INTERMON-IST-2001-34123

Evaluation of Inter-Domain QoS Modeling, Simulation and Optimization

(Deliverable 19)

WORK-PACKAGE NO. / TITLE	WP5 – Modelling and Simulation
Planned Issueing Date	2004-04-24
Distribution	WP5 Partners
Document Identifier	im-wp5-v100-UniBe-D19-pf
File name	im-wp5-v100-UniBe-D19-pf.doc
Version	V 1.0
Editor/Author	**University of Bern**
Contact Person(s)	Matthias Scheidegger <mscheid@iam.unibe.ch >
Authors	Maurizio Bartoli (Tilab) Christof Brandauer (SAR) Peter Dorfinger (SAR) Donato Emma (CINI) Ulrich Hofmann (SAR) Sándor Kardos (BUTE) Fabrizio Orlandi (Tilab) Antonio Pescapè (CINI) István Ráncsik (BUTE) Matthias Scheidegger (UniBe)

Change History

V0.01	Empty document with TOC
V0.02	Improved TOC
V0.1	Inserted available contributions for Budapest
V0.11	Contribution from SAR included
V0.12	Included hybrid simulator evaluation and CINI's references
V0.8	Everything included, Executive Summary still missing.
V0.9	Wrote Exec. Summary, included small corrections from CINI & Tilab. Fixed references. Adapted to Word's "interoperability".
V1.0	Layout fixes and last changes to the Executive Summary.

Table of Contents

List of Figures

List of Tables

Executive Summary

This document describes the evaluation of the Intermon modeling and simulation toolkit, which was described in Deliverables 6 (D6: “Modelling and Simulation Toolkit Specification”) and 11 (D11: “Integration of the Inter-Domain Modelling and Simulation Toolkit”).

Deliverable 6 described several modelling and simulation approaches. The common basis for these approaches is the inter-domain point of view. Traditional simulation approaches try to model the network as exactly as possible, making simulation scenarios of inter-domain networks virtually impossible due to scalability problems. Choosing a suitable abstraction reduces this scalability problem and enables the simulation of large- scale inter-domain scenarios, at the cost of reduced accuracy. Four approaches were implemented, NS2-Hybrid by the University of Bern, TSSIM by Budapest University of Technology, RTC-FSIM by Salzburg Research and Inter-IP (called “Planisfero” in D6) by Telecom Italia Lab.

In contrast to the other evaluation deliverables this document focuses purely on the evaluation of models and simulators. Issues related to the global Intermon architecture (the messaging system, generation of simulation requests through the GUI) or the visualization of simulation results are out of scope, even though they were used to perform the experiments described in this deliverable.

The evaluation task was split into two parts: The first part contains basic model and simulator implementation testing, simulator-specific evaluation and the evaluation of scalability. However, the ability to choose from several approaches is only valuable if the advantages and disadvantages of the choices are known. Because of the lack of measurements from real networks there was also a lack of real-world validation for most approaches. The second part of evaluation consequently focused on a common testbed provided by Salzburg Research, which on one hand provided measurement data to validate the simulators and on the other hand serves as reference scenario for the comparison of the approaches.

The main conclusions from the evaluation are that all four modelling and simulation tools perform satisfyingly. With the Inter-IP tool the user can evaluate scenarios very quickly and with good results. However, the rather strict requirements for scenarios restrict the possible situations when this tool can be used. Alternatively, the time series based TSSIM can be applied to study such scenarios. While this tool does not simulate delays very accurately it is very well suited to find “problematic” network load situations, which can then be further studied with either NS2-Hybrid or RTC-FSIM. NS2-Hybrid is both scalable and accurate but has shown to be sensitive to inaccurate scenarios (e.g. missing or bad packet length estimations). RTC-FSIM also accurately simulates QoS on the network and it also scales well with respect to link bandwidth and topology. However, the scalability of its numerical algorithms remains an open question.

The remainder of this document is structured as follows: Section 1 gives a brief overview of the evaluated modelling and simulation approaches and also gives a short description of the procedures and criteria for simulator evaluation. It also contains a description of the common testbed for reference in later sections. Part one of evaluation (simulator-specific evaluation) is described in Section 2. Section 3 contains the evaluation results from the common testbed scenario and presents a usage scenario for the simulation toolkit based on the results in this section. Finally, Section summarizes the document.

1 Introduction

The modelling and simulation workgroup aims at enhancing the capabilities of the Intermon system for inter-domain QoS analysis. In order to achieve this, several approaches to scalable inter-domain modelling and simulation have been developed and specified in Deliverable 6 ("Modelling and Simulation Specification", December 2002). Since each of the approaches has different strong points and weaknesses we did not unify them but decided to develop four different approaches simultaneously. Section 1.1 gives a brief overview of them.

This document describes the evaluation of the four modelling and simulation approaches in terms of accuracy, scalability, and performance. The evaluation of the integration into the global Intermon architecture, e.g. the automated retrieval of IPFIX measurements from the database or the visualisation of the results, is part of other deliverables and out of the scope of this document.

Because several independent modelling and simulation approaches were implemented we cannot restrict ourselves to individually evaluate each of them. We also need a comparison of the approaches based on a common scenario to find the strong points and weaknesses of each of them. Using this knowledge a user can then choose a suitable simulator for a specific scenario and/or question.

This document is structured accordingly: Section 1 gives an overview of the common scenarios and discusses suitable evaluation criteria. Section 2 presents the evaluation and tests done independently for each approach, and Section 3 compares the modelling and simulation approaches based on a common measurement based scenario.

1.1 Modelling and Simulation Approaches

Hybrid Simulation

The hybrid simulation module combines packet-based simulation of ns-2 with analytical models by using a hot-plug mechanism, which makes single ns-2 nodes behave like whole networks (e.g. autonomous systems). This abstraction allows the user to simulate large topologies in a fraction of the time a full scale packet-based simulation would take.

The NS2 Hybrid simulator is expected to perform well in the following application areas:

- End-to-end QoS evaluation of single flows – simulated using traditional packet-based models – over a complex backbone network.
- The effect of changes in a backbone network (e.g. addition/removal of links, capacity changes, big changes of network load due to new SLAs, etc.) on flows traversing the domain.

RTC-FSIM

The Rate and Time Continuous Fluid Simulation (RTC-FSIM) is a novel fluid simulation approach developed within the InterMON project. In contrast to other fluid-based modelling techniques, which require an event-based simulator for the execution of the model, the RTC-FSIM approach models all traffic as continuous signals and describes the signal transformations by means of differential equations.

As there are no events in the RTC-FSIM approach it does not suffer from the performance drawbacks that may arise from the ripple effect [Liu] in event-based fluid simulators. The performance of RTC-FSIM basically depends on the implementation for solving the differential equations.

As RTC-FSIM performance is completely independent of the link speeds and amount of traffic it is particularly well suited for scenarios with high link speeds and large amounts of aggregated traffic. Another key feature of RTC-FSIM is the ability to increase the simulation speed by allowing for less accurate results.

It is therefore suggested to employ RTC-FSIM in large inter-domain scenarios. The simulator produces queuing delay, loss, and throughput results. As the simulator is fully integrated with IPFIX measurements it is very useful for evaluating alternative routes for portions of the observed traffic. For the definition of that "portion" the full flexibility of the IPFIX meter rules is available. As an example, the "portion" could be defined as all traffic going to destination domain X; it could be all IP telephony flows; it could be all Web traffic coming from domain Y.

RTC-FSIM can also be used to simulate the effect of additional traffic beside the measured traffic. The addition can be a general load increase by some factor, a previous IPFIX measurement or a general traffic model (e.g. of Web traffic).

Inter-IP

The INTER-IP module evaluates the end-to-end packet or volume transfer delay performance for a traffic relation (i.e. a flow identified by the source and destination IP address and by the service class if differentiated service is used) that crosses multiple domains in a Differentiated Services context.

A suggested application in the InterMON context is the rapid comparison (on the base of the delay metric) of alternative routes for the same flow. The tool is able to compute the end-to-end delay performance in a small amount of time (if compared to simulation) because is based on a pure analytical model resolved in a closed form.

Time series simulator (TSSim)

The time series simulator evaluates the QoS state of the network in terms of throughput, drops, delay and jitter. Since the simulator is based on aggregate load information the QoS values represent an average for the traffic as a whole.

TSSim has the following suggested application areas:

- Analysing the effect of additional traffic on the QoS state of the network.
- Analysing the effect of rerouting a part of the traffic on the QoS state of the network.

1.2 Evaluation Procedures and Criteria

Modelling and simulation traditionally are the two approaches to evaluate network protocol design and network performance [FP01].

Modelling is often difficult with today's large networks and complex traffic patterns, so researchers have turned increasingly to simulation [Bre00]. As far as Intermon targets, in order to evaluate inter-domain QoS (Quality of Service) in multi-domain Internet infrastructures, we must be able to perform large-scale simulations. Large-scale simulation suffers of scalability problems and one approach to the scalability is to apply parallel and distributed simulation [CNO99], dividing tasks into parts coordinated over numbers of machines: this fact can require expensive hardware (CPU and memory) and have high overhead. Parallelism can improve simulation scale in ratio to the number of machines added, but this linear growth is not sufficient to add several orders of magnitude scaling needed. A complimentary solution is to slim down simulations by abstracting out details [HEH98]. The idea is to analyze simulations, identify the bottleneck, and eliminate it by abstracting unnecessary details. The risk of abstraction is that simulation results may be distorted.

In this section we describe best current practices for validating simulations approaches. Validation is the process of assuring that a model provides meaningful answers to the questions being investigated.

Regardless of the particular simulator employed, the user must understand whether the results of the simulation will be valid for the question at hand. One obvious approach is to compare the simulation results to results from a particular real-world implementation of a network. When network topologies are large or when protocols are under-specified, validation through direct comparison can prove difficult. Thus, in order to define a general framework we select a set of best practices to validate a specific simulation [HMK01]:

- Simulation results must be reproducible.
- Results must be interpreted appropriately, even at different scales.

- Used models and implementations must emphasize several (or all) aspects of the observed system.
- Where the model involves interactions over time among various independent entities, be sure to introduce asynchrony where needed to imitate the operation of real systems.
- Where the size of the simulation must be reduced to execute within memory and CPU cycle limitations, care must be exercised to avoid introducing artificial boundaries into the used model.
- Public availability of simulation source code is also important to allow examination for correct operation.
- The simulator must be easy to extend if its users are to add new functionality, explore a range of scenarios, and study new protocols.
- Visual representations of simulation states and results could be useful. A simple and intuitive GUI is also helpful.
- Varying simulation granularity allows a single simulator to accommodate both detailed and high-level simulations.
- An abstracted model must still be validated against a more detailed model running at slower speed, or against field experiments of sufficiently large scale. Further, new phenomena might emerge from interactions as networks increase in size.
- Simulation time should be a short as possible.
- The simulator should support automatic configuration and/or remote control.

In the Intermon project simulations are most useful as a tool for building understanding of dynamics, or to illustrate a point, or to explore unexpected behaviour. For these reasons, in a network simulator a wide range of available modelling and simulation approaches are needed. Thus, as far as measured parameters, in the field of simulation tool for an inter-domain scenario, we select the following set of properties:

- A per flow end-to-end QoS (delay, jitter, throughput, packet loss, packet wrong) parameters evaluation.
- Aggregate traffic data analysis could be available (i.e. average value of delay, jitter, throughput, packet loss and packet wrong over the traffic trunk).
- Simulation tools must provide the possibility to model ASs (Autonomous Systems), border routers, core routers, backbone links, access networks, etc. Indeed, a model should describe a set of domains, characterize their interconnection topology, and simulate user behaviour or packet traffic at a large-enough scale to accommodate spatial correlations that might arise in the real network.
- Dependencies between network dynamics (e.g. topology changes, bandwidth changes, load changes, rerouting, etc.) and flows exiting in the domain must be taken into account.
- Accuracy can be traded against performance by adjusting the abstraction level.

1.3 Testbed Layout and Scenarios

A goal of the work presented here is to make an in-depth comparison of the simulation tools against network measurements. Additionally, the measurement based simulation tools are compared against each other on the basis of a common set of measurement data.

To enable these studies it is necessary to have full control over the measured network. One must know the critical traffic and router parameters and have the ability to place measurement agents at the desired interfaces. In order to enable such evaluation, we set up a laboratory testbed in Salzburg. This testbed is used in two ways: On one hand it is used for evaluation of some of the simulation tools.

Additionally it serves as a common scenario for all modeling and simulation approaches, in order to be able to compare the performance and accuracy of the different simulation tools.

The general setup of the testbed is described in the following; details on the scenarios and the traffic mixes will be specified in the sections that make use of the testbed in some form.

1.3.1 Testbed setup

The testbed consists of 3 Cisco routers (2 Cisco 3600, 1 Cisco 7204), 3 Linux machines configured to emulate network delay and 4 Linux end system for traffic sources and sinks. The topology of the testbed is depicted in Figure 1.

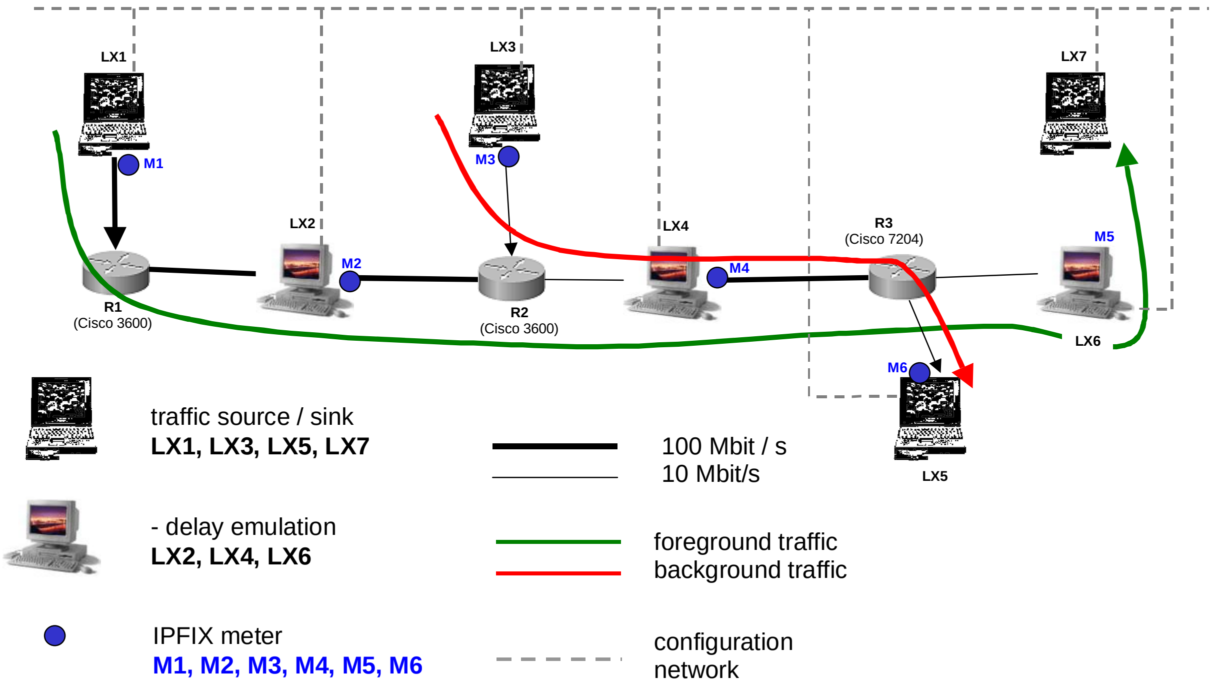


Figure 1 – Measurement testbed

The delay emulation on hosts LX2, LX4, and LX6 is realized with the Nistnet tool [SR-Nist]. It is configured to add a constant delay of 50ms to emulate a round trip time that is somewhere in the range of typical wide-area connections.

The blue circles in Figure 1 indicate the locations of the IPFIX meters. All meters M1 – M6 report their measured data to a common collector (not shown in the figure). This collector, as well as all Linux hosts, is connected to a separate configuration network indicated by dashed grey lines. Using this approach, the IPFIX result data is never sent on a link where the traffic to be measured is flowing. Therefore, the measurements do not influence the measured traffic in any way.

In order to enable high-precision 1-way delay measurements the hosts LX1 and LX7 are synchronized through a GPS clock signal.

1.3.2 Scenario

The basic scenario is that delay sensitive foreground traffic is sent from LX1 to LX7. At the same time, background traffic is sent from LX3 to LX5. Depending on the amount of background traffic, a bottleneck at the link from R2 to LX4 is created.

The type of traffic employed, depends on the specific test and will be described in the appropriate sections.

The foreground traffic is made of up Voice over IP flows (VoIP) that are sent from LX1 to LX7. The background traffic is made up of MPEG4 video flows that are sent from LX3 to LX5. More specifically, the following traffic is used for the measurements:

VoIP	G.711 flows (160 Byte payload every 20 ms)
Video	High quality MPEG 4 traffic. We make use of the trace file repository provided at [SR-TR1]. In particular, we use the high quality MPEG 4 trace of the Mr. Bean movie that can be found at [SR-TR2]. As the same file is used for several instances of voice flows the starting point within the video trace is randomly chosen for each flow.

Several measurement runs are made in order to study the impact of a varying amount of video cross traffic on the quality of the VoIP traffic. In each run, 10 voice flows are started. Over the different runs, the number of video flows is varied:

Run ID	voice flows	video flows
1	10	0
2	10	5
3	10	6
4	10	7
5	10	8
6	10	9
7	10	10
8	10	11
9	10	12
10	10	13
11	10	14
12	10	15

Table 1 – Testbed traffic configurations

1.3.3 Measurements

As described above, IPFIX meters M1 – M6 are used to measure the load at the various locations. The meters M1 – M5 are all configured with the same rule file:

```
ruleid 300

ver 4

start <start-time>

stop <stop-time>

traffic filter out
```

Of course, the `<start-time>` and `<stop-time>` are set to the appropriate values to capture the traffic of each measurement run.

The rule file of meter M6 is basically the same as for the others, except the last line that makes it measure all incoming traffic:

```
traffic filter in
```

All meters report their measured data to a common IPFIX collector. This collector stores the received data directly into the INTERMON database located in Berlin. The communication is done over the virtual private network that connects all INTERMON partner locations.

For the VoIP flows, the 1-way delay is measured with CM Toolset [SR-CMT].

1.3.4 Database Storage

Since the testbed measurements and topology are used for the evaluation of all four simulators they were made available on the central database in Berlin. The scenario descriptions for each simulator could then be generated by extracting the required information from the database and converting it into a simulator-specific format. The physical topology from Figure 1 was stored using the BGP topology seen in Figure 2. The traffic measurements from the testbed were stored in the IPFIX tables of the database. Consequently, the scenario is also selectable in the Intermon GUI, and simulation runs can also be started from there.

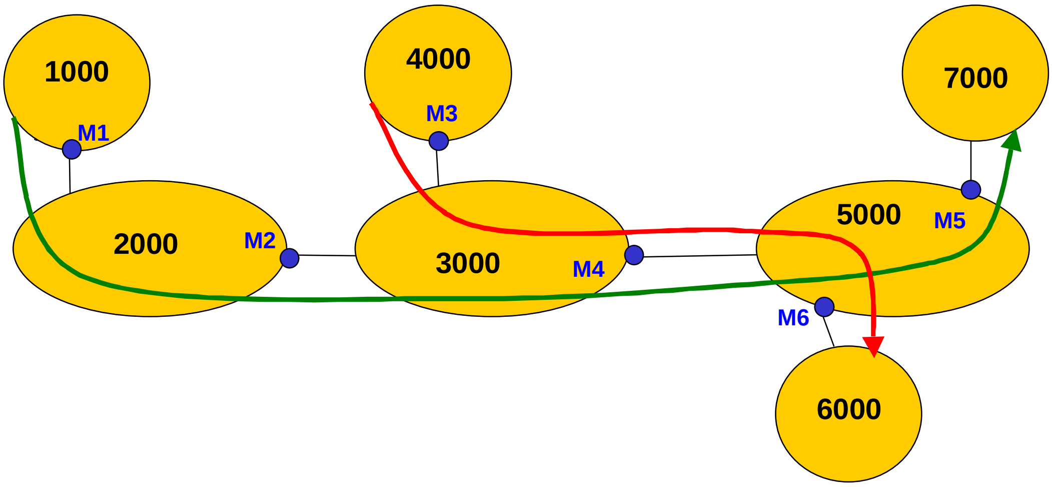


Figure 2 – BGP representation of the testbed topology

2 Evaluation of the Modeling and Simulation Approaches

In this section we present the validation and evaluation work done by the developers of the four modelling and simulation approaches during development. As there was little or no coordination between the development teams and the modelling and simulation approaches differ in many aspects, the experiments performed vary greatly.

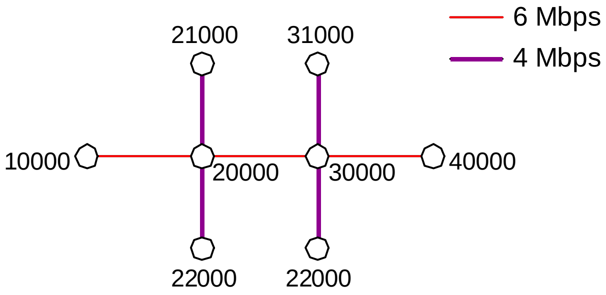


Figure 3 – Scalability test: Small scenario

Nevertheless, we decided to use common scenarios for scalability testing because some of the approaches require real-world trace data, which could be emulated using the ns2 simulator. In order to study the scalability of the approaches two scenarios of very different sizes were chosen. Figure 3 and Figure 4 show the topologies of the small and big scenarios, respectively. Both are classical cross traffic scenarios. In the small one FTPDATA streams flow from 21000 to 32000 and from 22000 to 31000. In the big scenario a great number of FTPDATA cross traffic flows were generated algorithmically. In both scenarios, a G.711 reference stream flows from 10000 to 40000. Since the aim of this experiment was to study the scalability of the modelling and simulation approaches, we did not evaluate the simulation results in detail.

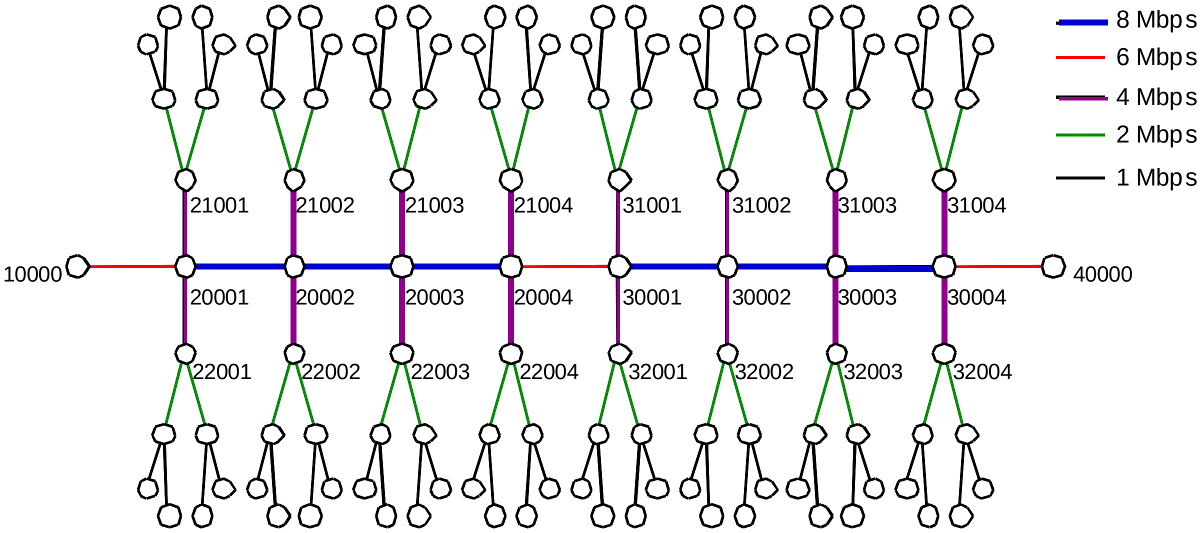


Figure 4 – Scalability test: Big scenario

2.1 NS2 Hybrid

Evaluation of NS2 Hybrid simulation was done in three parts. First, the basic building blocks of the approach (Inter-domain link models and domain models) were tested and compared to their counterparts in the original ns2 simulator. Second, the scalability of the approach was evaluated using the scenario described above.

2.1.1 Inter-Domain Link Models

Inter-domain link models are used to simulate bottleneck links where packet loss occurs. Consequently, we compared the throughput of an ns2 link to the throughput of an inter-domain link model in an extended 2 Mbps single-link scenario, considering five different traffic mixes. The configuration of the traffic mixes can be found in Table 2. Figure 5 shows the results of the test. Both link models performed practically identically for the first four traffic mixes: In the CBR-LOW case all traffic is forwarded. With the CBR-FIT traffic mix, the ns2 link forwards all packets while the analytical link drops a very small number of packets. The reason for this is that the analytical model assumes an exponential arrival process, which in reality is deterministic. For the CBR-HI traffic mix the behavior is identical again.

The five FTPDATA sources in the FTP scenario totally occupy the link. Consequently, the same effects as in the CBR-FIT case can be observed.

CBR-LOW	5 sources totalling to 1 Mbps
CBR-FIT	5 sources totalling to 2 Mbps
CBR-HI	5 sources totalling to 4 Mbps
FTP	5 FTPDATA sources
MIX	2 CBR sources totalling to 2 Mbps, 3 FTPDATA sources

Table 2 – Traffic mixes for IDL model evaluation

Surprisingly, there is a noticeable difference in the second traffic mix with TCP traffic, the MIX scenario. The two CBR sources leave 1 Mbps for the 3 FTPDATA sources who fail to occupy the whole remaining bandwidth. Here, with little bandwidth and small number of sources, there is a significant difference between the ns2 link and the analytical link. It can be explained as follows: The ns2 link drops packets from the tail of its queue, whereas the analytical link randomly selects packets to be dropped, creating a similar effect as random early detection (RED) queues [FJ93].

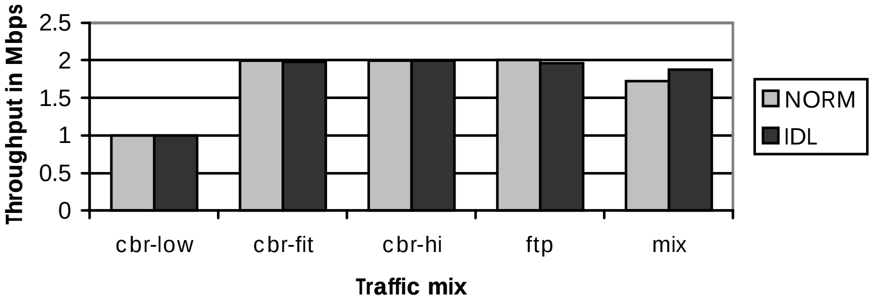


Figure 5 – Data rates of NS2 and analytical links

2.1.2 Domain Models

While the dynamic network effects are modelled by inter-domain link models, domain models simulate areas of the network with quasi-constant behavior. This implies negligible packet loss ratios and queuing. Domain models therefore only simulate delay according to delay distributions based on measurements taken from the real network, or traces generated by a simulator.

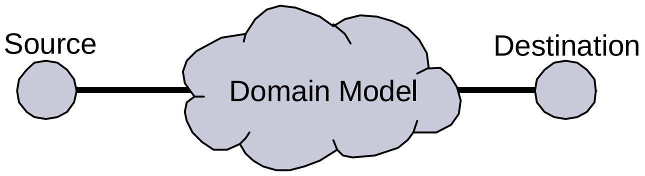
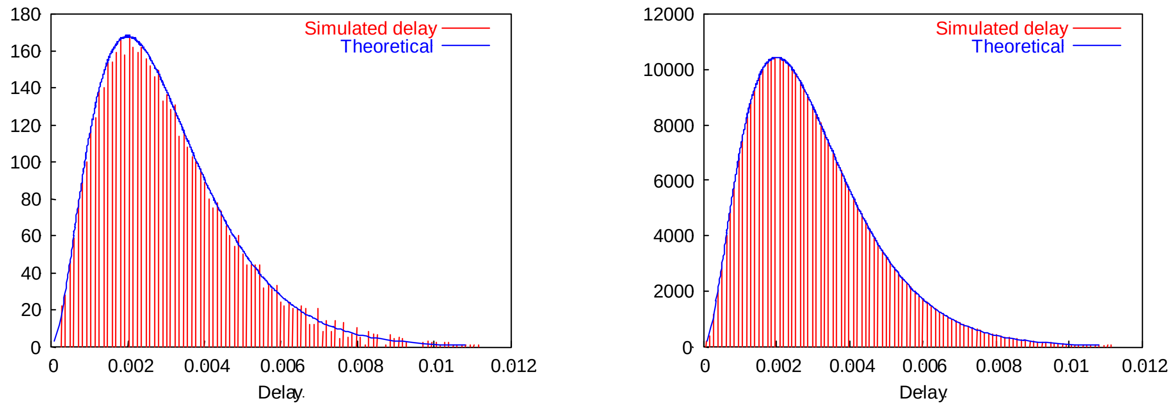


Figure 6 – Domain evaluation scenario

Early versions of the domain model proved to distort the distribution under high traffic loads with non-CBR packet arrival processes. This has been improved in recent versions. To verify this we created a very simple scenario with only a single domain model, connected to a source and a destination ns2 node (Figure 6). The domain model was parameterized using a theoretical gamma distribution. We ran the simulation two times, once with a low bandwidth CBR stream and once with a high bandwidth stream with exponential packet interarrival times. Figure 7 illustrates the results: Both experiments yielded clearly gamma-distributed delays. The bigger sample on the right hand side now leads to a better approximation of the gamma distribution without any distortions.



a) With big interarrival times

b) Very small interarrival times

Figure 7 – Delay behavior of a domain model

2.1.3 Scalability

For scalability testing we modeled the big scalability scenario (Figure 4) introduced above using a multi-domain model consisting of two domain models (A and B), and an inter-domain link model as bottleneck between them (Figure 8). The delay distributions for the domain models were extracted from the original trace: The delay from 20001 to 20004 for domain A, and the delay from 30001 to 30004 for domain B. The inter-domain link represents the link in between 20004 and 30001. Since this setup neglects the dynamics of the original scenario we further added two FTPDATA cross traffic streams, from XS1 to XD2 and from XS2 to XD1.

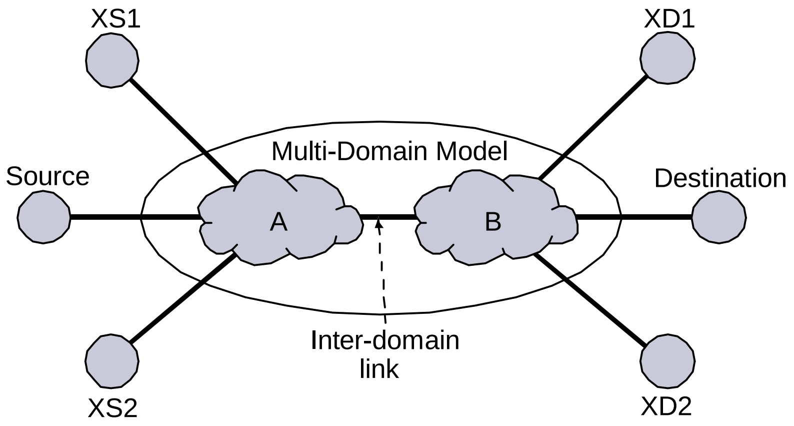


Figure 8 – NS2 Hybrid model of the scalability scenario

The respective model for the small scalability scenario from Figure 3 can be modelled identically, with one change: the domain models are configured to have a “zero” delay distribution, i.e. packets passing the domains are not delayed at all.

When we look at Figure 9 we can see the computation times of both scenarios for NS2 and NS2 Hybrid. The increase in computation time for NS2 is circa factor 11. For NS2 Hybrid, computation time remains nearly constant, which is not surprising given the nearly identical model. More generally, computation will not change if the number of nodes represented by a domain model remains the same. Furthermore, scalability is mainly achieved by replacing NS2 traffic by analytical models since the bandwidth estimation used in hybrid simulation is inefficient when applied to a small number of flows. It is easy to make hybrid simulation scale worse than packet-based simulation if the choice of models is not appropriate.

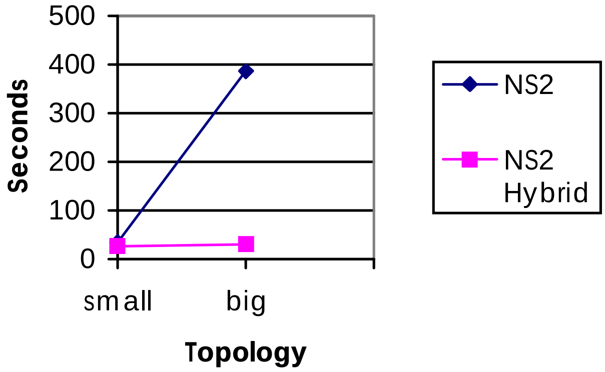


Figure 9 – Computation times for the scalability scenarios

2.2 RTC-FSIM

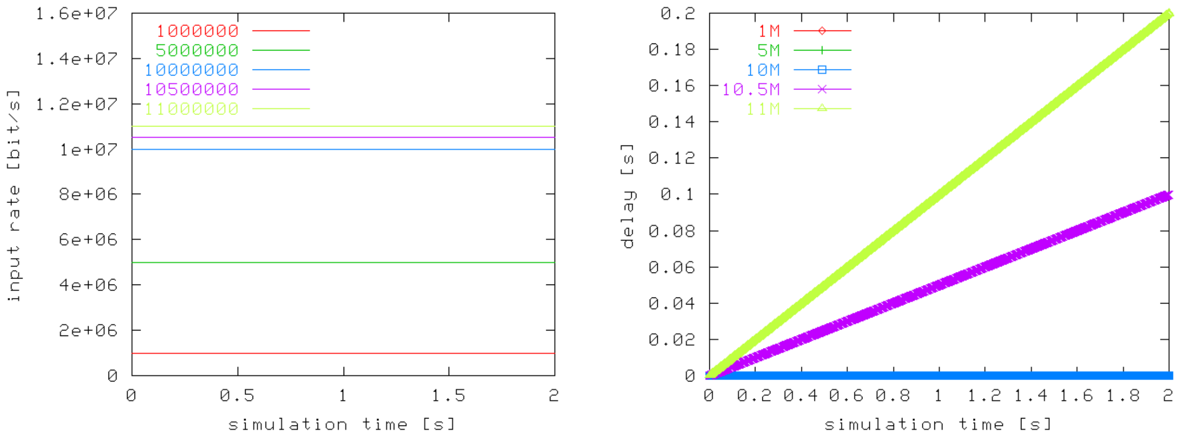
In this section the RTC-FSIM tool is evaluated in 3 parts. In the first part, the correctness of the implementation is evaluated on the basis of simple scenarios where the correct simulation result can be easily computed a priori. In the second part, measurement based simulations are performed and the results are compared to the testbed measurements. Finally, an analysis of the simulators scalability properties is performed.

2.2.1 Correctness of implementation

For the simulations shown in this section, a service rate of 10 Mbit/s is configured for the server – unless noted otherwise.

2.2.1.1 Constant input signal

The input signal to the server is constant. It is expected that the delay signal is zero as long as the input rate is smaller than the service rate and that it grows linearly, proportional to the input rate, if the input rate exceeds the service rate. For each of the input rates 1 Mbit/s, 5 Mbit/s, 10 Mbit/s, 10.5 Mbit/s, and 11 Mbit/s, see Figure 10a, a separate simulation is executed. The resulting delay signals are plotted in Figure 10b. It can be seen, that the results are as expected.



a) Input signals

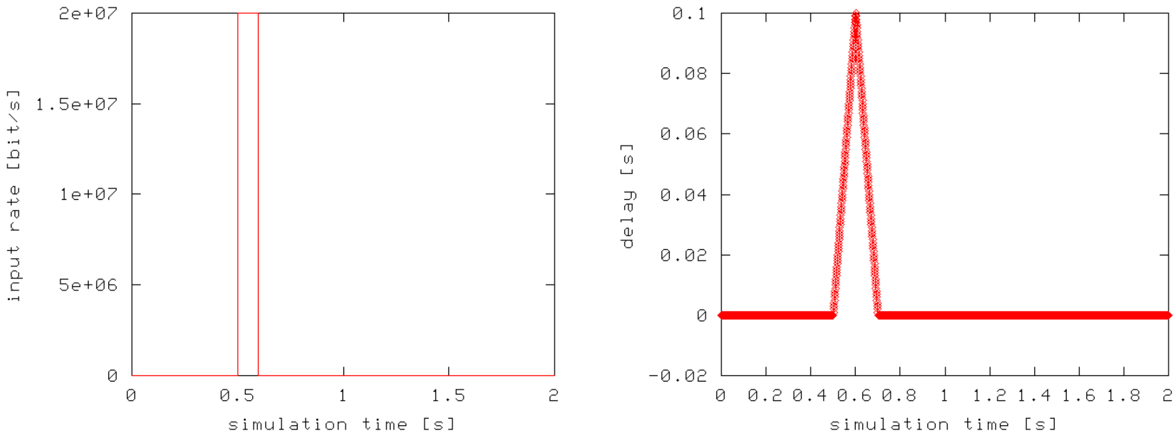
b) Delay Signals

Figure 10 – Constant input signal

2.2.1.2 Single impulse

The input signal has the form of a single impulse. The amplitude and duration of the impulse is such that the server's buffer size is not exceeded. It is expected, that as long as the signal has its high level the delay will increase. After the signal returns to zero the delay must return to zero. The rate of delay increase has to be proportional to the input rate minus the service rate. The rate of delay decrease has to be proportional to the service rate.

The input signal to the server is shown in Figure 11a, the resulting delay signal is shown in Figure 11b. The results are as expected.



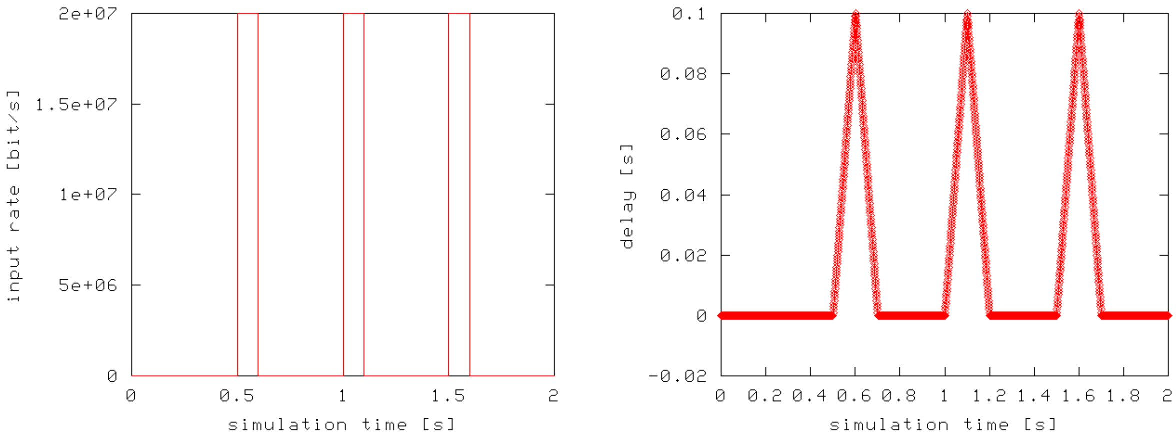
a) Input signal

b) Delay signal

Figure 11 – Single impulse

2.2.1.3 Multiple consecutive impulses

The input signal consists of multiple (*N, N=3*) impulses that are separated by periods of a zero level. The amplitude and duration of the impulse is such that the server's buffer size is not exceeded. It is expected that the resulting delay signal contains *N* delay peaks, one for each input impulse. In between these peaks there must be periods of zero delay. The input signal to the server is shown in Figure 12a, the resulting delay signal is shown in Figure 12b. The results are as expected.



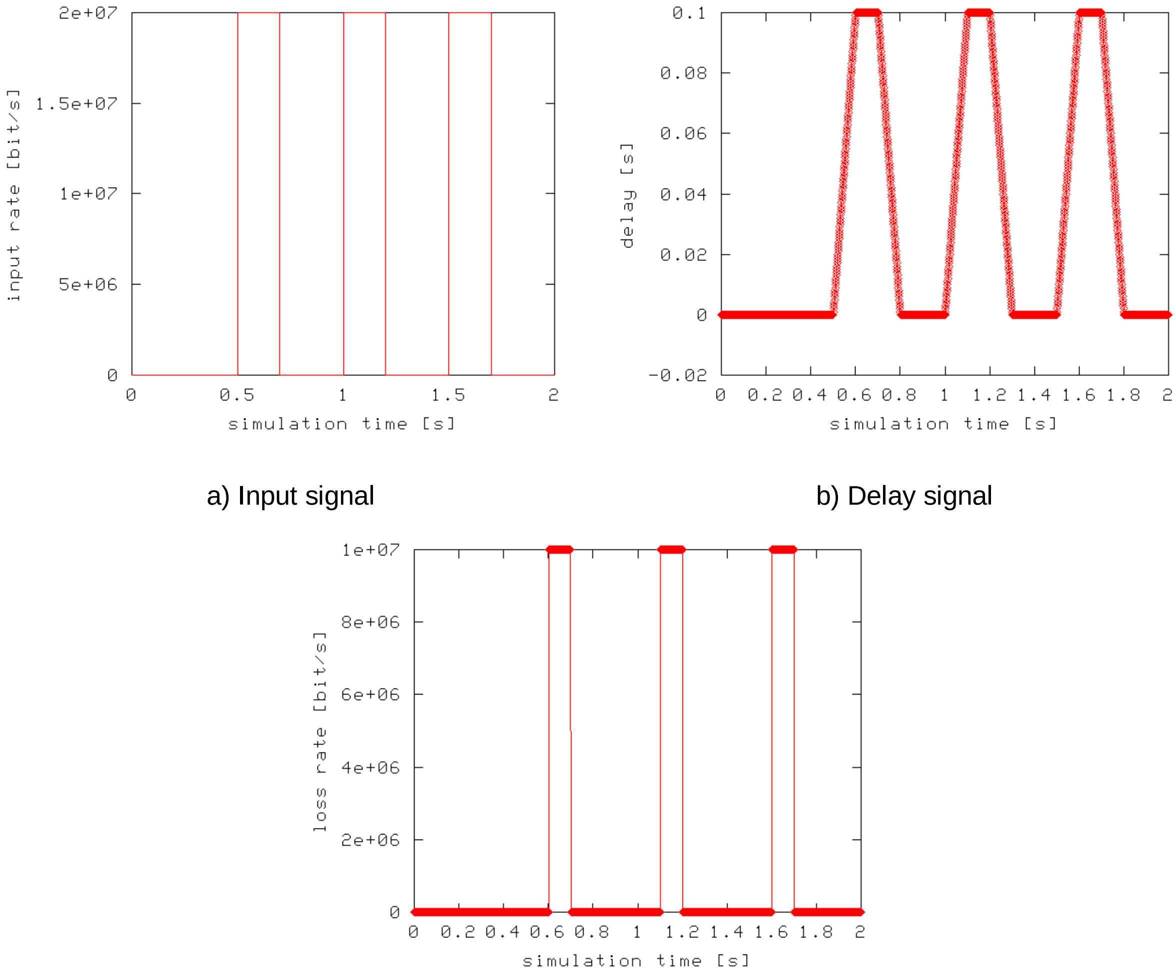
a) Input signal

b) Delay signal

Figure 12 – Multiple consecutive impulses

2.2.1.4 Multiple consecutive impulses exceeding the server's capacity

The input signal consists of multiple (*N, N=3*) impulses that are separated by periods of a zero level. The amplitude and duration of the impulse is such that the server's buffer size is exceeded. As in the previous test, the delay output must consist of N delay changes. However, unlike the previous test, the delay signal must reach a maximum and keep that maximum level until the input impulse ends. The maximum delay value must be equal to the server's buffer size (1 Mbit) divided by its service capacity (10 Mbit/s), thus equal 100 ms. Additionally, when the maximum delay is reached, the server must produce a loss signal. As the high level of the impulse equals to 20 Mbit/s, the loss signal must reach a high of 10 Mbit/s (the input signal minus the service capacity). The input signal to the server is shown in Figure 13a, the resulting delay signal is shown in Figure 13b, the loss signal is shown in Figure 13c. The results are as expected.



a) Input signal

b) Delay signal

c) Loss signal

Figure 13 – Multiple consecutive impulses exceeding the server's capacity

2.2.1.5 Multiple servers, no loss

In this test case, multiple servers are placed one after the other in a chain, i.e. the output signal of server 1 is connected to the input port of server 2 and the output signal of server 2 is connected to the input port of server 3. This simple topology is shown in Figure 14.

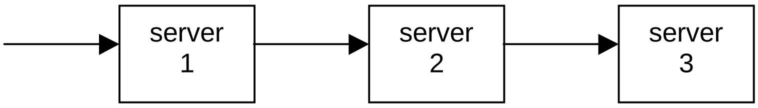
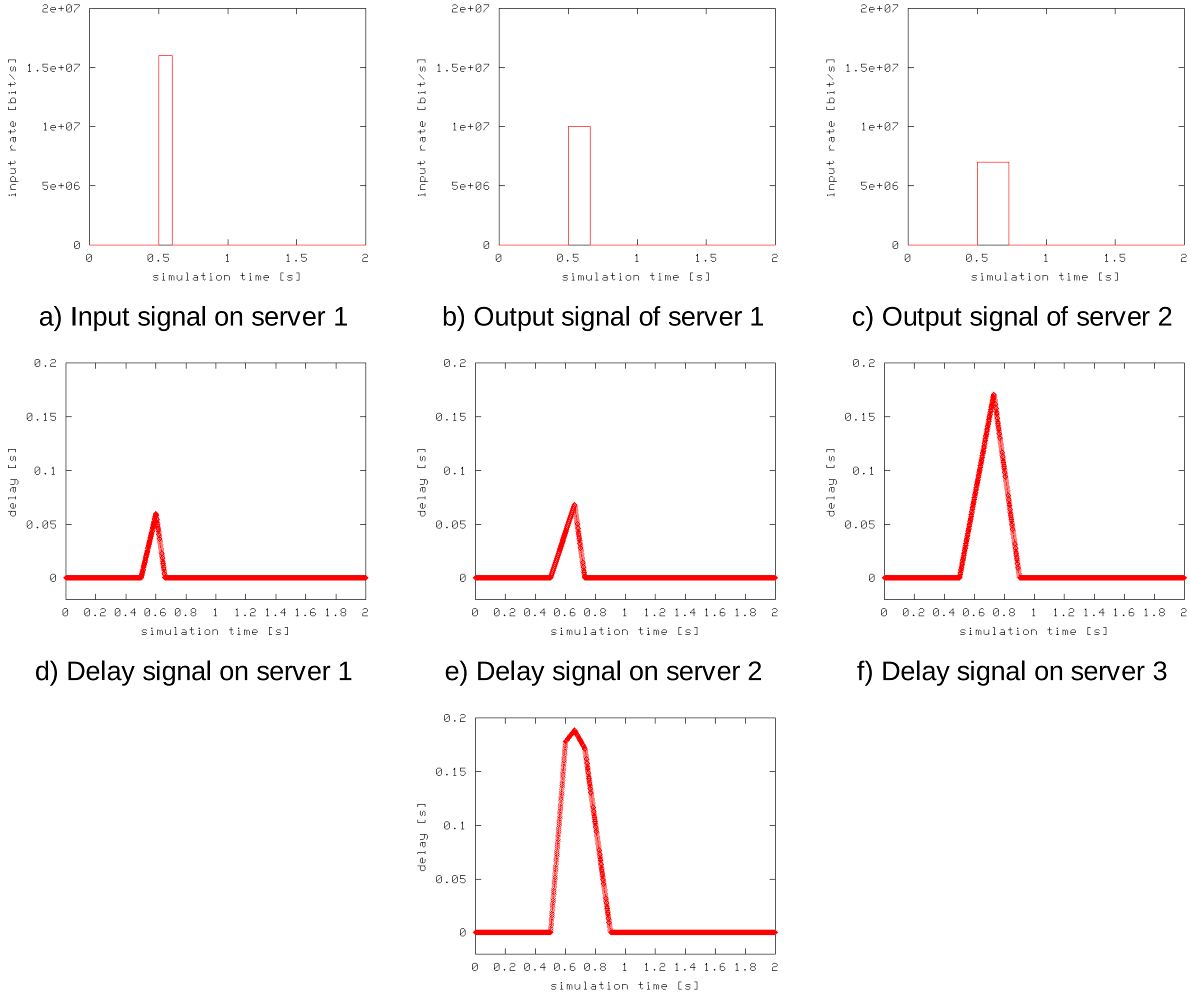


Figure 14 – Multiple server topology

Here, the service capacity is 10 Mbit/s for server 1, 7 Mbit/s for server 2, and 4 Mbit/s for server 3. The amplitude (16 Mbit/s) and duration (100 ms) of the input signal on server 1 (Figure 15) is such that the server's buffer can accommodate the impulse without loss. Naturally, the output signal of each server is shaped to the server's service rate. As the service rate decreases along the path, each server must produce a delay signal that is greater than zero for some amount of time.

The output signals of server 1 and 2 are shown in Figure 15b and Figure 15c, respectively. The delay output is show in Figure 15d-g. The results are as expected.



a) Input signal on server 1

b) Output signal of server 1

c) Output signal of server 2

d) Delay signal on server 1

e) Delay signal on server 2

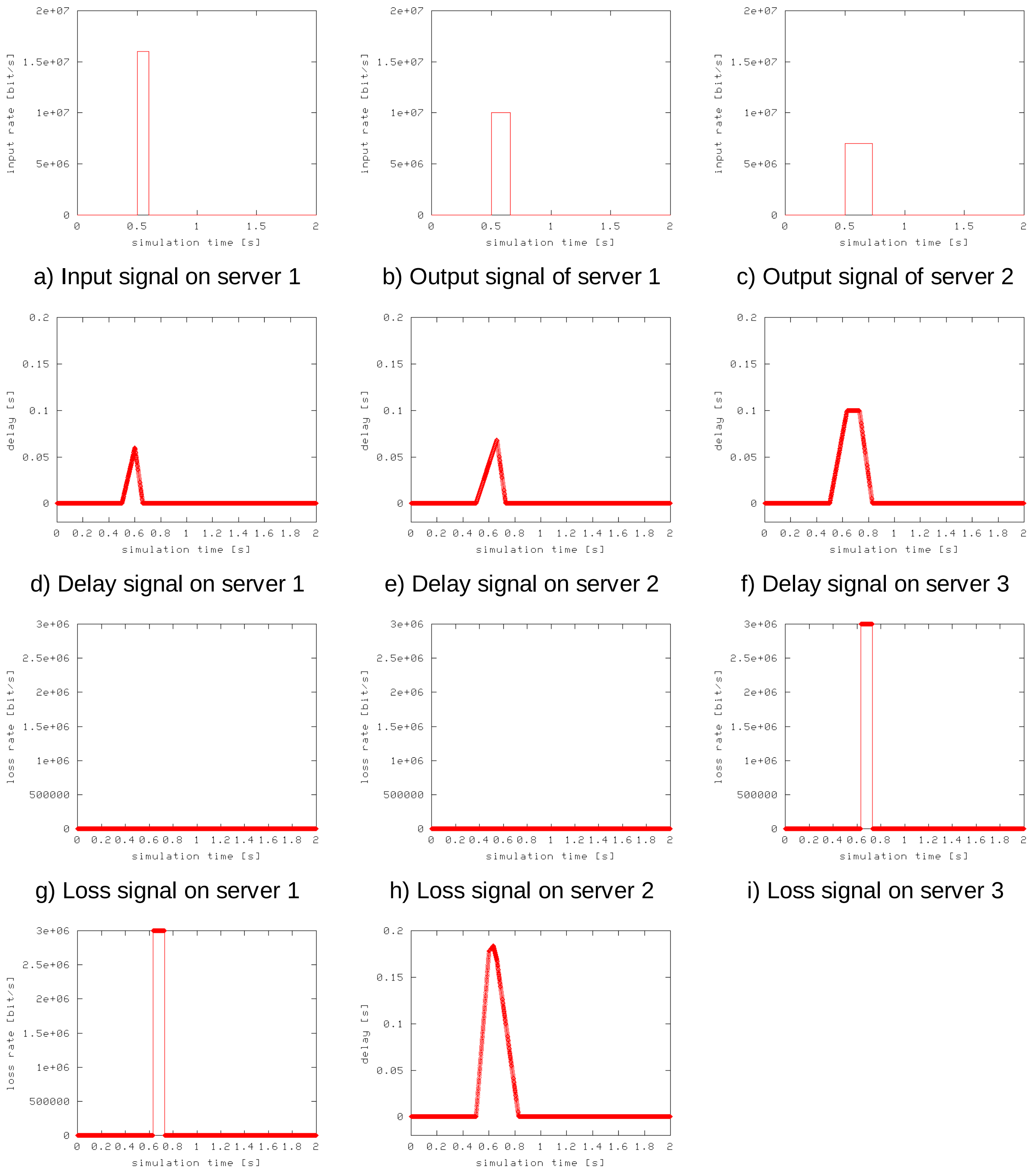
f) Delay signal on server 3

g) End-to-end delay

Figure 15 – Multiple servers, no loss

2.2.1.6 Multiple servers, server capacity exceeded

This test case is similar to the previous one with the difference that the server's buffer capacity is exceeded and loss occurs. The buffer size is configured such that the maximum delay at each server equals 100 ms. Server 3 has a capacity of only 4 Mbit/s and is not able to accommodate the output of server 2 without loss. Thus, the delay at server 3 reaches the maximum possible value (100ms) and a loss signal greater than zero is produced. The input, delay, and loss signals are shown in Figure 16. The results are as expected.



a) Input signal on server 1
b) Output signal of server 1
c) Output signal of server 2
d) Delay signal on server 1
e) Delay signal on server 2
f) Delay signal on server 3
g) Loss signal on server 1
h) Loss signal on server 2
i) Loss signal on server 3
j) Sum of loss signals
k) End-to-end delay

Figure 16 – Multiple servers, capacity exceeded

2.2.1.7 One server, multiple input signals

In this test case a single server is fed by the sum of two input signals (see Figure 17). One signal completely overlaps the second one. The delay output must reflect the arrivals and departures of the signals. The input signals are shown in Figure 18a, the resulting delay output is shown in Figure 18b. The results are as expected.

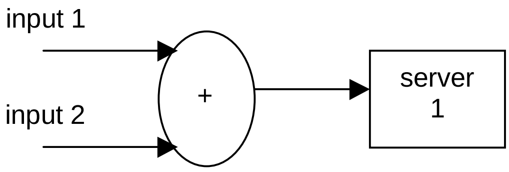
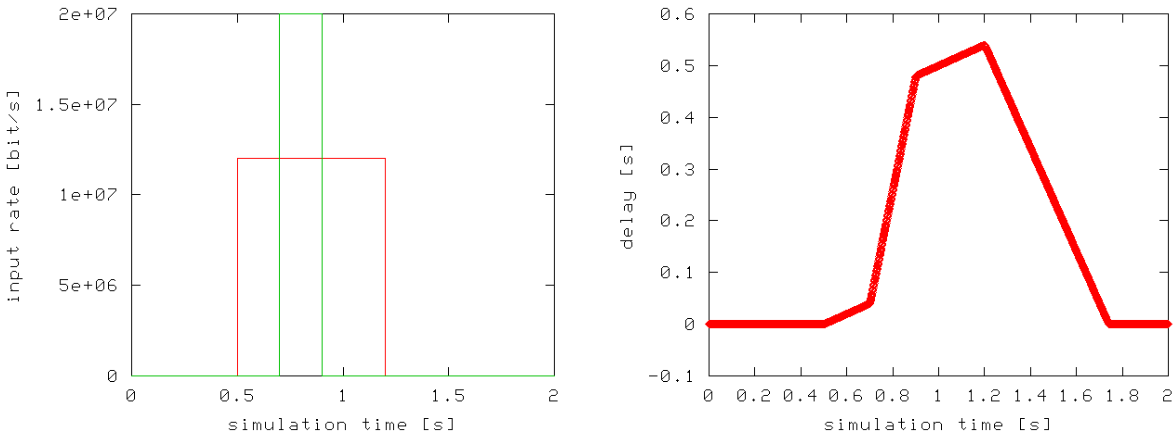


Figure 17 – Two input signals to 1 server



a) Input signals

b) Delay signal

Figure 18 – One server, multiple input signals

2.2.2 Scalability analysis

The time it takes for an RTC-FSIM simulation to complete is *independent* of the simulated link speeds and the amount of traffic on the links. This provides a for a great scalability advantage over traditional packet based simulations.

For RTC-FSIM, the run time depends

a) on the simulated time

b) the size of the network

c) the sampling interval for the differential equation solver

d) the efficiency of the differential equation solver

Concerning a) the run time grows linearly with the simulated time.

Concerning b): in the RTC-FSIM modelling approach, each AS is represented by a single server station. The maximum number of ASs along a path is in the order of 10s and therefore the simulated networks are rather small. There is no ripple effect that significantly reduces the scalability of event based fluid simulators. As each server station along the path has to solve the differential equations that describe the transformation of the input signals into the output signals. The time it takes to solve the equation system does not depend on the state of the other servers in the path. Therefore, the simulation run time grows linearly with the length of the path.

Concerning c) and d): at the moment we use a Runge Kutta 45 differential equation solver with a variable step size. The maximum step size is limited by the sampling interval *h*. Due to the variable step size the simulation run time does not solely depend on *h* but also on the nature of the input signals. We observed simulations where a h of $h_0/10$ had about the same execution time as a simulation with $h = h_0$. On the other hand, the same simulation with $h = h_0/10$ sometime needed up to 20 times as long as with $h = h_0$. If the value of h is very small (< 10e-3) numerical instabilities were observed that led to an explosion of the simulation time. In discussion with experts in the field of numerical mathematics it became obvious that the issue would have to be studied in great detail. This is outside the scope of this project and is therefore left for further study.

In any case, this issue is not of practical relevance as it has already been shown in [Ber02] that such small sampling intervals are not necessary because the improved accuracy of the results is negligible. The sampling interval should anyway be set in relation to the IPFIX measurement interval. We found that a *h* = 0.1 is a good choice.

As a practical example, the big topology shown in Figure 4 was simulated using RTC-FSIM. For a simulated time of 50 s the pure simulation time (excluding the DB queries and the Matlab startup time) was approx. 0.5s.

2.3 Inter-IP

This tool is based on an analytical queue model initially developed for planning purposes. The reason to use an analytical model was the need for using it in an automatic procedure for dimensioning a network supporting differentiated quality of services with the DiffServ approach [DSC].

The model was derived from the M/G/1/Priority queue model [Kle76] with some modifications that take into account the behaviour of the GPS class scheduling algorithms. Further modifications are brought for supporting an evaluation of performance as seen by the end-user. For example for data transfer application the packet delay is not so an intuitive parameter, the user demands a granted bandwidth, so this parameter or the equivalent time to download a document of a defined volume can be used to describe the network performance. For interactive application as, for example, VoIP the packet delay is used because in this case it is really the relevant parameter.

The use of a closed formula to compute the delay allows for a near real-time evaluation. Moreover no scalability problem exists. In fact the evaluation time is linearly proportional to the number of links crossed by the flow of which we are evaluating the performance. The delay is computed separately for each link crossed and it depends only on the link load, the traffic mix and the scheduling configuration.

The tool provides for each of the selected traffic relation (a couple of source and destination domains) the average end-to-end delays for all the traffic classes with traffic crossing the path from the source to the destination. The tool can't provide any information about the confidence level of the delays. This directly derives from the fact that this tool is not exactly a simulator but an estimator: a unique sample (the computed value) is available and there is no possibility at all to apply the common methods to derive a confidence interval.

Thus, the provided values have to be considered only an indication about the expected values: the model has been developed favouring simplicity over precision because in a planning task such precision is not a critical issue.

This suggests using the tool in those situations where a rapid analysis has to be performed. For example an operator often has to perform analysis on complex network scenarios with a large numbers of parameters and the effect of combining all the parameters could not be easy to understand "a priori". In such a situation our tool can be effectively used for skimming off the most interesting cases, which can result in saving a considerable amount of time and resources with respect to the case when a traditional simulator should analyse all the scenarios.

Another typical application of our tool is the "what-if" analysis and in general all the situations when a quick comparison between different alternatives is requested.

The tool uses the BGP topology data provided by InterRoute and the traffic measurements provided by the IPFIX collectors. Additional inputs are statistical characteristics of traffic and schedulers configuration.

2.3.1 Validation of implementation

In this paragraph we describe a simulation study where a simple DiffServ network with realistic traffic load has been modelled. The purpose of this study is the validation of the model presented in the previous paragraph. The use of a realistic traffic load is important especially for taking into account the effect of TCP flow control, considering that the majority of data applications are based on it. The reference scenario is depicted in Figure 19.

On the right side there are some workstations modelling users that requests the service to the network, on the left there are the servers. This situation is valid for client-server applications as ftp, http and video-conferencing. There are also some interactive applications that are simulated as ToIP and VoIP. For these applications the terminals (represented as phones) are located on both sides.

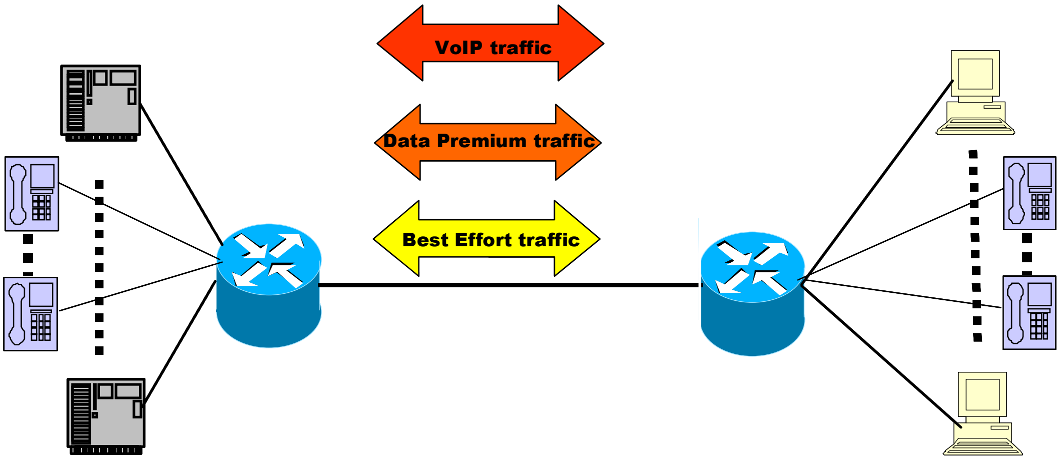


Figure 19 – The simulation scenario

For the simulation study the OPNET simulator program [OPNET] is used and the terminals at the sides are modelled using the “switched LAN” equivalent node present in the OPNET model library. This permits to have fewer interfaces on the routers. The links that connect the routers to the LAN nodes are modelled as Gigabit Ethernet links, and then the link connecting the two routers is intended to be a bottleneck. In the simulations a T1 (1544 Kb/s) has been used. The OPNET simulator has allowed us to have a very detailed model for the application traffic because each level of protocol is modelled in detail. Therefore it can be considered a valid alternative to real world measurements.

The WFQ scheduler is operating only at the T1 router interfaces. No low-latency queue has been used.

The traffic offered by applications simulated has the characteristics described below:

- Ftp traffic: Poisson session (one file request per session) arrival process, only “get” modelled (file transferred only in the directions from server side to the client side), normally distributed file size with mean equals to 50 KBytes;
- VoIP traffic: Poisson call arrival process, exponentially distributed call duration (mean equals to three minutes), G.711 codec with two frames for packet;

Three classes of traffic are used. One is reserved to the VoIP traffic, the other two are used for data traffic. Two different data traffic classes have been defined: the first is identified as Data Premium class, the other as Best Effort class.

The weight configuration used assigns 70% of the link capacity to the VoIP traffic, 20% of the link capacity to Data Premium traffic and the remaining 10% to the Best Effort traffic.

Two different scenarios have been studied. In the first scenario the total load of the bottleneck link varies but the VoIP traffic offers a fixed quota of traffic equal to 15% of the bottleneck link capacity. Moreover the others two classes offer the same quotas of traffic. The results are showed in the Figure 20, Figure 21 and Figure 22.

In the figures are compared the measures collected by OPNET with the results for the traditional M/G/1/Priority model and for the new “GPS model”. It can be noticed that the GPS model is more conservative than the M/G/1 priority model (this is important especially for the planning purposes) and the results are quite similar to those measured. The difference between the two models is less evident for the Best Effort class, and this can be explained with the weight configuration and the traffic mix used, because this class is very disadvantaged and it is in a situation similar to the last class in an M/G/1/Priority model.

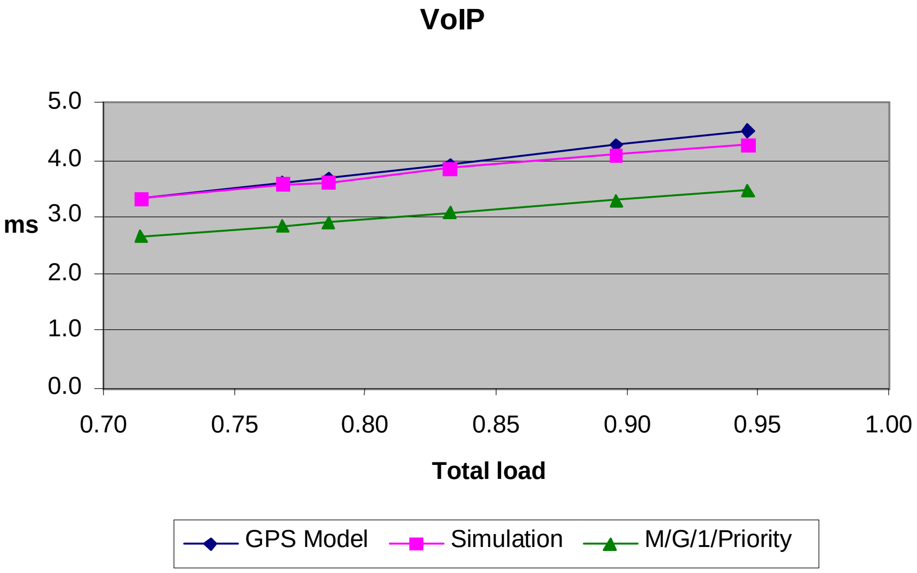


Figure 20 – VoIP packets mean delay vs. total load

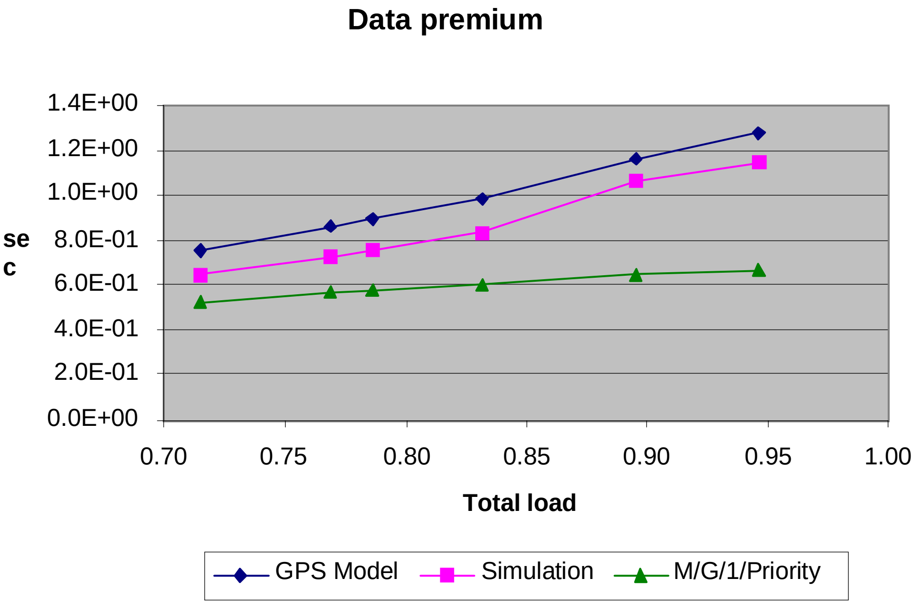


Figure 21 – Mean download time for Data Premium class vs. total load

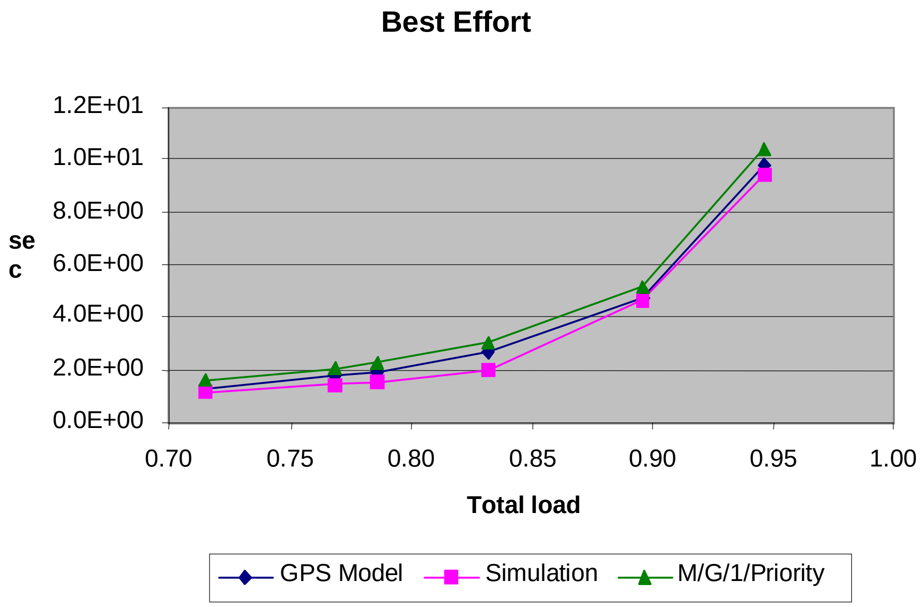


Figure 22 – Mean download time for Best Effort class vs. total load

In the second scenario the total load of the bottleneck link is fixed and equal to the 80% of the link capacity. The traffic offered by the VoIP class is fixed too and equal to the 20% of the total load. In the different runs the ratio between the loads offered by the data classes varies between two limits representing the cases when only one class offers all the traffic. The results are showed in Figure 23 and Figure 24 (the results for VoIP class are not displayed because they are similar to the previous ones).

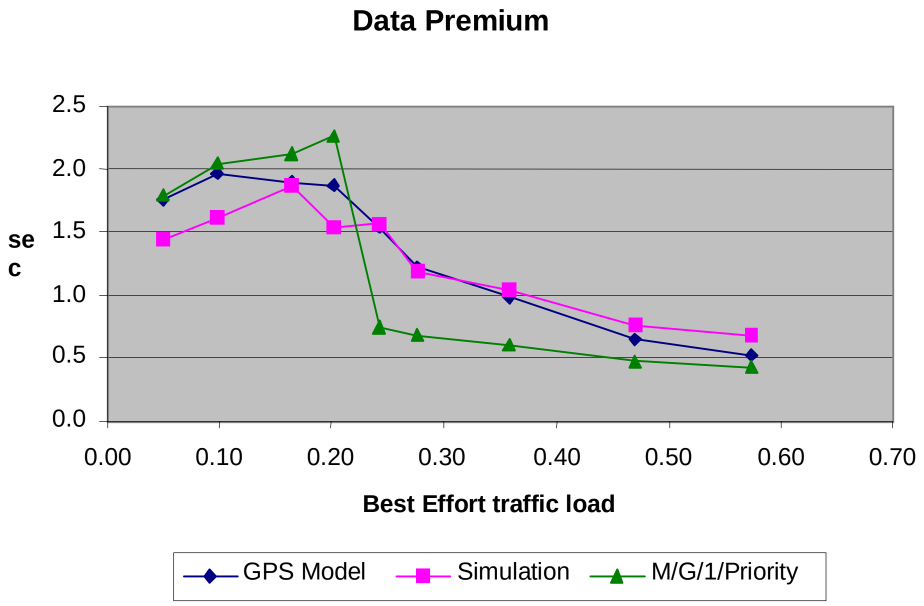


Figure 23 – Mean download time for Data Premium class vs. Best Effort offered traffic

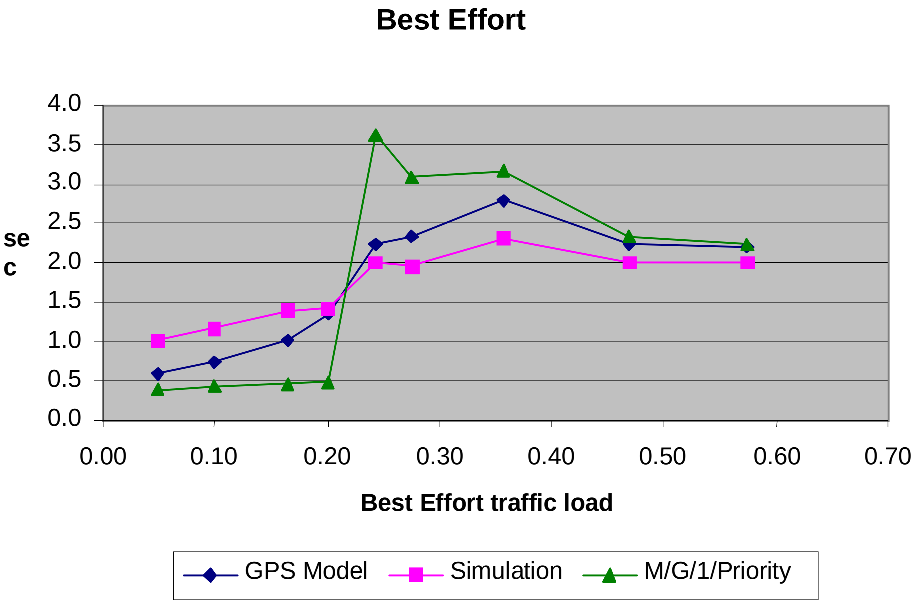


Figure 24 – Mean download time for Best Effort class vs. Best Effort offered traffic

For this scenario the advantage of the GPS model vs. the M/G/1/Priority model is particularly evident near the transition zone, the point where a class crosses from the zone where it is advantaged with respect to the other classes to the zone where it is disadvantaged (this zone is approximately the point where the load is equal to the capacity quota assigned by the scheduler).

2.3.2 Scalability analysis

Since the tool evaluates end-to-end delays by composing the results of a closed formula, it is computationally very fast. The evaluation time depends linearly on the number of inter-domain connections the traffic traverses in the end-to-end path, each of them being processed by the software in less than 1ms. This means that if we considered only the Inter-IP tool response time, we can safely declare that in less than 1 second the results would be ready for analysis.

This time is negligible compared to the overhead the InterMon system introduces in terms of delay when it processes the simulation request: the distributed nature of the architecture imposes connections between the modules and also with the database, with operational times that are bandwidth dependent and can greatly vary due to network conditions.

2.4 TSSIM

In this section the Time Series Simulator (TSSIM) is evaluated. TSSIM is a simulator architecture that handles network state descriptors in an aggregate, time series form. The simulator can be used to investigate load and packet drop rates at any point in the network, or delays and jitter along a specific path. As the state descriptors can come from real world measurements done by other tools of the Intermon toolkit, the simulator can be of high value when adding new traffic to the network, or rerouting traffic in certain parts of the network.

Since the simulator is based on aggregate load information, the simulated QoS values represent an average for the traffic as a whole. The more detailed the load information is the more accurate the QoS values will be with an increasing run time.

Similarly to the evaluation of RTC-FSIM, the evaluation of the TSSIM tool has been performed in three steps. Firstly, the correctness of the implementation has been evaluated on a simple scenario, where the correctness of the simulation result can be easily checked. Secondly, the simulator has been assessed by comparing the simulation results for a scenario to the measurement results on a testbed of the same scenario. Finally, the scalability of the simulator has been evaluated.

2.4.1 Correctness of implementation

To evaluate the correctness of the implementation, we have used a simple scenario that is composed of only one network domain (AS) with three border routers modeled by different router models. We use different models for the routers so that the correctness of all the available models can be evaluated in one scenario (Figure 25).

The input traffic of the domain enters the domain through a Null Model, which only adds a constant (10 ms) delay, and distributes the traffic according to the input distribution file. The distribution is set such that 30% of all incoming traffic is forwarded to the router modeled by the Leaky Bucket Model with a queue length of 30 bytes. The remaining 70% is forwarded to the router of Class 4 Leaky Bucket Model with a queue length of 20 bytes. Both the Leaky Bucket and the Class 4 Leaky Bucket models start with an empty queue at the beginning of the simulation, and both have an output rate of 5 bytes/sec. The simulations are run for 20 seconds.

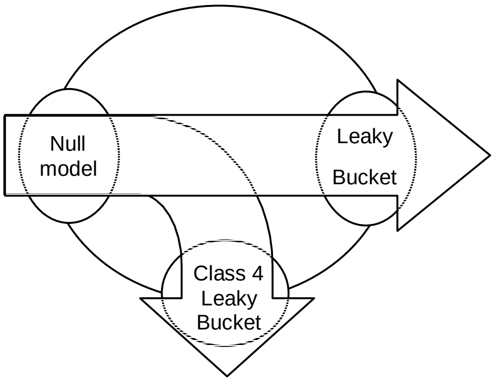


Figure 25 – TSSIM evaluation scenario

In the first setup, the input traffic of the domain is constant 10 bytes/sec, which produces a 3-bytes/sec load for the Leaky Bucket Model and a 7-bytes/sec load for the Class 4 Leaky Bucket Model. This way the former model is not overloaded, while the second one has 2 bytes in excess to its maximum output in each second. Now let us see the results produced by the simulator (Figure 26).

As it can be seen in the figures the Null Model simply copies its input to its output, and does not drop packets. It introduces a constant delay, and no jitter. The Leaky Bucket model is not overloaded; therefore it outputs all the traffic it receives, without dropping any packets. It introduces an average queuing delay of 0.4 seconds with 0 seconds of jitter. Finally, the Class 4 Leaky Bucket model outputs 5 bytes/sec. It starts dropping packets when its buffer becomes full at the tenth second. With the increase of the queue size, the introduced delay is increased until the maximum is reached. It calculates jitter as the difference between the delay in the previous time slot and the current time slot, which gives a constant value of 0.4 seconds until the buffer gets full, then this value is changed to 0. All operations are as expected.

In the second setup, the input traffic of the domain is a burst of 140 bytes with a peak at the third time slot. The peak overloads both the Leaky Bucket Model and the Class 4 Leaky Bucket Model (Figure 27).

As it can be seen in the figures, the Null Model outputs the burst to its neighbors. The Leaky Bucket receives a smaller portion of the traffic, and it has a larger buffer, therefore it only drops two bytes, while the Class 4 Leaky Bucket drops 63 bytes all together. The delay values are in accordance with the mathematical models used in each model, calculating delay as a function of the queue length in the previous time slot. The jitter values may seem a bit random, but this behavior is understandable, as jitter is calculated as the difference between the delay in the previous time slot and the current time slot. So, all operations function as expected.

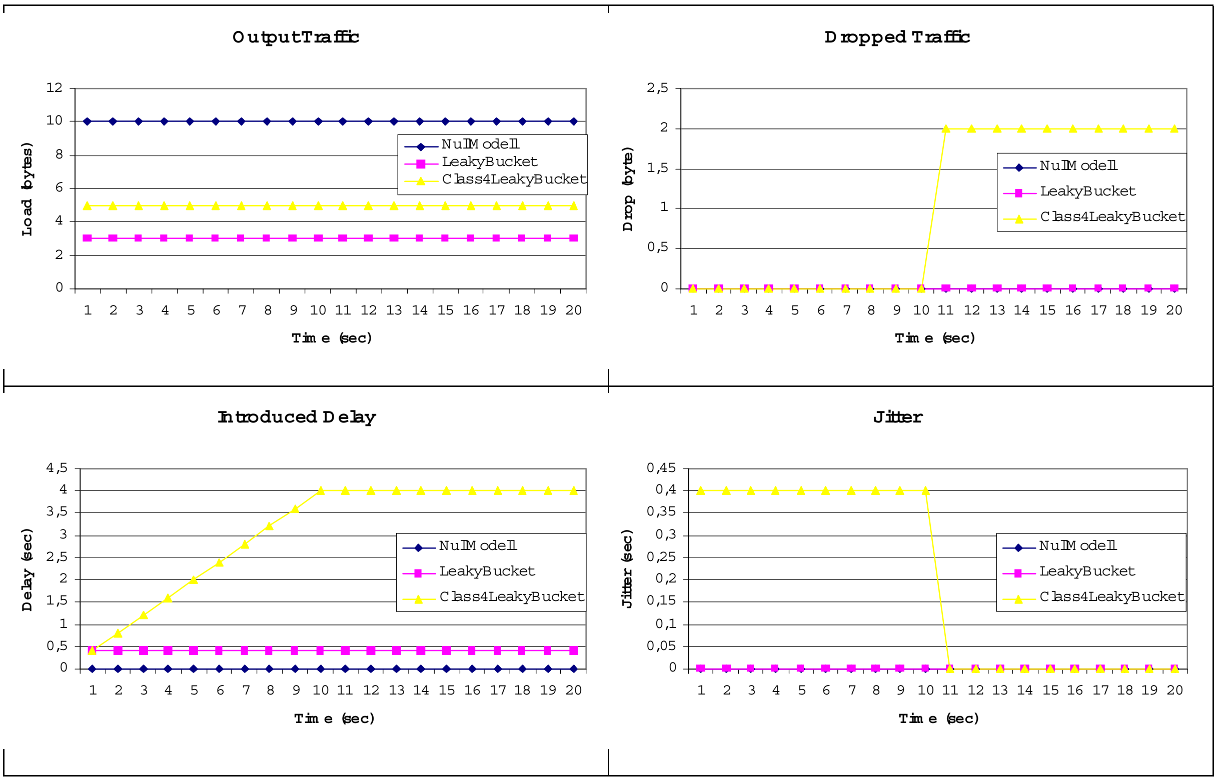


Figure 26 – Output of the TSSIM router models for constant bit rate input

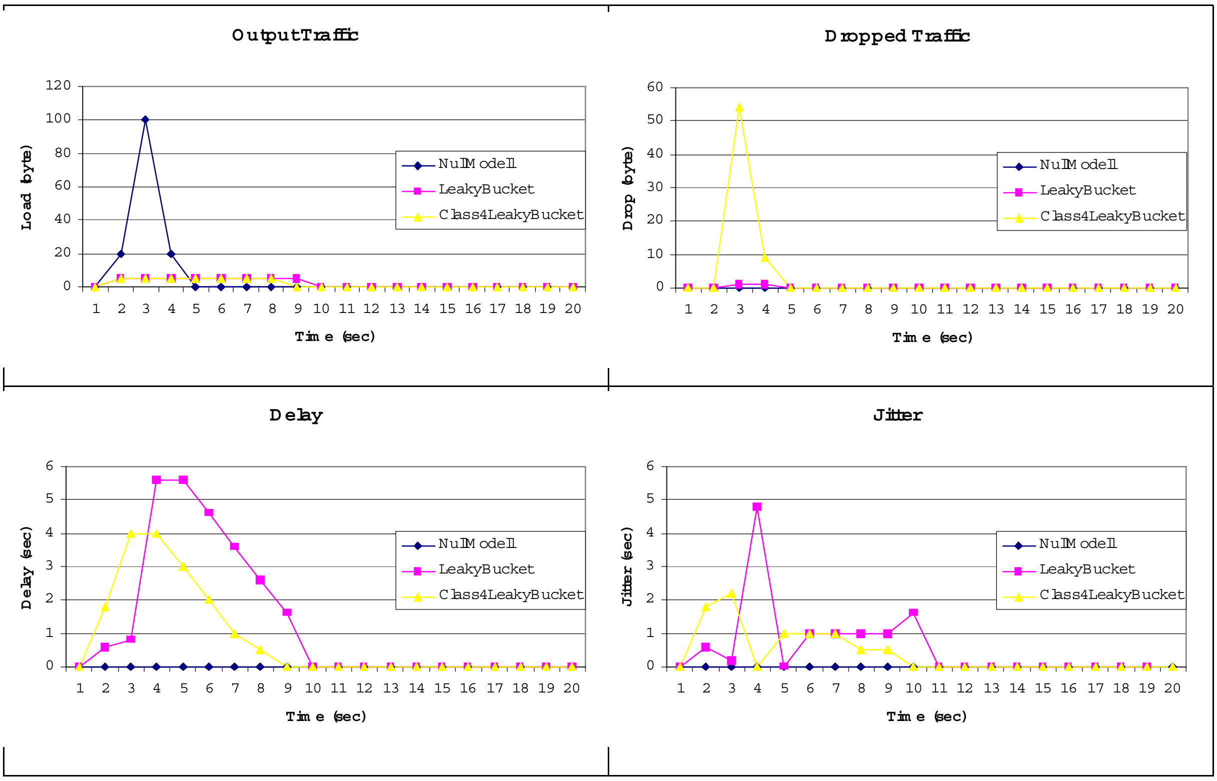


Figure 27 – Output of the TSSIM router models for a single impulse input

2.4.2 Scalability test

The main advantage of this simulator architecture is that the simulation time does not depend on the amount of traffic in the simulated network; therefore the scalability problems of packet level simulators do not arise with the time series simulator. During the execution of a simulation, the input time series are processed by each border router. Hence, the simulation runtime depends on the length of the time series, the number of border routers and the complexity of the border router models. The simulator assumes that the length of data traces to be processed does not allow them to be kept in memory; therefore the speed of file handling also has a crucial effect on the runtime of the simulator.

In Figure 28, we can see the runtimes of executing the small and the big topologies with different output models (the time required for transmitting the simulation requests and replies, and time for database access is not included). The complexity of the models is clearly reflected in the results, but the difference between the runtime of different models is not significant.

The runtime values of the time series simulator are much lower than the runtime values of a traditional packet based simulator, NS-2 (34 seconds for the small topology, 386 for the big one), so we can say the simulator produces acceptable results with good performance for analysing large networks.

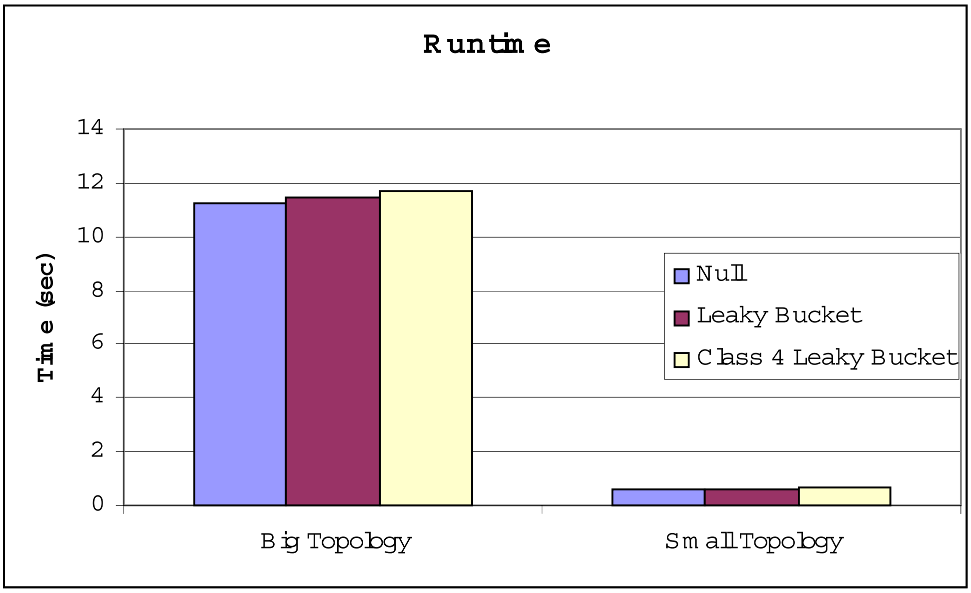


Figure 28 – Runtime for the small and big topologies on a PIII 800MHz machine

3 Comparison of the Modeling and Simulation Approaches

3.1 Results

3.1.1 NS2 Hybrid

3.1.1.1 Simulation Model

The simulation scenario for NS2-Hybrid was derived from the BGP topology and the IPFIX traces stored in the central Intermon database. However, the topology had to be changed to obtain good simulation results. The BGP topology from the database (see Figure 2) models the physical topology (see Figure 1) in a way that hides the bottleneck link between R2 and LX4 inside AS 3000, which runs contrary to the basic assumption made by the NS2-Hybrid simulator that bottleneck links are modelled *between* ASs. By explicitly modelling the potential bottlenecks (see AAA) the scenario could be suitably modelled, however.

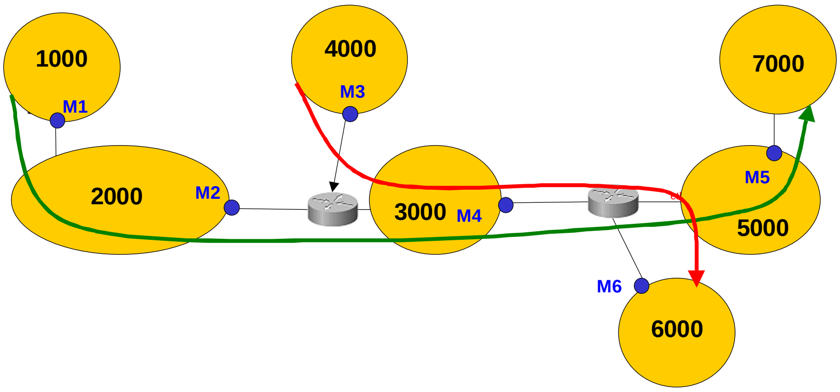


Figure 29 – Adapted testbed BGP topology

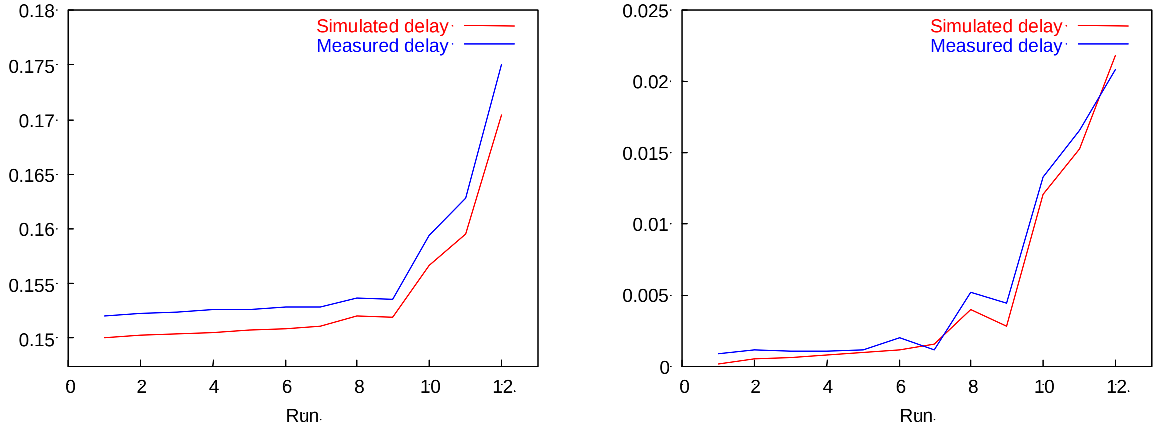
Since the main focus of evaluation was on our analytical extensions to the original NS2 the scenario was modelled almost purely analytically. The only packet-based element was a reference stream from AS 1000 to AS 7000. All elements of the testbed were modelled using a multi-domain model. VoIP and video traffic was modelled using traffic generator models, which were parameterized with the IPFIX traces from the database.

3.1.1.2 Results

Simulating the testbed scenario using the NS2 Hybrid simulator yielded good results. Both, simulated delay and the simulated standard deviation of delay (and thus jitter) were close to the measurements.

However, simulated delay was always lower than real delay, by an almost constant value (see Figure 30 a)). This difference can also be observed in the RTC-FSIM results described below and can be explained with several factors. First, the link and processing delays were ignored in the simulation since no suitable data was available. It would be easy to adapt the scenario accordingly if the data were available, however. Second, the 5-second intervals used in the IPFIX measurements have a “smoothing effect”, which filters out the effects of short-term traffic bursts and leads to an underestimation of the real burstiness of traffic, and thus an underestimation of delay. This interpretation is supported by the slightly growing difference with increasing load between measured and simulated delays. Burstiness grows as more video flows are added to the scenario.

Also the small difference in standard deviation of delay that can be observed in runs 8 and 9 of Figure 30 b) can be explained by underestimated burstiness. However, the difference is too small to be sure. Further investigation would be required.



a) Mean delay (in sec) **b) Standard deviation of delay (in sec)**

Figure 30 – Comparison of testbed measurements with NS2 Hybrid simulation

The mean packet size of the streams proved to be a critical factor for the simulation of queuing delay. We know that the greater the mean packet size, the greater the queuing delay. Unfortunately, the IPFIX-trace-based traffic model we used does not support packet size estimation. We manually configured a rough estimate of the mean packet size, which resulted in acceptable accuracy of simulation. However, an automated solution would be required to make the simulator fully operational in the integrated Intermon architecture. Adequate tools are available, as discussed in the following section, but they still have to be integrated with the NS2-Hybrid simulator.

3.1.2 RTC-FSIM

3.1.2.1 Simulation model

The simulation model that corresponds to the measurement topology (see Figure 1) is depicted in Figure 31. Each of the routers R1, R2, and R3 is modelled by an RTC-FSIM server. The data of measurement point M5 is needed to derive what portion of M4 traffic flows to LX 5 and LX7, respectively. The emulated delay of 50ms is not integrated into the model; instead a delay of 151ms (the minimum 1-way delay from LX1 to LX7) is added to the resulting delay values. The simulation is run for 500 s, the sampling interval for the differential equation solver is 0.1.

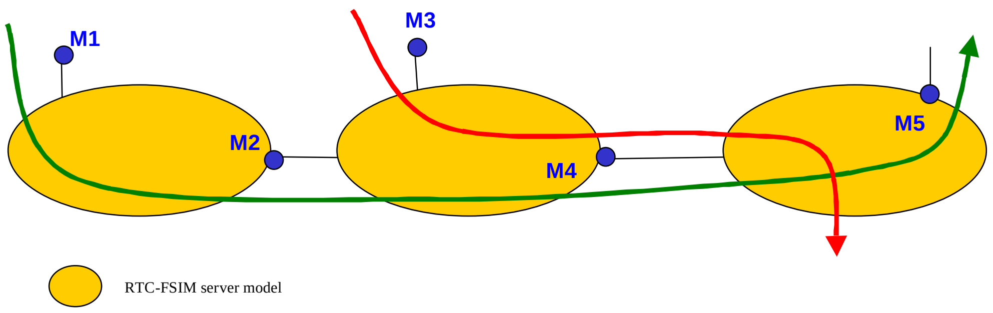
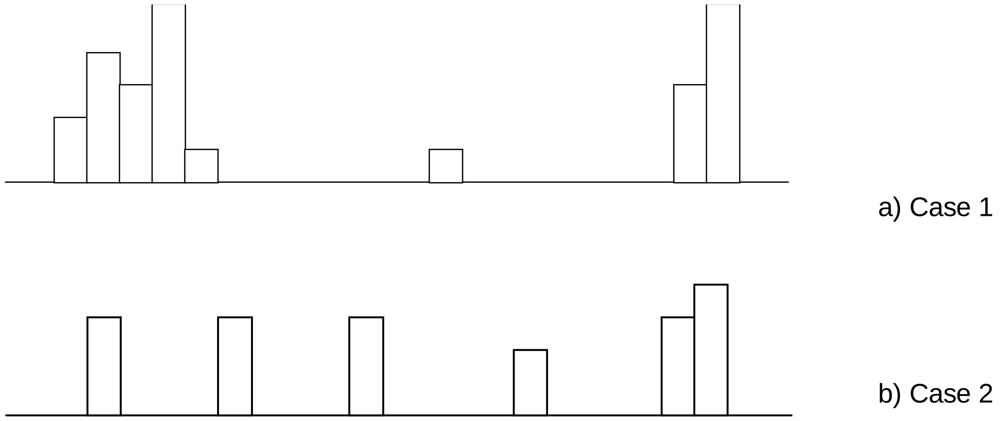


Figure 31 – Simulation model

The simulation tool should be useable for QoS evaluation of e.g. audio flows. A typical scenario of interest could be: "if the load increases by 50%, will the maximum delay stay below a certain threshold?"


a) Case 1

b) Case 2

Obviously, when new audio traffic is added in the two scenarios sketched above, the probability that a delay threshold will be exceed is higher for a bursty load (case 1) than for a smoother load (case 2). It is therefore necessary that the simulation model reflects the

a) length of the packets in a burst

b) duration of the bursts.

To reproduce a similar load for the simulation it is thus necessary to capture the variance (a) and the autocorrelation function (b) of the real traffic. To accomplish this in the evaluation trial we implemented a modified traffic meter that is able to measure rates over small time intervals in addition to the longer IPFIX intervals (e.g. 5 s). For the variance and autocorrelation function estimation we use a 40ms interval as this is the time scale of the video background load. In a real inter-domain scenario, passive measurement tools [DAG, SCAMPI] can be used. The importance of accurate variance estimation for traffic engineering is discussed in [DS03].

3.1.2.2 Results

In this section results are shown for a selection of load scenarios from low to very high load. In Figure 32, each subfigure shows the measured delay of each packet (red colour) as well as the end-to-end delay signal that is produced by the simulator (green colour). On the x-axis, the delay signal from the simulator has been converted from unit time to unit packet number such that it can be plotted along the measurement results. For the same reason, the minimum 1-way delay (151 ms) from LX1 to LX7 has been added to the simulation output.

These figures are used to make a first visual judgement on the similarity between measurement and simulation results. Generally, it can be seen that the simulation output does not reproduce the small delay variations of the voice flows. This is due to several reasons.

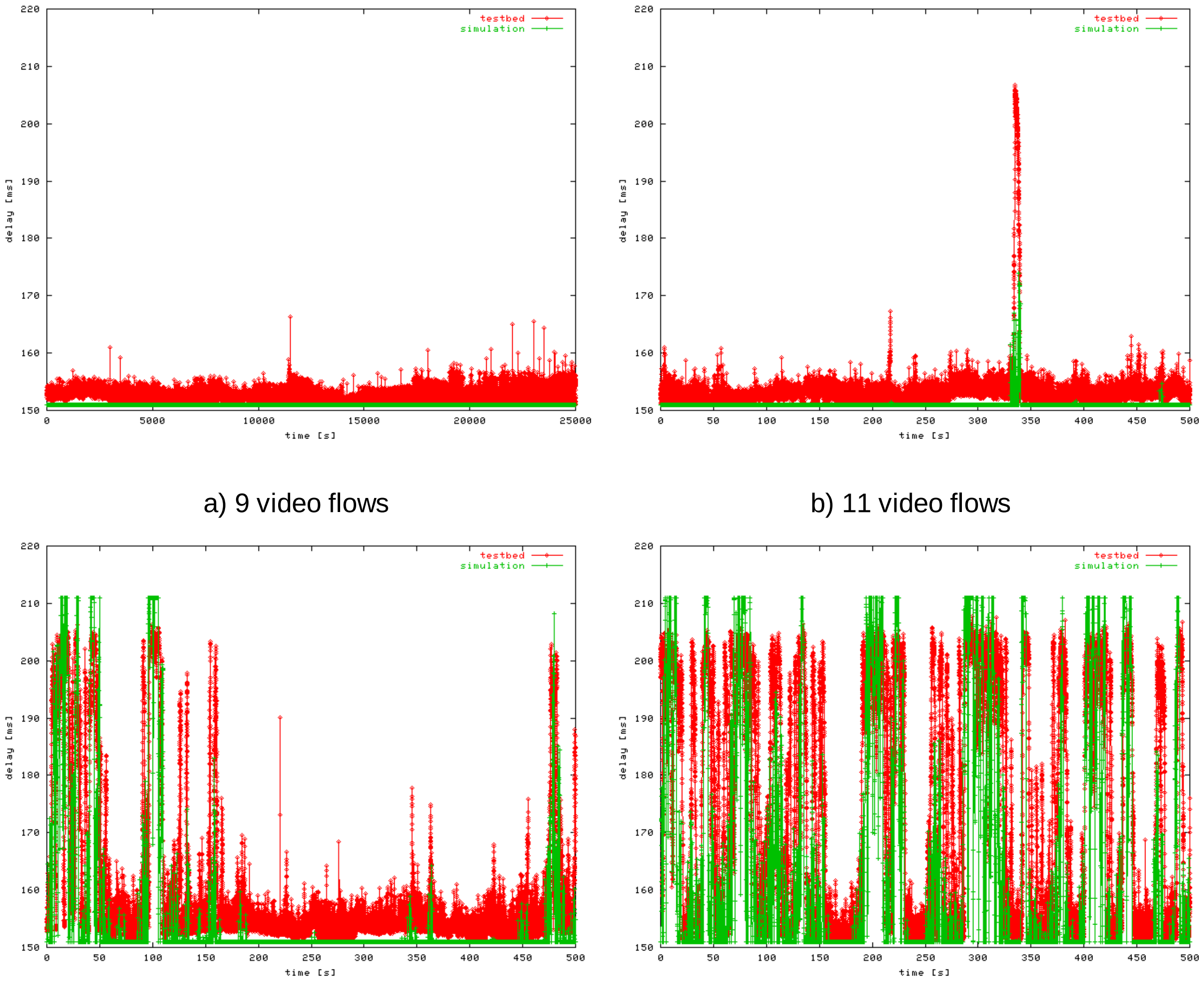
1. Real-world effects

   The RTC-FSIM approach models routers as servers with a fixed service rate and a limited buffer size. This is of course a high-level abstraction that does not capture several properties (e.g. processing delays, influence of the operating system) that exist with real-world routers.

2. Measurement of header data

   The IPFIX measurements are performed on the IP layer. The link layer encapsulation overhead (Ethernet framing) is not accounted for in the IPFIX data.

   For the voice traffic, 160 Byte of payload are produced every 20 ms. This results in 202 Bytes every 20 ms on the wire (160 Byte payload, 8 Byte UDP, 20 Byte IP, 14 Byte Ethernet). Thus, the rate of 1 voice flow equals 80800 bit/s.



a) 9 video flows

b) 11 video flows

c) 13 video flows

d) 15 video flows

Figure 32 – Measurement and simulation results

As the IPFIX meter measures at the IP layer, it misses the 14 Byte Ethernet header. Therefore, 188 Byte (160 payload, 8 UDP, 20 IP) are seen every 20 ms which results in a rate of 75200 bit/s for each voice flow. For the whole aggregate of 10 voice flows the rate reported rate is 752000 instead of 808000, a difference of 56000 bit/s.

Our MPEG 4 video traffic has an average packet payload size of 1169 Bytes. Again, on the wire this creates 1211 Bytes, the IPFIX meter reports only 1197 Bytes. Due to this rather large packet size, the exclusion of the Ethernet header in the IPFIX measurement has only a small effect. On average, the IPFIX rate if approximately 7000 bit/s below the wire rate for each video flow.

In a scenario with 10 video flows the total measured rate is 56000 + 70000 = 126000 bit/s smaller than the wire rate. Given a maximum bandwidth of 10 Mbit/s, the measurement error is in the range of 1.26 %

3. Multiplexing

In a real router, the multiplexing of input streams is done on a packet-by-packet basis. If a voice packet arrives at the router just after a video packet, the voice packet can not be transmitted before the full video packet is sent. The voice packet is thus delayed by the transmission time of the video packet.

In the RTC-FSIM fluid flow model, the multiplexing of different inputs translates into the summation of the input signals. This summation can be interpreted as a bit-by-bit multiplexing as the packet effects vanish in the fluid model. This leads to somewhat smaller delays in the simulation.

As an example, consider the arrival of a 1500 Byte video packet just before a 200 Byte voice packet at router R2 which has an outgoing link capacity of 10 Mbit/s. The transmission delay of the video packet equals 1.2 ms.

4. Delay measurement

   In the testbed, the delay is measured as the difference in time between the arrival of the last bit of a packet at the traffic source minus the time when the first bit of that packet was sent by the source.

   In the fluid model, no such thing as a packet delay exists. The delay signal can be interpreted as the delay of each bit of traffic. The average delay difference that occurs due to this different kind of measurement technique equals the transmission delay of half a voice packet. On a 10 Mbit/s link, this error equals 0.08 ms.

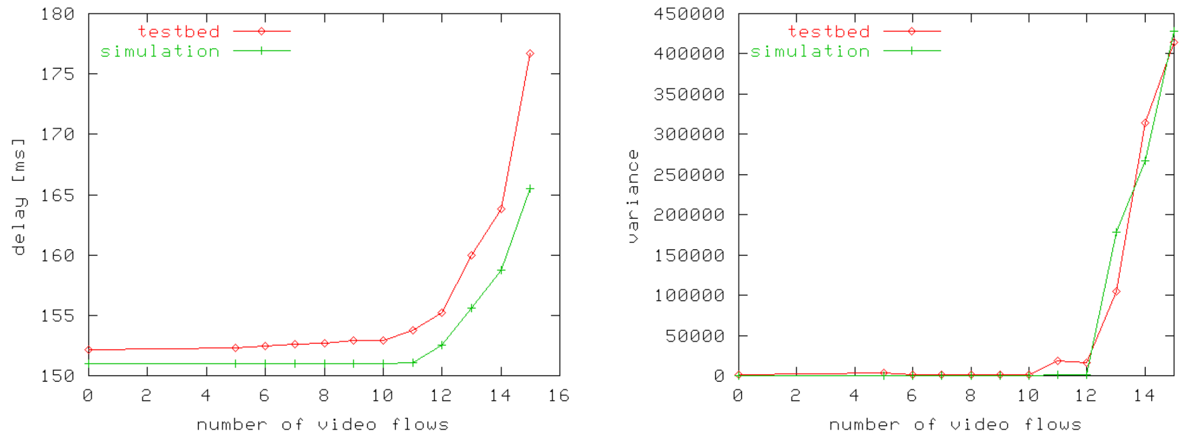
5. Transmission delay

   In the RTC-FSIM fluid model, delay is always due to queueing delay. Processing delays as well as transmission delays are not part of the model. In the testbed, the voice flows traverse 3 10 Mbit/s and 4 100 Mbit/s links. The sum of transmission delays for a 202 Byte voice packet is approximately 0.5 ms.

All these factors contribute to the effect that the simulation delay is lower than the measured delay. Note that the impact of most of the factors decreases as the link speed increases. On inter-domain links with high physical link speeds, the impact is much smaller than on the 10 Mbit/s bottleneck link in the testbed.

A second and more important observation from Figure 32, is that if the measured delays go up higher, the simulator produces similar delay values, too. It can be seen that the delay peaks from the testbed also exist in the simulation results. As a general rule, traffic peaks that are much shorter than the reporting interval of the meter will not have a strong influence on the simulation results as opposed to traffic bursts, that persist longer.

To quantify our first observation, the average queueing delay and its variance are shown for the testbed and simulation results in Figure 33. The spectrum of load scenarios goes from 0 to 15 video flows. As mentioned above, the simulation delays are permanently a bit lower than in the testbed and therefore the green average delay curve is somewhat lower than the red curve. Nevertheless, the main delay characteristics are reproduced well by the simulator.



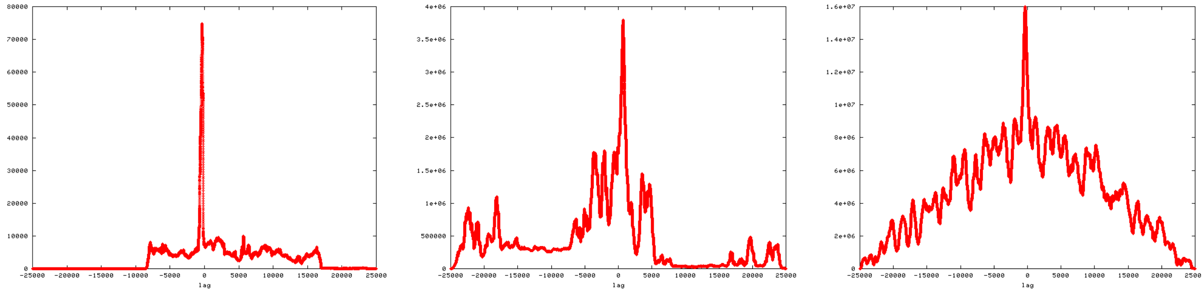
a) Average delays

b) Variance of delays

Figure 33 – Comparison of average and variance of delay

Also, the variance of the delay signal is very similar to the one of the testbed measurements. This is another important issue as it shows that the simulator also follows that short time fluctuations of the delay.

To further evaluate the simulation result we compute the cross-correlation of the delay time-series for the measurement and simulation results. The motivation is that we expect the two time-series to exhibit strong similarities at a small time lag. If a traffic peak occurs at the beginning of an IPFIX measurement interval, this peak will be reflected in the data reported at the end of the interval. Therefore it can be expected that the simulation delay time series is phase-shifted compared to the measured delay by about 1 measurement interval. This is where the cross-correlation function should have its high peak.



a) 11 flows

b) 13 flows

c) 15 flows

Figure 34 – Cross-correlation functions

The cross-correlation functions for load scenarios 11, 13, and 15 flows, respectively, are shown in Figure 34. The graphs clearly show that the highest cross-correlation is at about lag 0. The results are as expected.

The essential outcome of the evaluation of RTC-FSIM is that the simulation results reflect the main characteristics of the measured delay values. We have shown that the simulation delays have a similar (though somewhat smaller) average delay, a similar delay variance, and an evolution that is only slightly phase-shifted compared to the measurements. Considering the high level of abstraction of the RTC-FSIM modelling approach these results are very encouraging. Given these strong similarities with the measured values the simulator can indeed be used to evaluate "what-if" scenarios to e.g. forecast the effect of additional traffic on a link.

3.1.3 Inter-IP

The Inter-IP tool has been used to evaluate the end-to-end performances of the VoIP flows in the scenario described in 1.3; the chart in Figure 35 is a graphical summary of the results. Each bullet represents the average delay (y-axis) of the VoIP traffic aggregate for a certain run of the test (x-axis). The outputs of the estimator are close enough to the measured values in order to give a good indication about the user perception of the Quality of Service in this particular scenario.

What we can conclude from this analysis is that the model behaviour is quite good also when the arrival process of the traffic is unlikely to be poissonian like in this case. The principal causes for this consideration are two:

the type of traffic involved: voice and video packets are generated regularly, only if we have a very large number of independent voice and video flows we can think that the total traffic can be approximated with a poissonian process;

in the real word in a border router the traffic coming from a large numbers of incoming links is split over a large numbers of outgoing links. This multiplexing-demultiplexing process produces a sequence of independent arrivals and a poissonian model in this case is a very good approximation.

In spite of these considerations the model has shown to be able to provide a quite good estimation of the QoS performance parameter that we have evaluated. One reason of this fact could consist in the modifications brought to the original M/G/1/Priority model to take into account the application level (in our model the arrival process observed is at this level and not at the packet level). At this level if many different users request some service to the network it is likely that the aggregated demand of the service can be approximated with a poissonian process.

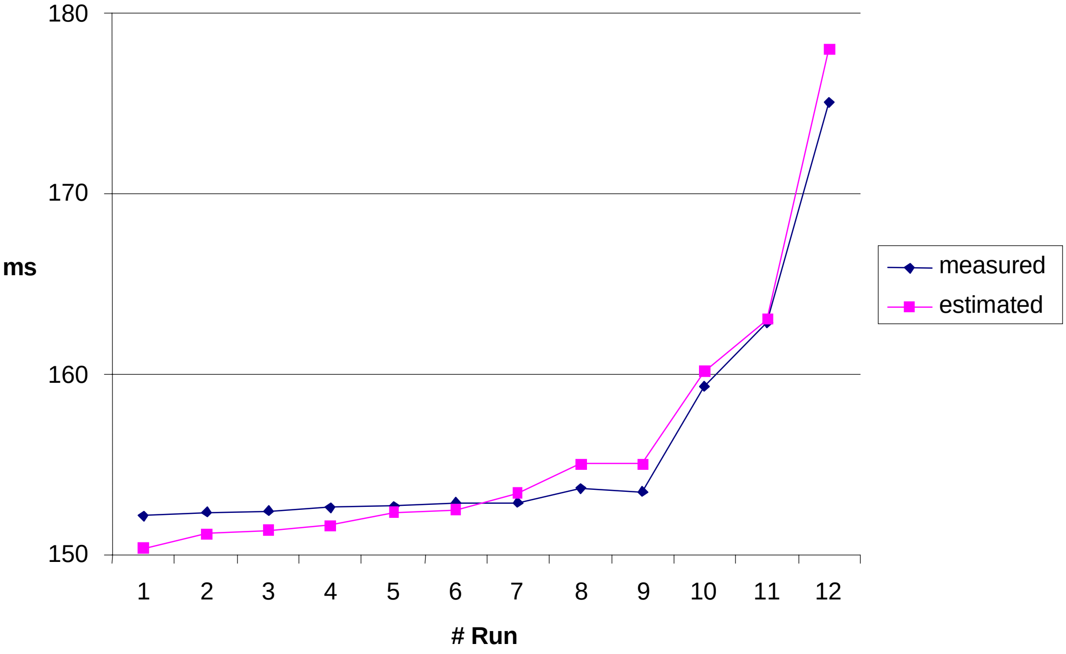


Figure 35 – Inter-IP end-to-end performances estimation

This analysis confirms that, in consideration of the negligible time to obtain a quite good estimation of the end-to-end delay (of the packet for a real time application like in this case or of the volume to be transferred for a data application), like anticipated in 2.3 this tool is useful for “a priori” analysis of what we can espect before to run a simulation also for those cases (see 3.2) where the use of this tool is considered not suitable. In certain cases this “a priori” analysis could support the decision on some parameters to use for configure the simulation, for example.

3.1.4 TSSIM

To validate the simulator, we use the results of measurements performed on a testbed at Salzburg Research. The investigated network has the following configuration (Figure 36).

The network has four domains with Cisco 3600 and 7204 routers. Links L1-L4 have 100Mbps capacity, while links L5-L7 have 10Mbps capacity. The queue management mechanism is Drop Tail, the queue size is 100 packets (150000 bytes). There is only one class of traffic (best effort). Two traffic types are introduced: constant bit rate VoIP traffic from L1 to L5 and varying bit rate video traffic from L3 to L6. Flows share the capacity of L4. For VoIP, 10 flows (G.711 like) that produce 160 bytes every 20 ms are used. For video, high quality MPEG4 trace files from the Mr. Bean film are made use of. The traffic produced by the flows at the sources and destinations are measured using IPFIX meters [4], M3-M6. Meters are configured to report measurement data every 5 seconds. Seven measurements are performed, 10 minutes each. The average of the measurements is used in the following figures.

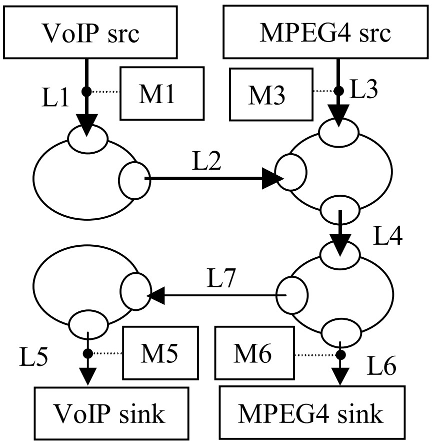


Figure 36 – Investigated network

Simulations shown in Figure 37 are performed using Null Model as input, and Class 4 Leaky Bucket as output model for all border routers with parameters set as in the measurement scenario. The following figures compare the output of the simulator at the destinations to the data measured in the real network.

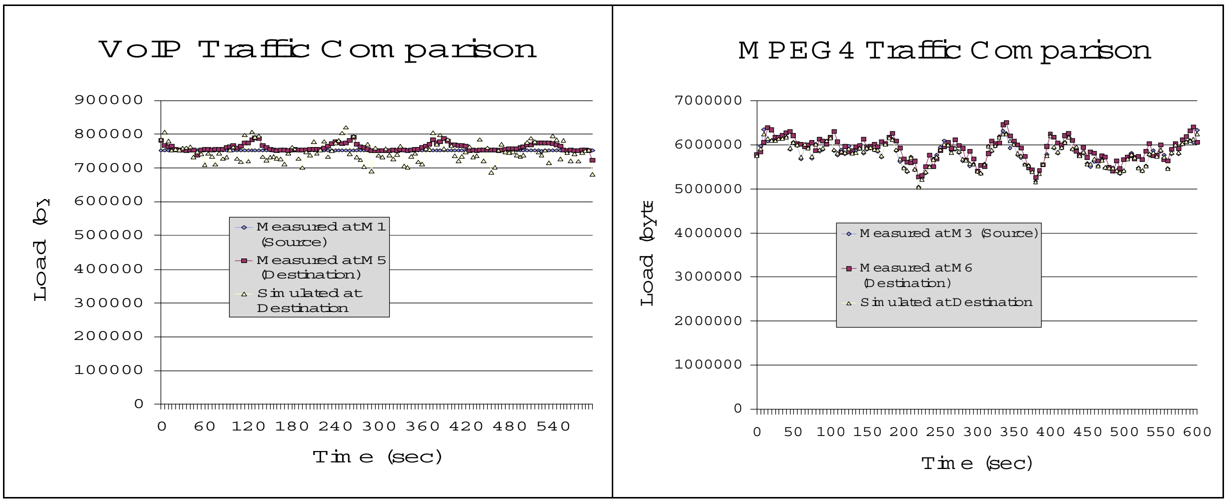


Figure 37 – Comparison of measured and simulated load results

As it can be seen in the figure, for the high volume MPEG4 streams, the output of the simulator almost exactly matches the data measured. For the VoIP traffic, however, it can be noticed that the simulated results show a much larger deviation than the measured data. This is caused by the fact that TSSIM does not handle flows, but traffic aggregates and therefore a correlation between the two flows is introduced. When comparing the two figures, it can be seen that the simulated VoIP traffic somewhat resembles the shape of the MPEG4 traffic. The effect of the VoIP streams is not noticeable on the MPEG4 streams due to the difference in their volume.

To compare the delay results produced by the simulator and measured on the testbed (Figure 38), we used different output models for the border routers and Null Model as input model. Looking at the figure, it can be seen that the Null Model is not suitable as an output model as it produces a constant delay regardless of the traffic of the domain. This model is intended for modeling the congestion free interiors of a domain, and it is primarily used as input model. However, the Class 4 Leaky Bucket and the Leaky Bucket models provide a good estimation of the real delay.

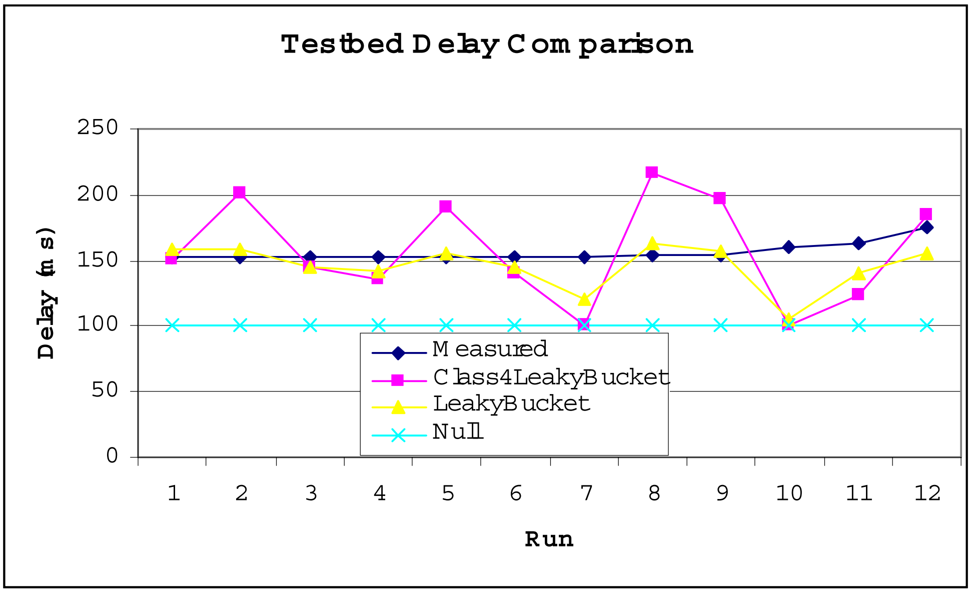


Figure 38 – Comparison of measured and simulated delay results

3.2 Implications

The above results as well as recent discussions between the developers of the modelling tools helped identifying the synergies and special pros/cons of the several modelling and simulation approaches. This chapter describes how the developed tools can be combined into a powerful interaction scenario. Suppose a user is interested in QoS interdomain analysis and uses the Intermon toolkit to model and simulate or calculate it. Because each of the modelling and simulation tools covers different situations we propose a three-step approach for future development:

Step 1: System Analysis (SA)

Step 2: Tool Selection (TS)

Step 3: Processing (P)

Why does this approach offer new potential for increasing the accuracy and performance of simulation/modelling?

1.: Real network measurements [SPRINT] have shown that the link transmission delay has to be modelled by a general distribution (there is a large percentage of long 1500 Byte packets) and that the load factor is about 10%. The */G/1 queuing model covers this real network lossless packet transmission (e.g. via a peering link) well. If the system's input can be suitably modelled by a poisson arrival process the Inter-IP modelling tool is a good choice. Otherwise, Inter-IP can be used to have a first evaluation of what we can expect and the simulators are used for detailed analysis (Step 2).

2.: Suppose we recognized in Step 1 that the analytical tool cannot be applied. Then, we start the analysis by simulating the system in the time interval „of interest". We begin the investigation by starting TSSim. This simulator produces the bandwidth demands for the different „what-if" scenarios which are of interest (route change, new link capacity, or new transit traffic). This simulation runs at a very high abstraction level with simple algorithms. Not all load situations in an operational system are of interest for the analysis. In 90% of the time nothing interesting will happen, but there are different load patterns per day, per link and when simulating with TSSim we will eventually get load values, which are high enough the be further investigated. These temporarily occurring high-load intervals are important for QoS analysis. Here, TSSim has reached its limitations (inaccurate predictions of loss and delay, due to its high-level nature) and we start with the fine-grained simulation using NS2-Hybrid or RTC-FSIM. These simulators give us information about the delay and loss values in the problematic time interval. After we have analyzed the problematic time frame using these simulators TSSIM is employed again to discover the next problematic time interval.

What is a problematic interval? This depends on the context of the analysis. For instance, if we are interested in loss probabilities higher than 10E-4, we can use the methodology of the effective bandwidth estimation based on the known link load (from TSSim) and additional information about the variance. These variance values can be either obtained via measurements (variance sampling) or by using a priori knowledge (e.g. known codecs of aggregated VoIP flows). Then, we can use the EBW-formulas to derive an appropriate threshold value. Figure 39 shows this scenario of modelling tool interaction.

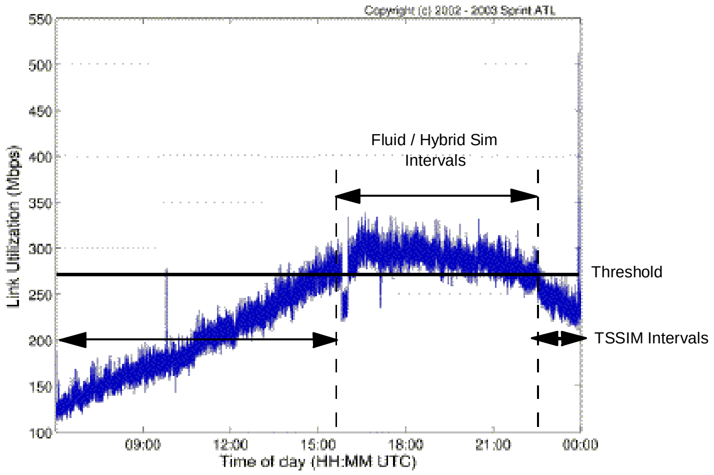


Figure 39 – Link utilization [SPRINT]

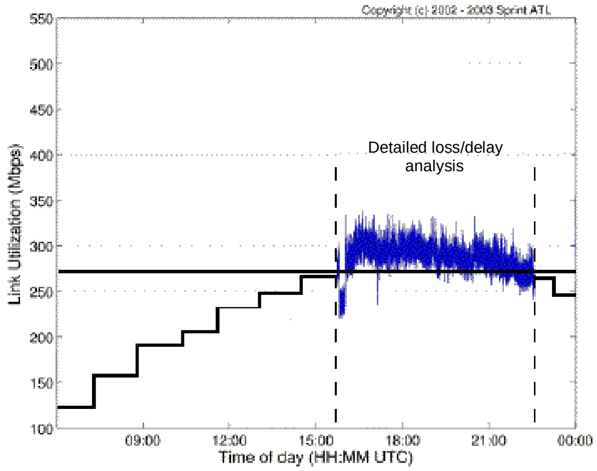


Figure 40 – Replacing TSSIM with RTC-FSIM or NS2-Hybrid for critical intervals

Figure 40 shows the link utilisation [1s] on a high speed OC-48 2.5 Gbit/s link. A mean value of 234 Mbit/s is about 10%. The 1s mean load ranges between 120 Mbit/s and 300 Mbit/s (15%) with ± variations of about 10% in average and 100% peaks. Obviously, in an inter-domain simulation scenario a high load factor can occur because in an inter-domain „what if I offer transit for this traffic“ scenario the after-lunch load peaks of the joint traffic may be additive. If we add two such traffic aggregates the resulting 1s load values reach the 30% level. Now, the decision whether or not to stop TSSim and to proceed with the fluid or hybrid simulator depends on the microscopic variances of the traffic (measured or well known) and the loss probability threshold. The same approach may be used for both, loss analysis and delay analysis.

4 Summary

In this document we presented the evaluation criteria, scenarios, and results of the Intermon modeling and simulation toolkit. We discussed best practice evaluation procedures and criteria and described the common evaluation testbed in Salzburg, which was used to validate and compare the modeling and simulation approaches. In a first step we then evaluated the correctness and scalability of each approach. Then, we used these tools to model and simulate the Salzburg testbed to be able to compare the results. Finally, we proposed a way to take advantage of the pros and cons of the several approaches, based on the experience gained during evaluation. Depending on the task and situation at hand, either Inter-IP, TSSIM, NS2-Hybrid or RTC-FSIM may be the optimal choice.

5 Appendix

5.1 References

[DSC] http://www.ietf.org/html.charters/diffserv-charter.html

[Kle76] L. Kleinrock, “Queuing Systems”, Vol.2: Computer Applications, John Wiley & Sons, 1976

[OPNET] http://www.mil3.com

[FP01] Floyd, S; Paxson, V – “*Difficulties in Simulating the Internet*” – IEEE/ACM Transaction on Networking, Volume: 9, Issue: 4, Aug. 2001, pages: 392-403

[Bre00] Breslau, L.; Estrin, D.; Fall, K.; Floyd, S.; Heidemann, J.; Helmy, A.; Huang, P.; McCanne, S.; Varadhan, K.; Ya Xu; Haobo Yu - “*Advances in Network Simulation*” – IEEE Computer, Volume: 2, Issue: 2 March-April 2000

[CNO99] Cowie, J.H.; Nicol, D.M.; Ogielski, A.T. – “*Modeling the global Internet*” – Computer Science & Engineering, Volume: 1, Issue: 1, January-February 1999, pages: 42-50

[HEH98] Huang, P.; Estrin, D.; Heidemann, J. - ”*Enabling large-scale simulations: selective abstraction approach to the study of multicast protocols*” - Proceedings of IEEE Sixth International Symposium on Analysis and Simulation of Computer and Telecommunication Systems, 19-24 July 1998, Montreal, Que. Canada, pages: 241 - 248

[HMK01] Heideman, J.; Mills, K.; Kumar, S. – “*Expanding Confidence in Network Simulation*” – IEEE Network Magazine, volume: 15, number: 5, September-October 2001, pages: 58-63

[FJ93] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Transactions on Networking, August 1993.

[DS03] Marek Dabrowski, Felix Strohmeier, Measurement-Based Admission Control in the AQUILA Network, Proceedings of the Workshop on Architectures for Quality of Service in the Internet. Art-QoS 2003, Warsaw, Poland, March 24-25, 2003.

[DAG] http://dag.cs.waikato.ac.nz/

[SCAMPI] http://www.ist-scampi.org/

[Ber02] G. Bergholz, Signalflußsimulation für Nachrichtenverkehrsmodelle, Technical Report SR-ANC-B0I1, Salzburg Research, September 2002

[SPRINT] http://ipmon.sprint.com/