Introducing PRouST: The PNNI Routing and Simulation Toolkit

Bilal Khan, David Talmage, Sean Mountcastle, Abdella Battou, Spencer Marsh [*][†]

September 14, 2003

Abstract

Despite the increased availability of affordable ATM hardware in recent years, the scientific research community has often found it difficult to engage in much needed basic research in the areas of ATM protocol design and network performance optimization. We believe that one major cause for this has been the absence of affordable, publicly available source-level implementations of the ATM switch protocol stack. The PNNI Routing and Simulation Toolkit (PRouST) attempts to remedy this. PRouST is a freely distributed, extensible environment for research and development in ATM switch signaling and routing. PRouST includes a complete source-level release of the ATM switch PNNI protocol stack, conformant to version 1.0 of the ATM Forum specification [1]. It is our hope that PRouST will serve as the starting point for bold new initiatives in research and development for ATM technologies. In this paper, we describe the design of PRouST and the features it supports.

1 Introduction

PRouST is used both to simulate networks of ATM switches, and to emulate ATM switches on hosts that reside on an ATM network. In fact, PRouST permits mixing simulated networks with emulated switches and real hardware switches. Each PRouST switch can specify its own custom policies for call admission, path selection, and topology aggregation using a plug-in mechanism. PRouST provides researchers with a language called FATE, which can be used to to specify network scenarios for simulatation. A researcher can conduct PRouST simulation experiments by registering their his/her own custom Investigator objects with PRouST. PRouST reports all network events of interest to registered Investigators for analysis. At present, FATE cannot be used to describe mixed simulated-emulated configurations. Instead, PRouST provides C++ libraries enabling programmers to construct such systems. The libraries facilitate the instantiation of switches, their interconnection, and their attachment to special adapter objects which translate between packets on a network interface card and PRouST's internal data objects.

2 PRouST Software Architecture

PRouST switches are the software analogues of hardware ATM switches. As such, a PRouST switch has a switch fabric, physical ports for connecting to the outside world, a controlling authority that makes decisions in response to the state of the network as perceived by the switch, and programmatic means for external monitoring of switch state. PRouST was designed and implemented using the Component Architecture for Simulating Network Objects (CASiNO). CASiNO is a framework for rapid prototyping of com-

[*] Bilal Khan, David Talmage, and Sean Mountcastle are with ITT Industries, Advanced Engineering & Sciences, Advanced Technology Group, at the Center for Computational Sciences of the Naval Research Laboratory, Washington D.C.

[†] Abdella Battou and Spencer Marsh are with Princeton Optical Systems, Greenbelt MD.

Table 1: The NetStatsCollector reports these PNNI events

Event Type	Meaning
Call_Submission	A call setup request has been submitted at a switch
Call_Arrival	A call setup has reached and was accepted at its destination
Call_Admission	A call has been admitted by ACAC at a switch
Call_Rejection	A call has been rejected by ACAC at a switch
DTL_Expansion	The route and designated transit list for an ATM setup message has been computed at the border node entry point of a peergroup
Crankback	Crankback has occurred
Start_Elections	A switch has started participating in elections
I_am_PGL	A switch believes it is peergroup leader
Voted_Out	A switch that was peergroup leader is no longer peergroup leader
Lost_Election	A switch believes it has lost the election in a peergroup
Hlink_Aggr	Some lower level links are being aggregated at a higher level
NSP_Aggr	Peergroup topology is being aggregated as the complex representation of the logical group node
Hello_Up	A Hello FSM has entered 2-way inside or common hierarchy
Hello_Down	A Hello FSM has left 2-way inside or common hierarchy
NP_Full	A Nodepeer FSM has entered Full State
NP_Down	A Nodepeer FSM has left Full State
NP_Exchanging	A Nodepeer FSM has entered Exchanging State