ELSEVIER

Computer Networks 36 (2001) 115–116

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Guest Editorial

Overlay networks

Virtual networks (VNs) extend the abstraction of virtual memory (VM) to apply to communication systems. They go by many names, including virtual private networks (VPNs) and overlays, and share a vision of networks on networks and interfaces within interfaces. Initially, Internet VNs were developed as sandboxes in which to test new protocols, or as ways to incrementally deploy these protocols in the greater underlying networks. The M-Bone is the most widely known of these Internet VNs, and is still in operation. Commercial VNs continue to be driven by security, as a way to extend private physical infrastructure over shared (Internet) resources, such as to staff located out of house or to selected business partners. VNs are often distinguished by their use of tunnels, encapsulation of packets inside packets, which allows the components of the VN to communicate via the underlying network.

VNs have a number of other uses besides isolating new protocols. Existing protocols can be tested in a variety of environments, including different topologies and link characteristics. Concurrent experiments can be performed on shared equipment, providing the network equivalent of VM and multiprocessing. Large networks can be emulated by smaller networks to examine the effects of scale, while avoiding recoding a real implementation into a simulation system.

Virtualizing a network raises issues at all protocol levels. Support for virtual interfaces and links must extend into the underlying network infrastructure, to support control over multiplexing and policy of use. Network services such as dynamic routing, QoS, and security must be extended to accommodate multiple, concurrent, layered use. Hosts and routers must extend their software interfaces to allow applications and routing protocols to select individual VNs, and control VNs as needed. There are also issues in the specifics of virtualizing hosts, interfaces, links, forwarding engines at routers, and routing tables.

The following three papers examine a variety of these issues from different perspectives. The first paper describes our project at the University of Southern California's Information Sciences Institute (USC/ISI): the X-Bone, a system for the integrated deployment of overlay networks, which configures both hosts and routers. The X-Bone deploys IP overlays using generic Internet protocols and services, including current dynamic routing, IP security protocols, and DNS naming.

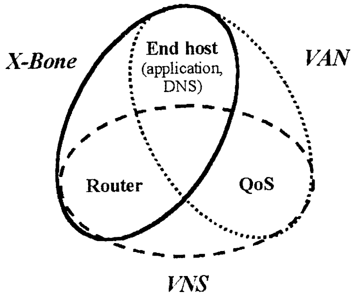
The second paper, from Carnegie-Mellon University (CMU), describes the virtual networking extensions to Darwin, known as VNS. VNS examines the requirements for supporting QoS-based overlay networks. It virtualizes routers to deploy a virtual backbone with translation gateways to convert traffic to and from hosts into backbone format, and monitor the pace of traffic flow into the network.

The third paper, from Columbia University, describes a system for virtual active networking (VAN). Active Networks are programmable at the packet level, where packets carry code to configure the network dynamically (or tags to create a similar effect). VAN focuses on abstracting the raw packet interfaces, rather than any specific network protocol or service, focusing on support for deploying distributed applications.

All three projects address a variety of issues, including router and host configuration, applica-

PII: S1389-1286(01)00171-2

tion and service deployment, resource allocation, reservation, and isolation, and topology management. The three projects cover many of these issues and focus on subsets as shown in the following figure:



These systems share the goal of abstracting network resources, and providing a service for deploying and coordinating the configuration of components that results in a VN. The landscape harkens back to the emergence of VM, with its goal of automated allocation and configuration of storage. Our VN current goals are as challenging, and, if achieved properly, as seamless, invisible, and invaluable.

Joe Touch*
University of Southern California Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6601, USA
E-mail address: touch@isi.edu

Joseph D. Touch received his B.S. (Hons.) degree in biophysics and computer science from the University of Scranton in 1985, an M.S. from Cornell University in 1988 and Ph.D. from the University of Pennsylvania in 1992, both in computer science. He joined the USC/Information Sciences Institute, Marina del Rey, California, in 1992, and is currently the Director of the Postel Center for Experimental Networking in the Computer Networks Division. Joe has led projects ranging from gigabit LANs (ATOMIC2), NIC design (PC-ATOMIC), and multicast Web caching (LSAM), to his current project in the automated deployment and management of VNs (XBONE) and optical Internet WANs and LANs. He is also a Research Assistant Professor in the Department of Computer Science, University of Southern California, where he taught Advanced Operating Systems for several years, and now runs a program for summer students (SGREP) and advises a number of graduate students. Joe also serves on the editorial boards of IEEE Network and Elsevier's Computer Networks, and is a member of several program committees, including IEEE Infocom (since 1994) and Sigcomm, and was the cochair of the IFIP/IEEE Protocols for High-Speed Networks Workshop 1999.

* Postel Center for Experimental Networking, USC/Information Sciences Institute (ISI). Tel.: +1-310-448-9151. http://www.isi.edu/touch.