Network Access in a Diversified Internet

Michael Wilson, Fred Kuhns, and Jonathan Turner
Department of Computer Science and Engineering
Washington University, St. Louis MO. 63130
{mlw2, fredk, jst}@arl.wustl.edu

Abstract. There is a growing interest in virtualized network infrastructures as a means to enable experimental evaluation of new network architectures on a realistic scale. The National Science Foundation's GENI initiative seeks to develop a national experimental facility that would include virtualized network platforms that can support many concurrent experimental networks, with the goal of reducing barriers to new network architectures. This paper focuses on how to extend the concept of virtualized networking through LAN-based access networks to the end systems. We demonstrate that our approach can improve performance by an order of magnitude over other approaches and can enable virtual networks that provide end-to-end quality of service.

1 Introduction

Today's Internet has grown far beyond the original design. New requirements have grown almost as rapidly as the scale of the Internet. Unfortunately, the Internet is owned by no single stakeholder, making it difficult or impossible to upgrade the underlying architecture. [1] As recognized in [2], the inability of the current Internet architecture to meet new needs has led to the development of numerous *ad hoc* solutions to legitimate problems.

The Internet needs a means of deploying potentially disruptive technologies alongside existing technologies. Virtualized networks and protocols could be deployed side-by-side but would be isolated by the virtualization mechanisms. The GENI [3] initiative seeks to use virtualization to create a national experimental facility for experimentation based on these very ideas.

Overlay networks have been proposed as one method of virtualizing the network. However, overlay networks exist on top of existing networks and protocols. We believe that overlay networks should be regarded as a temporary migration solution to allow legacy networks to participate in new services. We propose to make network virtualization a core capability of a next generation *diversified internet* (in the remainder of this paper, we use the term diversification in place of virtualization, because the "*V*-word" has been so overloaded that it is often misinterpreted). In our diversified internet model, the underlying network provides a minimal set of services and a thin provisioning layer upon which new protocols may be developed. More details can be found in [4].

The fundamental abstractions for a diversified network are *substrate routers*, which are connected to each other by point-to-point *substrate links*; and *metarouters*,

which are hosted on substrate routers and are connected to each other by point-to-point *metalinks* carried over substrate links. Collectively, a set of connected metarouters form a *metanet* exchanging *metaframes* adhering to a *metaprotocol*. We refer to the software components that support these abstractions as the Network Diversification Architecture.

In this paper, we focus on the impact of internet diversification on the access network and end systems.

2 Diversification of the Access Network

The access network provides the connection between a network endpoint and the first substrate router. We expect that Ethernet will continue to be one of the most common underlying technologies for access networks, and we focus our attention on the Ethernet context in this paper.

2.1 Objectives

The overarching objective for the access network in a diversified network infrastructure is to make it possible for end systems to take advantage of any network services that may be provided by metanetworks. This objective leads us to the following specific goals.

- *Enable provisioned access*. To support applications which need QoS guarantees, and to enable isolation between metanets, access links must be provisioned.
- *Enable dynamic reallocation of access capacity*. Access network traffic is inherently more dynamic than backbone traffic, and the model should support changes.
- *Support existing Internet protocols*. The existing Internet protocols should be able to operate within a diversified network environment with no loss of functionality and no significant performance degradation.

3 Diversification of the Hosts

Host diversification mechanisms allow the introduction of new *Metanet Protocol Stacks* (MPSs) that provide metanet-specific services to applications. These mechanisms include a common *substrate* which is independent of metanets, but can be configured on behalf of individual metanets.

3.1 Objectives

There are several key objectives that drive the design of the host diversification architecture.

- *Security*. A MPS should have no more privileges than any ordinary application. Administrative access should not be necessary for MPS operation. Applications using a MPS need no administrative access and should not be trusted by a MPS.
- *Traffic Isolation*. Provisioned metalinks must be isolated from one another and from other network traffic. Hosts must ensure that MPSs do not exceed assigned rates, nor does other traffic impact MPS provisions.
- *Efficiency*. The performance of a metanet protocol stack should be comparable to the performance of a stack integrated into the OS kernel.
- *Support Commodity Operating Systems*. We can't expect users to use non-standard operating systems in order to use metanetworks. The software must run on standard OS platforms, including Linux and Windows.
- *Developer Ease*. Applications using a new MPS should use familiar APIs.
- *Ease of Adding New Metanet Stacks*. Installing a new MPS should be no more difficult (or dangerous!) than installing an application program.

3.2 Software Design

In most systems today, network protocol stacks are integrated into the OS kernel and are accessed through the socket interface. This gives the network code unprotected access to kernel data structures. We expect many organizations to develop MPSs. Requiring that new MPSs be added to the OS kernel brings unacceptable security risks.

We solve this problem with a hybrid approach: a user-space implementation of metanet control together with trusted, metanet-independent OS kernel extensions for the data plane.

The *Substrate Kernel Module* (SKM) is a loadable kernel module that coordinates control plane transactions with a metanet control daemon, but handles all metanet data plane operations within the kernel.

The metanet control daemon runs in user space in an *unprivileged* context. The daemon handles control functionality for the MPS, but is divorced from the data path.

User applications interact with a MPS using the standard socket interface. Control requests are forwarded from the SKM to the control daemon; send and receive operations pass through the SKM.

4 Prototype Performance

Our initial prototype was developed on Linux 2.6.16. We currently support a subset of the socket operations, as some operations are nonsensical in our model. Our choices and reasoning are discussed in more detail in the expanded technical report [5].

To test the performance of the system, we created a metanet protocol resembling a combined UDP/IP. We created a test network with two 2.4 GHz machines connected via a 1000 Mb/s switch. Using our new metanet protocol, we measured CPU utilization vs. sending rate limit for rates from 1 Mb/s to 1000 Mb/s, using maximum size packets (1500 octets).

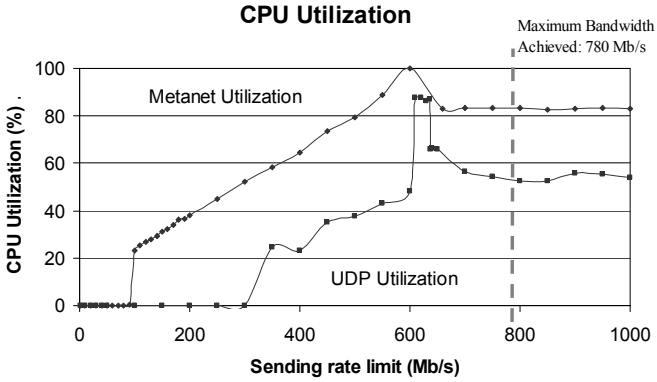

Fig. 1. CPU utilization vs. sending rate as limited by egress queues for metanet and native UDP. Senders were limited by token buckets until 780 Mb/s, where system I/O limits governed.

As shown in Fig. 1, our CPU utilization is largely linear with respect to bandwidth. The spike near 600 Mb/s is due to the implementation of the Linux Token Bucket. To see if there are sufficient tokens to allow sending traffic, the token bucket first dequeues a packet and checks the length. If there are insufficient tokens available, it requeues the packet. This process is repeated every time a packet is queued and at every clock tick. At speeds of 600 Mb/s, we saw upwards of 50,000 requeues per second. At higher rate limits, the queue never has a chance to run out of tokens, so packets are never requeued.

Because of additional outbound validation overhead, our CPU utilization is always worse than native UDP. Comparable systems such as Oasis [6] and PL-VINI [7] become CPU-bound at 3 Mb/s and 200 Mb/s respectively. We regard our system as a worthwhile gain in efficiency.

Further evaluation of our system may be found in the technical report [5].

Acknowledgements
This work is supported by the National Science Foundation (CNS 0325298, 0520778 and 0626661).

REFERENCES

[1] T. Anderson, L. Peterson, S. Shenker, J. Turner, "Overcoming the Internet Impasse through Virtualization," *IEEE Computer Magazine*, Apr. 2005.
[2] Report of NSF Workshop on Overcoming Barriers to Disruptive Innovation in Networking. (January 2005) http://www.arl .wustl.edu/netv/noBarriers_final_report.pdf.
[3] GENI web site. http://www.geni.net
[4] J. Turner, D. Taylor, "Diversifying the Internet," *Proceedings of Globecom*, Nov. 2005.
[5] M. Wilson, F. Kuhns, J. Turner, "Network Access in a Diversified Internet," Washington University Technical. Report. WUCSE-2007-14, Feb. 2007
[6] H. V. Madhyastha, A. Venkataramani, A. Krishnamurthy, T. Anderson, "Oasis: an overlay-aware network stack," *SIGOPS Oper. Syst. Rev.* 40, 1 (Jan. 2006), pp. 41-48.
[7] A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford, "In VINI Veritas: Realistic and Controlled Network Experimentation," SIGCOMM 2006.