GENI

Global Environment for Network Innovations

Milestone 3
Discrete-event network simulations

Document ID: GENI-MS3-ERM-March09-v1.0

February 27, 2009

Prepared by:
F. Fidler, C.P. Lai, and K. Bergman
Dept. of Electrical Engineering, Columbia University New York,
500 W. 120th Street,
New York City, NY 10027

under Project Nr. 1631
“Embedding real-time measurements for cross-layer communications”

TABLE OF CONTENTS

1 Document Scope

This section describes this document's purpose, its context within the overall GENI project, the set of related documents, and this document's revision history.

1.1 Executive Summary

This technical note presents the results obtained in work package "Milestone 3: Discrete-event network simulations" of Project Nr. 1631, "Embedding real-time substrate measurements for cross-layer communications".

The purpose of this milestone is to enable and perform discrete-event network simulations to quantitatively evaluate performance impact under several scenarios of cross-layer information exchange. In cooperation with *Alcatel-Lucent Bell Labs*, we therefore developed new simulations and simulation modules into the ns-2 open source network modeling environment which are described in detail in Section 2.

1.2 Related Documents

The following documents are related to this document, and provide background information, requirements, etc., that are important for this document.

1.2.1 GENI Documents

Document ID	Document Title and Issue Date
GENI_QR_ERM_Dec08	4Q08 Status Report

1.3 Document Revision History

The following table provides the revision history for this document, summarizing the date at which it was revised, who revised it, and a brief summary of the changes. This list is maintained in chronological order so the earliest version comes first in the list.

Revision	Date	Revised By	Summary of Changes
1.0	27 February 09	F. Fidler	Initial draft

2 Discrete-event network simulation

2.1 Motivation

In the scope of this work, we want to enable discrete-event network simulations to quantitatively evaluate performance impact under several scenarios of cross-layer information exchange. Simulations are being developed into the ns-2 [ns08_1] open source network modeling environment. The open source software ns-2 is a popular discrete-event network simulator supporting a number of network architectures and protocols, while lacking the implementation of a realistic physical layer model. Implementation of such a model in a unified manner is complicated due to the mere variety of impairments, which for wireless and optical channels may be of static nature (such as attenuation, chromatic dispersion,…) or dynamically varying on different time scales (e.g. fading, polarization mode dispersion, nonlinear crosstalk, power transients,...) However, in order to study optical networks with a systems approach, such physical layer modeling would be highly desirable to:

- identify the key differences of several types of networks when enabled with cross-layer communications,
- evaluate the performance in terms of quantitative measurements such as packet loss or throughput,
- provide a comparison of different quality-of-service (QoS) requirements and cross-layer based routing protocols and network control and management efforts,
- facilitate system architecture planning by reducing system design time.

2.2 Implementation

The notion of a clean-slate network design with a cross-layer optimized protocol stack has been explored to some extent within the wireless networking domain; as such, some modules enabling cross-layer simulations have been previously implemented for wireless networks. We endeavor to implement additional cross-layer enabling modules for the optical physical layer to show that these functionalities will provide significant performance enhancements and to enable the GENI community to conduct their own packet network simulations tailored to the individual researcher's needs.

While the *direct* implementation of physical layer models in a network simulator is prone to be suitable only for a very limited number of scenarios (modulation formats, ...) and impairments, it also might become outdated after some time. Therefore the usage of external software (proprietary or third party software such as Matlab, OptSim,VPI Transmission Maker,...) or even real-time measurements for modelling BER variations in time offers the possibility to take into account various physical impairments with freely chosen fidelity. Without substantial changes to the network simulator, the network performance can be compared when using different models or measurement series. BER variations can simple be exported to a data file (*.tcl, *.csv,...) and then linked to the desired links within the network simulation environment.

The *open source* software ns2 allows integrating additional new modules based on the programming language C++. As an important step towards cross-layer simulations, we have implemented the following functionalities (cf. Figure 2-1): intra-packet BER variations, a forward error correction (FEC) module, and a local control plane. The modules, which can be downloaded from the GENI webpage [geni09_1], are described in detail in the next sections.

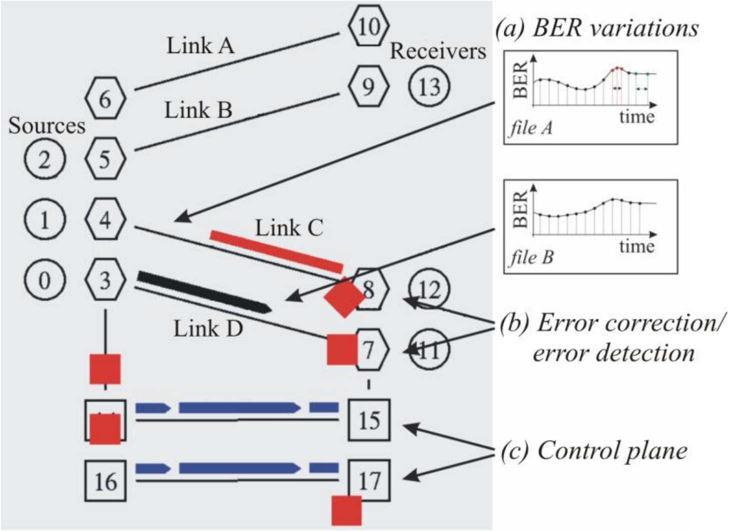


Figure 2-1: New ns2 modules: (a) intra-packet BER variation for every link, (b) error correction and error detection, (c) control plane.

2.2.1 Intra-packet BER variations

The current version of ns2 (ns-2.33) does not allow for the simulation of individual bit-errors on a time scale shorter than the packet duration. The decision to drop a packet is based on a given packet loss ratio and its statistics. The number of bit errors per packet is not calculated or recorded.

Therefore, we have implemented a new software module that calculates the number of bit errors and burst errors per packet according to the variations of the BER with respect to time. These variations may be on a time scale which is less than the packet duration, i.e. the intra-packet variations of the BER can now be taken into account. The time course of the BER is accessible to the simulation environment by means of an external file. Therefore, any simulation tool that allows for the modeling of BER variations over time or even recorded real-time measurements of a physical link may be used for our simulations. In this way, individual files, i.e. BER variations, can be assigned to each network link in the ns2 network topology. As a packet traverses a link (Figure 2-2), the number of bit errors and the number of burst errors that the packet incurs are calculated according to the timestamps of transmit and receive times; these are based on the BER variations that occur during this time period.

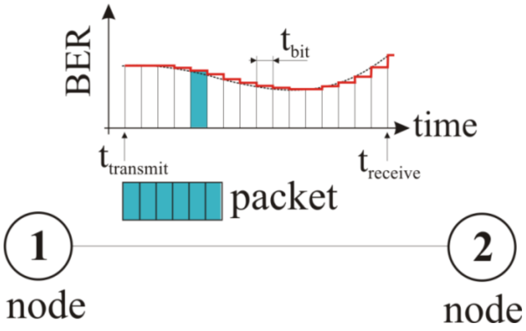


Figure 2-2: Calculation of bit errors within a packet traversing a link based on BER variations stored in external file.

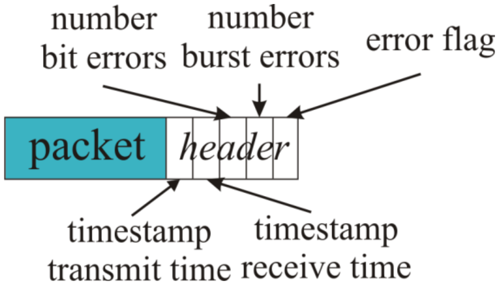


Figure 2-3: New ns2 header added to packet to allow for simulation of intra-packet BER variations.

During simulation, the calculated bit errors and burst errors within each packet are recorded. Each receiving node can then decide whether these errors can be corrected by means of an FEC or whether the packet must be dropped. To allow for the BER variation functionality, an additional ns2 header was added to each packet that stores information about the transmission and receive time of each packet, the number of bit errors, the number of burst errors, and an error flag (cf. Figure 2-3).

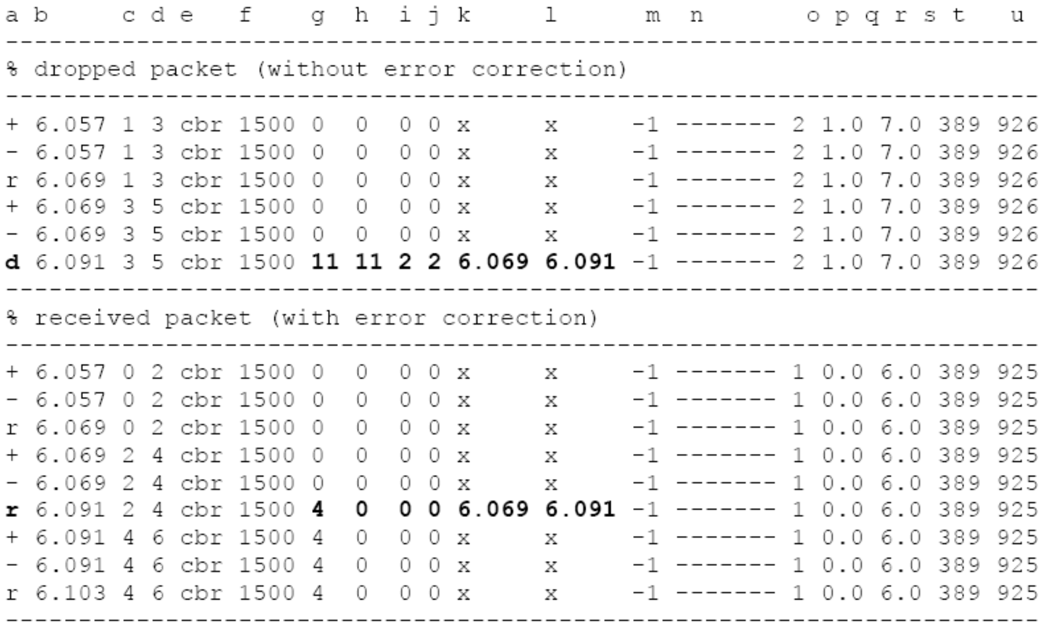
2.2.2 Forward error correction module

As mentioned above, the decision to drop a received packet depends on the FEC scheme used in the receiver. We have implemented a new ns2 module that allows for the specification of whether an FEC for error correction is used in the receiving node, whether a cyclic-redundancy check (CRC) is performed for error detection, if bit errors should be ignored, or if the packet should be dropped whenever it contains bit errors. The error correction and error detection capability can be enabled or disabled independently by means of a flag. The ns2 user can also specify the maximum number of correctable errors, the maximum number of detectable errors, and the maximum number of consecutive erroneous bits within a packet which can still be corrected (i.e. the maximum burst error length). If any of these values is exceeded by the number of errors and burst errors stored in the ns2 packet header, the packet is dropped and/or marked with an error flag (cf. example trace file, Figure 2-4).

2.2.3 Local control plane

The “local control plane” is a module aiming to monitor the QoS requirements for each packet, in addition to the performance on a link, and takes appropriate action if the signal quality degrades, i.e. if the actual BER exceeds a certain, predefined target BER. The control plane module constitutes an essential extendable entity for future cross-layer simulations in the ns2 network simulator and is in an ongoing development stage. The modules’ main intent is to provide a starting point for GENI researchers to implement their own cross-layer optimized network protocols before conducting ns2 network simulations.

The position of the new modules (errmodel_ , errcorr_, and controlplane) within the logical structure of an ns-2 simple link is depicted in Figure 2-5.



```
a b     c d e   f    g  h  i j k     l      m  n      o p q r s t   u
-----------------------------------------------------------------------
% dropped packet (without error correction)
-----------------------------------------------------------------------
+ 6.057 1 3 cbr 1500 0  0  0 0 x     x     -1 ------- 2 1.0 7.0 389 926
- 6.057 1 3 cbr 1500 0  0  0 0 x     x     -1 ------- 2 1.0 7.0 389 926
r 6.069 1 3 cbr 1500 0  0  0 0 x     x     -1 ------- 2 1.0 7.0 389 926
+ 6.069 3 5 cbr 1500 0  0  0 0 x     x     -1 ------- 2 1.0 7.0 389 926
- 6.069 3 5 cbr 1500 0  0  0 0 x     x     -1 ------- 2 1.0 7.0 389 926
d 6.091 3 5 cbr 1500 11 11 2 2 6.069 6.091 -1 ------- 2 1.0 7.0 389 926
-----------------------------------------------------------------------
% received packet (with error correction)
-----------------------------------------------------------------------
+ 6.057 0 2 cbr 1500 0  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
- 6.057 0 2 cbr 1500 0  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
r 6.069 0 2 cbr 1500 0  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
+ 6.069 2 4 cbr 1500 0  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
- 6.069 2 4 cbr 1500 0  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
r 6.091 2 4 cbr 1500 4  0  0 0 6.069 6.091 -1 ------- 1 0.0 6.0 389 925
+ 6.091 4 6 cbr 1500 4  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
- 6.091 4 6 cbr 1500 4  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
r 6.103 4 6 cbr 1500 4  0  0 0 x     x     -1 ------- 1 0.0 6.0 389 925
-----------------------------------------------------------------------
```

Figure 2-4: Trace file - Dropped packet (**d**) sent at t=**6.069**s, received at **6.091**s with **11** bit errors and max. burst error length **2**. Succesfully received packet (**r**) with **4** bit errors (**0** after FEC) and no burst errors. (Columns a: packet status [+..in output queue, -..on link, r..received, d..dropped], b: timestamp, c: source node, d: receive node, e: packet type, f: packet length in byte, g: bit errors, h: bit errors after correction, i: burst error length, j: burst errors after correction, k: transmit time, l: receive time, m: interface, n: flags, o: flow id, p: origin source node, q: origin source port, r: destination receive node, s: destination receive port, t: sequence number, u: packet id)

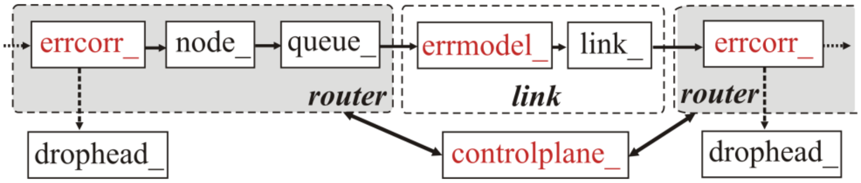


Figure 2-5: Updated ns-2 transmission link (errcorr_...FEC module, errmodel_...intra-packet BER variations, controlplane_...control plane)

2.3 Resources

The software modules for the ns2 discrete-event network simulator, which were specifically developed for its latest version ns-2.33, can be downloaded from the project's GENI Wikipedia page at

http://groups.geni.net/geni/wiki/Embedded%20Real-Time%20Measurements

The repository currently stores the following resources:

Documentation

- Documentation of ERM software tools for ns-2.33
- GNU general public license (version 2)

Software

- All-in-one package download link for standard version of ns-2.33
- New ns-2.33 ERM software modules
- Example Tcl programs for ns2 using the ERM software modules
- Previous versions of the ERM software modules

It is our intention to frequently update this repository with the latest version and further developments of the ERM software, as well as with future publications of our research group which include simulation results using the new software package to give examples. Such updates will be posted on GENI Wikipedia. The network simulator is licensed for use under version 2 of the GNU General Public License and so are the ERM software modules.

THE PROBLEM-FREE FUNCTIONING OF THE MODULES WAS TESTED WITH A WIDE SET OF PARAMETER VALUES AND EVERY EFFORT WAS UNDERTAKEN TO PROVIDE A RELIABLE TOOL. HOWEVER, THE OCCURRENCE OF AN ERROR CANNOT COMPLETELY BE EXCLUDED. THESE PROGRAMS ARE FREE SOFTWARE; YOU CAN REDISTRIBUTE IT AND/OR MODIFY IT UNDER THE TERMS OF THE GNU GENERAL PUBLIC LICENSE AS PUBLISHED BY THE FREE SOFTWARE FOUNDATION; EITHER VERSION 2 OF THE LICENSE, OR (AT YOUR OPTION) ANY LATER VERSION. THESE PROGRAMS ARE DISTRIBUTED IN THE HOPE THAT IT WILL BE USEFUL, BUT WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. SEE THE GNU GENERAL PUBLIC LICENSE (VERSION 2) FOR MORE DETAILS.

3 Bibliography

[1] [geni09_1] Global Environment for Network Innovations – Wikipedia, "Embedded Real-Time Measurements (Project Nr. 1631)" (2009, January) [Online]. Available: http://groups.geni.net/geni/wiki/Embedded%20Real-Time%20Measurements

[2] [geni08_2] GENI Project Office, "Spiral 1 substrate catalog", 05. January 2009

[3] [ns08_1] The network simulator ns2 - Wikipedia (2008, December) [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/Main_Page