Galileo: a Tool for Simulation and Analysis of Real-Time Networks

Edward W. Knightly and Giorgio Ventre [*]

International Computer Science Institute, Berkeley, CA 94704
and EECS Department, University of California, Berkeley, CA 94720

Abstract

Galileo is a flexible tool for simulation of heterogeneous real-time communication networks and for development and validation of network protocols. Galileo provides several unique features that make it particularly suitable for the simulation and analysis of networks that provide quality-of-service guarantees. First, its object-oriented programming environment provides the means for a modular, hierarchical, heterogeneous description of networks. Second, its multimedia device interface provides the tools for a qualitative analysis of network protocols. Finally, Galileo's network interface provides interaction with actual networks to access real data and simulate realistic multimedia scenarios.

1 Introduction

In this paper we present Galileo, a tool designed and implemented to address several problems unique to the simulation and analysis of real-time communication networks. The tool is based on Ptolemy [2], an object-oriented software environment that serves as a foundation for simulation of heterogeneous, hierarchical systems. Galileo has several unique features that distinguish it from the vast number of existing network simulators such as Netsim, REAL, and NEST.

First, Galileo is object-oriented. This feature, inherited from Ptolemy, effectively exploits the modular and hierarchical nature of communication network protocols and facilitates their rapid prototyping and testing. The simulation environment, together with Ptolemy's graphical interface, provides a block diagram representation of the network where modules may be encapsulated into higher level modules. This representation allows easy configuration of networks and network protocols. The object-oriented paradigm extends to the protocol code (written in C++) and facilitates an easy exchange of a protocol's algorithms. The modular description of networks also provides the capability of simulating the heterogeneity of internetworks. Because future networks will offer a wide range of communication architectures and services in a rapidly changing scenario, Galileo provides easy integration of new network models into the existing ones, and supports modularity in their development.

The second feature of Galileo is quality-of-service simulation with an emphasis on multimedia applications. The primary goal of the real-time network simulator is to show the quantitative effects of network management decisions on the traffic, and to verify both the correctness and efficiency of the resource allocation schemes. In addition, Galileo facilitates the *qualitative* evaluation of protocols by interacting directly with multimedia devices and applications. For example, Galileo supports the sending of video packets across the simulated network showing the effects of the network on transmitted video streams. This feature is especially useful to the network designer because it provides the capability of analyzing the network's impact on the quality of the communication provided to end-users.

Finally, Galileo provides the tools for interacting with the network itself. For example, if a user wishes to simulate a video conference with another node on the network, Galileo reads the actual information on the current state of the network by interacting with the network protocols. With this information, Galileo is able to simulate the proposed video connection and to show the impact of the network on the communication. Thus, this feature provides the means for the study of realistic scenarios. Indeed, a tool for simulation and analysis of networks should not be isolated from the external world. The availability of actual data from the network components can dramatically improve the quality of the simulation results, particu-

---

[*] On leave from Dipartimento di Informatica e Sistemistica, Universitá degli Studi di Napoli "Federico II", Napoli, Italy

larly for multimedia applications, where precise models for traffic sources are still under development.

This paper is organized as follows. First, some issues unique to real-time networking and simulation are examined along with the Tenet protocol suite (under development by the Tenet Group at the University of California at Berkeley and the International Computer Science Institute). Second, Galileo's architecture is described in terms of its functional modules. Next, Galileo's main modules are described: the Simulation Module, the Multimedia Interface Module, and the Network Analysis Module. Finally, we present a brief example simulation and conclude.

2 Real-time communication

Real-time communication protocols must present mechanisms for the reservation and control of the network's resources. In this section, the general principles of real-time communication are presented along with the Tenet protocols, a real-time protocol suite.

2.1 General principles

The architecture of real-time (i.e., guaranteed performance) communication network protocols has two primary components. First, real-time protocols must partition the network's resources among various clients. That is, the protocols need an algorithm to allocate the network's bandwidth and buffer space to its clients according to their corresponding quality-of-service (QoS) requirements (as well as some pricing scheme). A series of tests based on analysis, measurements, or both are performed to check if sufficient resources are available to meet the client's QoS requirements. The calculations involve transforming the client's requirements for bandwidth, end-to-end delay, delay-jitter, and loss probabilities into local node parameters so that each node of the network guarantees local performance, thus ensuring global performance. Providing statistical bounds on local performance requires calculating the effects of heterogeneous traffic sources (sources with different statistical properties and different priorities) on each other. The client's traffic characteristics (such as minimum and average packet interarrival times and an averaging interval) are also used in the calculations. The real-time channel may then be viewed as a contract between the client and the network. If the channel is accepted, the network guarantees the requested performance bounds provided that the client obeys its specified traffic characteristics.

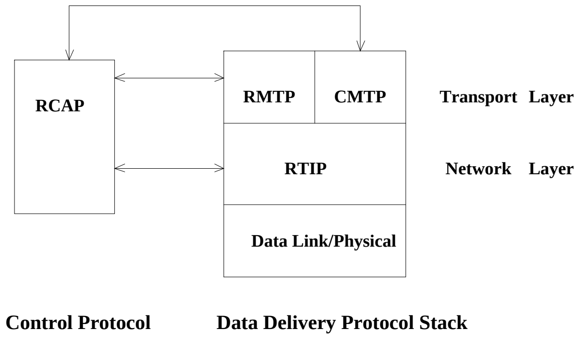

Figure 1: Tenet protocol suite

The second component of real-time communication is scheduling. A scheduling algorithm is needed that can provide the performance guarantees promised by the network. There are several options available, with the underlying requirement that packets are scheduled with some manner of fairness or priority. This is necessary to ensure that a non-real-time channel does not degrade the performance of a real-time channel and that a malicious real-time channel (one exceeding its promised traffic specification) does not degrade the performance of other real-time channels. The scheduling algorithm and admission control algorithm are interdependent with (for example) an HRR (Hierarchical Round Robin) scheduler requiring different tests than an EDD (Earliest Due Date) scheduler. Though the mechanisms and implementation details of various real-time protocols differ, these two components are inherent to guaranteed performance communication (see [4] for a survey).

2.2 The Tenet protocols

The Tenet real-time protocol suite consists of RMTP (Real-Time Message Transport Protocol), CMTP (Continuous-Media Transport Protocol), RTIP (Real-time Internet Protocol), and RCAP (Real-time Channel Administration Protocol) [1]. The logical structure of the protocols is shown in figure 1. Currently, of the four Tenet protocols, Galileo simulates RTIP and RCAP. The primary functions of RTIP and RCAP are as follows:

- RTIP is a simplex, sequenced, unreliable, guaranteed-performance packet service providing rate control, jitter control, packet scheduling, and data transfer. It also supports best-effort traffic in a separate, lower priority, FCFS queue.
- RCAP performs the admission control tests to provide mathematically provable guarantees for

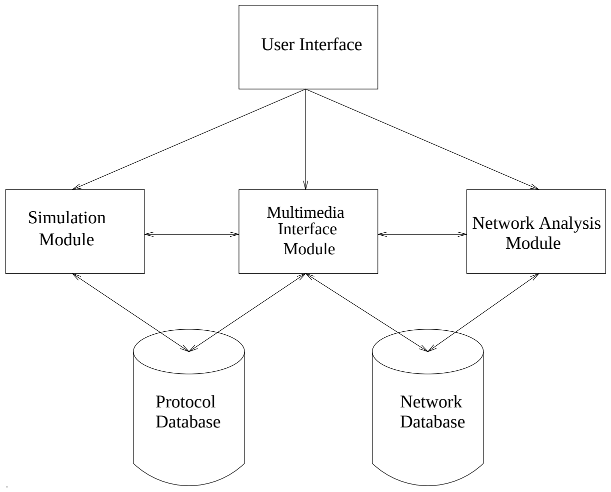

Figure 2: Galileo architecture

throughput, end-to-end delay, delay-jitter, and loss rates.

Galileo is flexible enough to support a variety of communication protocols (both real-time and non-real-time) in addition to the Tenet real-time protocols. Irrespective of the particular protocols involved, the unique requirements of simulating real-time traffic are the same. The simulator must provide quality-of-service validation in addition to performing admission control calculations. For real-time applications, this QoS validation must be qualitative as well as quantitative. Furthermore, the need for an object-oriented programming environment is also evident. With the current variety of algorithms for network functions such as admission control and scheduling, an object-oriented environment allows a comparison of these algorithms with minimal programming difficulty. In addition, these algorithms must coexist in order to simulate heterogeneous networks. The sections below describe how these simulation needs are met.

3 Galileo's architecture

Galileo's architecture is shown in figure 2. At the highest level is the graphical user interface shown in figure 3. The interface, provided by Ptolemy, allows networks to be created graphically by placing and connecting icons. The User Interface interacts with three modules: the Simulation Module, the Multimedia Interface Module, and the Network Analysis Module.

The Simulation Module consists of a library of programming modules such as protocols and network ar-