Composite Protocols and Networking Services

Gary J. Minden
The University of Kansas

Active Networking is concerned with the rapid definition and deployment of innovative, but reliable and robust, networking services. Towards this end we have developed a composite protocol and networking services architecture that encourages re-use of protocol functions, is well defined, and facilitates automatic checking of interfaces and protocol component properties. The architecture has been used to implement common Internet protocols and services. We will report on this work at the workshop.

A number of projects [1, 2, 3, 4, 5] have considered deconstructing networking protocols into simpler components. Once decomposed, one has a library of components from which to build new operational protocols. The advantages of decomposition are that components can be reused, one component can easily substitute for another, and verification of component operation is simple. We developed our architecture and implementation based on the Ensemble system [4]. We removed the group communications components and built new protocol components to implement useful Internet services.

In our architecture, protocols are composed of protocol components. A typical protocol component fragments a packet into smaller message transfer units (MTUs) with a corresponding component at the receiver that reassembles fragments into a complete packet. A protocol component is represented by an augmented finite state machine (AFSM) [6] and a local memory [7]. The AFSM describes which actions to take when a packet is transmitted or a packet is received. The actions on the packet and interactions with local memory are formally specified, and that enables us to consider using automated tools for protocol analysis.

Protocol components are composed to form protocol stacks. A protocol stack is similar to our normal concept of a protocol, for example transmission of a TCP message, or a UDP message. In a multicast environment, we think of protocol stacks for joining/leaving a multicast group; dispersing packets (multicast); and arranging the multicast tree (routing and router placement). These protocols interact to form a network service. Our architecture defines how these separate protocols interact, via a well-defined interface, to form a network service.

Our architecture facilitates reuse of protocol components, rapid development of new protocols and services, and uses rigorous definition to enable formal analysis of protocols and interactions between protocols. We believe basing networking protocols and services on fundamental computing science foundations of well-defined structures and interfaces will facilitate rapid deployment of innovative networking services. Our architecture is demonstrated by several protocol implementations including TCP, UDP, FTP, and reliable multicast.

Our contribution to the workshop will review our composite protocol architecture, show how common protocols are implemented, and show how innovative services can be introduced into a network using a formal approach to defining protocols and services.

[1] N. C. Hutchinson and L. L. Peterson, “The x-Kernel: An architecture for implementing network protocols,” IEEE Transactions on Software Engineering, 17(1):64-76, Jan. 1991.

[2] Gary T. Wong, Matti A. Hiltunen, and Richard D. Schlichting, “A Configurable and Extensible Transport Protocol,” Proceedings of the 20th Annual Conference of IEEE Communications and Computer Societies (INFOCOM 2001), Anchorage, Alaska, (April 2001), pages 319-328.

[3] Robbert van Renesse, Ken Birman, Roy Friedman, Mark Hayden and David Karr, “A Framework for Protocol Composition in Horus,” Proceedings of the 1995 Principles of Distributed Computing, August 1995.

[4] Hayden, Mark. The Ensemble System, Ph.D. Dissertation, Cornell Computer Science Department, January 1998.

[5] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek, “The Click modular router,” ACM Transactions on Computer Systems 18(3), August 2000, pages 263-297.

[6] Yuri Gurevich, “Sequential Abstract State Machines Capture Sequential Algorithms,” ACM Transactions on Computational Logic, vol. 1, no. 1, July 2000, 77-111.

[7] G. J. Minden, E. Komp, S. Ganje, M. Kannan, S. Subramaniam, S. Tan, S. Vallabhaneni, J. B. Evans, "Composite Protocols for Innovative Active Services,” IEEE, Proceedings DARPA Active Networks Conference and Exposition (DANCE 2002), San Francisco, California, May 29-31, 2002.