Plan for Simulating the Test Cases Studied in the Integrated Testing Activity

Karl-Johan Grinnemo
Ericsson Infotech AB
Lagergrens gata 2
SE-651 15 Karlstad, SWEDEN
einkjgr@ein.ericsson.se

October 16, 2000

1 Introduction

During the first year of the PRTP project, the main focus was on developing a framework for evaluating the PRTP protocol. A prototype of the protocol was developed in Linux [1]; a robust JPEG coder which, in contrast to standard JPEG coders, works over partially reliable channels was built, and finally a dedicated Web proxy server was developed as part of a Bachelor's project [4]. The experiments performed on the PRTP protocol during the first year of the project were limited to bulk-data transfers between a sender and a receiver in a two-node topology. Thus, only the steady-state behavior of the PRTP protocol was studied.

Now, at the second year of the PRTP project, we have reached a point were it is possible to put together the components developed in the first year, and study PRTP in a more realistic situation. More specifically, we will explore how PRTP behaves in a client/Web server application [3]. Three test cases will be studied [2]: browsing over the Internet, browsing over a modem connection, and browsing over GSM. We believe the experiments performed will not only give us some appreciation of how PRTP functions in a real-world situation, but also give some insights into the transient behavior of PRTP.

The same test cases as used in the client/Web server experiments will also be simulated in ns2, making it possible to gain a deeper understanding of PRTP's transient behavior than possible from the experiments. Also the stationary behavior of PRTP in these test cases will be simulated. We will, however, not simulate a client/Web server application. ns2 doesn't lend itself easily to that kind of applications. Instead, we will simulate something resembling the last part of a client/Web server scenario, i.e., the Web server responds to a former request from the client. Since our interest in the experiments as well as the simulations lies in how PRTP on the client side behaves when the Web server responds with the requested data, this simplification shouldn't lead to any particular loss in the validity of the outcome of the simulations.

The remainder of this document describes in detail how the simulation of the three test cases will be performed. The next section, section 2, elaborates on terminology and parameters common to all simulations. Section 3 provides the simulation plan for the test case: browsing over the Internet. The simulation plan for the test case: browsing over a modem connection, is given in section 4. In section 5, the simulation plan for the test case: browsing over GSM, is presented.

2 Preliminaries

Before presenting the simulation plans for the test cases mentioned in the introduction, some concepts need to be defined.

A *simulation scenario* connotes a particular topology together with ns2 agents and applications, neither having any unspecified parameters. If the topology have an unspecified parameter, or something concerning the agents or applications are left unset, then we end up with something representing a range of scenarios. One for each parameter assignment.

In all simulation scenarios, the same packet-loss patterns are used. These packet-loss patterns will from now on be termed *loss profiles*. We define the following six loss-profiles (see also [3]):

- Loss profile 1 $\equiv \emptyset$,
- Loss profile 2 $\equiv \{8\}$,
- Loss profile 3 $\equiv \{10, 33\}$,
- Loss profile 4 $\equiv \{6, 22, 38\}$,
- Loss profile 5 $\equiv \{4, 15, 18, 24, 44\}$,
- Loss profile 6 $\equiv \{5, 6, 9, 15, 22, 28, 33, 35, 42, 47\}$.

Each loss profile is periodic with a periodicity of 48 packets, i.e. after 48 packets have been received, we start over, and begin counting packets from one again. The loss-profile sets consist of the sequence numbers of the packets to be dropped within a period. For example, if the third loss profile is used this entails the 10th and the 33rd packet in a period are dropped. From now on, we let L denote the set of all loss profiles.

In all test cases, the same set of use cases are studied. All use cases derive from the same template, which we hereafter denote u(b). In u(b), an FTP application sends b bytes to a receiver. The union of all use cases is denoted U, and is formally defined as:

$$U = \{u(5kB), u(8kB), u(12kB), u(20kB), u(35kB), u(60kB), u(1MB)\}$$

Simulation scenarios are represented as cartesian products of sets. We let T denote the set consisting of the topologies to be studied. A particular topology is a function of its parameters, and is denoted $t(\cdot)$. The set of simulation scenarios, denoted S, are calculated as $S = T \times U \times L$.

Some parameters are kept constant in all simulations. The MTU size is always 1500 bytes; the aging factor and the required reliability level are always set to 1.0 and 85%, respectively.

In contrast to the the experiments [3], none of the two options SACK and timestamp is used in the simulations. SACK is not implemented in the PRTP protocol agent (`FullPRTPAgent`), and it seems to us as if the timestamp option is only partially implemented in the TCP agent from which our PRTP agent is derived (`FullTcp`).

3 Browsing over the Internet

This test case strives to mimic the behavior of a fixed Internet connection. In the particular scenario we study, an FTP application, residing on the sender node, sends a file to a receiver. Figure 1 shows a schematic of the topology used in this test case.

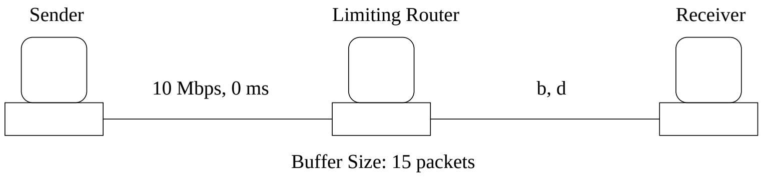

Figure 1: Topology used in test case: browsing over the Internet

The limiting router represents the queue limits existing in intermediary routers. In all scenarios, the queue size of the router is set to 15 packets. The link between the sender and the limiting router is only an ns2 makeup in order to be able to include the router queue limit into the simulation model in the first place. It is dimensioned in a way making the link between the limiting router and receiver be the one governing the bandwidth and delay of the connection between the sender and the receiver.

There are two parameters in the topology used in this test case. b denotes the bandwidth of the link between the limiting router and the receiver, and d the delay experienced over the same link. In the remainder of this section, we let t(b, d) denote the topology depicted in figure 1. The argument b in function t(b, d) stands for the bandwidth of the link between the limiting router and the receiver, and argument d, the delay experienced over the same link.

In reality, the bandwidth between two nodes in the Internet varies over time. These variations are induced in our simulations by letting b be a uniformly distributed stochastic variable.

The following simulation scenarios will be performed on this test case:

$$\begin{aligned} T_1 &\equiv \{t_i(b, 12ms); 1 \leq i \leq 20 \wedge b \in Uni(250kbps, 550kbps)\} \\ T_2 &\equiv \{t_i(b, 30ms); 1 \leq i \leq 20 \wedge b \in Uni(100kbps, 200kbps)\} \\ T_3 &\equiv \{t_i(b, 180ms); 1 \leq i \leq 20 \wedge b \in Uni(30kbps, 90kbps)\} \\ T &= T_1 \cup T_2 \cup T_3 \\ S &= T \times U \times L \end{aligned}$$

4 Browsing over a Modem Connection

The test case described in the previous section was about a fixed Internet connection. Most people do not have a fixed Internet connection. Instead, they dial up an ISP. In this test case, we look at a scenario where an FTP application sends a file over a 56 kbps dial-up connection to a receiver. The topology used in this test case is shown in figure 2 below. The limiting router in figure 2 serves

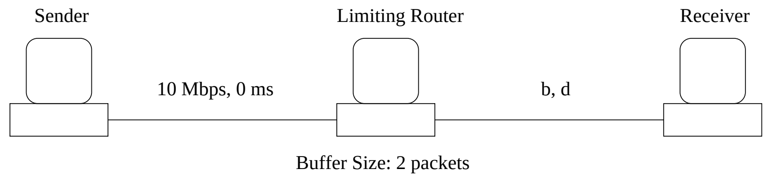

Figure 2: Topology used in test case: browsing over a modem connection

the same purpose as in section 3.

This test case differs from the corresponding experiment (see document [3]) in that the proxy server has been left out. Since the proxy server operates at the application layer and do not start coding/decoding until an image has been completely received, the two activities:

- the Web server sends an image to the proxy server which encodes it,
- the proxy server forwards the encoded image to the client,

happen in series with each other. Thus, the experiment could be decomposed into two simulations. One simulating the traffic between the Web server and the proxy server, and the other simulating the traffic between the proxy server and the client. We could see our test case as simulating the connection between the proxy server and the client. Since the variations in bandwidth and delay over the Web server-proxy server link are neglectible, the traffic over this link could be reduced to a time offset to be added to the proxy server-client simulation. Accordingly, we need only simulate the traffic over this link once.

This test case comprises the following simulation scenarios:

$$\begin{aligned} T &= \{t(33.6kbps, 35ms), t(48kbps, 35ms)\} \\ S &= T \times U \times L \end{aligned}$$

t(b, d) denotes the topology in figure 2. Argument b is the bandwidth, and d the delay of the link between the limiting router and the receiver.

5 Browsing over GSM

In the last test case, we try to simulate a GSM connection. Simulating a wireless connection is not without problems. To make it tractable, we make two assumptions about the GSM connection. First, the test case is confined to a 3.1 kHz modem connection. Second, the connection is non-transparent.

The scenario studied in this test case is the same as the one studied in the previous two test cases, i.e. an FTP application sending a file to a receiver.

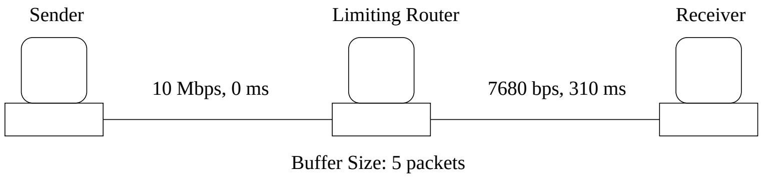

Figure 3: Topology used in test case: browsing over GSM

The topology used in this test case is displayed in figure 3. The limiting router serves the same purpose as in the previous two test cases (see section 3 and 4), and the proxy server is omitted from this test case for the same reason it is omitted in test case: browsing over a modem connection (see section 4).

In this test case, we study the following simulation scenarios:

$$\begin{aligned} T &= \{t\} \\ S &= T \times U \times L \end{aligned}$$

t denotes the topology shown in figure 3 above.

References

[1] K. Asplund, J. Garcia, and A. Brunstrom. Decreasing Transfer Delay Through Partial Reliability. *Proceedings of PROMS 2000*, 2000.

[2] Anna Brunstrom. Analysis and Implementation of a Partially Reliable Transport Protocol for Multimedia Applications: Project Specification. *Project Specification, University of Karlstad*, Dec 1998.

[3] Johan Garcia. Integrated testing - test setup. *Working Report KaU*, October 2000.

[4] Jens Landgre and Katrin Nilsson. Evaluation of proxy vs plug-in and experimental implementations for prtp. *Bachelor's Project 1999:15, Karlstad University*, June 1999.