# AIRA: Additive Increase Rate Accelerator

Ioannis Psaras and Vassilis Tsaoussidis

Dept. of Electrical and Computer Engineering
Democritus University of Thrace,
12 Vas. Sofias Str., 67100, Xanthi, Greece
{ipsaras,vtsaousi}@ee.duth.gr
http://comnet.ee.duth.gr

**Abstract.** We propose AIRA, an Additive Increase Rate Accelerator. AIRA extends AIMD functionality towards adaptive increase rates, depending on the level of network contention and bandwidth availability. In this context, acceleration grows when resource availability is detected by goodput/throughput measurements and slows down when increased throughput does not translate into increased goodput as well. Thus, the gap between throughput and goodput determines the behavior of the rate accelerator.

We study the properties of the extended model and propose, based on analysis and simulation, appropriate rate decrease and increase rules. Furthermore, we study conditional rules to guarantee operational success even in the presence of symptomatic, extra-ordinary events. We show that analytical rules can be derived for accelerating, either positively or negatively, the increase rate of AIMD in accordance with network dynamics. Indeed, we find that the "blind", fixed Additive Increase rule can become an obstacle for the performance of TCP, especially when contention increases. Instead, sophisticated, contention-aware additive increase rates may preserve system stability and reduce retransmission effort, without reducing the goodput performance of TCP.

## 1 Introduction

We introduce a new paradigm for the responsive behavior of flows when bandwidth availability changes due to varying network contention. We call this paradigm, Additive Increase Rate Accelerator (AIRA), to reflect its operational perspective, namely to adjust the transmission rate of flows to current conditions of network contention. More precisely, we propose an algorithm, which progressively adjusts the Additive Increase factor of AIMD according to the current level of network contention. Typical systems adjust their rate, inevitably, only when contention leads to congestion; then, timeouts and multiplicative decreases force flows to reduce their windows. However, the transmission policy itself (i.e., increase/decrease rules) is not adjusted according to network contention. That is, although systems are adaptive to network dynamics, this adaptivity is limited: window size can be regulated but the window increase rate (i.e., Additive Increase factor) cannot. This is similar to a car regulating its velocity scale but with fixed acceleration.

By and large, the impact of contention on network dynamics, as well as the need for adjusting network parameters accordingly, is known to the networking community. Indeed, the conclusion is straightforward if we consider that the aggregate system rate increase is different in a system with 2 or 200 flows, which increases with a fixed rate of 1 packet per window for every successfully-acknowledged window. Consequently, our sample system with the same fixed increase/decrease parameters exhibits significantly different properties: it may reach stability or fail [15]; it may exploit bandwidth rapidly or waste available resources; it may require major retransmission effort or minimize overhead [13], [16].

In this context, we rely on two major concepts, to move beyond the confined perspective of predetermined and inflexible network parameters: (i) effort-based contention estimation and (ii) contention-oriented adaptive Additive Increase transmission. Effort, which is expressed as the ratio of Throughput over Goodput, reflects the efficiency of transmission strategy; when both throughput and goodput increase, bandwidth availability is clearly indicated, otherwise, as their gap widens, transmission effort is wasted. Consequently, increased contention will be reflected by higher protocol effort. In simple terms, a protocol that monitors transmission effort could approximate the dynamics of contention. That said, responses can be triggered accordingly. We investigate precisely the responsive strategies that correspond to various contention dynamics.

The proposed paradigm requires a number of issues to be addressed, prior to deployment. How accurately can we estimate changes of contention? What is practically the gain from a hypothetical transition to AIRA paradigm? What is the adaptation scheme of choice to varying contention? Are the properties of stability and fairness violated in favor of efficiency?

We address the aforementioned issues based on analysis and simulation experiments. In particular, we show that (i) contention can be estimated coarsely; coarse estimation suffices to enhance system properties; (ii) system fairness and stability can be preserved if we retain the Multiplicative Decrease response to congestion; and (iii) system efficiency can be increased if we adopt an adaptive Additive Increase response to available bandwidth. Adaptive increase allows for more sophisticated utilization of bandwidth, through more aggressive increase, when contention is low and less retransmission effort when contention is high. Furthermore, we go beyond conclusive statements to investigate the particular responsive strategy that corresponds to varying contention.

During our investigation, we uncovered a number of dynamics associated with heterogeneous RTTs, as well as with long- and short-lived flows. We show that adaptive increase favors short-lived flows, which is a desirable property if we consider a utilization-oriented (i.e., time-oriented) notion of fairness. Also, flows that experience long propagation delays tend to decrease their rate slower than flows that experience short propagation delays, which is another desirable property. Therefore, our proposal demonstrates high potential for deployment.

We organize the remaining paper as follows: In Section 2, we justify our research perspective and motives; in Section 3, we define our system model, which

includes definitions, observations on the dynamics of Additive Increase with diverse rates and proposed solution framework. We present AIRA in Section 4 and we propose adjustments to the proposed algorithm, due to practical constraints in Section 5. In Section 6, we evaluate the performance of the proposed algorithm through simulations and, finally, in Section 7, we conclude the paper.

## 2   Motivation

Deployment of AIMD is associated with two operational standards: (i) the fixed increase rate and decrease ratio and (ii) the corresponding selection of appropriate values.

Recent research has focused on altering the Additive Increase ($a$) and Multiplicative Decrease ($b$) values (e.g., [5], [8], [7], [21], [22], [18], [4], [19], [9], [2], [10], [11], [12], [24], [20], [6], [17], [1], [23]) but has not questioned really the validity and efficiency of fixed rates throughout the lifetime of participating flows. In this context, research efforts cannot address questions such as: Why do flows increase their rate by $a$ packets instead of $2a$ packets, even when half users of a system leave and bandwidth becomes available?

One possible justification for not highlighting that research direction is that:

*The Additive Increase factor of AIMD does not contribute to the long-term Goodput performance of TCP.*

In Figure 1, we present the *cwnd* evolution for two TCP flavors: Figure 1(a), where $a = 1$ (regular TCP) and Figure 1(b), where $a = 0.5$. The area underneath the solid *cwnd* lineplot (Area 1 and 2) represents the Goodput[1] performance of the protocols. In Figure 1(c), we show that both protocols achieve the same Goodput performance, since $A1 = A2$ and $A3 = A4$[2].

However,

*Additive Increase affects significantly the Retransmission Effort of flows, which impacts overall system behavior, as well.*

For example, TCP $a = 1$, in Figure 1, experiences 4 congestion events, while TCP $a = 0.5$ experiences only 2. Assuming that each congestion event is associated with a fixed number of lost packets, regular TCP (i.e., $a = 1$) will retransmit twice as many packets as TCP with $a = 0.5$, without any gain in Goodput performance.

Clearly, there is a tradeoff between *Aggressiveness* and *Retransmission Effort*. The degree of *Aggressiveness* that a transport protocol can achieve is tightly

---

[1] We define the system Goodput as $\frac{Original\ Data}{Connection\ Time}$, where *Original Data* is the number of bytes delivered to the high level protocol at the receiver (i.e., excluding retransmissions and the TCP header overhead) and *Connection Time* is the amount of time required for the data delivery. Instead, system Throughput includes retransmitted packets and header overhead (i.e., $\frac{Total\ Data}{Connection\ Time}$).

[2] In Figure 1(c) grey areas are common for both protocols; white areas are equal ($A1$ is similar to $A2$ and $A3$ is similar to $A4$).

(a) Increase Factor = 1   (b) Increase Factor = 0.5     (c) Difference
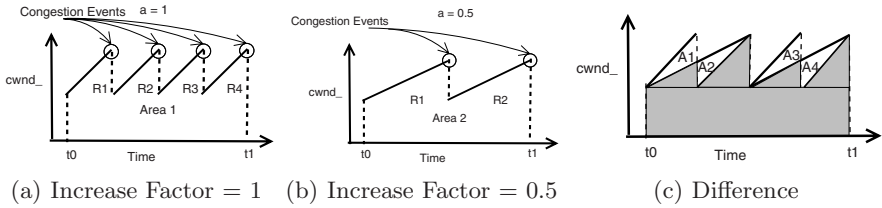
**Fig. 1.** Different Increase Factors

associated with its Retransmission Effort. The higher the Additive Increase factor, the more the retransmission effort of the transport protocol.

## 3   System Model

The initial study that investigated the operational properties of AIMD is [3]. In that study, the authors assume a feedback model, where all flows become aware of congestion events synchronously. In the current study, we extend this model to reflect more realistic situations. For example, in our model, different flows may become aware of congestion events at different points in time (i.e., congestion feedback is received asynchronously). A synchronous model, inherently assumes that flows do not experience queuing delays and hence, the duration of a *Round* (which is defined here as the interval between two *cwnd* multiplicative decreases) is in-varying for all flows. Instead, we allow for the possibility of queuing delays, which further means that the duration of a *Round* may differ among flows. Finally, we also allow for the possibility of multiple packet losses at the end of a *Round*.

### 3.1   Definitions

We define the following terms:

1. A *Round* is defined as the interval between two *cwnd* multiplicative decreases.
2. *Round Loss Rate ($p_i$)* is the ratio of the lost packets over the total number of sent packets, within *Round i*. The *Round Loss Rate* is calculated at the end of each *Round*.
3. The *Throughput Slope* within a Round is defined as $\frac{a}{cwnd}$. Obviously, the *Throughput Slope* is identical to the *cwnd Slope* (see Figure 2).
4. Assuming a *Round Loss Rate* $p_i$ and a *Round duration* $t_i$, the *Desired Throughput Slope* within a Round is defined as the *hypothetical cwnd Slope*, which would result in zero packet losses, within $t_i$, but without causing bandwidth underutilization. The *Desired Throughput Slope* is, therefore, determined by $\frac{a}{cwnd} \cdot (1 - p_i)$ (see Figure 2).
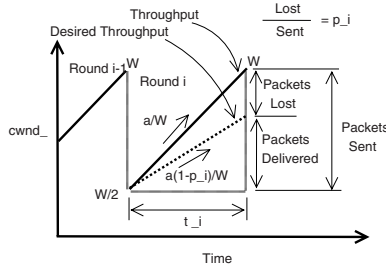
**Fig. 2.** Throughput / Desired Throughput Slopes at Round $i$

## 3.2 Observations on the Dynamics of Additive Increase with Diverse Rates

1. When $p_i > p_{i-1}$ then the *Throughput Slope* exceeds the *Desired Throughput Slope*, for *Round i*.
   – The greater the distance between $p_i$ and $p_{i-1}$, the wider the gap between the *Throughput* and *Desired Throughput Slopes* (i.e., the protocol is too aggressive).
2. When $p_i < p_{i-1}$ then the *Throughput Slope* is underneath the *Desired Throughput Slope*, for *Round i*.
   – The greater the distance between $p_i$ and $p_{i-1}$, the wider the gap between the *Desired Throughput* and *Throughput Slopes* (i.e., the protocol is too conservative).

Hence, our primary objective is to reduce the gap between the *Throughput* and *Desired Throughput Slopes*, in order to avoid extensive retransmission effort, or bandwidth under-utilization, respectively.

## 3.3 Solution Framework

We provide a solution framework in order to determine the primary requirements of our system model. We require from our system to:

1. *Converge to Fairness.*
   – We evaluate the Fairness properties of a protocol using the Fairness Index introduced in [3]:

$$Fairness = \frac{\sum (Throughput_i)^2}{n \sum (Throughput_i{}^2)},\qquad(1)$$

   where $Throughput_i$ is the Throughput performance of the $i^{th}$ flow and $n$ is the number of participating flows.
   – We introduce the *ALPHA Fairness Index* of multi-rate systems. We define *ALPHA Fairness Index* as:

$$ALPHA\ Fairness = \frac{\sum (a_i)^2}{n \sum (a_i{}^2)},\qquad(2)$$

where $a_i$ is the Additive Increase factor of the $i^{th}$ flow and $n$ is the number of participating flows.

The ALPHA Fairness Index has the following properties: In a homogeneous/synchronous static (contention-wise) system, increased ALPHA Fairness Index leads to increased System Fairness Index as well. In a heterogeneous/asynchronous dynamic system, however, the properties of ALPHA Fairness Index are not straightforward. For example, in a diverse-RTT system, reduced ALPHA Fairness Index may correspond to either increased or decreased system Fairness (i.e., reduced Additive Increase factor for longer-RTT flows results in reduced Fairness, while increased Additive Increase factor for longer-RTT flows results in increased system Fairness). We explore the above properties through simulations in the following sections.

2. *Guarantee Stability.*
   *Stability* is a quality measure, which we attempt to simplify and furthermore quantify by measuring the frequency of congestion events. Therefore, a protocol is said to achieve higher *Stability*, when it minimizes retransmission overhead.
3. *Exploit available resources efficiently*, which means:
   – *faster* (i.e., aggressively) when contention is low (i.e., utilize high percentage of available bandwidth).
   – *slower* (i.e., conservatively) when contention is high (i.e., minimize overhead/transmission effort).

*Efficiency* is achieved through high resource utilization (i.e., high goodput performance) *and* minimal retransmission overhead.

## 4   AIRA: Additive Increase Rate Accelerator

We assume that a flow can adjust its rate at the end of each *Round*. The flow at the end of *Round i* can determine the rate for round $i+1$ exploiting, recursively, the behavior of its rate during *Round i − 1*. In particular, the Decrease rule applies when $p_i > p_{i-1}$.

### 4.1   The Decrease Rule

**- Decrease Rule.** *In order to reduce the gap between the Throughput and Desired Throughput Slopes, the Additive Increase factor should decrease, according to Equation:*

$$a_{i+1} = a_i \cdot (1 - p_i). \tag{3}$$

We present the process followed by the AIRA *Decrease Rule* in Figure 3. At the end of *Round i*, the sender calculates the error rate $p_i$ and applies Equation 3 to the *cwnd* update function. Since $p_i > p_{i-1}$, the system operates in a moderated congestion environment, where the level of contention within the current round has increased compared to the previous one. Therefore, the sending rate at *Round*
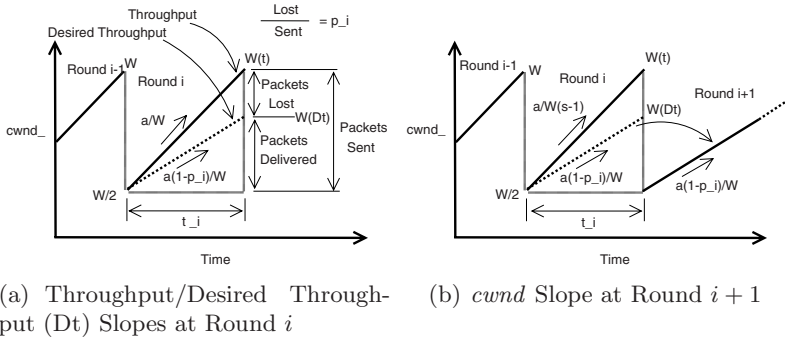
(a) Throughput/Desired Through-put (Dt) Slopes at Round $i$    (b) *cwnd* Slope at Round $i+1$

**Fig. 3.** Decrease Rule

$i + 1$ decreases in order to improve stability and system fairness, according to the *Desired Throughput* slope indicated by AIRA. Justification regarding the rationale behind the AIRA *Decrease Rule* setting can be found in [14].

## 4.2   The Increase Rule

According to the *Decrease Rule*, the Additive Increase factor of TCP (i.e., AIRA) may get non-increasing values. In that case, however, the system may never reach equilibrium; flows already existing in the system, possibly transmitting with $a < 1$, will not have the opportunity to compete (fairly) with new, incoming flows. Therefore, we introduce an *Increase Rule*, which applies when $p_i < p_{i-1}$.

**- Increase Rule.** *In order to reduce the gap between the Desired Throughput - Throughput Slopes, the Additive Increase factor should increase, according to Equation:*

$$a_{i+1} = a_i \cdot (1 + p_{i-1} - p_i). \tag{4}$$

AIRA interprets the decrease of the *Round* error rate (i.e., $p_i < p_{i-1}$) as a sign of contention decrease and therefore increases the Additive Increase factor
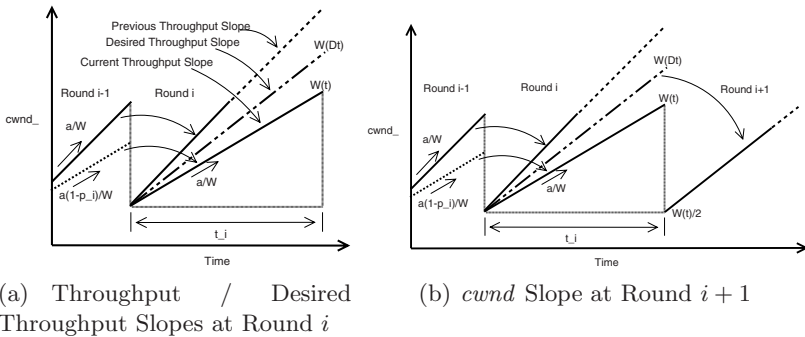


(a) Throughput / Desired Throughput Slopes at Round $i$    (b) *cwnd* Slope at Round $i+1$

**Fig. 4.** Increase Rule

of TCP in order to exploit available resources. This increase, however, should not be extended in order to preserve system stability. Therefore, the Additive Increase factor is increased according to the error rate difference during the last two rounds. Justification regarding the rationale behind the AIRA *Increase Rule* setting can be found in [14].

## 5    Adjustments due to Practical Constraints

### 5.1    Lower Bounds

We set lower bounds for AIRA for the following reason: Progressive reduction of the Additive Increase factor may result in transmission rate stabilization (i.e., $a = 0$). This, however, will inevitably cause system in-stability, protocol in-efficiency and flow starvation. We, therefore, bound $a$ to 0.1, attempting to avoid the above un-desirable system property. That said, the greatest possible reduction of the Additive Increase factor is 0.9 (i.e., $a_{greatest\ reduction} = 0.9$, when $a = 0.1$).

Furthermore, we complement this absolute bound with another, dynamically-adjustable lower bound, which depends on the number of completed *Rounds* and is called *Minimum Additive Increase Limit (MAIL)*. The reason is twofold:

1. A flow may lose several back-to-back packets, due to sudden traffic-bursts or symptomatic events. In order to limit drastic and systematic responses to symptomatic events, lower bounds need to be introduced. Otherwise, recovery from such symptomatic events may require significant effort and time or may even become impossible.
2. When contention reaches extra-ordinary levels, AIRA lower bound (i.e., $a = 0.1$) dominates and the rest of AIRA functionality is practically suspended. In that case, MAIL will prevent short-flow starvation, adhering to a time-oriented notion of Fairness (e.g., a flow at the second *Round* operating with $a = 0.1$). That is, assuming that short flows need to be favored over long, time-insensitive ones, *MAIL* is designed to provide more opportunities for data transmission to short, probably time-sensitive flows.

**Minimum Additive Increase Limit (MAIL).** We use the following equation to adjust MAIL:

$$MAIL = a_{min} = 1 - a_{greatest\ reduction} \cdot R_N\%, \qquad (5)$$

where, $R_N$ is the number of completed *Rounds* and $a_{greatest\ reduction} = 0.9$ (see Figure 5).

As a secondary effect, MAIL exhibits one desirable property, which was not an initial design goal: it favors long-RTT flows over shorter-RTT ones. That is, short-RTT flows complete a *Round* faster than long-RTT flows. Therefore, MAIL will get a lower value (i.e., Additive Increase factor) for a short-RTT flow faster than for a longer-RTT one. Hence, in case of moderated or high contention the short-RTT flows will operate with lower Additive Increase factor, avoiding this way starvation of long-RTT flows. We verify the above assumption through simulations in Section 6.
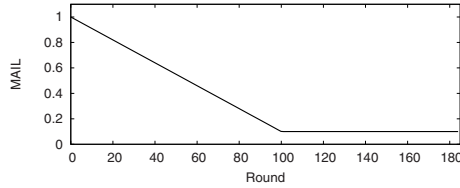
**Fig. 5.** Minimum Additive Increase Limit (MAIL)

### 5.2   Response to Contention Decrease

Although the AIRA *Increase Rule* (Section 4) proves to operate efficiently in static as well as in contention-increase scenarios, it fails to exploit extra available resources when contention decreases (i.e., $p_{i-1}-p_i$ is a small value, incapable of accelerating $a$ fast enough, see Equation (4)). Inline with our Solution Framework (Section 3.3), we apply a *Reset Condition* to AIRA, in order to prevent bandwidth wastage in case of contention decrease.

**AIRA Reset Condition.** Assuming that a flow's *cwnd* oscillates between $W/2$ and $W$ before the contention decrease event, the flow will have the opportunity to expand its *cwnd* to $W + \frac{1}{2}W = \frac{3}{2}W$ after $\frac{1}{3}$ of the participating flows leave the system. When this happens, AIRA resets $a$ to 1.

   System-wise, we assume that AIRA *should* exploit bandwidth fastly *iff* $\frac{n-x}{n} \leq \frac{2}{3}$, where $n$ is the total number of participating flows and $x$ is the number of flows who end their task and leave the system. In this case (i.e., when $x \geq \frac{1}{3}n$), AIRA resets $a$ to 1. Clearly, AIRA is slower in exploiting available resources compared to regular TCP (i.e., $a = 1$), but we consider this delay to be acceptable, according to our solution framework (see Section 3.3). In [14], we present an analysis regarding the convergence properties of AIRA in case of contention decrease.

## 6   Simulation Results

We attempt to verify the above considerations through simulations. Due to space limitations, we cannot present extended performance evaluations for AIRA in the current manuscript. We refer the interested reader to [14] for a variety of simulation scenarios and the corresponding results. Initially, we simulate 4 flows over the "Diverse-RTT Network Topology" (Figure 6(a)). We use Drop Tail routers, with buffer sizes equal to the Bandwidth-Delay Product of the outgoing links. We note that results are similar in case of Active Queue Management schemes, like RED for example. The round trip propagation delay for flows 1 and 2 is 60ms, while for flows 3 and 4 is 220ms. Simulation time is 300 seconds. Figure 6(b) depicts the Additive Increase factor for each flow. We see that MAIL provides more transmission opportunities to long-RTT flows (see Fairness Index in Table 1), than to short-RTT flows.

   We extend the above scenario to include more flows. The results are presented in Figure 7. In all cases (i.e., 4-flow scenario, Table 1 and extended scenario,
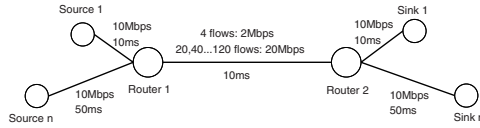
(a) Diverse-RTT Network Topology
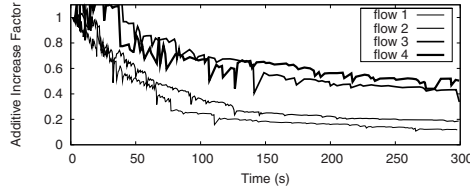


(b) Additive Increase Factor

**Fig. 6.**

**Table 1.** Performance Difference - RTT Un-Fairness

|  | Goodput | Retransmissions | Fairness | $a$ Fairness |
|---|---|---|---|---|
| AIMD | 200.6 KB/s | 2009 pkts | 0.6594 | 1.0 |
| AIRA | 209.5 KB/s | 943 pkts | 0.8768 | 0.787 |



(a) System Goodput



(b) Retransmission Overhead



(c) Fairness Index
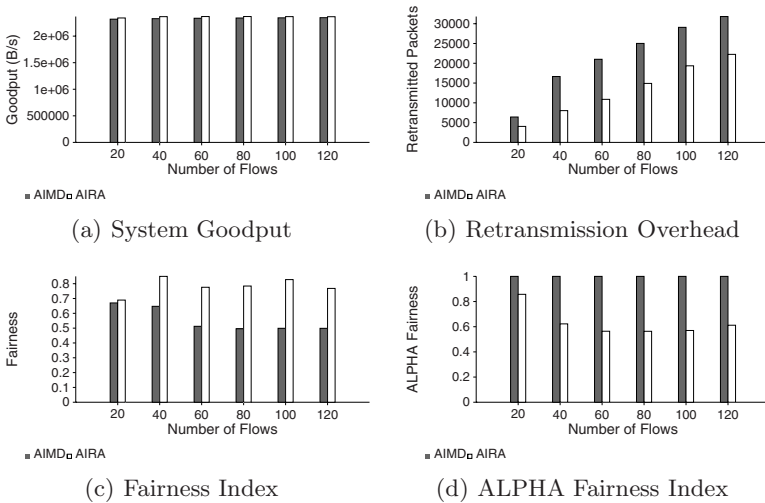


(d) ALPHA Fairness Index

**Fig. 7.** RTT Un-Fairness Scenario

Figure 7), we see that the proposed algorithm improves *Stability* (through far less retransmissions) and *Efficiency* (through less retransmissions *and* slightly increased Goodput). As contention increases, we notice considerable decrease of

the *ALPHA Fairness Index*, which indicates more fair resource allocation among short and long-RTT flows, when AIRA is used.

## 7    Conclusions

We have shown that analytical rules can be derived for accelerating, either positively or negatively, the increase rate of AIMD in accordance with network dynamics. Indeed, we found that the "blind" Additive Increase rule can become an obstacle for the performance of TCP, especially when contention increases. Instead, sophisticated, contention-aware additive increase rates may preserve system stability and reduce retransmission effort, without reducing the goodput performance of TCP.

Based on specific criteria, namely efficiency, fairness and stability, we proposed and evaluated an adaptive additive increase rate scheme, which we call Additive Increase Rate Accelerator (AIRA). We have shown two major results: (i) Fairness is possible even in the context of varying increase rates within the same system. Furthermore, the problematic balance among short and long flows, as well as long- and short-RTT flows, can be better handled. (ii) Efficiency can be improved. Efficiency is judged not only on the basis of goodput performance but mainly against retransmission effort and overhead.

Our primar future work direction is the theoretical evaluation of AIRA estimations regarding their accuracy. For instance, progressive contention is here estimated and is not explicitly communicated by some central network authority. The accuracy and precision of this estimation depends on two main factors: (i) the granularity of measurements and (ii) the accuracy of the monitoring functions. For example, the way throughput is calculated and the frequency of calculating throughput may have some impact. Higher (e.g., per packet) measurement granularity will probably increase precision, but it will also increase system entropy, leading to reduced system stability. Among others, this observation calls for further investigation.

## References

1. Bansal, D., Balakrishnan, H.: Binomial congestion control algorithms. In: INFO-COM, pp. 631–640 (2001)
2. Brakmo, L., O'Malley, S., Peterson, L.: TCP Vegas: New techniques for congestion detection and avoidance. In: Proceedings of SIGCOMM (1994)
3. Chiu, D.-M., Jain, R.: Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. Computer Networks and ISDN Systems 17(1), 1–14 (1989)
4. McCullagh, G., Leith, D.J., Shorten, R.N.: Experimental Evaluation of Cubic-TCP. In: Proceedings of PFLDnet (2007)
5. Floyd, S.: HighSpeed TCP for Large Congestion Windows, RFC 3649 (December 2003)
6. Floyd, S., Kohler, E.: Tcp friendly rate control (tfrc): the small-packet (sp) variant, rfc 4828, experimental (April 2007)

 7. Jin, S., Guo, L., Matta, I., Bestavros, A.: Tcp-friendly simd congestion control and its convergence behavior. In: Proc. 9th IEEE International Conference on Network Protocols (ICNP 2001), Riverside, CA, November 2001, vol. 5 (2001)
 8. Kelly, T.: Scalable TCP: Improving Performance in Highspeed Wide Area Networks. ACM SIGCOMM Computer Communication Review 33(2), 83–91 (2003)
 9. King, R., Baraniuk, R., Riedi, R.: Tcp-africa: An adaptive and fair rapid increase rule for scalable tcp. In: Proceedings of INFOCOM 2005, vol. 3, pp. 1838–1848 (2005)
10. Lahanas, A., Tsaoussidis, V.: Exploiting the efficiency and fairness potential of aimd-based congestion avoidance and control. Computer Networks 43(2), 227–245 (2003)
11. Lahanas, A., Tsaoussidis, V.: $\tau$-aimd for asynchronous receiver feedback. In: Proceedings of the Eighth IEEE International Symposium on Computers and Communications, ISCC (2003)
12. Marfia, G., Palazzi, C., Pau, G., Gerla, M., Sanadidi, M.Y., Roccetti, M.: TCP Libra: Exploring RTT-Fairness for TCP. UCLA Computer Science Department Technical Report TR050037 (2005)
13. Psaras, I., Tsaoussidis, V.: WB-RTO: A Window-Based Retransmission Timeout for TCP. In: In Proc. of the 49th IEEE GLOBECOM (November 2006)
14. Psaras, I., Tsaoussidis, V.: Aira: Additive increase rate accelerator. Technical Report, TR-DUTH-EE-2007-12 (2007),
    http://comnet.ee.duth.gr/comnet/files/aira-tech-report.pdf
15. Psaras, I., Tsaoussidis, V.: Why tcp timers (still) don't work well. Computer Networks 51(8), 2033–2048 (2007)
16. Psaras, I., Tsaoussidis, V., Mamatas, L.: CA-RTO: A Contention-Adaptive Retransmission Timeout. In: Proceedings of ICCCN (October 2005)
17. Rhee, I., Ozdemir, V., Yi, Y.: Tear: Tcp emulation at receivers – flow control for multimedia streaming. April 2000. NCSU Technical Report (2000)
18. Rhee, I., Xu, L.: CUBIC: A New TCP-Friendly High-Speed TCP Variant. In: Proceedings of PFLDnet (2005)
19. Tan, K., Song, J., Zhang, Q., Sridharan, M.: A compound tcp approach for high-speed and long distance networks.
20. Tsaoussidis, V., Zhang, C.: The dynamics of responsiveness and smoothness in heterogeneous networks. IEEE Journal on Selected Areas in Communications (JSAC) 23(6), 1178–1189 (2005)
21. Wei, D.X., Jin, C., Low, S.H., Hegde, S.: FAST TCP: motivation, architecture, algorithms, performance. IEEE/ACM Transactions on Networking (to appear, 2007)
22. Xu, L., Harfoush, K., Rhee, I.: Binary increase congestion control (bic) for fast long-distance networks. In: Proceedings of INFOCOM 2004, vol. 4, pp. 2514–2524 (March 2004)
23. Yang, Y.R., Lam, S.S.: General aimd congestion control. In: Proceedings of ICNP 2000 (October 2000)
24. Zhang, C., Tsaoussidis, V.: Tcp smoothness and window adjustment strategy. IEEE Transactions on Multimedia 8(3), 600–609 (2006)