AFRL-IF-RS-TR-2004-41
Final Technical Report
February 2004

RENDEZVOUS: SELF-ORGANIZING SERVICES IN AN ACTIVE NETWORK

University of Washington

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. G412

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-41 has been reviewed and is approved for publication

APPROVED: /s/

SCOTT S. SHYNE
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	FEBRUARY 2004	Final May 98 - Mar 02

4. TITLE AND SUBTITLE
RENDEZVOUS: SELF-ORGANIZING SERVICES IN AN ACTIVE NETWORK

5. FUNDING NUMBERS
C - F30602-98-1-0205
PE - 62301E
PR - G412
TA - 00
WU - 01

6. AUTHOR(S)
David J. Wetherall and Thomas E. Anderson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Washington
Department of Computer Science & Engineering
Box 352350
185 Stevens Way
Seattle Washington 98195-2350

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency AFRL/IFGA
3701 North Fairfax Drive 525 Brooks Road
Arlington Virginia 22203-1714 Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2004-41

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Scott S. Shyne/IFGA/(315) 330-4819/ Scott.Shyne@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
" APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED"

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

We propose to explore a rendezvous model that has three characteristics: self-organization, heterogeneity, and administrator for friendliness. Not only do we believe that active network techniques have much to offer the overlay, proxy and interposition models that are prevalent in the Internet, but for the most part previous active networks research has not explored models with these characteristics. Our approach builds on our earlier experience with active networks, and specifically the ANTS active network toolkit. We model the network as an Internet-like arrangement of IP routers, some of which are co-located with active nodes that support additional computation in the form of a well-known runtime. We assume this to be ANTS, although our strategies will be applicable to other runtimes too.

14. SUBJECT TERMS
Active Networks, Heterogeneous Operation, ABONE, Dynamic Routing, Congestion Signaling

15. NUMBER OF PAGES
19

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

ACKNOWLEDGMENT:

This work was funded by the Defense Advanced Research Project Agency (DARPA) under the direction of Dr. Douglas Maughan. The executive agent for this work was the Air Force Research Laboratory/ Information Directorate with Mr. Scott S. Shyne as the Laboratory Program Manager (LPM).

EXECUTIVE SUMMARY:

This is the final technical report for "Rendezvous: Self-Organizing Services in an Active Network" (Grant No. F30602-98-1-0205). Over the past three and a half years, work on this project at the University of Washington has furthered the development of active network technology. The overall thrust of the project has been to help to narrow the gap between the promise of active network technology and its present value in practice. To do so the project was tasked to investigate three broad aspects of active networks infrastructure: automated configuration via self-organization, understanding the impact of node and network heterogeneity, and support for node privileged or "administrator friendly" network services that complement "user specified" services.

The three themes above were explored by both constructing active network infrastructures and through measurement studies that serve to evaluate the heterogeneity of the environment and performance of algorithms running in it. During the course of our investigation, the development of other overlay services (namely P2P networks for file-sharing, end-system multicast and so forth) and new technologies for building large-scale overlays (namely Distributed Hash Tables (DHTs) such as Chord and CAN) influenced research in the active network area. We accommodated these new research directions where possible, e.g., by developing a self-organizing overlay algorithm that did not distinguish between an active network and a P2P one.

The project broadly succeeded at meeting its goals. Notable results from our work include: the ANTS2.0 reference platform, developed and distributed by the University of Washington and the University of Utah for use on the ABONE and elsewhere; measurement studies of bandwidth and client heterogeneity in large overlay networks, as seen in Gnutella and Napster; measurement tools and techniques, including an estimator (King) for the latency between two arbitrary points; a self-organizing overlay algorithm and its comparison by simulation with DHT-based alternatives; and the Denali virtual machine infrastructure for hosting many services on a single node. Publication of our research results and software artifacts has served to distribute these results. Taken as a whole, these results further the development of active networks, overlays, and service hosting.

INTRODUCTION:

The goal of our project is to develop the technologies that will allow the construction of a system in which clients, servers and capabilities that are embedded in the network infrastructure can *rendezvous* to provide Internet middleware services that are requested by applications. For example, in our system, an application might specify that a document be retrieved by a Web service, that a nearby replica is sufficient, and that lossy transcoding or compression be applied if the result is large. The network would then determine which caches, servers, transcoders and compressors should be visited in what order and arrange for their combination. The existing models that are used to insert processing between client and server in the Internet, such as proxies or transparent Web caches, are ad hoc and limited. At the other extreme, research active network prototypes offer great flexibility, but for the most part have not tackled the issue of how processing functionality contributed by different users should be combined.

We believe that an effective programming model for composing network services must possess the three system characteristics described below. Our contract work is designed to explore these characteristics by leveraging active network techniques.

Self-Organization. A major problem with existing software frameworks for active networking is the system administration overhead of maintaining the active network as an overlay on top of the inactive Internet. Automatic configuration via self-organization will reduce the effort needed to deploy new applications and potentially increase the robustness of the overlay and decrease its overheads.

Heterogeneous Operation. We believe that widely varying node and link capabilities are fundamental for technical, economic and administrative reasons and cannot be eliminated simply by using active network techniques to download new functionality. Thus differing capabilities cannot be ignored by the network programming model. This is in contrast with IP, which is a lowest common denominator service in the sense that it is universally available at network nodes.

Administrator Friendly. While the capsule model is suited to introducing user-specified processing, it is clear that administrators as well as users must be able to deploy new services. This requires network programming strategies that not only support both kinds of services but allow them to be combined within the network.

Our contract work has explored each of these areas, using the methods outlined below and with the results discussed in later sections of this report.

METHODS, ASSUMPTIONS, AND PROCEDURES:

Our aim is to prototype and release real Internet software infrastructure that can be freely used by the community in an ongoing fashion, rather than toy examples or short-lived experiments. Our approach has been to build on our earlier experience with active networks, and specifically the ANTS active network toolkit, by developing infrastructure in stages that are staggered over the project lifetime. We have also coupled the development of new platforms with Internet measurement studies of the characteristics of large-scale overlays and trace-based evaluation to better inform ourselves (and other researchers) of the problems we seek to solve.

The initial task we faced was that of revamping the ANTS distribution to provide a basis for subsequent research. The initial version had become widely used both in the ABONE and by the broader research community, but existed in many different versions (Utah, TIS, ICSI, UCLA, TASC, SRI, and MIT). Some of these variations were inevitable as ANTS was used as infrastructure to explore different research goals. However, some of the differences were upgrades and improvements that were more widely applicable. In the first year of our contract work, we merged several enhancements that were mutually compatible and of widespread interest into the reference distribution and, in conjunction with the University of Utah, publicly released ANTS2.0.

In the second and third years of the contract, we faced two further tasks to support self-organizing, heterogeneous active networks, and to support administrator-defined services as well as capsules. To tackle the former, our approach has been to develop

measurement-based clustering algorithms by which ANTS nodes can efficiently, robustly and completely automatically organize themselves into a connected topology that is (a subset of) the ABONE. These algorithms must be fully distributed and require no manual intervention. They must be based on an understanding of the heterogeneous environment in which the ABONE is embedded, and for this reason we undertook a large-scale measurement study to characterize large-scale overlays. The algorithms we developed are also applicable to any overlay, rather than ANTS-specific, and so can be used by other active network nodes as well as separate overlays such as the MBONE.

To tackle the latter and allow administrators to install network services, our approach has been to develop and leverage lightweight virtual machine technology. This will allow many Executive Environments (EE) and Active Applications (AA) variations to be simultaneously hosted by an active node while providing good isolation, so that administrators may add or remove nodes services by simply controlling the set of available EEs. At the same time, users will still be able to program nodes at the AA level. We also refined our approach in response to the shortened end date of the contract and consequent reduction in the scope of work. Specifically, for the preceding task, we developed the base technology but not novel network applications that make use of it.

RESULTS AND DISCUSSION:

We discuss the major results of the project below, by area of contribution.

ANTS2.0 Reference Platform. Working with the University of Utah, we completed the integration of most variants of ANTS, including the merge of ANTS1.3 and ANTSR, and released the result into the public domain as the ANTS2.0 reference platform [1]. ANTS1.3 was an interim checkpoint produced under this contract that consolidated select changes contributed by TASC, Georgia Tech (EnhANTS), ISI (Dante), Utah (multiple classloaders), SRI (anted), MIT (many, sundry), and NAI (no AWT dependence). ANTSR was the Utah train that provided strong resource management functionality. The ANTS2.0 release is publicly available from:

http://www.cs.washington.edu/research/networking/ants/, and
http://www.cs.utah.edu/flux/janos/ants.html.

The main new features of ANTS2.0 are:

- A clean separation between the NodeOS and EE portions of the runtime, with the Janos Java NodeOS (distributed separately) being used as the underlying NodeOS
- Many internal changes for intelligent per-protocol and per-application resource controls.
- A nearly complete restructuring of the ANTS internals, yet few changes to the public ANTS API.
- Inclusion of a new prototype security subsystem that allows the runtime to make fine-grained access control decisions on a per-application and per-protocol basis.
- Separation of the dynamic routing protocol out of the ANTS core. It will operate as a regular application and protocol. This is a nice example of the extensible network philosophy at work.
- Several new demo applications, and more robust scripts for running them.
- Lots of bug fixes!

At the same time, ANTS2.0 preserves the following features of previous distributions:

- ABONE and DANTE support.
- Close to the original API for capsules and applications.
- Example applications.
- JDK1.1.x compatibility.
- A simple BSD-style license.

Containing Runaway Protocols with Icarus. The ANTS2.0 release above includes a mechanism called Icarus for detecting and terminating runaway network protocols that enter forwarding loops across multiple nodes. This work was presented at the ANETS June 2001 PI meeting and published at OPENARCH 2002 [5].

The goal of Icarus is to detect when the execution of an active network program (or more generally an implementation of a network protocol no matter how it is deployed) has become unsafe for the network as a whole. The safety condition that we focus on is a generalized definition of packet loops that can lead to excessive resource consumption if not halted. This problem had not previously received much attention and represents a security vulnerability in active networks: a simple but authentic buggy program could appear locally reasonable every time a packet visits a node, yet could cause an extremely large aggregate amount of work for the overall network that adversely affects legitimate traffic in a manner similar to a denial-of-service attack. Icarus avoids this vulnerability for unicast packets by detecting loops when they are formed, rather than when a TTL expires, and for multicast packets, by checking the distribution tree as it is formed. Our insight was to use Bloom filters to provide a lightweight, probabilistic primitive for detecting cycles. The result is useful both in active networks, and to provide added safety when experimenting with new protocols in other settings such as an overlay.

Self-Organizing Overlays. We developed techniques and algorithms to help the ABONE and other overlays to scale without prohibitive administrative or performance overheads. As indicated at ANETS PI meetings, the lack of self-organizing algorithms has been an impediment to making the ABONE easy to use, and better support for overlays is key to attracting a large user community. The challenge here is to develop a method of organizing overlays that both scales to large numbers of nodes and provides efficient paths. When we began our work in this area, existing algorithms that provide high quality paths were only suited to small scale overlays of less than 50 nodes, and current techniques used for large peer-to-peer overlays were ad hoc and provided poor quality paths.

We developed a hierarchical clustering algorithm that aims to provide paths that are bandwidth efficient as well as latency efficient [4]. There is a natural tension between overlay paths that provide low latency and overlay paths that make good use of bandwidth for broadcast. For example, a fully-connected mesh provides low latency, but causes many copies of a message to travel over the same underlying network links during broadcast. Our approach is to construct an overlay by selecting low latency links, but to bound the outdegree of overlay nodes to avoid a full mesh. Adjusting the bound allows a tradeoff between bandwidth efficiency and latency efficiency.

We used a simulation framework to understand the qualities of the overlay generated by our algorithm. We find that a self-organizing algorithm can greatly improve the quality of the overlay compared to manual, ad hoc arrangements. Not only are paths within the overlay much more performance effective (typically by a factor of at least three) but the hierarchy provides information hiding that limits the propagation of changes, which leads to a more stable overlay structure. We also used simulation to compare our algorithm with Distributed Hash Table (DHT) technology, publishing the results at USITS [6]. Our study concentrates on comparing the quality of the paths DHTs and measurement-based algorithms are able to provide in overlays of around 100-10000 nodes, a range in which both technologies are viable, though DHTs have considerably less overhead for control messages. We find that the DHT-based overlays perform surprisingly well and should be considered candidates for the ABONE and other large overlays. Their "latency stretch" penalty is a factor of two greater than measurement-based overlays, but their "bandwidth hotspot" penalty is comparable to measurement-based overlays. Furthermore, we find that likely improvements in DHT-based algorithms that use topology-aware node identifier assignment or proximity-based routing heuristics have the potential to bring the performance of DHT-based overlays to a level that is comparable to that of measurement-based overlays.

Measurements of P2P Services. To inform the above effort, we conducted a large-scale Internet measurement study on the characteristics of large, real world overlays, focusing primarily on P2P networks such as Napster, Gnutella, and Kazaa. The motivation for this work is to understand the kind of heterogeneity that is present in real overlays so that we can exploit it to manage overlay topologies for various active network applications in a manner that provides for effective performance. The study is also interesting in its own right because Gnutella is one of the largest peer-to-peer overlays in existence, yet it is relatively unstudied because of its newness.

We were able to instrument and measure the bottleneck bandwidth and delays between Gnutella and Napster hosts, providing evidence of their skewed distributions and dynamics. We found significant variations in the reliability of Gnutella nodes, suggesting that only nodes with proven high availability be placed in key roles to minimize the churn due to failures. This work is reported in [9], with a follow-up study on P2P, CDN and Web traffic in [3]. As a side-effect of the effort, we developed new measurement techniques. These include a fast, accurate and practical tool (SProbe) for measuring the bottleneck link bandwidth to and from a variety of Internet hosts, and a tool (King) for estimating the latency between arbitrary Internet hosts without special-purpose end-system support. A paper on King was published at the Internet Measurement Workshop, where it received the Best Student Paper Award, and the source code is available for download from our website [2]. We believe that King will be of value to other researchers and network operators as a building block for tasks that require latency information. Previous methods for estimating latencies extrapolate the measurements between landmark hosts and the hosts being measured. Instead, King estimates the latency between hosts by measuring the latency between their name servers using recursive DNS queries. This technique has two primary advantages over existing methods: (a) it uses the existing DNS infrastructure and hence does not require the deployment of additional infrastructure, and (b) unlike existing techniques, it does not

make assumptions about Internet topology (e.g., triangular inequality over network distances).

Lightweight Virtual Services. We reported results on a platform that moves beyond the capsule model to allow administrators to install network services at nodes in a manner that is synergistic with user requested network processing. Specifically, we developed *Denali*, a platform for executing a large number of simultaneous and mutually untrusted EEs or AAs on a single machine. Our approach is to isolate each application in its own virtual machine (VM), allowing communication only via the network. This provides a high degree of isolation since VM's do not share high-level abstractions like file systems. Thus, security breaches or malicious behavior in one VM need not infect the other VM's on the host machine. What distinguishes Denali from existing approaches in the operating systems community (e.g., VMWare) is our emphasis on scale: we envision running a thousand virtual machines on commodity hardware. What distinguishes it from existing active network approaches (e.g., early versions of ANTS and language-based isolation) is strong isolation and a traditional "whole machine" API for service developers. Denali represents a substantial undertaking in its own right. We successfully completed the design, construction and evaluation of the base Denali technology, and the work is the subject of a paper that appeared at OSDI [7].

CONCLUSIONS:

In this report, we have described the scope of the project "Rendezvous: Self-Organizing Services in an Active Network" (Grant No. F30602-98-1-0205), its methods, and its major results. Over the past three and a half years, work on this project at the University of Washington has furthered the development of active network technology. The overall thrust of the project has been to help to narrow the gap between the promise of active network technology and its present value in practice. To do so the project was tasked to investigate three broad aspects of active networks infrastructure: automated configuration via self-organization, understanding the impact of node and network heterogeneity, and support for node privileged or "administrator friendly" network services that complement "user specified" services. The three themes above were explored by both constructing active network infrastructures and through measurement studies that serve to evaluate the heterogeneity of the environment and performance of algorithms running in it.

The project has broadly succeeded in meeting its goals. Notable results from our work are discussed above and include: the ANTS2.0 reference platform, developed and distributed by the University of Washington and the University of Utah for use on the ABONE and elsewhere; measurement studies of bandwidth and client heterogeneity in large overlay networks, as seen in Gnutella and Napster; measurement tools and techniques, including an estimator (King) for the latency between two arbitrary points; a self-organizing overlay algorithm and its comparison by simulation with DHT-based alternatives; and the Denali virtual machine infrastructure for hosting many services on a single node. Publication of our research results and software artifacts has served to distribute these results.

Taken as a whole, these results further the development of active networks, overlays, and service hosting. We also note that during the course of our project work in the active

network area diffused with that of other areas with the development of other overlay services (namely P2P networks for file-sharing, end-system multicast and so forth) and new technologies for building large-scale overlays (namely Distributed Hash Tables (DHTs) such as Chord and CAN). We view this as a healthy progression, and were able to accommodate these new research directions to some extent, e.g., by developing a self-organizing overlay algorithm that did not distinguish between an active network and a P2P one, and virtual machine technology that can be used in both active networks and other domains.

REFERENCES:

[1] Whitaker, Andrew and David Wetherall. Documentation for ANTS 2.0. *http://www.cs.washington.edu/research/networking/ants/*, and *http://www.cs.utah.edu/flux/janos/ants.html*, 2001.

[2] Krishna P. Gummadi, *"King: Estimating Latency Between Arbitrary Internet End Hosts."* SIGCOMM Internet Measurements Workshop, November 2002. Best Student Paper.

[3] Stefan Saroiu, *"An Analysis of Internet Content Delivery Systems."* The 5th OSDI, December 2002. Best Student Paper.

[4] Jain, Sushant, Ratul Mahajan, David Wetherall, and Gaetano Borriello. "*Scalable Self-Organizing Overlays*." Technical Report UW-CSE-02-06-04, 2002.

[5] Whitaker, Andrew and David Wetherall. *"Forwarding Without Loops in Icarus."* IEEE OpenArch, June 2002.

[6] Jain, Sushant, Ratul Mahajan, and David Wetherall. *"A Study of the Performance Potential of DHT-Based Overlays."* USITS, March 2003.

[7] Andrew Whitaker, *"Scale and Performance in the Denali Isolation Kernel."* Presented at the 5th OSDI December 2002.

[8] Whitaker, Andrew, Marianne Shaw, and Steven D. Gribble. *"Denali: A Scalable Isolation Kernel."* Proceedings of the Tenth ACM SIGOPS European Workshop, September 2002.

[9] Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble: A Measurement Study of Peer-to-Peer File Sharing Systems, *Proceedings of Multimedia Computing and Networking (MMCN) 2002*, San Jose, CA, USA, January 2002. Best Student Paper.

APPENDIX

Appendix A: Publications

Grimm, Robert and Brian Bershad. *"Future Directions: System Support for Pervasive Applications."* Proceedings of FuDiCo (International Workshop on Future Directions in Distribtuion Computing), June 2002, pp. 56-59.

Grimm, Robert, Janet Davis, Eric Lemar, and Brian Bershad. *"Migration for Pervasive Applications."* Submitted to OSDI, June 2002.

Gummadi, P. Krishna, Stefan Saroiu, and Steve Gribble. *"King: Estimating Latency Between Arbitrary Internet Hosts."* Accepted for publication in SIGCOMM-IMW, November 2002.

Gummadi, P. Krishna, Stefan Saroiu, and Steve Gribble. *"King: Estimating Latency Between Arbitrary Internet Hosts."* Accepted for publication in SIGCOMM-IMW, November 2002.

Jain, Sushant, Ratul Mahajan, David Wetherall, and Gaetano Borriello. *"Scalable Self-Organizing Overlays*." Submitted to Infocom 2002.

Jain, Sushant, Ratul Mahajan, David Wetherall, and Gaetano Borriello. *"Scalable Self-Organizing Overlays*." Submitted to Infocom 2002, August 2001.

Jain, Sushant, Ratul Mahajan, and David Wetherall. *"A Study of the Performance Potential of DHT-Based Overlays."* Accepted to USITS, September 2002.

Mahajan, Ratul, Sally Floyd, and David Wetherall. *"Controlling High-Bandwidth Flows at Congested Routers."* Submitted to International Conference of Network Protocols, May 2001.

Mahajan, Ratul, Steven M. Bellovin, Sally Floyd, Vern Paxson, Scott Shenker, and John Ioannidis. *"Aggregate Congestion Control."* Accepted poster for SIGCOMM '01, August 2001.

Saroiu, Stefan. *"An Analysis of Peer-to-Peer File-Sharing Systems."* Submitted to Multimedia Computing and Networking 2002, June 2001.

Saroiu, Stefan, P. Krishna Gummadi, and Steven D. Gribble. *"SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments."* Submitted to Infocom 2002, August 2001.

Saroiu, Stefan. *"A Measurement Study of Peer-to-Peer File-Sharing Systems."* To appear in Multimedia Computing and Networking 2002, Best Paper Award, August 2001. Also available as Technical Report UW-CSE-01-06-02.

Saroiu, Stefan, Krishna Gummadi, and Steve Gribble. *"Exploring the Design Space of Distributed and Peer-to-Peer Systems: Comparing the Web, TRIAD, and Chord/CFS."* Submitted to IPTPS 2002, December 2001.

Saroiu, Stefan, P. Krishna Gummadi, Steve Gribble, and Henry Levy. *"An Analysis of Internet Content Delivery Systems."* Submitted to OSDI, June 2002.

Saroiu, Stefan, P. Krishna Gummadi, Steve Gribble, and Henry Levy. *"An Analysis of Internet Content Delivery Systems."* Accepted to OSDI, December 2002.

Whitaker, Andrew and David Wetherall. *"Containing Runaway Network Protocols with the Icarus Network Architecture."* Submitted to USITS, October 2000.

Whitaker, Andrew and David Wetherall. *"Containing Runaway Network Protocols with the Icarus Network Architecture."* Submitted to USENIX, December 2000.

Whitaker, Andrew. "Containing runaway network protocols with Icarus", Quals Report, University of Washington, February 2001.

Whitaker, Andrew and David Wetherall. *"Forwarding Without Loops in Icarus."* Submitted to Infocom 2002, August 2001.

Whitaker, Andrew and David Wetherall. *"Forwarding Without Loops in Icarus."* Submitted to OpenArch Workshop, December 2001.

Whitaker, Andrew and David Wetherall. Documentation for ANTS 2.0. On ANTS website, 2001.

Whitaker, Andrew and David Wetherall. Documentation for ANTS 2.0. On ANTS website, 2001.

Whitaker, Andrew, Marianne Shaw, and Steve Gribble. *"Denali: Lightweight Virtual Machines for Distributed and Networked Applications."* Submitted to Usenix 2002, December 2001.

Whitaker, Andrew and David Wetherall. *"Denali: A Scalable Isolation Kernel."* SIGOPS European Workshop, June 2002.

Whitaker, Andrew, Marianne Shaw, and Steven Gribble. *"Scale and Performance in the Denali Isolation Kernel."* Submitted to OSDI, June 2002.
Whitaker, Andrew and David Wetherall. *"Forwarding Without Loops in Icarus."* IEEE OpenArch, June 2002.

Whitaker, Andrew: I thought about re-submitting a version of my Icarus work with David Wetherall. We will submit a version of this paper to Infocom at the end of July.

Whitaker, Andrew, Marianne Shaw, and Steven D. Gribble. *"Denali: A Scalable Isolation Kernel."* Proceedings of the Tenth ACM SIGOPS European Workshop, September 2002.

Whitaker, Andrew, Marianne Shaw, and Steven D. Gribble. *"Scale and Performance in the Denali Isolation Kernel."* Accepted to OSDI, December 2002.

Copies of the project papers are available off the UW CSE Network Research webpage, www.cs.washington.edu/research/networking/, in several formats.

Appendix B: Presentation at Meetings and Conferences

Gribble, Steve. *"Denali: Lightweight Virtual Machines for Distributed and Networked Systems."* Presented at HP and Sprint Labs, September 2001.

Gribble, Steve, Andrew Whitaker, and Marianne Shaw. *"Denali: Lightweight Virtual Machines for Distributed and Networked Systems."* Presented at SOSP 2001, October 2001.

Grimm, Robert. *"Future Directions: System Support for Pervasive Applications."* Presented at FuDiCo 2002, Italy, June 2002.

Gummadi, P. Krishna, Stefan Saroiu, and Steven Gribble. *"A Measurement Study of Napster and Gnutella*." Poster at ACM SIGCOMM 2001, August 2001.

Gummadi, P. Krishna. *"Caching in Peer-to-Peer Systems."* Presented at the Systems/ Networking Workshop, University of Washington, June 2002.

Gummadi, P. Krishna. *"King: Estimating Latencies Between Arbitrary End Hosts in the Internet."* Presented at the Systems/Networking Workshop, University of Washington, June 2002.

Gummadi, P. Krishna. *"Measurement and Analysis of Content Delivery Systems in Today's Internet."* Presented at ICIR, ICSI, Berkeley, CA, June 2002.
Jain, Sushant, Ratul Mahajan, Andy Collins, Gaetano Borriello, Steven D. Gribble, David Wetherall. *"Self-Organizing Overlays"*, Poster at University of Washington, Computer Science and Engineering Annual Industrial Affiliates Meeting, February 2001.

Saroiu, Stefan. *"A Measurement Study of Gnutella"*, Talk at University of Washington, Computer Science and Engineering Annual Industrial Affiliates Meeting, February 2001.

Saroiu, Stefan. *"SProbe: Another Tool for Measuring Bottleneck Link Bandwidth"*, Work-In-Progress at the Third USENIX Symposium on Internet Technologies and Systems (USITS 2001), San Francisco, California, USA, March 26-28, 2001.

Saroiu, Stefan and Michael B. Jones "Predictability Requirements of a Soft Modem." SIGMETRICS 2001, Boston, MA, June 2001.

Saroiu, Stefan, P. Krishna Gummadi, and Steven Gribble. *"Characteristics of Current P2P File-Sharing Systems* (with a brief excursion into network measurement tools)." Presented at Microsoft Research, August 16th, 2001.

Saroiu, Stefan. *"Studying Today's Internet Content Delivery Systems."* Presented at the Systems/Networking Workshop, University of Washington, June 2002. Also presented at Computing and Communications Center, University of Washington, June 2002.

Shaw, Marianne. *"Denali: Design and Implementation."* Presented at the Systems/Networking Workshop, University of Washington, June 2002.

Wetherall, David. *ANTS in the ABONE Status Report*, Active Networks PI Meet, May 2000, Portland.

Wetherall, David. *"Containing Runaway Network Protocols with Icarus"*, Talk at DARPA ANets PI Meet, Jackson Hole, June 2001.

Whitaker, Andrew and David Wetherall. *"Containing runaway network protocols with Icarus"*, Poster at University of Washington, Computer Science and Engineering Annual Industrial Affiliates Meeting, February 2001.

Whitaker, Andrew. *"Containing runaway network protocols with Icarus"*, Talk at University of Washington, Computer Science and Engineering Annual Industrial Affiliates Meeting, February 2001.

Whitaker, Andrew. *"Denali Future Directions."* Presented at the Systems/Networking Workshop, University of Washington, June 2002.

Whitaker, Andrew. *"Forwarding Without Loops in Icarus."* Presented at the IEEE Open Arch Workshop, New York, June 2002.

Whitaker, Andrew, *"Denali: A Scalable Isolation Kernel."* Proceedings of the Tenth ACM SIGOPS European Workshop, September 2002.

Whitaker, Andrew and Steve Gribble "Issues in Machine Virtualization." Networking Workshop, Gig Harbor, WA, June 2001.

SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AA ---------- Active Application
ABONE ---------- Active Network Backbone
ANTS ---------- Active Node Transfer System
ANTS2.0 ---------- ANTS Reference Platform (Version 2.0)
ANTSR ---------- ANTS with Resource Controls
API ---------- Application Programmer Interface
CAN ---------- Content Addressable Network
CDN ---------- Content Distribution Network
DHT ---------- Distributed Hash Table
DNS ---------- Domain Name System
EE ---------- Execution Environment
ICSI ---------- International Computer Science Institute
IP ---------- Internet Protocol
MBONE ---------- Multicast Backbone
MIT ---------- Massachusetts Institute of Technology
OPENARCH ---------- Open Architectures and Programmable Networks
OSDI ---------- Operating Systems Design and Implementation
P2P ---------- Peer-to-Peer
SRI ---------- Stanford Research Institute
TIS ---------- Trusted Information Systems
UCLA ---------- University of California, Los Angeles
USITS ---------- Usenix Symposium on Internet Technologies and Systems
Utah ---------- University of Utah
VM ---------- Virtual Machine