Simulation of IPA Gradients in Hybrid Network Systems

Benjamin Melamed

Rutgers University
Rutgers Business School –
Newark and New Brunswick
Department of MSIS
94 Rockafeller Rd.
Piscataway, NJ 08854

Shuo Pan

Rutgers University
RUTCOR- Rutgers Center
for Operations Research
640 Bartholomew Rd.
Piscataway, NJ 08854

Yorai Wardi

Georgia Institute of
Technology
School of Electrical and
Computer Engineering
Atlanta, GA 30332

October 26, 2005

Abstract

Infinitesimal Perturbation Analysis (IPA) provides formulas for derivatives (gradients) of performance measures with respect to parameters of interest, computed from sample paths of stochastic systems. In practice, IPA derivatives may be computed either from simulation runs or from empirical field data (when the formulas are non-parametric). Nonparametric IPA derivatives in fluid-flow queues have been recently derived for the loss volume and time average of buffer occupancy, with respect to buffer size, and arrival-rate or service-rate parameters. Additionally, these IPA derivatives have been shown to be unbiased in the sense that their expectation and differentiation operators commute, while their discrete counterparts have long been known to be generally biased. Recent work has further shown how to map the computation of IPA derivatives from a fluid-flow queue to its discrete counterpart without an appreciable loss of accuracy in performance measures. Thus, this work holds out the promise of potential applications of IPA derivatives to gradient-based optimization of objective functions involving performance metrics parameterized by settable parameters in queueing network context.

This paper is an empirical study of IPA derivatives for individual queues within queueing systems which model telecommunications networks and some of their protocols. As a testbed, we used HNS (Hybrid Network Simulator) -- a hybrid Java simulator of queueing networks with traffic streams subject to several telecommunications protocols. More specifically, the hybrid feature of HNS admits models with mixtures of discrete (packet) flows and continuous (fluid) flows, and collects detailed statistics and IPA derivatives for all flow types. The paper outlines the mapping of IPA derivatives from the fluid domain to the packet domain as implemented in HNS, and studies the accuracy of IPA derivatives in compatible fluid and packet queueing models, as well as the stabilization of their values in time. Our experimental results lend empirical support to the contention that IPA derivatives can be accurately computed from discrete versions by adopting a fluid-flow view. Furthermore, the long-run values IPA derivatives are empirically shown to stabilize quite fast. Finally, the results provide the basis and motivation for IPA applications to optimization of telecommunications network design and to potential new open-loop protocols that take advantage of IPA information.

Keywords: Infinitesimal Perturbation Analysis (IPA), IPA Derivatives, IPA Gradients, Fluid-Flow Models, Fluid-Flow Simulation, Hybrid Simulation

1. Introduction

Monte Carlo simulation methods are widely used in analyzing complex queueing networks, where current analytical methods cannot provide closed-form or numerical solutions [Bratley et al. (1987), Law et al. (1991)]. In addition to standard queueing statistics (e.g., sojourn time, lost workload and buffer occupancy), ***infinitesimal Perturbation Analysis*** (***IPA***) can be used to obtain sensitivity information of statistics with respect to parameters of interest. ***IPA*** is a sample path technique for computing gradients (derivatives) of performance metrics (formulated as means) with respect to design/control parameters [Ho and Cao (1991), Cassandras and Lafortune (1999)]. Formally, let $L(\theta)$ be a real-valued random variable that depends on a parameter $\theta \in \Theta \subset \mathbb{R}$. Let further $l(\theta) = E(L(\theta))$ be a parameter-dependent expectation (performance function). The ***IPA derivative*** (***gradient***) of $L(\theta)$ is the random variable $L'(\theta) = \frac{d}{d\theta} L(\theta)$. If $E(L'(\theta)) = l'(\theta)$, then the IPA derivative is said to be unbiased (when it exists).

In queueing-system simulation, performance metrics include loss-related and workload-related metrics, while design parameters include buffer size and parameters of service and arrival processes. In principle, IPA derivatives provide a basis for research on control applications and design optimization applications for simulated systems, since simulation-based mean-derivative estimates can be used to optimize objective functions formulated in terms of performance metrics of interest. Such objective functions are often expressed in terms of cost functions of performance metrics, such as link loss rate, and time average of link buffer occupancy (or equivalently, of mean waiting time, by Little's Formula [Kleinrock (1975)]). The requisite IPA derivatives then can be used in simulation-based gradient-descent techniques to optimize system performance. Moreover, if the IPA derivatives can be shown to be nonparametric, then the aforementioned approaches can be applied to real-life systems. For example, one can imagine a telecommunications router that computes IPA derivatives, which are updated at packet arrival times. For an online management and control application, one may try to use the observed values of ***performance metrics and their IPA derivatives*** as a component of a control policy that adjusts network parameters, such as source arrival rate (e.g., for access control), link buffer size (e.g., buffer size allocation), and link service rate (e.g., service rate allocation).

It is known that in discrete queueing setting, IPA gradient estimators are biased in the majority of cases [Ho and Cao (1991), Cassandras and Lafortune (1999)]. Consequently, attention has recently shifted to Continuous Flow Model (CFM) setting [Wardi and Melamed (2001), Cassandras et al. (2001)] (see Section 2.1), where IPA gradients are unbiased [Wardi et al. (2002), Cassandras et al. (2002), Cassandras et al. (2003), Sun et al. (2004)]. Several of these references contain examples of simple optimization problems where the IPA gradients, while derived based on fluid-flow models, are applied to packet-based models. The results indicate convergence to the respective optima, thus raising the following question: *Can IPA formulas obtained in a CFM setting be reproduced, or adequately approximated, in the corresponding discrete settings*? For example, many basic telecommunications models are packet-based and hence discrete in essence. Recall that the IPA-derivative formulas derived from such models tend to be biased, in contrast, their fluid-model counterparts, which tend to be unbiased and nonparametric. Thus, an affirmative answer to the above question may render the IPA technique applicable to a significant class of network control and management problems. Note that this approach requires us to map a discrete-flow (e.g., packets) to a corresponding continuous-flow (fluid) counterpart, and such corresponding flows will be referred to as ***compatible flow versions***. Thus, a queueing system may consist entirely of discrete flows (***pure-packet*** model), entirely of fluid flows (***pure-fluid*** model) or a mixture thereof (***hybrid*** model). Finally, any set of queueing models (pure-packet, pure-fluid, or hybrid) consisting entirely of compatible flows will be referred to as ***compatible model versions***.

In addition to fluid-flow analytical models for queueing networks (e.g., [Kobayashi et al. (1992)], fluid models have been proposed for telecommunications networks under specific protocols [Hall (2000)]. For example, fluid simulation for ATM networks [Kyas (1996)] is discussed in [Kesidis (1996)]. This model describes a fluid event-driven ATM network simulator that used Markov-modulated fluid models for the sources as well as fluid leaky buckets and fluid bandwidth schedulers. An overview of fluid-flow network modeling can be found in [Milidrag et al. (2000)], including fluid versions of four common workload-processing schedulers: work-conserving Generalized Processor Sharing (GPS) schedulers, non work-conserving idling schedulers, FIFO schedulers, and priority schedulers. In a similar vein, a fluid-flow model for the TCP protocol is proposed in [Nicol (2001)]. This model captures the TCP congestion control mechanism both in slow start and congestion avoidance modes, as well as acknowledgements, lost traffic, timeouts, and retransmissions. Finally, [Liu et al. (1999)] compares the simulation complexity of packet-based and fluid-based simulations.

In a previous paper [(Melamed et al. (2004)], we described a software environment, dubbed ***Hybrid Network Simulator*** (***HNS***), which provides a discrete-event simulation testbed for experimentation with compatible queueing networks (especially, telecommunications networks). HNS is designed to facilitate a fundamental modeling tradeoff between simulation running speed and its accuracy. Modern high-bandwidth telecommunications networks are a case in point. Simulated at the packet level, accurate models of such networks can require prohibitively high CPU times in consequence of the enormous numbers of packets processed by them. In contrast, a compatible fluid-flow simulation is less accurate, but can run much faster than its discrete-flow counterpart (speedups of some 3 orders of magnitude are reported in [Melamed et al. (2004)]). HNS seamlessly integrates the packet (discrete) and fluid (continuous) paradigms into a ***Hybrid Simulation paradigm*** (***HS paradigm***). Modelers can explore the speed/accuracy tradeoff by freely selecting flow types (packet or fluid), and benefit from the ease of switching from one flow type to the other, as well as from variance reduction afforded by using the same random number stream. IPA derivative computation is incorporated into HNS for all three types of compatible model versions (pure-packet, pure-fluid, or hybrid). These IPA derivatives are based on formulas derived in CFM setting in Wardi et al. (2002) and Cassandras et al. (2002).

This paper studies IPA derivatives computed from a suite of compatible model versions in telecommunications settings. It compares the accuracy of the IPA long-run derivatives across versions and studies the stabilization of their values under various stopping rules. The results lend empirical support to the contention that IPA derivatives can be accurately computed from discrete versions by adopting a fluid-flow view. Furthermore, as suggested by the functional form of the IPA derivatives, their long-run values stabilize quite fast. The results reported here summarize the work in Pan (2005).

The rest of the paper is organized as follows. Section 2 presents a mathematical description of the CFM construct. Section 3 provides a brief overview of IPA gradient formulas for the basic CFM, while Section 4 describes briefly their implementation in HNS. Section 5 studies the accuracy and stabilization of IPA gradients across compatible model versions. Finally, Section 6 concludes the paper.

2. Basic CFM

A basic CFM [Wardi and Melamed (2001)] is shown in Figure 2.1. Mathematically, a CFM is determined by a set of (given) stochastic processes, referred to as ***defining processes*** and defined over a common probability space, as follows.

- $\{\alpha(t)\}$ is the input flow (inflow) rate process into the CFM, where $\alpha(t)$ is the arrival rate at the system at time t.
- $\{\beta(t)\}$ is the service rate process, where $\beta(t)$ is the service rate at time t.
- $\{c(t)\}$ is the buffer capacity (buffer size) process (usually a deterministic quantity), where $c(t)$ is the buffer capacity at time t.

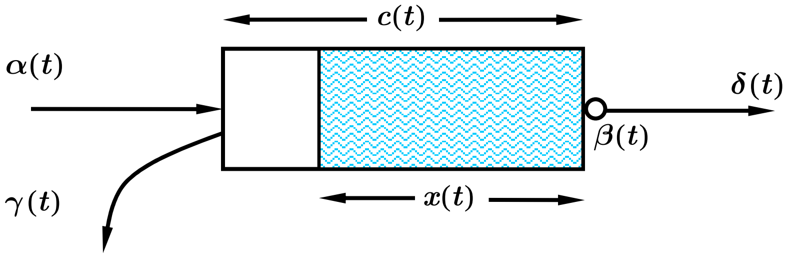


Figure 2.1 A basic Continuous Fluid Model (CFM)

The defining processes above determine all other stochastic processes of interest, referred to as ***derived processes***. These processes include

- $\{x(t)\}$ is the buffer workload (occupancy) process, where $x(t)$ is the fluid volume in the buffer at time t, governed by the stochastic differential equation

$$\frac{d}{dt^+}x(t)=\begin{cases}0, & \text{if } x(t)=0 \text{ and } \alpha(t)-\beta(t)\le 0\\ 0, & \text{if } x(t)=c(t) \text{ and } \alpha(t)-\beta(t)\ge 0\\ \alpha(t)-\beta(t), & \text{otherwise}\end{cases}$$

- $\{\delta(t)\}$ is the fluid discharge (outflow) rate process at the server, where $\delta(t)$ is the departure rate of fluid at time t, given by

$$\delta(t)=\begin{cases}\beta(t), & \text{if } x(t)>0\\ \alpha(t), & \text{if } x(t)=0\end{cases}$$

- $\{\gamma(t)\}$ is the loss (overflow) rate process due to a full buffer, where $\gamma(t)$ is the rate of lost fluid at time t, given by

$$\gamma(t)=\begin{cases}\alpha(t)-\beta(t), & \text{if } x(t)=c(t)\\ 0, & \text{if } x(t)<c(t)\end{cases}$$

The computation of IPA gradient formulas from CFM sample paths in the next section requires the partitioning of the simulation horizon, $[0, T]$. The resulting partition consists of three types of periods, in accordance with the state of the buffer, as follows:

- An ***empty*** period is a closed extremal interval during which the buffer is empty.
- A ***partial*** period is an open extremal interval during which the buffer is neither full nor empty.
- A ***full*** period is a closed extremal interval during which the buffer is full.

By an ***extremal interval*** we mean one with end points obtained via the **inf** and **sup** operations, respectively. We refer to partial and full periods as non-empty (buffering) periods, and refer to empty and partial periods as non-full periods. Furthermore, a ***lossy buffering period*** is one that experienced some loss.

Figure 2.2 illustrates such a partition of the time horizon, where

- $[\mathbf{0}, \xi_1]$, $[\eta_1, \xi_2]$ and $[\eta_2, T]$ are empty periods.
- $[\xi_1, u_{1,1}]$, $[v_{1,1}, u_{1,2}]$, $[v_{1,2}, u_{1,3}]$, $[v_{1,3}, \eta_1]$ and $[\xi_2, \eta_2]$ are partial periods.
- $[u_{1,1}, v_{1,1}]$, $[u_{1,2}, v_{1,2}]$ and $[u_{1,3}, v_{1,3}]$ are full periods.

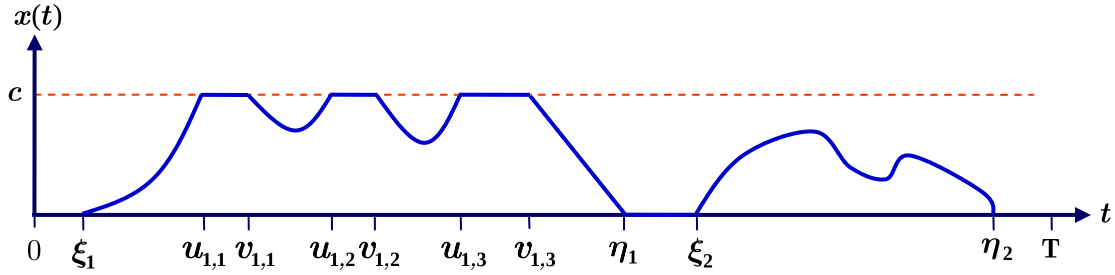


Figure 2.2 Partitioning a time horizon

Here, we assume constant buffer capacity, i.e., $c(t) = c$, and omit the parameter θ in the variables demarcating period boundaries on the horizontal axis (see next section).

We shall be concerned with the following performance metrics over $[0, T]$:

- Loss rate: $L_V = \frac{1}{T}\int_0^T \gamma(t)\, dt$.
- Time average of buffer workload: $L_W = \frac{1}{T}\int_0^T x(t)\, dt$.

The IPA derivatives of these metrics will be exhibited in the next section.

3. IPA Gradients for Basic CFM

Suppose that the CFM under consideration depends on a real-valued parameter θ, assumed to belong to a compact (bounded and closed) interval, Θ. For instance, θ might be a parameter related to the buffer size, service rate, or inflow rate.

Explicit formulas of various IPA derivatives in open-loop CFM setting (i.e., without feedback flows) have been derived in a number of papers [Wardi et al. (2002), Cassandras et al. (2002)] using sample path analysis. This work has shown various IPA gradients to be fast to compute, unbiased. Furthermore, by virtue of the sample path analysis, these IPA gradients are nonparametric in the sense that they can be computed from data without any knowledge of the underlying probability law. Consequently, these properties hold out the promise of utilizing IPA gradient estimates as an ingredient of on-line management and control of telecommunications networks.

Before exhibiting the aforementioned formulas of IPA derivatives, we introduce some notation. Let the buffering periods in $[0,\ T]$ be denoted by

$$B_k = (\xi_k(\theta), \eta_k(\theta)), \quad k = 1, \ldots, K,$$

for some $K > 0$, where $\xi_k(\theta)$ and $\eta_k(\theta)$ are the boundary points of B_k, respectively. The buffering periods can be classified according to whether or not they experienced some loss. Define the index set

$$\Phi(\theta) = \{1 \leq k \leq K\text{: some loss occurred during } B_k\},$$

and let

$$N_{\mathrm{T}}(\theta) = |\ \Phi(\theta)\ |,$$

where vertical bars denote cardinality.

For each $k \in \Phi(\theta)$, let $M_k(\theta)$ be the number of full periods in B_k, and denote the associated full periods in increasing order by

$$F_{k,m}(\theta) = [u_{k,m}(\theta),\, v_{k,m}(\theta)],\ m = 1, \cdots, M_k(\theta),$$

where $u_{k,m}(\theta)$ and $v_{k,m}(\theta)$ denote, respectively, the start time and end time of $F_{k,m}(\theta)$.

Finally, for a given $\theta \in \Theta$, we define a ***sample path event*** as an occurrence in a sample path at a particular time point. We mention that the computation of IPA derivatives makes use of two types of sample path events:

- An exogenous sample path event is a jump in either $\{\alpha(\theta;t)\}$ or $\{\beta(\theta;t)\}$.
- An endogenous sample path event corresponds to the buffer becoming either empty or full.

We now proceed to exhibit the IPA derivatives of interest. We shall assume that $\theta \in \Theta$ for a given closed and bounded interval Θ whose left-hand point is positive and that for every $\theta \in \Theta$,

- With probability 1, the process $\{\alpha(t) - \beta(t)\}$ is piecewise continuously-differentiable in the interval $[0,\ T]$.
- With probability 1, no multiple events occur simultaneously.
- The sample derivatives $L'_V(\theta)$ and $L'_W(\theta)$ exist, with probability 1.

3.1 IPA Derivatives with Respect to Buffer Size

Throughout this subsection, the parameter θ is the buffer size, i.e., $c(\theta) = c$, and we make the assumption that the processes $\{\alpha(t)\}$ and $\{\beta(t)\}$ are independent of θ.

Proposition 1 [cf. Wardi, et al. (2002) Proposition 3.1]:
For every $\theta \in \Theta$,

$$L_V'(\theta) = -\frac{1}{T} N_T(\theta) . \tag{1}$$

In words, the requisite derivative is a ratio, where the numerator is the negative number of lossy nonempty periods in the time horizon $[0, \mathbf{T}]$, and the denominator is the time horizon, $\mathbf{T}$.

Proposition 2 [cf. Wardi et al. (2002) Proposition 3.2]:
For every $\theta \in \Theta$,

$$L_W'(\theta) = \frac{1}{T} \sum_{k \in \Phi(\theta)} [\eta_k(\theta) - u_{k,1}(\theta)] . \tag{2}$$

In words, only lossy nonempty periods contribute to the requisite derivative, and the contribution to of each lossy nonempty period, B_k, is a ratio, where the numerator is the length of the time interval starting at the first point in B_k at which the buffer becomes full and ending at the last point of B_k, and the denominator is the time horizon, T.

3.2 IPA Derivatives with Respect to a Service Rate Parameter

Throughout this subsection, the parameter θ is a parameter of the service rate process, $\{\beta(\theta;\ t)\}$, and we make the following assumptions:

- The inflow process, $\{\alpha(t)\}$, and the buffer size, c, are independent of θ.
- For all $\theta \in \Theta$ and for all $t \in [0,\ T]$,

$$\frac{d}{d\theta} \beta(\theta;t) = \beta'(\theta;t) = 1 .$$

Proposition 3 [Wardi et al. (2002) Proposition 4.1]
For every $\theta \in \Theta$,

$$L_V'(\theta) = -\frac{1}{T} \sum_{k \in \Phi(\theta)} [v_{k,M_k}(\theta) - \xi_k(\theta)] . \tag{3}$$

In words, the contribution to the requisite derivative of each nonempty period B_k, during which some loss occurred, is the negative ratio of the length of the time interval from the start of B_k until the last time point in B_k at which the buffer is full, to the time horizon, T.

Proposition 4 [Wardi et al. (2002) Proposition 4.2]:
For every $\theta \in \Theta$,

$$L'_W(\theta) = -\frac{1}{2T}\sum_{k=1}^{K}\sum_{m=1}^{M_k+1}[u_{k,m}(\theta) - v_{k,m-1}(\theta)]^2 \ . \tag{4}$$

In words, the contribution to the requisite derivative of each nonempty period B_k, during which some loss occurred, is half of the negative ratio of the sum of squares of the length of the partial periods, to the time horizon, T.

3.3 IPA Derivative with Respect to an Inflow Rate Parameter

In this section, the parameter θ is a parameter of the inflow rate processes $\{\alpha(\theta;\, t)\}$, and $\theta \in \Theta$ for a given closed and bounded interval Θ. Assume further that for all $\theta \in \Theta$ and for all $t \in [0,\ T]$,

$$\frac{d}{d\theta}\alpha(\theta;\, t) = \alpha'(\theta;\, t) = \begin{cases} 1, & \text{if } \alpha(\theta;\, t) > \ 0, \\ 0, & \text{if } \alpha(\theta;\, t) = \ 0. \end{cases} \ .$$

Throughout this section, we make the following assumptions.

Assumption 3.

(a) For every $\theta \in \Theta$, with probability 1, the function $\alpha(\theta;\, t) - \beta(t)$ is piecewise continuously-differentiable in the interval $[0,\ T]$.

(b) For every $\theta \in \Theta$, with probability 1, no multiple event occur simultaneously.

(c) For every $\theta \in \Theta$, the sample derivatives $L'_V(\theta)$ and $L'_W(\theta)$ exist, with probability 1.

(d) The jump points of $\alpha(\theta;\, t)$ (as function of t) do not depend on θ.

(e) For every $t \in [0,\ T]$, the function $\alpha(\theta;\, t)$ is continuously differentiable in θ. There exists $K < \infty$ such that, with probability 1,

$$\textbf{sup}\ \{\ |\ \alpha'(\theta;\, t)\ | : \theta \in \Theta;\ t \in [0, T]\} \le K$$

Proposition 5 [Wardi et al. (2002) Proposition 5.1]:
For every $\theta \in \Theta$,

$$L'_V(\theta) = \frac{1}{T}\sum_{k\in\Phi(\theta)}\int_{\xi_k(\theta)}^{v_{k,M_k}(\theta)}\alpha'(\theta;\tau)dt\,. \tag{5}$$

In words, the contribution to the requisite derivative of each nonempty period B_k, during which some loss occurred, is the ratio of the total durations with positive inflow rates from the start of B_k until the last time point in B_k at which the buffer is full, to the time horizon, T.

Proposition 6 [Wardi et al. (2002) Proposition 5.2]:
For every $\theta \in \Theta$,

$$L'_W(\theta) = \frac{1}{T}\sum_{k=1}^{K}\sum_{m=1}^{M_k+1}\int_{v_{k,m-1}(\theta)}^{u_{k,m}(\theta)}\int_{v_{k,m-1}(\theta)}^{t}\alpha'(\theta;\tau)d\tau dt\,. \tag{6}$$

4. Implementation of IPA Gradients in HNS

The IPA formulas above were derived for the pure-fluid paradigm. However, the implementation of IPA gradients in discrete-event simulation requires algorithms for their computation not only in a pure-fluid setting, but also in pure-packet and hybrid settings.

In this section we briefly describe the implementation of IPA gradient computations for single queues in the HNS (Hybrid Network Simulator) application [Melamed et al. (2004)]. First, we describe briefly the functionality of HNS (see ibid. for the full details).

In HNS paradigm, a ***network*** is a (fixed) directed graph consisting of ***nodes*** and ***links***, where a node represents a network location and a link connects a pair of nodes. Each link houses a service facility (a shared server with a prescribed service speed and a shared buffer with a prescribed size) for processing (serving) workload carried by transactions. ***Workload*** can be ***discrete*** (e.g., packets, jobs, etc. as in traditional queueing theory) or ***continuous*** (fluid). While discrete workload is represented in the standard way, continuous workload is represented by piece-wise constant flow rates and their associated durations.

The network interacts with an exogenous environment consisting of sources and sinks, which are attached to nodes. A ***source*** is an ingress point for workload to enter the network, while a ***sink*** is an egress point for workload to drain from the network.

A source generates a transaction stream, according to the interarrival process, and has the following attributes:

1. an arrival process of transactions and their associated workload distribution
2. an injection process (piece-wise constant stochastic process of injection rates with random durations)
3. the workload type (discrete or continuous)
4. a priority in contending for buffer and server resources
5. an itinerary (path through the network starting in a source and terminating in a sink)
6. a protocol (rules that govern the transport of workload, such as flow control)

Once generated, each transaction is injected from the source into the network, where its workload traverses the network along its itinerary. It then contends for server and buffer resources at each link according to its priority, and finally departs from the network at its itinerary's sink.

In HNS, a link can be in one of the following states at any given time: **IDLE-EMPTY**, **BUSY-EMPTY**, **BUSY-PARTIAL**, and **BUSY-FULL**. They are essential in determining the flow workload admission logic at the link buffer. HNS's ***parceling algorithm*** seamlessly integrates the processing of workload at a link buffer for both discrete and continuous transactions. Flows at a link buffer are organized as multiparcels, which is a vector of volumes of multiple inflows into a buffer, each with its own piecewise-constant inflow rate. The period of simultaneous constancy of constituent inflow rates is called a **duration**. Multiparcels are created by the flows' ***effective inflow rates*** (rates at which the flow actually enters the current link buffer), which is calculated based on the flows' ***offered arrival rates*** (rates at which the flow leaves the upstream link) and the current link's state and service rate.

As it turns out, the main task in computing IPA sample gradients is the identification of the time boundaries of various queue states. In HNS, these are the time points at which the link state changes. In pure-fluid setting, these states can be identified unambiguously in a natural way. However, whenever packets are involved (pure-packet setting or hybrid setting) some conceptual problems arise in identifying the packet counterparts of the states **BUSY-EMPTY** and **BUSY-FULL**. The problems involved and their solutions will be explained next.

4.1 Time Boundaries Between Empty and Non-Empty Periods

Consider a packet being routed from some origination link to some destination link. Recall that in this case, the entire packet workload is routed in zero time (in contrast, a fluid workload is routed over a period of time). When contemplating the mapping of a packet routing to the fluid-flow setting, the following discordance emerges. Focusing on the destination link, it is clear that packet routing would cause its state to alternate between empty and nonempty states. However, its fluid-flow counterpart would not cause that effect as a continuous fluid flow would usually stay in the same state. What is conceptually required is to "fluidize" packet behavior, so as to remove the state alternations inherent in packet routing, but only for the purpose of mapping the computation of IPA derivatives from the fluid paradigm to the packet or hybrid paradigms.

To this end, we associate with a packet in service a so-called ***nominal arrival rate***. This rate is defined as either the ***effective injection rate*** (in case the packet is injected into the network from a source), or the packet's ***allocated service rate*** at the origination link (in all other cases). As soon as the packet enters the service group in the origination link, the packet's nominal arrival rate is incorporated into the inflow rate of the destination link, and is used to determine its state transitions.

4.2 Time Boundaries Between Full and Non-Full Periods

In a hybrid network, a link buffer is considered full either of the following two scenarios:

1. A fluid buffer-full event occurs.
2. A packet is discarded when the destination link buffer cannot accommodate the packet's workload.

The latter case causes the following discordance when contemplating the mapping of a packet routing to the fluid-flow setting. Upon the arrival of the packet at the destination link buffer, the packet is discarded (due to insufficient buffer capacity) and we declare the link state full. However, the moment the packet is discarded, the link state changed back to non-full as there is still some residual capacity in that buffer. Hence, the link state will undergo two instantaneous state transitions. However, in pure-fluid networks, this kind of state transitions is absent.

In order to compute conceptually compatible IPA derivatives for packet and hybrid streams, it is essential to harmonize the fluid and packet worldviews for the second scenario above. To this end, we again use the notion of packet ***nominal arrival rate***. We then harmonize the packet and fluid worldviews by distinguishing between two cases:

1. If $\alpha(t) > \beta(t)$, where $\alpha(t)$ includes all relevant packet nominal arrival rates, then we gloss over the instantaneous state transition to a **non-FULL**, and simply declare the link buffer to stay in the **FULL** state.
2. Otherwise, if $\alpha(t) \leq \beta(t)$, then we declare the link buffer to undergo a state transition from **FULL** to **non-FULL**.

5. Experimentation with IPA in HNS

In this section, we describe simulation experiments with IPA statistics using HNS, summarizing the work in Pan (2005). Recall that the closed form formula for IPA statistics calculations in

Section 3 are derived for ***open-loop*** flows, i.e., network flows with no feedback loops. In particular, the UDP transport protocol implemented in HNS meets this requirement, and consequently, the experimentation in this chapter studies telecommunications networks with UDP-type streams only. Recall that the three compatible model versions of interest are the pure-packet, pure-fluid and hybrid versions.

The goal of this chapter is to study statistically the six variants of IPA gradients of Section 3, collected from compatible simulation model versions in various network models:

1. Loss rate as function of buffer size (**IPA-1**)
2. Workload time average as function of buffer size (**IPA-2**)
3. Loss rate as function of a service rate parameter (**IPA-3**)
4. Workload time average as function of a service rate parameter (**IPA-4**)
5. Loss rate as function of an arrival rate parameter (**IPA-5**)
6. Workload time average as function of an arrival rate parameter (**IPA-6**)

Recall that the IPA derivative formulas of Section 3 are valid only when the initial state of a sample path consists of an empty buffer (cold start). All replications in this section have no warm-up period to ensure IPA statistics collection from a cold start.

Since all the IPA derivatives above are time averages of the form

$$G(t) = \frac{C(t)}{t}, \tag{7}$$

where $C(t)$ is monotone non-decreasing, it is reasonable to expect them to stabilize (approximately converge) after a sufficiently long simulation horizon of minimal length, T_{min}, to be determined by the modeler. More precisely, when the system is ***ergodic***, then the long-run value of the (time average) IPA derivative coincides with its invariant (equilibrium) mean [Breiman (1968), Chapter 6]. From now on, allusions to estimating the stable value of IPA derivatives will refer to the estimation of the aforementioned invariant mean. Experimentation with these IPA derivatives in HNS supports this expectation. Consequently, in order to estimate an IPA derivative from a replication, it is necessary to devise a good stopping rule which can be employed to determine the convergence of the IPA derivatives.

Since detecting the stopping point in time is computationally intensive, we considered the following two computationally efficient stopping rules:

HSR: This is a heuristic stopping rule based on inspection of iterative pilot runs and requires human interaction. A heuristic simulation horizon, T_H, is selected conservatively, such that all the IPA derivatives appear to stabilize "sufficiently". Such IPA values will be referred to as the ***almost stable IPA derivatives***.

MSR: This is a two-stage mixed (heuristic and algorithmic) stopping rule, computed from a replication per IPA derivative as follows:

Stage 1: Determine a minimal time length, T_{min}, applicable to all IPA derivatives by heuristic experimentation with pilot runs, in the vein of ***HSR***.

Stage 2: Once T_{min} elapses, this stage starts to test for convergence. To economize on the computational effort involved, it only tests every $\frac{N}{100}$ IPA derivative updates, where N is the total number of IPA derivative updates observed during $[0, T_{min}]$.

Let $t_k, k = 0, 1, 2, \ldots$ denote the simulation time at which update number $N + \left\lfloor \frac{Nk}{100} \right\rfloor$ occurs, where $\lfloor x \rfloor$ is the floor function. The iteration stops as soon as the relative deviation of two successive IPA derivative observations is below $\epsilon = 0.005$, namely, when

$$\left| \frac{G(t_k) - G(t_{k-1})}{G(t_{k-1})} \right| < \epsilon. \tag{4.2}$$

The rationale for T_{min} is the need to add a measure of stabilization to the IPA derivatives and prevent premature stopping of its computation. Such IPA values will be referred to as the ***approximately stable IPA derivatives***.

The following subsections describe four studies:

1. ***HSR Accuracy Study***: compares the relative deviations of almost stable IPA derivatives across simulation model versions, based on one long replication subject to stopping rule ***HSR***.
2. ***MSR Convergence Study***: studies the convergence of approximately stable IPA derivatives as well as stopping time values, based on multiple replications subject to stopping rule ***MSR***.
3. ***MSR Accuracy Study***: compares the relative deviations of various statistics of IPA derivatives across simulation model versions, based on multiple replications subject to stopping rule ***MSR***.
4. ***MSR-to-HSR Comparison Study***: compares the relative deviations of corresponding almost stable and approximately stable IPA derivatives within versions, based on one compatible replication subject to their respective stopping rule.

The accuracy measure used in studies 1, 3 and 4 above utilize the relative deviation of Eq. (7.3) in Chapter 7. Unlike Chapter 7, this chapter always uses the pure-fluid version as the baseline version, to which the other two versions are compared. However, as in Chapter 7, the simulation model versions are made compatible in that each has identical network configurations, as well as the same arrival processes at network sources and service processes at network links. Thus, the versions are subjected to identical loads, and only differ in the transport mechanism used.

5.1 *HSR* Accuracy Study

The goal of this study is to gauge the relative accuracy of almost stable IPA derivatives across simulation model versions. For each version (pure-fluid, pure-packet and hybrid), this study runs one long replication of length T_H and computes all IPA derivatives at this time, and these values are used in the comparison. To this end, a suite of two network models were studied:

1. A simple 5-source, 4-link network.
2. A network 12-node, 11 link network with a bottleneck link.

The next two subsections describe the comparison study for each of the models above.

5.1.1 A Simple 5-Source 4-Link Network

Figure 5.1 depicts a simple network with 5 nodes (empty circles), 4 links (arrows), 4 sources (filled trapezoids), and 4 sinks (filled circles). Each source generates one UDP flow, whose itinerary (path in the network) is texture (and color) coded.

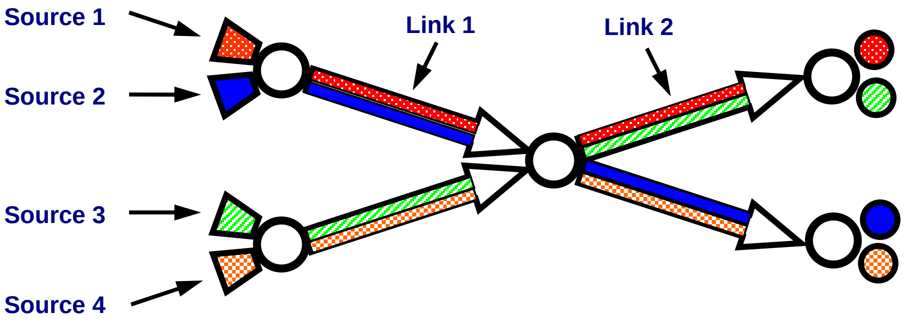


Figure 5.1 A simple 5-node 4-link network

Network model parameters are as follows:

- Incoming UDP message workloads are infinite (so only one message arrives at each UDP source and generates the entire workload).
- Each UDP source injection rate according to an ON/OFF process with iid exponential durations of mean 1 second and an ON rate of 10 Mbps.
- UDP packets are of size 8 Kb.
- All link rates are 18 Mbps.
- All links have buffer size 10Mb.
- In the hybrid simulation version, Source 1 and Source 4 inject packet streams, while Source 2 and Source 3 inject fluid streams. Therefore, each link carries one packet and one fluid stream.

Due to the symmetry of the network model, we collected the IPA statistics only for Link 1 and Link 2 (see Figure 4.1) over the same simulation interval, [0, 600].

Table 5.1 compares the IPA statistics, IPA-1 through IPA-6, of the three simulation model versions at Link 1.

IPA Statistics	Pure Fluid	Pure Packet (% deviation)	Hybrid (% deviation)
Loss Rate as Function of Buffer Size	-0.01907	-0.01904 (-0.131%)	-0.01907 (0%)
Workload Time Average as Function of Buffer Size	0.05701	0.05003 (-12.244%)	0.05702 (0.009%)
Loss Rate as Function of a Service Rate Parameter	-0.06225	-0.06226 (0.018%)	-0.06222 (-0.040%)
Workload Time Average as Function of a Service Rate Parameter	-0.46144	-0.48566 (5.249%)	-0.46148 (0.009%)
Loss Rate as Function of an Arrival Rate Parameter	0.06209	0.06210 (0.022%)	0.06021 (-3.029%)
Workload Time Average as Function of an Arrival Rate Parameter	0.45877	0.43319 (-5.575%)	0.45210 (-1.453%)

Table 5.1 Comparison of IPA statistics, subject to *HSR*, for Link 1 of the simple network

Table 5.2 compares the IPA statistics, IPA-1 through IPA-6, of the three simulation model versions at Link 2.

IPA Derivative	Pure Fluid	Pure Packet (% deviation)	Hybrid (% deviation)
Loss Rate as Function of Buffer Size	-0.00935	-0.00935 (-0.003%)	-0.00935 (0.017%)
Workload Time Average as Function of Buffer Size	0.01800	0.01877 (4.269%)	0.01775 (-1.371%)
Loss Rate as Function of a Service Rate Parameter	-0.02193	-0.02111 (-3.768%)	-0.02063 (-5.935%)
Workload Time Average as Function of a Service Rate Parameter	-0.41924	-0.41473 (-1.077%)	-0.44128 (5.257%)
Loss Rate as Function of an Arrival Rate Parameter	0.02193	0.01922 (-12.340%)	0.01935 (-11.774%)
Workload Time Average as Function of an Arrival Rate Parameter	0.39881	0.39781 (-0.250%)	0.42225 (5.879%)

Table 5.2 Comparison of IPA statistics, subject to *HSR*, for Link 2 of the simple network

5.1.2 A 12-Node 11-Link Network with a Bottleneck Link

Figure 5.2 depicts a 12-node, 11-link network with 5 UDP sources.

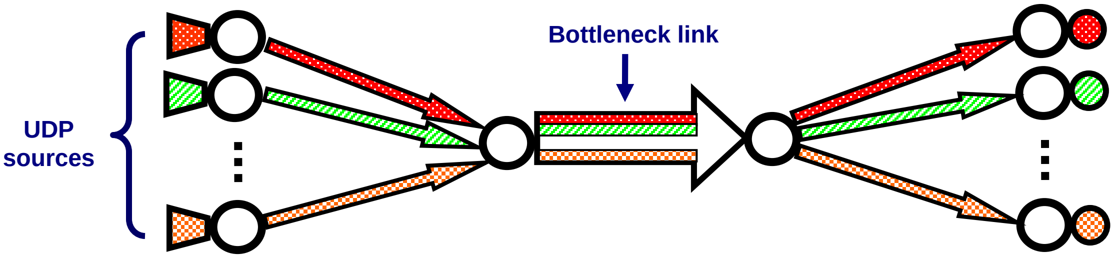


Figure 5.2 A network with a bottleneck link

The network model parameters are as follows:

- Incoming UDP message workloads are infinite (so only one message arrives at each UDP source and generates the entire workload).
- Each UDP source injection rate according to an ON/OFF process with iid exponential durations of mean 1 second and an ON rate of 990 Mbps.
- All UDP packets are of size 8 Kb.
- All non-bottleneck links have transmission rate of 1 Gbps.
- The bottleneck link has transmission rate of 1.5 Gbps.
- All links buffers have size 30Mb.
- In the hybrid version, one of the UDP source inject packet stream, while the other four UDP sources inject fluid streams.

Table 5.3 compares the IPA statistics, IPA-1 through IPA-6, at the bottleneck link of the three simulation model versions over the same simulation interval, [0, 600].

IPA Derivative	Pure Fluid	Pure Packet (% deviation)	Hybrid (% deviation)
Loss Rate as Function of Buffer Size	-1.11888	-1.12054 (0.15%)	-1.11888 (0%)
Workload Time Average as Function of Buffer Size	0.22905	0.22901 (-0.02%)	0.22905 (0%)
Loss Rate as Function of a Service Rate Parameter	-0.21814	-0.21806 (-0.04%)	-0.21843 (0.13%)
Workload Time Average as Function of a Service Rate Parameter	-0.00660	-0.00659 (-0.04%)	-0.00657 (-0.42%)
Loss Rate as Function of an Arrival Rate Parameter	0.21805	0.21706 (-0.45%)	0.21828 (0.11%)
Workload Time Average as Function of an Arrival Rate Parameter	0.00660	0.00659 (-0.04%)	0.00657 (-0.43%)

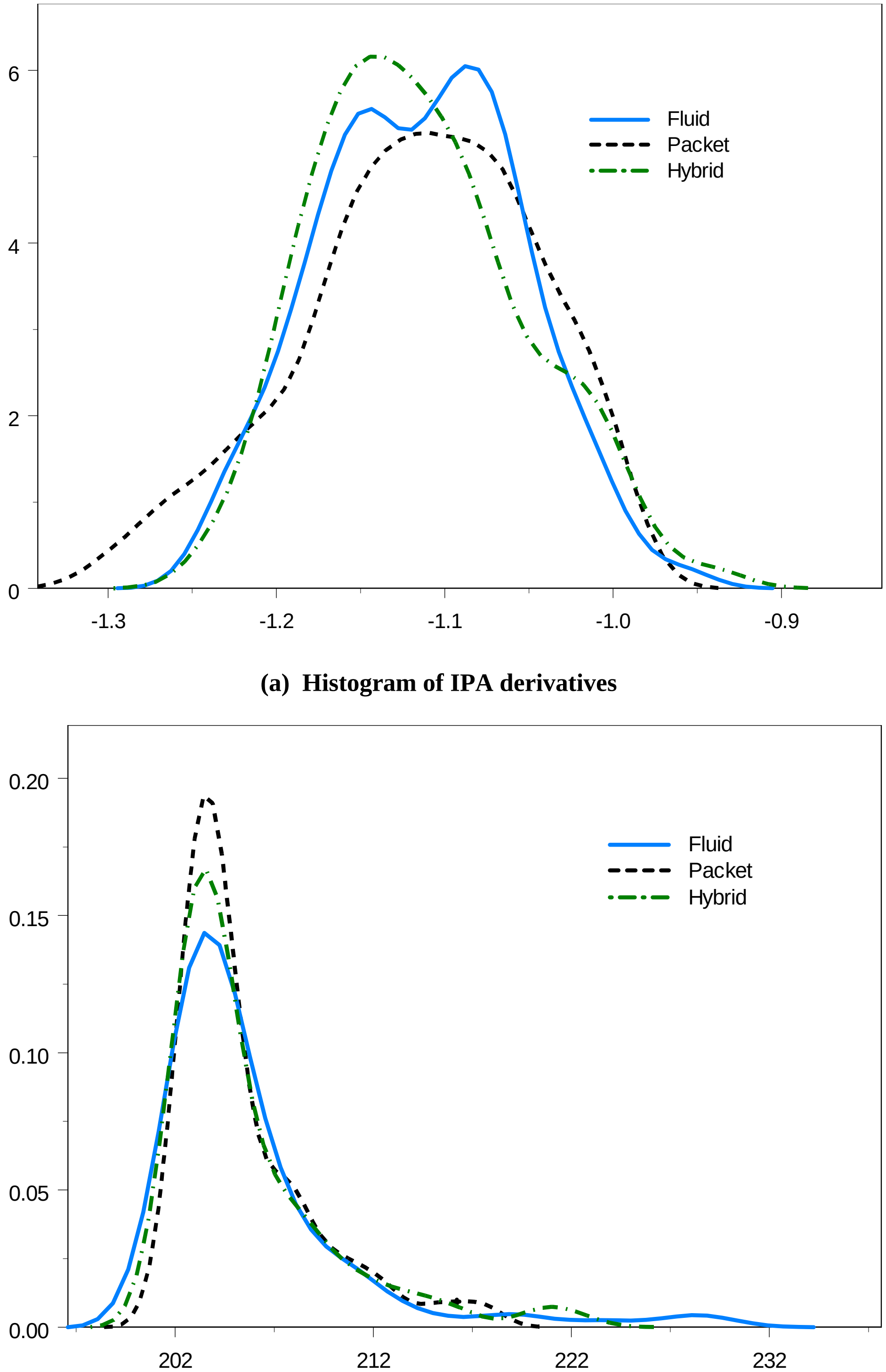
Table 5.3 IPA statistics, subject to HSR, for the network with a bottleneck link

The relative deviations of the IPA statistics in the above tables are all less than 1%, which indicates that the three different simulation model versions give rise to very close IPA derivatives. The higher accuracy of the IPA derivatives in this model as compared to the previous one is due to the fact that this model has higher multiplexing levels of packet flows as compared to the previous one, and consequently, the behavior of its packet streams is more fluid-like. Overall, this study attests to the efficacy of using fluid-based IPA derivatives as approximations for pure-packet and hybrid models.

5.2 *MSR* Convergence Study

The goal of this study is to gauge the convergence of the approximately stable IPA derivatives within simulation model versions, subject to stopping rule ***MSR***. It also compares the corresponding stopping time values. To this end, 100 replications are run for each simulation model version, and the approximately stable IPA derivatives as well as corresponding stopping time values are collected. Histograms of the approximately stable IPA derivatives and stopping time values are generated from the replications.

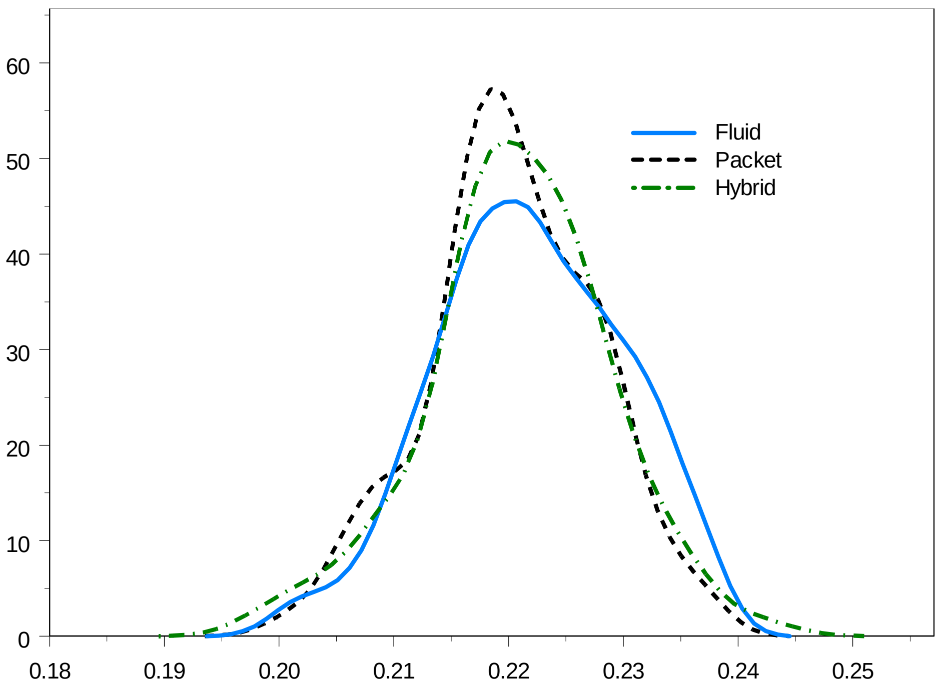
The network model we studied is the 12-node 11-link network with a bottleneck link, depicted in Figure 5.2 in Section 5.1.2. The simulation results for replications with $\boldsymbol{T_{min}} = 200$ seconds are summarized in Figure 5.3 through Figure 5.8. Each figure depicts the histogram of an IPA derivative computed from the 100 replications (left) and the histogram of the corresponding stopping times (right). In all these figures, statistics for the baseline pure-fluid version are depicted by solid curves, those for the pure-packet version are depicted by dashed curves, and those for the hybrid version are depicted by dashed and dotted curves.



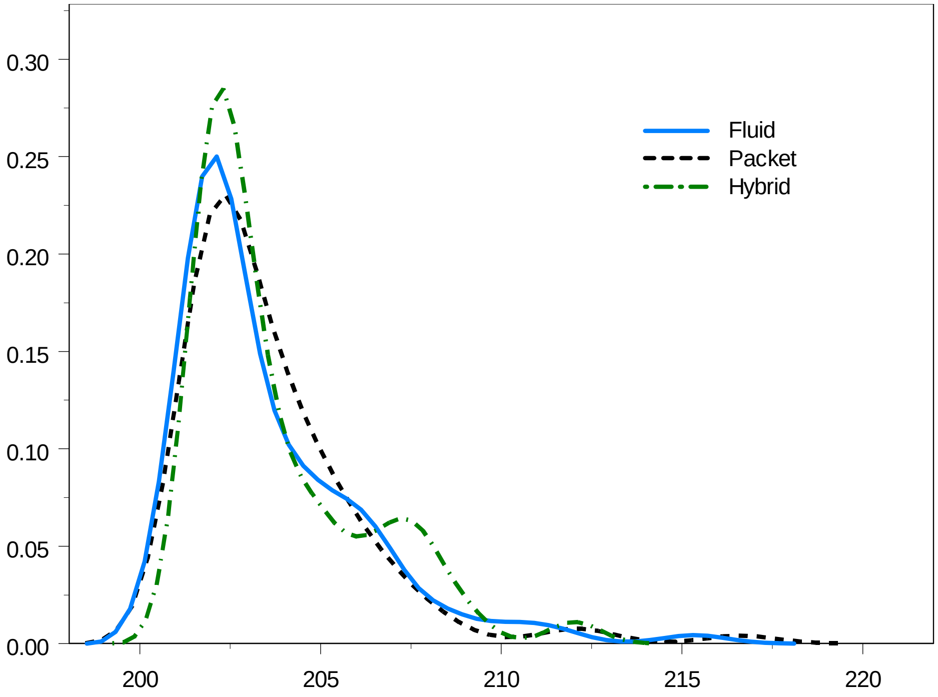
(a) Histogram of IPA derivatives

(b) Histogram of stopping times

Figure 5.3 Histograms of loss rate as function of buffer size

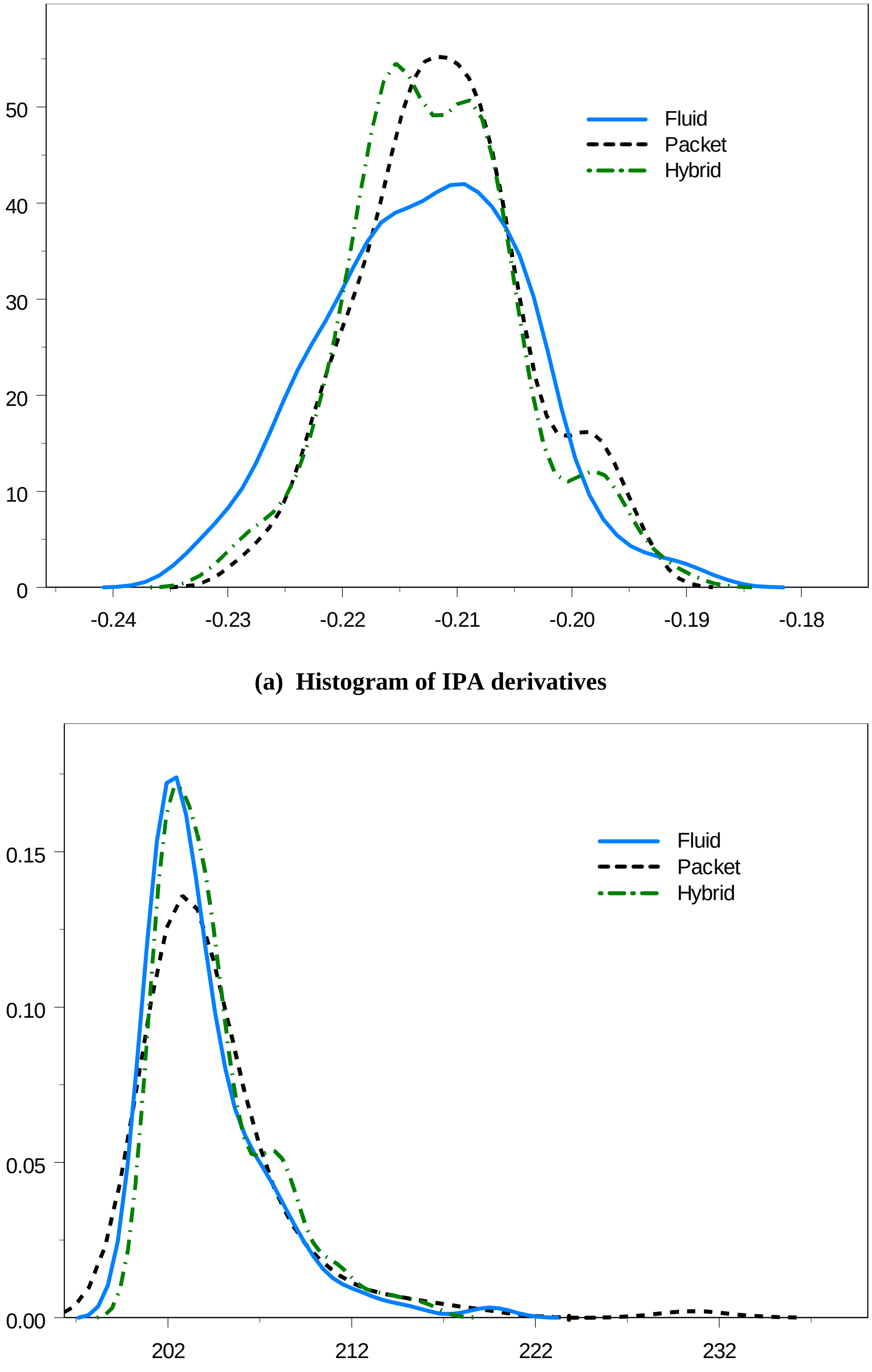


(a) Histogram of IPA derivatives



(b) Histogram of stopping times

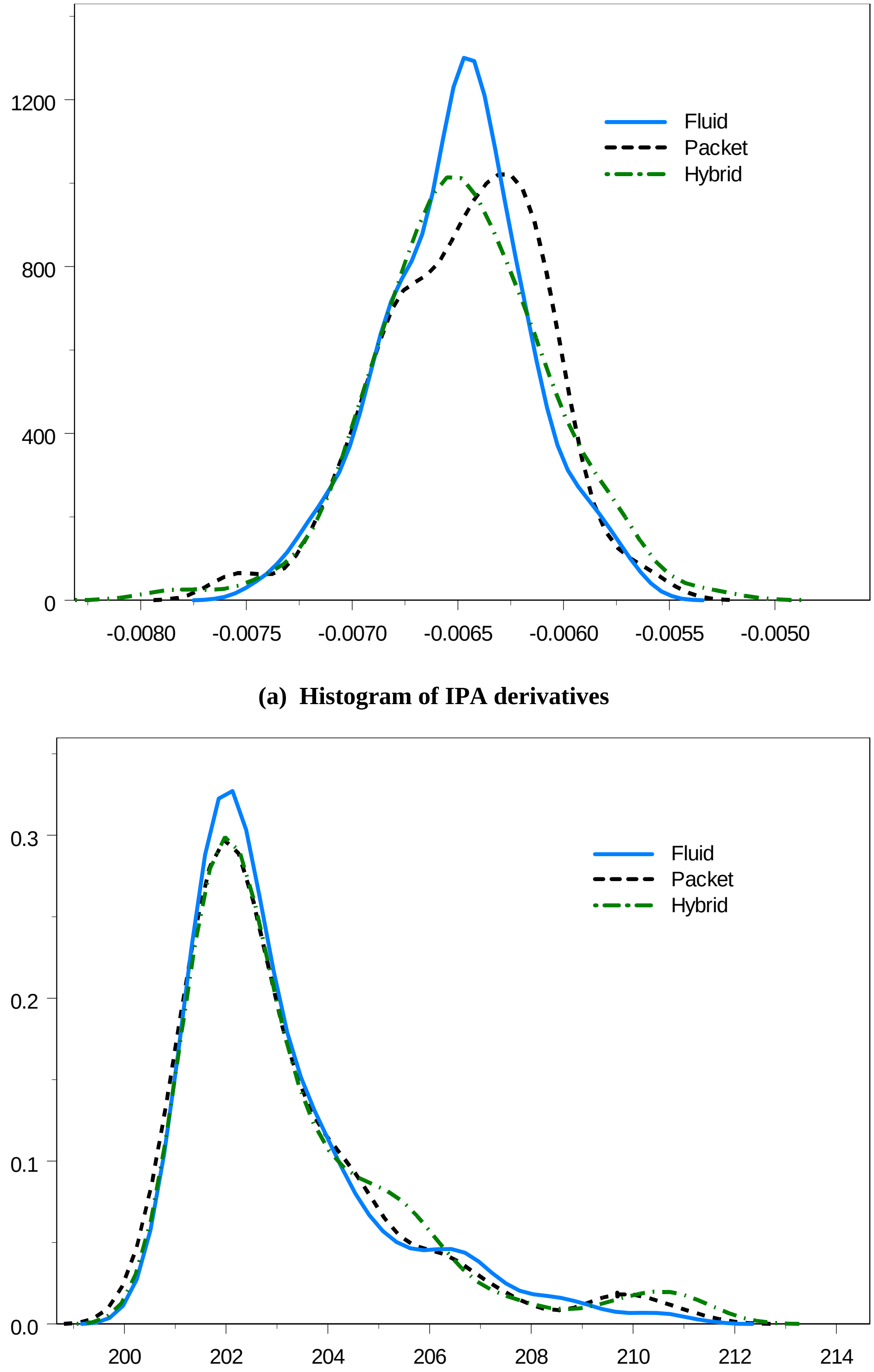
Figure 5.4 Histograms of workload time average as function of buffer size



(a) Histogram of IPA derivatives

(b) Histogram of stopping times

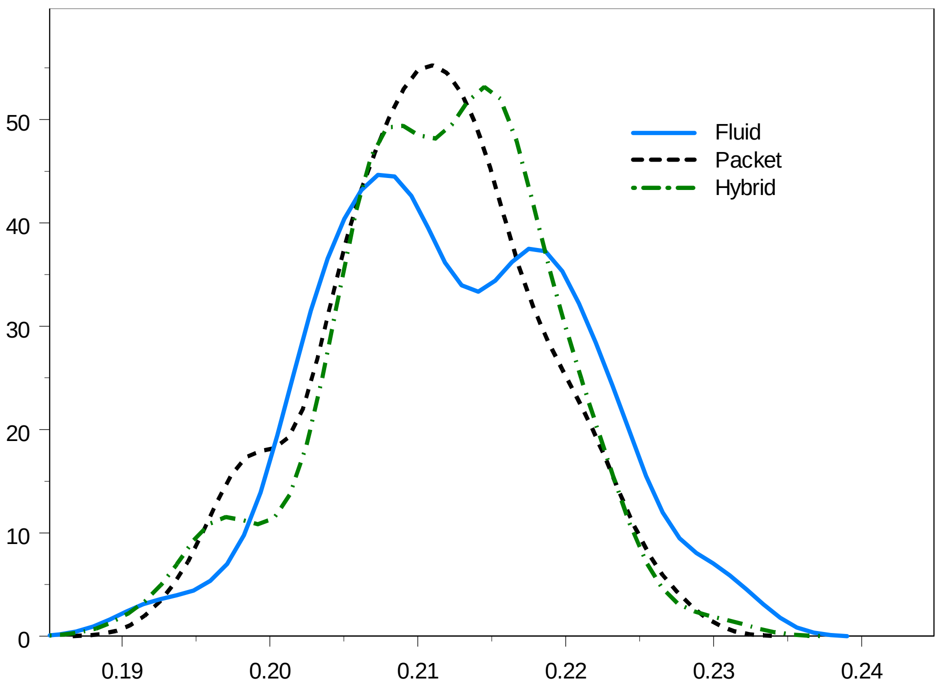
Figure 5.5 Histograms of loss rate as function of a service rate parameter



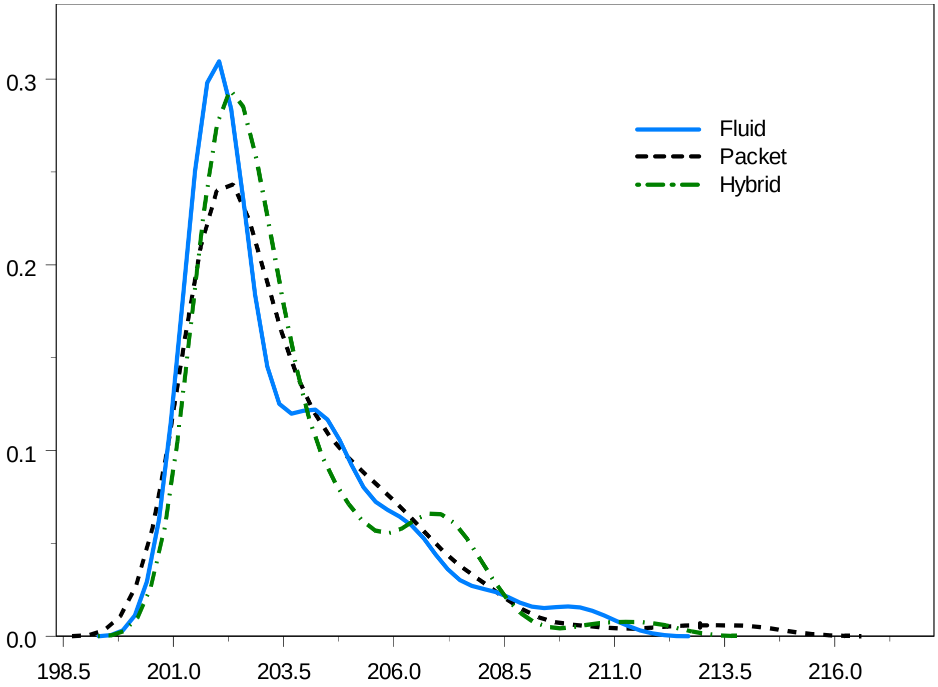
(a) Histogram of IPA derivatives

(b) Histogram of stopping times

Figure 5.6 Histograms of workload time average as function of a service rate parameter

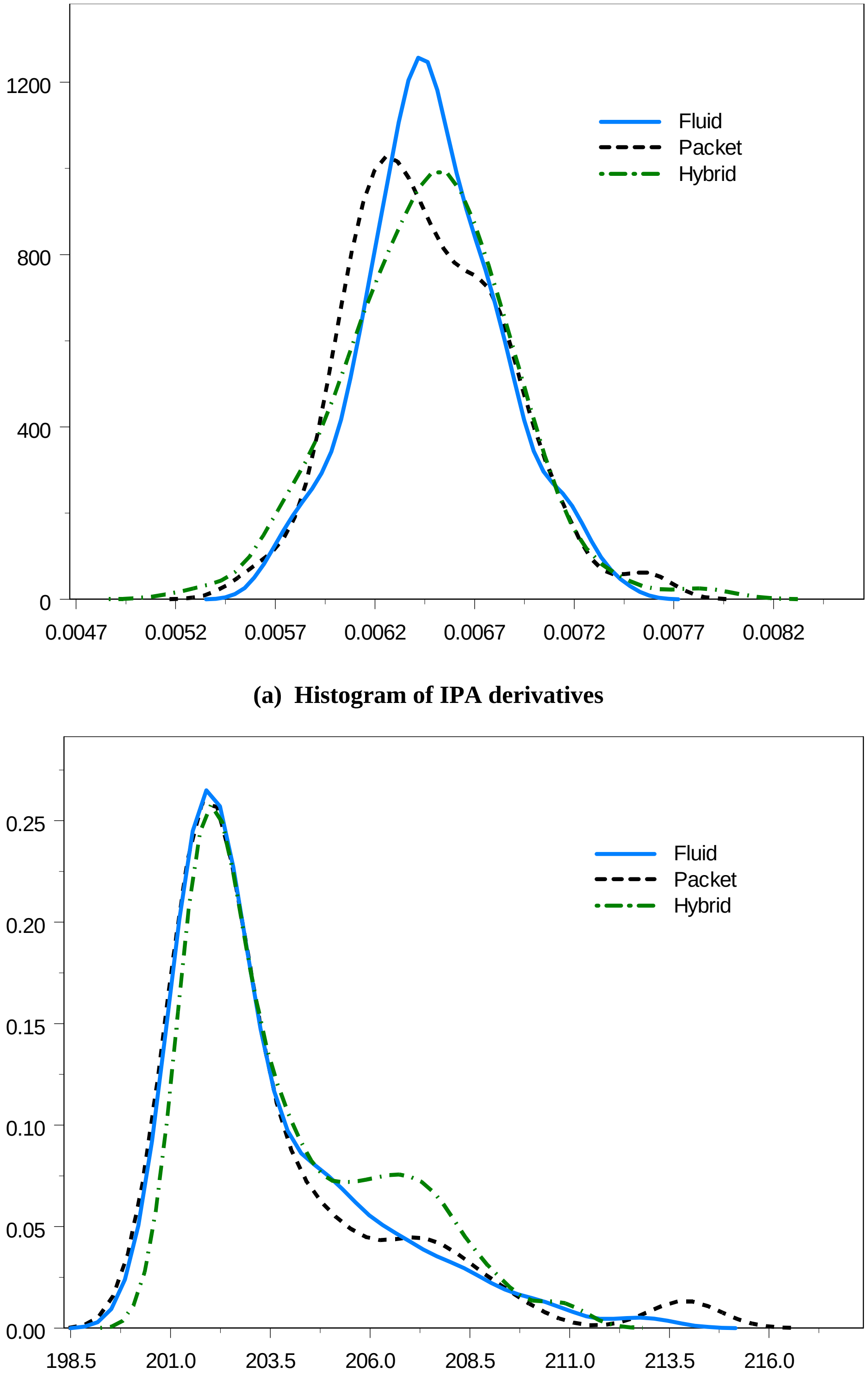


(a) Histogram of IPA derivatives



(b) Histogram of stopping times

Figure 5.7 Histograms of loss rate as function of an arrival rate parameter



(a) Histogram of IPA derivatives

(b) Histogram of stopping times

Figure 5.8 Histograms of workload time average as function of an arrival rate parameter

The figures show that the convergence to stable values has low variability. To wit, note that all the histograms of approximately stable IPA derivative values are roughly bell shaped and thintailed, with the bulk of observations distributed within $1 \pm 10\%$ of the mean values. Furthermore, their standard deviations (to be calculated in Section 8.3) are quite small relative to the corresponding mean values. Moreover, the histograms of stopping times of all three simulation model versions are pretty close. Interestingly, these histograms are not bell-shaped, but rather, are skewed to the right.

5.3 MSR Accuracy Study

The goal of this study is to gauge the relative accuracy of various statistics of approximately stable IPA derivatives across simulation model versions. For each version (pure-fluid, pure-packet and hybrid), this study runs 100 replications subject to stopping rule ***MSR***, with the $\boldsymbol{T_{min}}$ parameter set to 200 seconds. Each replication collects all IPA derivatives at the stopping time, as well as their following statistics: minimum, maximum, mean, and standard deviation. These statistics are then compared for relative accuracy.

The network model studied in this section is the 12-node 11-link network with a bottleneck link, depicted in Figure 5.2 in Section 5.1.2. The results are summarized in Table 5.4 through Table 5.9.

Statistic	Pure-Fluid	Pure-Packet (% deviation)	Hybrid (% deviation)
Minimum	-1.23953	-1.29985 (4.87%)	-1.23847 (-0.09%)
Maximum	-0.96017	-0.99596 (3.73%)	-0.94059 (-2.04%)
Mean	-1.114	-1.11624 (0.20%)	-1.11516 (0.10%)
Standard Deviation	0.058955	0.070113 (18.93%)	0.061811 (4.84%)

Table 5.4 Statistics of IPA-1 derivatives, subject to *MSR*

Statistic	Pure-Fluid	Pure-Packet (% deviation)	Hybrid (% deviation)
Minimum	0.200865	0.200699 (-0.08%)	0.198163 (-1.35%)
Maximum	0.237344	0.237231 (-0.05%)	0.242321 (2.10%)
Mean	0.221837	0.220336 (-0.68%)	0.220748 (-0.49%)
Standard Deviation	0.008131	0.00739 (-9.11%)	0.007979 (-1.87%)

Table 5.5 Statistics of IPA-2 derivatives, subject to *MSR*

Statistic	Pure-Fluid	Pure-Packet (% deviation)	Hybrid (% deviation)
Minimum	-0.23248	-0.22841 (-1.75%)	-0.22937 (-1.34%)
Maximum	-0.18992	-0.19438 (2.35%)	-0.19166 (0.92%)
Mean	-0.21285	-0.21107 (-0.84%)	-0.21189 (-0.45%)
Standard Deviation	0.00857	0.007238 (-15.54%)	0.007289 (-14.95%)

Table 5.6 Statistics of IPA-3 derivatives, subject to *MSR*

Statistic	Pure-Fluid	Pure-Packet) (% deviation)	Hybrid (% deviation)
Minimum	-0.00741	-0.00755 (1.89%)	-0.00783 (5.67%)
Maximum	-0.00568	-0.00557 (-1.94%)	-0.00535 (-5.81%)
Mean	-0.00651	-0.00647 (-0.61%)	-0.00648 (-0.46%)
Standard Deviation	0.000338	0.000377 (11.54%)	0.000398 (17.75%)

Table 5.7 Statistics of IPA-4 derivatives, subject to *MSR*

Statistic	Pure-Fluid	Pure-Packet (% deviation)	Hybrid (% deviation)
Minimum	0.191327	0.193387 (1.08%)	0.191425 (0.05%)
Maximum	0.231203	0.227628 (-1.55%)	0.229889 (-0.57%)
Mean	0.212309	0.210663 (-0.78%)	0.211316 (-0.47%)
Standard Deviation	0.008424	0.007225 (-14.23%)	0.007254 (-13.89%)

Table 5.8 Statistics of IPA-5 derivatives, subject to *MSR*

Statistic	Pure-Fluid	Pure-Packet (% deviation)	Hybrid (% deviation)
Minimum	0.005686	0.005564 (-2.15%)	0.005352 (-5.87%)
Maximum	0.007388	0.007576 (2.54%)	0.007835 (6.05%)
Mean	0.006488	0.006461 (-0.42%)	0.006471 (-0.26%)
Standard Deviation	0.000341	0.000383 (12.32%)	0.0004 (17.30%)

Table 5.9 Statistics of IPA-6 derivatives, subject to *MSR*

The tables above show excellent agreement across versions of the statistics of approximately stable IPA derivatives. All statistics other than the standard deviations fall within a few percentage points of each other (in particular, all means are within 1%). However, the coefficients of variations are very close. This study attests to the efficacy of using fluid-based IPA derivatives as approximations for pure-packet and hybrid models, subject to the ***MSR*** stopping rule.

5.4 *MSR*-to-*HSR* Comparison Study

The goal of this study is to compare compatible almost stable and approximately stable IPA derivatives for each simulation model version, in order to gauge the tradeoff between the accuracy of IPA derivatives subject to the ***HSR*** stopping rule and potential savings in time complexity afforded by the ***MSR*** stopping rule. To this end, we compare the IPA derivative values obtained from a replication subject to the ***HSR*** stopping rule and a compatible replication (using the same random number stream), subject to the ***MSR*** stopping rule.

The network model studied in this section is the 12-node 11-link network with a bottleneck link, depicted in Figure 5.2 in Section 5.1.2. The T_H parameter for stopping rule ***HSR*** was set to 600 seconds, while the T_{min} parameter for stopping rule ***MSR*** was set to 200 seconds. The results are summarized in Table 5.10 through Table 5.12.

IPA Derivative	T_H	***HSR***	***MSR*** Stopping Time	***MSR*** (% deviation)
Loss Rate as Function of Buffer Size	600	-1.11888	209.8461	-1.04839 (-6.30%)
Workload Time Average as Function of Buffer Size	600	0.22905	201.8987	0.22244 (-2.89%)
Loss Rate as Function of a Service Rate Parameter	600	-0.21814	201.5469	-0.20812 (-4.59%)
Workload Time Average as Function of a Service Rate Parameter	600	-0.00660	201.8987	-0.00676 (2.40%)
Loss Rate as Function of an Arrival Rate Parameter	600	0.21805	201.8987	0.20900 (-4.15%)
Workload Time Average as Function of an Arrival Rate Parameter	600	0.00660	201.82	0.00675 (2.30%)

Table 5.10 Pure-fluid version comparison of IPA derivatives subject to *HSR* and *MSR*

IPA Derivative	T_H	*HSR*	*MSR* Stopping Time	*MSR* (% deviation)
Loss Rate as Function of Buffer Size	600	-1.12054	209.8461	-1.04839 (-6.44%)
Workload Time Average as Function of Buffer Size	600	0.22901	201.9969	0.22250 (-2.84%)
Loss Rate as Function of a Service Rate Parameter	600	-0.21806	201.5469	-0.20813 (-4.56%)
Workload Time Average as Function of a Service Rate Parameter	600	-0.00659	201.9969	-0.00676 (2.54%)
Loss Rate as Function of an Arrival Rate Parameter	600	0.21706	201.9969	0.20845 (-3.97%)
Workload Time Average as Function of an Arrival Rate Parameter	600	0.00659	201.9303	0.00676 (2.48%)

Table 5.11 Pure-packet version comparison of IPA derivatives subject to *HSR* and *MSR*

IPA Derivative	T_H	*HSR*	*MSR* Stopping Time	*MSR* (% deviation)
Loss Rate as Function of Buffer Size	600	-1.11888	209.8461	-1.04839 (-6.30%)
Workload Time Average as Function of Buffer Size	600	0.22905	201.6441	0.22145 (-3.31%)
Loss Rate as Function of a Service Rate Parameter	600	-0.21843	201.2899	-0.20864 (-4.48%)
Workload Time Average as Function of a Service Rate Parameter	600	-0.00657	201.6441	-0.00671 (2.15%)
Loss Rate as Function of an Arrival Rate Parameter	600	0.21828	201.6441	0.20834 (-4.55%)
Workload Time Average as Function of an Arrival Rate Parameter	600	0.00657	201.8247	0.00672 (2.28%)

Table 5.12 Hybrid version comparison of IPA derivatives subject to *HSR* and *MSR*

The tables above show excellent agreement across stopping rules of compatible almost stable and approximately stable IPA derivatives. All IPA derivative values are within 6% regardless of version (interestingly, the deviations are very close across versions). However, the time complexity of computing IPA derivatives under ***MSR*** is some 1/3 of their counterparts under ***HSR***. This study attests to the efficacy of using the ***MSR*** stopping rule to compute approximately stable IPA derivatives instead of almost stable IPA derivatives, but at a fraction of the time complexity.

6. Conclusion

The empirical studies described in this paper can be summarized as follows.

First, **pure-fluid IPA derivatives can be accurately computed from pure-packet and hybrid models**. This conclusion is supported by the study in Section 5.1 (under ***HSR***) and Section 5.2 (under ***MSR***), where various compatible simulation model versions (pure-fluid, pure-packet, and hybrid) gave rise to corresponding IPA derivatives within a few percentage points of relative deviations. Since the IPA derivatives are non-parametric, this opens the door for possible practical applications of IPA derivatives to control and optimization of real-life systems as well as simulation models.

Second, **the *MSR* stopping rule yields IPA derivatives with relatively little variability across replications**. This conclusion is supported by the study in Section 5.2, and motivates using this rule to approximate stable (long-run) IPA derivatives.

Finally, **the *MSR* stopping rule is superior to the *HSR* stopping rule in that it provides comparable computational accuracy at a fraction of the computational complexity**. This conclusion is supported by the study in Section 5.4, which compared identical replications with different stopping rules.

References

[1] Bratley, P., Fox, B.L. and L.E. Schrage, (1987) *A Guide to Simulation*, Springer-Verlag.

[2] Breiman, L. (1968) *Probability*, Addison-Wesley, Reading, Mass.

[3] Cassandras, C.G. and S. Lafortune (1999) *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, Boston, Massachusetts.

[4] Cassandras, C.G., G. Sun and C.G. Panayiotou (2001) "Stochastic Fluid Models for Control and Optimization of Systems with Quality of Service Requirements" *Proc. 40th IEEE Conference on Decision and Control (CDC01)*, Orlando, Florida.

[5] Cassandras, C.G., Y. Wardi, B. Melamed, G. Sun and C.G. Panayiotou (2002) "Perturbation Analysis for On-Line Control and Optimization of Stochastic Fluid Models" *IEEE Trans. On Automatic Control*, AC-47(8), 1234-1248.

[6] Cassandras, C.G., G. Sun, C.G. Panayiotou, and Y. Wardi (2003) "Perturbation analysis and control of two-class stochastic fluid models for communications networks", *IEEE Transactions on Automatic Control*, Vol. 48, 770-782.

[7] Hall, E.A. (2000) *Internet Core Protocols: The Definitive Guide*, O'Reilly & Associates, Inc.

[8] Ho, Y.C. and X.R. Cao (1991) *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer Academic Publishers, Boston, MA.

[9] Kesidis, G., A. Singh, D. Cheung and W.W. Kwok (1996) “Feasibility of Fluid-Driven Simulation for ATM Network”, *Proc. IEEE Globecom 3*, 2013--2017.

[10] Kleinrock, L. (1975) *Queueing Systems*, Vol. I: Theory, Wiley, New York, NY.

[11] Kobayashi, H. and Q. Ren (1992) “A Mathematical Theory for Transient Analysis of Communications Networks”, *IEICE Trans. on Communications*, E75-B, 1266-1276.

[12] Kyas, O. (1996) *ATM Networks*, second edition, International Thomson Computer Press.

[13] Law, A.M. and W.D. Kelton (1991) *Simulation Modeling & Analysis*, (second edition), McGraw-Hill.

[14] Liu, B., Y. Guo, J. Kurose, D. Towsley and W.B. Gong (1999) “Fluid Simulation of Large Scale Networks: Issues and Tradeoffs”, *Proc. Intl. Conf. on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada.

[15] Melamed, B., S. Pan and Y. Wardi, (2004) “HNS: A Streamlined Hybrid Network Simulator”, *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, , Vol. 14, No. 3251-277.

[16] Milidrag, N., G. Kesidis and M. Devetsikiotis (2000) “An Overview of Fluid-Based Quick Simulation Techniques for Large Packet-Switched Communication Networks”, *Proc. SPIE ITCom*, Denver, Aug. 2001.

[17] Nicol, D.M. (2001) “Discrete Event Fluid Modeling of TCP”, *Winter Simulation Conference (WSC 01)*, Arlington, Virginia.

[18] Pan, S. (2005) *Hybrid Network Simulation*, Ph.D. thesis, Rutgers Center for Operations Research (RUTCOR), Rutgers University.

[19] Sun, G., C.G. Cassandras, Y. Wardi, C.G. Panayiotou, and G. Riley (2004) “Perturbation Analysis and Optimization of Stochastic Flow Networks”, *IEEE Transactions on Automatic Control*, Vol. 49, No. 12, 2143-2159.

[20] Wardi, Y. and B Melamed. (2001) “Variational Bounds and Sensitivity Analysis of Continuous Flow Models”, *J. of Discrete Event Dynamic Systems*, 11(3), 249-282.

[21] Wardi, Y., B. Melamed, C.G. Cassandras and C.G. Panayiotou (2002) “On-Line IPA Gradient Estimators in Stochastic Continuous Fluid Models”, *J. of Optimization Theory and Applications*, 115(2), 369-405.