms and their comparison with binary algorithms lie beyond the scope of this paper.

The paper is organized as follows. First, we specify our model of fair networks in Section 2. The examined binary adjustment algorithms are presented in Section 3. Section 4 describes the theoretical foundations of our evaluation. Section 5 contains definitions and justifications for the chosen metrics of performance. Section 6 outlines our evaluation methodology. Analysis of the compared policies is provided in Section 7. Section 8 summarizes our conclusions.

2 Network Model

In this paper, we analyze the performance of a particular flow (called *the examined flow*) that employs a binary algorithm to adjust its *load* in a fair *network*. We model the network as the bottleneck link of this flow (see Figure 1). The network *capacity* C equals the capacity of this link and is a positive real number. The network is shared by n flows. At time t, flow k imposes load $l_k(t)$ on the network, where $l_k(t)$ is a positive real number. The *total load* on the network at time t equals:

$$L(t) = \sum_{k=1}^{n} l_k(t). \tag{1}$$

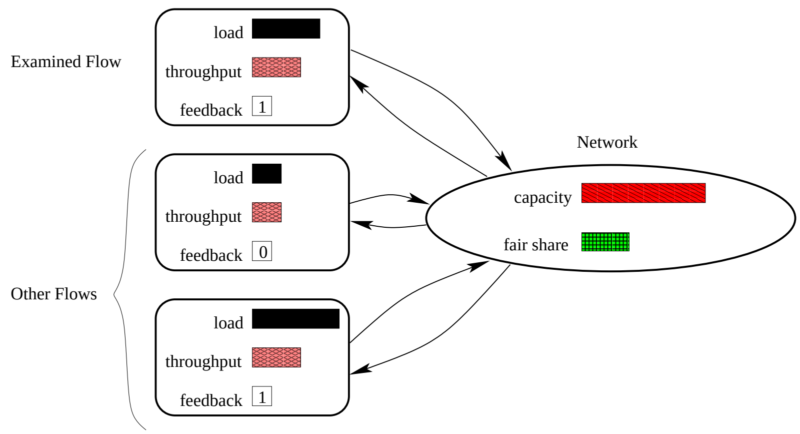

Figure 1: The network model.

The network splits its capacity between flows according to the principle of maxmin fairness [8, 11]. A recursive procedure for computing this fair allocation is given in [21]. The procedure assigns a *throughput* $b_k(t)$ to flow k based on the notion of *fair share* $s(t)$ at time t. If the flow demands less than the fair share, its demand is fully satisfied. Otherwise, the flow receives the fair share:

$$b_k(t) = \min\{l_k(t), s(t)\} \tag{2}$$

When $L(t) \leq C$, all the demands can be satisfied, and the fair share is assumed to be the maximum among the imposed loads. When $L(t) > C$, only the demands from a proper subset $p(t)$ of all the flows can be fully satisfied. The other flows split the remaining capacity equally:

$$s(t) = \begin{cases} \max\limits_{k=1}^{n}\{l_k(t)\} & \text{if } L(t) \leq C, \\ \frac{C - \sum\limits_{k \in p(t)} l_k(t)}{n - |\,p(t)\,|} & \text{if } L(t) > C \end{cases} \tag{3}$$

where

$$p(t) = \{\ k \mid l_k(t) \leq s(t)\ \}. \tag{4}$$

To facilitate efficient congestion control, the network provides flow k with binary *feedback* $f_k(t)$:

$$f_k(t) = \begin{cases} 0 & \text{if } l_k(t) \leq s(t), \\ 1 & \text{if } l_k(t) > s(t). \end{cases} \tag{5}$$

We examine the performance of a particular flow which adjusts its load in response to the network feedback. For succinctness of the notation, we omit the subscript when we refer to the characteristics of this flow: $l(t)$, $f(t)$, and $b(t)$ denote the load, feedback, and throughput of this flow respectively.

We model time as the number of adjustments performed by the examined flow. Thus, time is integer: $t = 0$ represents the moment when the examined flow imposes its initial load $l(0) = \lambda$ on the network; for $t > 0$, t corresponds to the t-th adjustment of the load for this flow.

The flow uses the following binary algorithm to adjust its load:

$$l(t+1) = \begin{cases} i(l(t)) & \text{if } f(t) = 0, \\ d(l(t)) & \text{if } f(t) = 1, \end{cases} \tag{6}$$

where i and d are an increase policy and decrease policy, respectively. We consider increase policies that always increase the load:

$$\forall l > 0 \quad i(l) > l \tag{7}$$

and are guaranteed to produce unbounded values if applied repetitively:

$$\forall x, l > 0 \quad \exists \tau \quad i^\tau(l) > x \tag{8}$$

where $i^\tau(l)$ is the result of τ consecutive applications of i to l.

Similar constraints are imposed on decrease policies: a decrease policy always decreases the load to a positive value:

$$\forall l > 0 \quad 0 < d(l) < l \tag{9}$$

and is guaranteed to produce a value below any positive number if applied repetitively:

$$\forall x, l > 0 \quad \exists \tau \quad d^\tau(l) < x \tag{10}$$

where $d^\tau(l)$ is the result of τ consecutive applications of d to l.

Constraints (7) and (9) implement the principle of negative feedback: when the load of the examined flow is below the fair share, the adjustment algorithm increases the load; when the load exceeds the fair share, the adjustment algorithm decreases the load of the examined flow [4]. Constraints (8) and (10) ensure that regardless of the initial load and fair share, the adjustment algorithm eventually brings the load of the examined flow to the fair share.

Our model does not make any assumptions about how and when the other flows adjust their loads on the network.

The next section presents the binary adjustment algorithms examined in this paper.

3 Binary Adjustment Algorithms

A binary adjustment algorithm consists of two components: an increase policy and a decrease policy. The following increase and decrease policies have been proposed in the literature.

- Increase policies:
    1. *Multiplicative Increase* (MI) policy: $i_\mu(l) = \mu l$ where $\mu > 1$ is a constant. This policy models the behavior of TCP during its slow start mode [1, 10].
    2. *Additive Increase* (AI) policy: $i_\alpha(l) = l + \alpha$ where $\alpha > 0$ is a constant. This policy models the increase behaviors of AIMD [22], GAIMD [30], and TCP congestion avoidance mode [1, 10].
    3. *Inverse-Square-root Increase* (ISI) policy: $i_\sigma(l) = l + \frac{\sigma}{\sqrt{l}}$ where $\sigma > 0$ is a constant. This policy represents the increase behavior of SQRT algorithm [2].
    4. *Inverse Increase* (II) policy: $i_\epsilon(l) = l + \frac{\epsilon}{l}$ where $\epsilon > 0$ is a constant. This policy is employed by IIAD algorithm [2].

    Note that all these four increase policies satisfy conditions (7) and (8).
- Decrease policies:

1. *Multiplicative Decrease* (MD) policy: $d_\beta(l) = \beta l$ where $0 < \beta < 1$ is a constant. This policy models the decrease behaviors of AIMD, GAIMD, and TCP.
2. *Square-root Decrease* (SD) policy: $d_\xi(l) = l - \xi\sqrt{l}$ where $\xi > 0$ is a constant. This policy is used by SQRT algorithm.
3. *Additive Decrease* (AD) policy: $d_\delta(l) = l - \delta$ where $\delta > 0$ is a constant. This policy models the decrease behavior of IIAD algorithm.

Note that neither AD nor SD satisfies condition (9): $d_\delta(l) < 0$ for $l = \frac{\delta}{2}$, and $d_\xi(l) < 0$ for $l = \frac{\xi^2}{4}$. To address this problem, one can modify AD and SD policies as follows: if the policy suggests a load that is not a positive value, then the load is set to some value $l^\star$ where $l^\star > 0$. However, these new decrease policies do not satisfy condition (10).

Since MD is the only proposed decrease policy that satisfies both conditions (9) and (10), we consider only binary adjustment algorithms that use MD as the decrease policy. Because fair networks eliminate the need for coupling the increase and decrease policies to achieve the fairness of resource utilization, our objective of comparing the binary adjustment algorithms reduces to comparison of the increase policies.

The next section provides a theoretical basis for our evaluation of the proposed increase policies.

4 Theoretical Foundations of Our Evaluation

We assess the performance of the examined flow controlled by a binary adjustment algorithm. The other flows in the network constitute cross traffic for the examined flow and can have diverse round-trip times, be bottlenecked at different links, employ various forms of congestion control (including, no congestion control at all), and transmit less data than suggested by their congestion control mechanisms. This is a realistic model of traffic for large heterogeneous networks. The fair allocation of resources in a fair network protects the examined flow from those flows that respond to congestion on a slower timescale or do not exercise any form of congestion control. If the examined flow is bottlenecked at a link with capacity C and n flows, then any binary adjustment algorithm that satisfies conditions (7) through (10) is guaranteed to raise the throughput of the examined flow in a fair network to $\frac{C}{n}$. This nice property holds regardless of the initial load for the examined flow or the behaviors of the other flows. We prove this property in Lemma 2 below and refer to

$$g = \frac{C}{n} \tag{11}$$

as a *guaranteed throughput*.

Lemma 1 *In the overloaded network, the fair share is at least the guaranteed throughput:*

$$(L(t) > C) \quad \Rightarrow \quad (s(t) \geq g). \tag{12}$$

Lemma 2 *The examined flow is assured to reach the guaranteed throughput:*

$$\exists t \quad b(t) \geq g. \tag{13}$$

Proofs for the lemmata are given in Appendix A. This appendix contains also a proof for the following theorem that shows why g is an important value:

Theorem 1 *g is the maximum throughput that the examined flow is guaranteed to reach.*

Since the other flows can be bottlenecked at different links and can transmit at smaller rates than suggested by their congestion control algorithms, the load of some flow can always be below the guaranteed throughput even if this flow employs the same binary adjustment algorithm as the examined flow. Thus, the fair share in the overloaded fair network can be anywhere between g and C. Due to the lack of assumptions about the behaviors of the other flows, Theorem 1 tells us as much as possible about either the assured or the expected performance of the examined flow in a fair network: g is the maximum throughput that the examined flow is guaranteed to reach. The ability to reach the guaranteed throughput serves as a foundation for our evaluation of the increase policies. The metrics for our evaluation are defined and justified in the next section.

5 Performance Metrics

We evaluate the increase policies with respect to their *responsiveness* – measured in terms of convergence time – and *smoothness* – measured in terms of overload.

- *Convergence time* $u(\lambda)$ of a policy refers to the amount of time it takes for the policy to increase the load of the examined flow from λ to the guaranteed throughput:

  $$u(\lambda) = \min_{b(t) \geq g} \{t\} \tag{14}$$

  This metric for convergence time can be expressed differently based on the following observation: as long as the throughput of the examined flow is below the guaranteed throughput, the load of the flow does not exceed the fair share, and the flow keeps increasing its load. Thus, we can transform (14) into a form which is more suitable for computation:

  $$u(\lambda) = \min_{i^t(\lambda) \geq g} \{t\} \tag{15}$$

- *Overload* v of a policy refers to the maximum relative increase produced by applying the policy to the fair share when the fair share reaches the guaranteed throughput:

  $$v = \max_{s(t) \geq g} \left\{ \frac{i(s(t)) - s(t)}{s(t)} \right\}. \tag{16}$$

  Our choice for the overload metric requires two clarifications.

  1. A seemingly better alternative is a measure that shows by how much the load of the examined flow exceeds its throughput after reaching the guaranteed throughput:

     $$v^\star = \max_{t > u(\lambda)} \left\{ \frac{l(t) - b(t)}{b(t)} \right\}. \tag{17}$$

     Unfortunately, as the following example illustrates, this measure depends on the behaviors of the other flows and is not suitable for representing the contribution of the evaluated increase policy to overload.

     Example 1 *Consider a fair network with capacity* 11 *and two flows. Let the examined flow employ the additive increase policy with parameter* $\alpha = 1$*. Assume that the load of the examined flow after* $(t-1)$ *adjustments is* $l(t-1) = 10$ *while the other flow imposes load of* 1 *at time* $(t-1)$*. Because* $s(t-1) = l(t-1) = 10$*, the examined flow increases its load at time t to* $l(t) = 11$*. If the other flow raises its load at time t to* 6*, then the fair share at time t becomes* $s(t) = 5.5$*. Since the examined flow exceeds the fair share at time* t*, its throughput* $b(t)$ *equals the fair share:* $b(t) = 5.5$*. Then, we have* $(l(t) - b(t))/b(t) = 100\%$*. According to (17), metric* $v^\star$ *is at least* 100%*. Note that such a high value of* $v^\star$ *is caused not by the increase policy of the examined flow (the examined flow increases its load from* 10 *to* 11*, i.e., by* 10%*) but by the drastic load increase of the other flow.* ■

     We would like to isolate the contribution of the evaluated policy to the overload from the contributions of the other flows. We believe that our metric (16) achieves this goal: it captures the scenarios when the examined flow has reached the guaranteed throughput, and the policy increases the load of the flow beyond the fair share.

  2. We measure relative overload rather than absolute overload because relative values are more suitable for reflecting the degree of congestion in a heterogeneous network. In real networks, overload manifests itself as high buffer occupancies – when binary feedback notifies flows about buffer buildups, as in Selective DECbit [21] – and packet losses. The same absolute loss rate of 20 $Kbps$ represents severe congestion for a 50 $Kbps$ link (the losses amount to 40% of the link capacity) but can be considered negligible for a 1 $Gbps$ link (0.002% of the link capacity). A similar observation can be made about overload evaluation in terms of buffer occupancies. Since the size of a link buffer is recommended to be proportional to the capacity of the link [28], buffer sizes can vary significantly in the network. In this situation of high heterogeneity, stating the relative buffer occupancy (e.g, 90%) reveals more information about the congestion status than providing the absolute buffer occupancy (e.g., 10 $Kbytes$). Due to these considerations, we report overload in relative units.

6 Evaluation Methodology

Since the capacities of links, the number of flows, and the locations of bottlenecks can vary dramatically in heterogeneous networks, we assume that the guaranteed load g is not known a priori but lies between some positive values g_{min} and g_{max}:

$$g \in [g_{min}, g_{max}]. \tag{18}$$

We refer to

$$\gamma = \frac{g_{max}}{g_{min}} \tag{19}$$

as a *heterogeneity index* of the network, $\gamma \geq 1$.

We assume that the values of g_{min} and g_{max} are known (either from gathered statistics on network usage or due to some form of admission control). We strive to evaluate each policy in terms of its ability to provide an acceptable behavior for every value of g that is between g_{min} and g_{max}.

Since the convergence time of the examined flow depends on its initial load λ, the choice of the initial load is an important issue. If the selected λ were such that $\lambda > g_{min}$, then the initial load of the examined flow could exceed the fair share in the scenarios when $g < \lambda$. We believe that such initial overload is undesirable. Thus, λ should be at most g_{min}. On the other hand, setting λ to a value below g_{min} does not seem appropriate because the convergence from this value to any g between g_{min} and g_{max} would include an additional time interval when the load is increased from λ to g_{min}. Since we assume that the value of g_{min} is known (from network statistics or due to admission control), we choose the initial load λ of the examined flow to be the minimum guaranteed throughput:

$$\lambda = g_{min}. \tag{20}$$

As it is shown in [4], the minimization of convergence time and the minimization of overload are conflicting objectives. Besides, as the following example illustrates, different parameter settings of a policy as well as different values of the guaranteed throughput produce different tradeoffs between convergence time and overload.

Example 2 *Consider an additive increase policy A_1 with parameter settings $\alpha = 1$. If $g_{min} = 4$ and $g = 10$, then convergence time for A_1 is 6 adjustments. Further, after reaching the guaranteed throughput of $g = 10$, Policy A_1 increases load to 11, thereby causing an overload of 10%. If $g = 20$ instead, then A_1 requires 16 adjustments to reach the guaranteed load and causes 5% overload.*

Now consider another additive increase policy A_2 with $\alpha = 2$. Policy A_1 is smoother but less responsive than A_2. When $g = 10$, A_2 converges from g_{min} to g after 3 adjustments but incurs an overload of 20%; and when $g = 20$, A_2 converges after 8 adjustments and incurs an overload of 10%. ■

To characterize the ability of a policy to provide a satisfactory behavior over the whole range of possible guaranteed throughputs, we introduce a notion of feasibility of an increase policy with respect to responsiveness and smoothness requirements:

Definition 6.1 *An increase policy is* feasible *with respect to* responsiveness η *and* smoothness ν *iff there exists such a single setting for the parameters of the policy that:*

$$\forall g \in [g_{min}, g_{max}] \quad u(g_{min}) \leq \eta \ \wedge \ v \leq \nu. \tag{21}$$

To compare two policies qualitatively, we define a relation "more feasible than" and denote it as "$\supseteq$":

Definition 6.2 *Policy A is* more feasible than *policy B iff whenever policy B is feasible with respect to some responsiveness η and smoothness ν, policy A is feasible with respect to the same η and ν:*

$$A \supseteq B \quad \equiv \quad \forall \eta, \nu \geq 0 \ \ (B \textit{ is feasible with respect to } \eta \textit{ and } \nu) \Rightarrow (A \textit{ is feasible with respect to } \eta \textit{ and } \nu). \tag{22}$$

To assess an increase policy quantitatively, we measure the responsiveness of the policy when this policy provides acceptable performance in terms of its smoothness. First, we consider such parameter settings of the policy that the overload does not exceed the smoothness requirement ν. We refer to them as ν-smooth settings:

Definition 6.3 *A parameter setting of a policy is* ν-smooth *iff:*

$$\forall g \in [g_{min}, g_{max}] \quad v \leq \nu. \tag{23}$$

policy	MI	AI	ISI	II
overload v	$\mu - 1$	$\frac{\alpha}{g}$	$\frac{\sigma}{g^{\frac{3}{2}}}$	$\frac{\epsilon}{g^2}$
convergence time $u(\lambda)$	$\left\lceil \log_\mu \frac{g}{\lambda} \right\rceil$	$\left\lceil \frac{g-\lambda}{\alpha} \right\rceil$	(15)	(15)
feasibility conditions	$\gamma \leq (1+\nu)^\eta$	$\gamma \leq 1 + \eta\nu$	(21)	(21)
ν-smooth parameter settings	$\mu \leq 1 + \nu$	$\alpha \leq \nu g_{min}$	$\sigma \leq \nu g_{min}^{\frac{3}{2}}$	$\epsilon \leq \nu g_{min}^2$
guaranteed convergence time ρ	$\left\lceil \log_{(1+\nu)} \gamma \right\rceil$	$\left\lceil \frac{\gamma-1}{\nu} \right\rceil$	(24)	(24)

Table 1: The performances of increase policies.

Then, in the set of ν-smooth settings of the policy, we distinguish such a setting that provides the policy with the smallest maximum convergence time. We refer to this time as the guaranteed convergence time of this policy and use it as a quantitative measure of the policy performance:

Definition 6.4 *The* guaranteed convergence time ρ *of an increase policy with respect to smoothness* ν *is the smallest among the maximum convergence times of the policy when the parameter setting belongs to the set* S *of* ν*-smooth settings of the policy:*

$$\rho = \min_S \left\{ \max_{g \in [g_{min}, g_{max}]} \{u(g_{min})\} \right\}. \tag{24}$$

We bound the overload to compare the convergence times (rather than limiting the convergence time to compare the overloads) because a specific bound on overload – e.g., the buffer size when overload is measured in terms of the buffer occupancy – can correspond to a boundary between two qualitatively different modes of network operation – e.g., lossless transmission versus packet drops. On the other hand, it is difficult to provide specific bounds on convergence times such that exceeding them results in qualitatively different performances.

Using the described methodology, we compare increase policies in the next section.

7 Analysis

We analyze the four increase policies introduced in Section 3: multiplicative increase (MI), additive increase (AI), inverse-square-root increase (ISI), and inverse increase (II). We present our findings as a series of lemmata below. While the proofs for the lemmata are given in Appendix B, Table 1 summarizes the results of our analysis. For some properties of ISI and II policies, closed-form expressions could not be obtained, and the table refers to the general definitions (15), (21), and (24) in these cases.

First, we derive the values of overload and convergence time for the considered policies:

Lemma 3 *The values of overload* v *for* MI*,* AI*,* ISI*, and* II *policies are* $(\mu - 1)$, $\frac{\alpha}{g}$, $\frac{\sigma}{g^{\frac{3}{2}}}$, *and* $\frac{\epsilon}{g^2}$ *respectively.*

Lemma 4 *The values of convergence time* $u(\lambda)$ *for* MI *and* AI *policies are* $\left\lceil \log_\mu \frac{g}{\lambda} \right\rceil$ *and* $\left\lceil \frac{g-\lambda}{\alpha} \right\rceil$ *respectively.*

Having obtained the closed-form expressions for both overload and convergence time of MI and AI policies, we can derive feasibility conditions as well as closed-form expressions for guaranteed convergence times of these policies:

Lemma 5 MI *is feasible with respect to responsiveness* η *and smoothness* ν *iff:*

$$\gamma \leq (1+\nu)^\eta. \tag{25}$$

Lemma 6 AI *is feasible with respect to responsiveness* η *and smoothness* ν *iff:*

$$\gamma \leq 1 + \eta\nu. \tag{26}$$

Lemma 7 *The values of guaranteed convergence time ρ for* MI *and* AI *policies are* $\left\lceil \log_{(1+\nu)} \gamma \right\rceil$ *and* $\left\lceil \frac{\gamma-1}{\nu} \right\rceil$ *respectively.*

The derived values of guaranteed convergence times for MI and AI policies do not depend on the minimum guaranteed throughput g_{min}. The following two lemmata show that ISI and II policies possess the same property:

Lemma 8 *The guaranteed convergence time of* ISI *policy does not depend on the minimum guaranteed throughput g_{min}.*

Lemma 9 *The guaranteed convergence time of* II *policy does not depend on the minimum guaranteed throughput g_{min}.*

The main results of our paper are formulated by the following three theorems:

Theorem 2 MI $\supseteq$ AI.

Proof:

$$
\begin{array}{ll}
 & \text{AI is feasible with respect to responsiveness } \eta \text{ and smoothness } \nu \\
\equiv & \{ \text{Lemma 6} \} \\
 & \gamma \leq 1 + \eta\nu \\
\Rightarrow & \left\{ \text{binomial series: } (1+\nu)^{\eta} = 1 + \eta\nu + \sum_{k=2}^{\eta} \binom{\eta}{k} \nu^k \right\} \\
 & \gamma \leq (1+\nu)^{\eta} \\
\equiv & \{ \text{Lemma 5} \} \\
 & \text{MI is feasible with respect to responsiveness } \eta \text{ and smoothness } \nu.
\end{array}
$$

According to Definition 6.2, MI $\supseteq$ AI. ■

Theorem 3 AI $\supseteq$ ISI.

Proof: Let us denote the convergence time and overload of ISI policy with parameter σ as u_σ and v_σ respectively. Then, consider AI policy with parameter $\alpha = \frac{\sigma}{\sqrt{g_{min}}}$ and denote its convergence time and overload as u_α and v_α respectively.

Let us compare the results of applying these ISI and AI policies to some g_1 and g_2 such that $g_{min} \leq g_1 \leq g_2$.

$$
\begin{array}{ll}
 & i_\sigma(g_1) \\
= & \{ \text{Definition of ISI policy} \} \\
 & g_1 + \dfrac{\sigma}{\sqrt{g_1}} \\
= & \\
 & g_1 + \dfrac{\sigma}{\sqrt{g_{min}}}\sqrt{\dfrac{g_{min}}{g_1}} \\
= & \left\{ \alpha = \dfrac{\sigma}{\sqrt{g_{min}}} \right\} \\
 & g_1 + \alpha\sqrt{\dfrac{g_{min}}{g_1}} \\
\leq & \{ g_{min} \leq g_1 \} \\
 & g_1 + \alpha \\
\leq & \{ g_1 \leq g_2 \} \\
 & g_2 + \alpha \\
= & \{ \text{Definition of AI policy} \} \\
 & i_\alpha(g_2).
\end{array}
$$

Thus, $(g_{min} \leq g_1 \leq g_2) \Rightarrow (i_\sigma(g_1) \leq i_\alpha(g_2))$. By induction, $\forall\tau \;\; i_\sigma^\tau(g_{min}) \leq i_\alpha^\tau(g_{min})$. Using (15), we derive:

$$u_\alpha(g_{min}) \leq u_\sigma(g_{min}). \tag{27}$$

Relying on (27), we obtain:

$$\begin{array}{cl}
& \textsf{ISI} \text{ is feasible with respect to responsiveness } \eta \text{ and smoothness } \nu \\
\equiv & \{ \text{ Definition 6.1 } \} \\
& \exists\sigma \quad \forall g \in [g_{min}, g_{max}] \quad u_\sigma(g_{min}) \le \eta \;\wedge\; v_\sigma \le \nu \\
\equiv & \{ \text{ Lemma 3 } \} \\
& \exists\sigma \quad \forall g \in [g_{min}, g_{max}] \quad u_\sigma(g_{min}) \le \eta \;\wedge\; \dfrac{\sigma}{g^{\frac{3}{2}}} \le \nu \\
\equiv & \\
& \exists\sigma \quad (\forall g \in [g_{min}, g_{max}] \quad u_\sigma(g_{min}) \le \eta) \;\wedge\; \dfrac{\sigma}{g_{min}^{\frac{3}{2}}} \le \nu \\
\Rightarrow & \{ (27) \} \\
& \exists\alpha = \dfrac{\sigma}{\sqrt{g_{min}}} \quad (\forall g \in [g_{min}, g_{max}] \quad u_\alpha(g_{min}) \le \eta) \;\wedge\; \dfrac{\sigma}{g_{min}^{\frac{3}{2}}} \le \nu \\
\equiv & \\
& \exists\alpha \quad (\forall g \in [g_{min}, g_{max}] \quad u_\alpha(g_{min}) \le \eta) \;\wedge\; \dfrac{\alpha}{g_{min}} \le \nu \\
\equiv & \\
& \exists\alpha \quad \forall g \in [g_{min}, g_{max}] \quad u_\alpha(g_{min}) \le \eta \;\wedge\; \dfrac{\alpha}{g} \le \nu \\
\equiv & \{ \text{ Lemma 3 } \} \\
& \exists\alpha \quad \forall g \in [g_{min}, g_{max}] \quad u_\alpha(g_{min}) \le \eta \;\wedge\; v_\alpha \le \nu \\
\equiv & \{ \text{ Definition 6.1 } \} \\
& \textsf{AI} \text{ is feasible with respect to responsiveness } \eta \text{ and smoothness } \nu.
\end{array}$$

According to Definition 6.2, AI $\supseteq$ ISI. ■

Theorem 4 ISI $\supseteq$ II.

Proof: Let us denote the convergence time and overload of II policy i_ϵ with parameter ϵ as u_ϵ and v_ϵ respectively. Then, consider ISI policy i_σ with parameter $\sigma = \frac{\epsilon}{\sqrt{g_{min}}}$ and denote its convergence time and overload as u_σ and v_σ respectively.

First, examine the derivative for policy $i_\epsilon(g)$ when $g \ge g_{min} + \frac{\epsilon}{g_{min}}$:

$$i'_\epsilon(g) = \left(g + \frac{\epsilon}{g}\right)' = 1 - \frac{\epsilon}{g^2} \ge 1 - \frac{\epsilon}{(g_{min} + \frac{\epsilon}{g_{min}})^2} = \frac{(g_{min}^2 - \epsilon)^2 + 3\epsilon g_{min}^2}{(g_{min}^2 + \epsilon)^2} > 0.$$

Therefore, $i_\epsilon(g)$ is an increasing function for $g \ge g_{min} + \frac{\epsilon}{g_{min}}$.

Now, let us compare the results of applying these i_ϵ and i_σ policies to some g_1 and g_2 such that $g_{min} + \frac{\epsilon}{g_{min}} \le g_1 \le g_2$:

$$\begin{array}{cl}
& i_\epsilon(g_1) \\
\le & \{ \; i_\epsilon(g) \text{ is an increasing function for } g \ge g_{min} + \frac{\epsilon}{g_{min}} \; \} \\
& i_\epsilon(g_2) \\
= & \{ \text{ Definition of II policy } \} \\
& g_2 + \dfrac{\epsilon}{g_2} \\
= & \left\{ \; \sigma = \dfrac{\epsilon}{\sqrt{g_{min}}} \; \right\} \\
& g_2 + \dfrac{\sigma}{\sqrt{g_2}}\sqrt{\dfrac{g_{min}}{g_2}} \\
\le & \{ \; g_{min} \le g_2 \text{ because } g_{min} + \frac{\epsilon}{g_{min}} \le g_2 \; \} \\
& g_2 + \dfrac{\sigma}{\sqrt{g_2}}
\end{array}$$

$$= \quad \{ \text{ Definition of } \mathsf{ISI} \text{ policy } \}$$
$$i_\sigma(g_2).$$

Thus, $(g_{min} + \frac{\epsilon}{g_{min}} \leq g_1 \leq g_2) \Rightarrow (i_\epsilon(g_1) \leq i_\sigma(g_2))$. Using this and the fact that $i_\sigma(g_{min}) = i_\epsilon(g_{min}) = g_{min} + \frac{\epsilon}{g_{min}}$, we derive by induction that $\forall \tau \;\; i_\epsilon^\tau(g_{min}) \leq i_\sigma^\tau(g_{min})$. Then, according to (15), we have:

$$u_\sigma(g_{min}) \leq u_\epsilon(g_{min}). \tag{28}$$

Relying on (28), we obtain:

$$\begin{aligned}
& \mathsf{II} \text{ is feasible with respect to responsiveness } \eta \text{ and smoothness } \nu \\
\equiv \quad & \{ \text{ Definition 6.1 } \} \\
& \exists \epsilon \quad \forall g \in [g_{min}, g_{max}] \quad u_\epsilon(g_{min}) \leq \eta \;\; \wedge \;\; v_\epsilon \leq \nu \\
\equiv \quad & \{ \text{ Lemma 3 } \} \\
& \exists \epsilon \quad \forall g \in [g_{min}, g_{max}] \quad u_\epsilon(g_{min}) \leq \eta \;\; \wedge \;\; \frac{\epsilon}{g^2} \leq \nu \\
\equiv \quad & \\
& \exists \epsilon \quad (\forall g \in [g_{min}, g_{max}] \quad u_\epsilon(g_{min}) \leq \eta) \;\; \wedge \;\; \frac{\epsilon}{g_{min}^2} \leq \nu \\
\Rightarrow \quad & \{ \text{ (28) } \} \\
& \exists \sigma = \frac{\epsilon}{\sqrt{g_{min}}} \quad (\forall g \in [g_{min}, g_{max}] \quad u_\sigma(g_{min}) \leq \eta) \;\; \wedge \;\; \frac{\epsilon}{g_{min}^2} \leq \nu \\
\equiv \quad & \\
& \exists \sigma \quad (\forall g \in [g_{min}, g_{max}] \quad u_\sigma(g_{min}) \leq \eta) \;\; \wedge \;\; \frac{\sigma}{g_{min}^{\frac{3}{2}}} \leq \nu \\
\equiv \quad & \\
& \exists \sigma \quad \forall g \in [g_{min}, g_{max}] \quad u_\sigma(g_{min}) \leq \eta \;\; \wedge \;\; \frac{\sigma}{g^{\frac{3}{2}}} \leq \nu \\
\equiv \quad & \{ \text{ Lemma 3 } \} \\
& \exists \sigma \quad \forall g \in [g_{min}, g_{max}] \quad u_\sigma(g_{min}) \leq \eta \;\; \wedge \;\; v_\sigma \leq \nu \\
\equiv \quad & \{ \text{ Definition 6.1 } \} \\
& \mathsf{ISI} \text{ is feasible with respect to responsiveness } \eta \text{ and smoothness } \nu.
\end{aligned}$$

According to Definition 6.2, ISI $\supseteq$ II. ■

Theorems 2, 3, and 4 establish an interesting chain of superiorities in terms of the abilities of the considered policies to satisfy the smoothness and responsiveness requirements:

$$\mathsf{MI} \supseteq \mathsf{AI} \supseteq \mathsf{ISI} \supseteq \mathsf{II} \tag{29}$$

i.e., MI is superior to AI which is superior to ISI which is superior to II. Thus, MI provides the best performance in fair heterogeneous networks in comparison to the other examined increase policies.

We assess quantitative advantages of MI over AI, ISI, and II in terms of the guaranteed convergence times of the compared policies. According to Lemma 7, the guaranteed convergence times of MI and AI policies depend only on the smoothness requirement ν and the heterogeneity index γ of the network. In particular, the guaranteed convergence times of these policies do not depend on the minimum guaranteed throughput g_{min}. Lemmata 8 and 9 show that ISI and II policies share the same property. Thus, we evaluate the guaranteed convergence times of the four compared policies as functions of the heterogeneity index (see Figure 2) and smoothness requirement (see Figure 3). Figure 2 shows that the larger heterogeneity index for the network, the larger advantage MI provides in comparison to the other considered policies. Figure 3 shows that MI consistently provides better performance than AI, ISI, and II policies for all considered smoothness requirements.

8 Summary and Discussion

In this paper, we analyze binary adjustment algorithms in fair heterogeneous networks. We introduce a network model where routers allocate link capacities among flows according to the principle of maxmin fairness. We evaluate four different increase