Toward Reflective Network Architectures

Andrew T. Campbell, and Michael E. Kounavis
COMET Group, Center for Telecommunications Research,
Columbia University, New York NY 10027, USA
`http://comet.columbia.edu/genesis`

1. Introduction

Existing network architectures (e.g., Internet, mobile, Telephone, ATM) exhibit lack of intrinsic architectural flexibility in adapting to new user needs and requirements. Difficulties arise, for example, because of the inability of the TCP protocol to match the high loss rate encountered in wireless networks or for mobile IP to provide fast handoff capabilities with low loss rates. Protocols other than mobile IP and TCP operating in wireless networks might help but their implementation and deployment is non-trivial. In addition, network architects lack software tools for systematically exploring the network design space. As a result, the deployment cycle is typically a "blind" iterative process based on "design, deploy and analyze". We argue that the concept of reflection can be applied to network architectures as an alternative methodology to the way we design, develop, observe and analyze distributed network algorithms. Reflective systems [1-3] provide access to their internal operations, and allow their modification at run-time. We propose the design and implementation of reflective network architectures having the capability to add modify or delete communication algorithms (e.g., routing, resource reservation, or network management) on demand. Reification of network architectures calls for new meta-models and meta-object protocols that deal with the complexity of visualizing and architecting networks. In this position statement we report our experience from the design of the Genesis Kernel [4], a virtual network operating system that supports reflection, and discuss the main challenges toward realizing reflective network architectures.

2. Reflective Network Architectures

Network architectures identify the network hardware and software components and show how they can be arranged to build a complete environment supporting service requirements. We broadly define a network architecture as having the following attributes: (i) network services (e.g. information transport, accounting services) which the network realizes and offers to the end-systems; (ii) multiple time scales which impact and influence the design of network algorithms, and (iii) network state management, which includes the state network algorithms operate on (e.g., routing, QOS state).

Reflective network architectures can be selected and visualized, exposing their implementation and state to network architects. Once selected, network architectures can be observed, and architecturally refined on-the-fly. Architectural refinement [4] includes the ability to modify network attributes using dynamic plug-ins (e.g., replacing a mobility management protocol) or by re-configuring existing network services (e.g., adjusting the configuration parameters of bandwidth brokers in a differentiated services architecture). Reflective network architectures can support the introduction or modification of complex distributed network services as well as specific communication protocols. For example reflection allows mobile terminals to re-program their protocol stacks in order to dynamically interact with heterogeneous wireless access networks [5].

3. The Genesis Kernel

An example of a network operating system that supports refinement of network architectures is the Genesis kernel [4]. A unique feature of the Genesis Kernel is its ability to automate the creation, deployment and management of network architectures. Through profiling, the Genesis Kernel captures the 'blueprint' of network architectures in terms of a comprehensive profiling script. Profiling captures addressing, routing, signaling, security, control and management requirements. Through spawning, the Genesis Kernel systematically sets up the topology and address space, allocates resources and binds transport, routing and network management objects to the physical network infrastructure. Through reflection the Genesis Kernel can exert control over multiple spawned network architectures, allowing designers to analyze the pros and cons of the network design space.

At the lowest level of its hierarchy, the Genesis Kernel transports packets from source to destination end-systems through a set of virtual router nodes called routelets. Routelets represent the lowest level operating system support dedicated to a reflective network architecture. Routelets process packets along a programmable data path at the internetworking layer, and can be composed dynamically during the spawning phase. On top of this transport environment, a programming environment enables the instantiation of distributed algorithms that control and manage the network. In addition the programming environment supports a set of life cycle services for profiling, spawning, and managing distinct network architectures. Thus a 'parent' network can dynamically create 'child' virtual networks with their own protocols for transport, control and management protocols. Architecting based of reflection enables network designers to add, delete, or modify architectural components on demand.

4. Challenges

A major challenge toward the realization of reflective network architectures is related to the complexity associated with reification. Many reflective systems [2,3] provide access to their internal operations in terms of a composition graph, describing the dependencies between their components. Such an approach requires the specification of all interfaces and objects involved. An alternative, more scalable approach, would be to separate the 'binding rules' describing information transport, network control, and network management systems (e.g., a rule for placing and configuring a bandwidth broker inside the network) from the 'binding data' (e.g., network topology, user preferences. We currently investigate the formulation of such a meta-model. In addition, the Genesis Kernel framework introduces the concept of inheritance of binding rules, architectural components or provisioning characteristics. Inheritance allows the network designer to leverage existing network services when constructing new child networks.

Another challenging issue is related to the computational efficiency and performance of reflective network architectures. Currently, transmission rates are increasing more rapidly than the computational power needed for routing and congestion control. Reflective network architectures require more computational resources than existing networks in order to support the introduction or modification of communication services. Reflective network nodes should be capable of forwarding packets as fast as any other handcrafted optimized code executing on routers today. Results from extensible router architectures [8] indicate that software-based packet scheduling and forwarding is viable.

5. References

[1] G. Kiczales, J.des Rivieres, and D. G. Bobrow, "The Art of Metaobject Protocol", *MIT Press*, 1991.

[2] G. S. Blair, G. Coulson, P. Robin, and M. Papathomas, "An architecture for next generation middleware", *IFIP International Conference on Distributed Systems Platforms, and Open Distributed Processing (Middleware'98)*, 1998

[3] F. Kon, R. H. Campbell, M. D. Mickunas, and K. Nahrstedt, "2K: A Distributed Operating System for Dynamic Heterogeneous Environments", *Technical Report, University of Illinois, Urbana-Champaign*, 1999

[4] A. T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. Vicente, and D. Villela, "The Genesis Kernel: A Virtual Network Operating System for Spawning Network Architectures", *Second International Conference on Open Architectures, and Network Programming (OPENARCH,99)*, New York, 1999.

[5] M. E. Kounavis, A. T. Campbell, G. Ito, and G. Binachi, "Supporting Programmable Handoff in Mobile Networks", *Sixth International Workshop on Mobile Multimedia Communications (MoMuC'99)*, San Diego, CA, 1999.

[6] A. A. Lazar, "Programming Telecommunication Networks", *IEEE Network Magazine*, September 1997

[7] D. Tennenhouse, and D. Wetherall, "Towards an Active Network Architecture", *Multimedia Computing and Networking*, San Jose, CA, 1996.

[8] F. Modoux, S.-H. Yun, M. E. Kounavis, A. T. Campbell, "A Programmable Router Architecture", *Technical Report*, *COMET Group*, *Center for Telecommunications Research*, *Columbia University*, 1999.