Report on WREN 2009 – Workshop: Research on Enterprise Networking

Nathan Farrington
UC San Diego
farrington@cs.ucsd.edu

Nikhil Handigol
Stanford University
nikhil.handigol@gmail.com

Christoph Mayer
University of Karlsruhe
mayer@tm.uka.de

Kok-Kiong Yap
Stanford University
yapkke@stanford.edu

Jeffrey C. Mogul
HP Labs, Palo Alto
Jeff.Mogul@hp.com
(report editor)

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.3 [**Computer-Communication Networks**]: Network Operations

General Terms

Management, Measurement, Performance, Reliability, Security

Keywords

Enterprise networks

1. INTRODUCTION

WREN 2009, the Workshop on Research on Enterprise Networking, was held on August 21, 2009, in conjunction with SIGCOMM 2009 in Barcelona. WREN focussed on research challenges and results specific to enterprise and data-center networks. Details about the workshop, including the organizers and the papers presented, are at `http://conferences.sigcomm.org/sigcomm/2009/workshops/wren/index.php`. Approximately 48 people registered to attend WREN.

The workshop was structured to encourage a lot of questions and discussion. To record what was said, four volunteer scribes (Nathan Farrington, Nikhil Handigol, Christoph Mayer, and Kok-Kiong Yap) took notes. This report is a merged and edited version of their notes. Please realize that the result, while presented in the form of quotations, is *at best* a paraphrasing of what was actually said, and *in some cases may be mistaken*. Also, some quotes might be mis-attributed, and some discussion has been lost, due to the interactive nature of the workshop.

The second instance of WREN will be combined with the Internet Network Management Workshop (INM), in conjunction with NSDI 2010; see `http://www.usenix.org/event/inmwren10/cfp/` for deadlines and additional information.

Also note that two papers from WREN were re-published in the January 2010 issue of *Computer Communication Review*: "Understanding Data Center Traffic Characteristics," by Theophilus A Benson, Ashok Anand, Aditya Akella, and Ming Zhang, and "Remote Network Labs: An On-Demand Network Cloud for Configuration Testing," by Huan Liu and Dan Orban.

2. SESSION 1: SECURITY

Session chair: Jeff Mogul, HP Labs
Session scribes: Nathan Farrington, Christoph Mayer

Practical Declarative Network Management

Authors: Timothy Hinrichs (University of Chicago); Natasha Gude, Martin Casado, John Mitchell (Stanford University); Scott Shenker (U.C. Berkeley and ICSI)
Presenter: Natasha Gude

Summary:

Natasha Gude presented a flow-based management language (FML), for the declaration of policies in enterprise networks. Today's networks have a vast amount of distributed configuration, including configuration of VLANs, subnets, ACLs, NAT, and routing policies. Handling such configuration in enterprise networks is complex, mainly because (1) configuration is low-level, and does not reflect the actual goals of an administrator, and (2) dynamic change in the network, such as growth, increases complexity. Also, current rule-based mechanisms require you to get the rules in the right order, which can lead to mistakes. Today, network administrators have in-depth knowledge on the complete setup of hosts, communication, and all applied mechanisms. As networks grow this approach becomes unfeasible or even impossible.

The goal of the FML policy language is to simplify network configuration, focussed on high-level goals (e.g., all guests must send HTTP traffic via a proxy) to support configurations that are easier to formulate, understand, and maintain, without sacrificing expressiveness. FML is based on non-recursive Datalog. They implemented FML using the NOX OpenFlow controller, and deployed their system on a production network with 200 hosts for 10 months. In a separate data-center implementation, they cache the policy in the switch rather than in the NOX OpenFlow controller.

Discussion:

Saikat Guha (MPI): Are the flow levels and actions complete? Answer: the granularity in terms of variables can express quite anything, but it is hard to tell whether they really are complete. Completeness has not been proved formally, but in actual deployment it has been shown that flow levels and actions are sufficient.

Michalis Faloustos (UC Riverside): Does a rule need to see multiple packets from the same flow before being enacted? E.g., one would need to see several packets of an HTTP request before being

able to classify it. Answer: they had such a case with HTTP redirect, so they used a single admin-defined action that would wait before pushing the decision down to the controller.

Saikat Guha (MPI): Can you share information between enterprises; e.g. user names shared between companies when classification is performed on per-user rules. Answer: it is necessary to get such information from different sources, such as LDAP, and that such sources must be defined beforehand.

Ant Rowstron (Microsoft): how does classification work with IPSEC, since the flow would be encrypted? Answer: In this case, the flow rules can only work on the unencrypted header.

Nathan Farrington (UCSD): Is it possible for multiple rules to be combined and unintentionally create a forwarding loop? Answer: This would be detected by the policy engine, and the flow would be denied.

Resonance: Inference-based Dynamic Access Control for Enterprise Networks

Authors: Ankur Nayak, Alex Reimers, Nick Feamster, Russ Clark (Georgia Tech)
Presenter: Ankur Nayak

Summary:

Ankur Nayak presented the Resonance system for dynamic access control in enterprise networks. The goal of Resonance is in-network access control, in contrast to today's networks, where a large number of components like firewalls, VLANs, etc. are bolted onto the network. Through the use of programmable switches that manipulate traffic at lower layers (dropping or redirecting packets), high-level policies can be enforced early on the path.

Resonance, based on OpenFlow and the NOX controller, tracks the state of each host on the network and updates the forwarding state of switches per host as these host states change. Resonance (1) associates hosts with states, (2) specifies a state machine for moving machines from one state to another, and (3) controls forwarding state in switches based on the current state of each machine.

A research testbed has been set up in a small part of the Georgia Tech campus networks to show the feasibility of Resonance, with plans for a larger deployment on the complete campus network. They have some performance issues with the OpenFlow implementation, and they have a security problem based on MAC-address spoofing.

Discussion:

Kok-Kiong Yap (Stanford): What is the scale of the measurements and how many flows are used in the testbed? Answer: We used actual traces from Georgia Tech campus, and looked at how many entries would actually be needed: about 1K entries, which is well below what OpenFlow can handle.

Jad Naous (Stanford): how do you monitor the hosts? Answer: We treat hosts as black boxes. Vulnerability scanning is performed by dedicated boxes that then report back to the switches; the controller then decides whether to put a host into an "isolated" state.

Lars Eggert: What is the difference between what you do and what someone else does? The IETF has worked on similar approaches and has defined protocols. Answer: There is no central authority that governs the entire network. Clarify: these guys have it.

Michalis Faloustos (UC Riverside): can you compare your work with the previous talk? Answer: Resonance provides a higher level of abstraction (machines) rather than FML's flow-level abstraction. FML is a policy language and that work comes into play when you define policies for the various states. Resonance states are Registration, Authenticated, Quarantined, and Operation. FML could be used with Resonance, they are complimentary.

Jeff Mogul (HP Labs): Sounds like the state machine is integrated into the system in Resonance.

Peter Druschel (MPI): What if you didn't have the problem of reassigning an IP address, do you still need to have flow-based access control instead of VLANs? Answer: we believe that VLAN-based access control is very static, inflexible, and very coarse-grained.

Aditya Akella (U Wisconsin): How important is the state machine? OpenFlow already gives you the ability, right? So what are you adding? Answer: We are adding an extra layer. We are looking at the network from a global view and looking at which machines are compromised and which are not. We are looking at what set of access control policies for each host. We want to identify machines as being infected or not. We want to have different kinds of states that might be possible in an Enterprise network, such as guest hosts.

Delegating Network Security Through More Information

Authors: Jad Naous, Ryan Stutsman, David Mazieres, Nick McKeown (Stanford University); Nickolai Zeldovich (MIT CSAIL)
Presenter: Jad Naous

Summary:

Jad Naous presented the ident++ protocol for requesting information from end-hosts and networks on the path of a flow. In today's networks, administrators cannot define complex and high-level policies (e.g. Skype calls can only reach other Skype clients; only a specific Email client may be used; or that only certain groups of users can talk to the complete Internet while others only to specific networks), mainly because the network does not provide enough information to make these decisions. Ident++ addresses the problem of specifying network security policies across administrative domains. The solution involves allowing individual users to specify their own security policies.

Ident++ allows parts of the system to 'ask' for such missing information on the path between sender and destination – for example, user names, application names, versions, patches, network names, location, or expected application behavior – in the form of key-value pairs. A firewall, for example, could use this info to install new rules. All routers in such an ident++-enabled network are OpenFlow switches that report packets to an OpenFlow controller. The controller then requests the additional information.

Discussion:

Ant Rowstron (Microsoft): The approach you are taking is to ship rules into the network, but if you had something more static then you could enforce it on the end hosts – wouldn't that be easier?. Answer: you are talking about distributed firewalls. If you have to enforce rules on the end host, then you can't reliably stop attacks, because the hosts are not trusted. Also, how would you collect information from different administrative domains to implement it on your firewall? Also, you may want a single point in the network where packets are dropped, especially in case of multiple administrative domains.

Ant Rowstron: How can the information given by the end-system on request be trusted, and (2) why it should be better to put such information into the network than rather keep it at the end-systems? (The discussion was taken offline at the request of the program chair.)

Albert Greenberg (Microsoft): In your experience, what kind of rules do people want to use? What do the rules look like? How many? Answer: this is just a proposal. We haven't actually deployed it.

Saikat Guha (MPI): How did you evolve to this design? Answer: a first packet that has all the information is sent first and cached by the firewall. [this response seems garbled]

Justin McCann (U Maryland): There is an open-source firewall that does this (netfilter) – it doesn't do exactly what you describe, but will authenticate the end user.

Jeff Mogul (HP Labs): If you could rely on having a trusted platform module, would that solve some of the problems? Answer: While you can get info from the end hosts, you can't necessarily trust them. If an admin's laptop is compromised, you can get access to lots of machines. So whenever you delegate, you are putting some trust in that information. If you don't trust the information, you shouldn't use it. But firewalls do make assumptions, and we are saying let's make this assumption explicit.

Impact of IT Monoculture on Behavioral End Host Intrusion Detection

Authors: Dhiman Barman (Juniper Networks); Jaideep Chandrashekar (Intel Research, Santa Clara); Michalis Faloutsos (UC, Riverside); Ling Huang, Nina Taft (Intel Research, Berkeley); Frederic Giroire (INRIA, Sophia Antipolis, France)
Presenter: Michalis Faloutsos

Summary:

Michalis Faloutsos described a study on the impact of "IT monoculture" on end-host intrusion detection. They concluded that it is advantageous to use more than one host intrusion detection policy for all users in the enterprise; this helps to detect a more diverse set of attacks. In the ideal case, there would be a separate intrusion detection policy per user, but they found that it sufficient to group users into a few groups, and to assign the same policy to the whole group. Their analysis showed that 8 groups was optimal for a particular set of 350 notebook computers.

Discussion:

Kok-Kiong Yap (Stanford): 8 groups is good. Where is the huge jump? 2, 3, 5, etc.? Answer: We did not do that thoroughly. It did vary a bit with the actual features used.

Aditya Akella (U. Wisconsin): How were the groups defined? Did you see if groups mapped to different properties of the operating system? Answer: could we identify particular trends? There were privacy issues so we didn't do that; the data sets were anonymized. They were all users with notebooks.

Huan Liu (Accenture): I'm surprised that you are looking at the number of TCP or UDP sessions. I would think you would want to look at the number of sites rather than sessions. Did you look at the destinations of the traffic? Answer: no. Part of the motivation was that the tool would run on the laptop with secure hardware, such as on the NIC that could count your network behavior. So keeping track of URLs becomes sort of expensive in this case. But if you add more features, then you could have a more powerful intrusion detection system. We were looking at a very simple feature set that existing hardware already used. (Justin McCann, Maryland, also asked about the window size for counting connections.)

Albert Greenberg (Microsoft): I didn't understand how you simulated the attacks. How did you decide if you had a false positive or false negative? Answer: we introduced the attack.

Justin McCann (Maryland): What timescales were you examining? Answer: I don't remember.

Jeff Mogul (HP Labs): I work in a company with 300K users. How would the number of groups scale with more users? Answer: I don't know but I would guess that even 8 groups would provide benefits. They could try to go to 100 or 200 and see what then happens to the number of groups.

3. SESSION 2: SYSTEM DESIGN

Session chair: Peter Druschel, MPI-SWS)

Session scribes: Nathan Farrington, Nikhil Handigol, Christoph Mayer

Hash, Don't Cache: Fast Packet Forwarding for Enterprise Edge Routers

Authors: Minlan Yu, Jennifer Rexford (Princeton University)
Presenter: Minlan Yu

Summary:

Minlan Yu presented a design for scaling the size of router forwarding tables by using bloom filters. The challenges include tables over 250K entries, ever increasing link speeds, and the high cost of large fast memories. Route caching is not a viable solution, especially because of poor performance on worst-case workloads such as malicious traffic or route changes.

Her technique exploits small-but-fast SRAM, and uses one Bloom filter per next-hop (or outgoing link), which stores all addresses forwarded to that next hop. Each Bloom filter is checked in parallel and hopefully exactly one of the returns true. If two or more return true, due to the probabilistic nature of Bloom filters, then there are multiple techniques used to handle the false positives.

They treated this as a convex optimization problem, with the goal of minimizing overall false positive rate, and constraints on memory size and maximum number of hash functions. It takes 50 ms to solve the optimization problem with 200K entries, 10 next hops, and 8 hash functions; 600KB of fast memory achieves a 0.01% false-positive rate.

Since Bloom filters do not directly support deletion, in order to handle route changes they used a Counting Bloom Filter (CBF) in slow memory to make updates and resizing easier. All non-zero entries in the CBF are mapped to 1 in the SRAM Bloom filter; all zero entries in the CBF are mapped to 0.

Discussion:

[missing]: What about cheap TCAMs?

[missing]: What do you mean by changing significantly?

Praveen Yalagandula (HP Labs): How do you handle the longest prefix matching? Answer: Based on SIGCOMM 2003 work: use one Bloom filter for each (next-hop, prefix length) pair, and do parallel lookups on all bloom filters.

Yanpei Chen (UC Berkeley): Isn't your work just shifting the bottleneck from RAM to CPU (to compute hash functions)? As router designs progress, you can shift the work one way or the other. Is memory getting cheaper or are we going to hit a CPU bottleneck? Answer: we are not significantly increasing CPU, and we use cheap hash functions that can easily be implemented in hardware.

Huan Liu (Accenture): It seems you need a lot of memory access to do this lookup: $8 * 10 = 80$ memory reads. Wouldn't it be easier to use a larger DRAM and do a single hashed lookup? Answer: Since we use the same group of hash functions, we will query the same positions of all the Bloom filters, so we store the column together in memory. Therefore, we only need S memory accesses for the S hash functions; the total access time is smaller than a single DRAM lookup.

Praveen Yalagandula (HP Labs): For the false-positive case, you will randomly pick one of the output ports. But if you randomly pick one of the next hops, it might be going internally or externally. If it matches on all 3, then if you randomly pick the internal port instead of the external port, then you will be incorrect. Answer: We can leverage the property of enterprise routers that the number of internal IP addresses is small. We can provide a hash-based solu-

tion for internal lookup, but use a Bloom filter for external lookup. That reduces the number of dropped packets.

Kok-Kiong Yap (Stanford): I'm concerned over the packet-processing time in hardware. With increased packet-processing time, you'll need to hold that many more packets in memory. Won't that increase the memory size? Answer: We want to bound the packet processing time. Hardware processing time can be used as a constraint in the optimization problem, and can be tuned based on hardware characteristics.

Crossbow: A Vertically Integrated QoS Stack

Authors: Sunay Tripathi, Nicolas Droux, Thirumalai Srinivasan, Kais Belgaied, Venu Iyer (Sun Microsystems)
Presenter: Sunay Tripathi

Summary:

Issues in host-based QoS solutions include performance (additional classification/queuing for all packets; existing QoS layers are generally high up in stack; packets need to be DMA'd into the system before any policy can be applied), management complexity, and the use of multiple VMs in a single host, in which case switches can't do much to manage QoS. This talk is about how you slice up the network resources, to provide QoS.

The crux of Crossbow is that you need to look at the packet before you can do anything, but the act of looking at the packet causes a lot of CPU overhead. You don't want to pay the overheads of QoS for normal packet processing. So, in Crossbow the policy decision is made before the packet enters the system. They use a flow classifier to classify traffic into separate hardware "lanes" with dedicated resources. Lanes can be defined in terms of services, transports, or IP addresses.

They gain efficiency through "dynamic polling" and "packet chaining" (no interrupts; whenever packets start arriving the lane is set to polling mode). Bandwidth allocation becomes easy, and Crossbow achieves TCP bandwidth close to configured bandwidth because packets are not dropped. Features include bandwidth limits, priorities, per-VM bandwidth configuration, and defense against DOS attacks.

Crossbow is implemented in the Solaris networking stack, supports multi-core CPUs and multi-10gig bandwidth, exploits advanced NIC features, and allows for better resource partitioning

Discussion:

Ernst Biersack (Eurecom): can you give some intuition as to why TCP performance does not suffer? Answer: We basically prevent the system from dropping the packets by queueing them up in the NIC, up to 200 packets. By letting the queue build up, we allow TCP to adjust itself.

Jad Naous (Stanford): Don't you need a hardware abstraction layer? What if the physical NIC does not have all the features required? Answer: The device driver written for crossbow is a little different. We require the device driver to support certain simple functionality, and that is the abstraction layer.

Jeff Mogul (HP Labs): How do you deal with the problem where you need more classifications than the number of available hardware lanes, normally 2, 4, or 8? Does the administrator have to figure out which lane needs to be allocated for what? Answer: Yes. And it can be dynamically changed. You do a power of 2 flows in hardware and the rest in software.

Parveen Patel (MSR): Can you dynamically change the number of buffers? Answer: Yes, although the number of receive and transmit descriptors is NIC-dependent. But the rest of the resources (CPU, bandwidth limit, etc.) can be assigned dynamically.

Alexandre Gerber (AT&T): In the data center, can you talk about the advantages of doing this in the host layer as opposed to the network layer? Answer: you can do this in many places in the network, which is why QoS became less important recently. But when you introduce virtualization, the switches can't do much, and you need host-based QoS. In the DC, there are multiple types of traffic, and you need to be able to assign priorities.

Huan Liu (Accenture): How different is this from the Linux networking stack? Answer: The idea of a vertically-integrated lane doesn't exist in other stacks, including Linux. The NIC vendors are writing the drivers. In Linux, you still bring packet into the system before deciding how to deal with it. There is NAPI, which allows polling, but doesn't use the NIC's ability to do classification. Crossbow is better in terms of both performance and fairness.

4. SESSION 3: MEASUREMENT AND MODELING

Session chair: Kashi Vishwanath, Microsoft Research
Session scribes: Nathan Farrington, Kok-Kiong Yap

Multicast Redux: A First Look at Enterprise Multicast Traffic

Authors: Elliott Karpilovsky (Princeton University); Lee Breslau, Alexandre Gerber, Shubho Sen (AT&T Labs – Research)
Presenter: Alexandre Gerber

Summary:

Multicast is an efficient 1-to-many distribution mechanism, but in spite of significant interest in 1990s, was not really successful on the Internet. It has reemerged in recent years, for IPTV, disseminating financial data, and in enterprise networks, which typically support an MPLS-based VPN. MPLS itself does not support multicast, so GRE encapsulation is used over the MPLS network; however, this leads to possible scalability issues.

Little is known about the deployed networks, hence this study. They found multicast traffic fit nicely into four categories, meaning that there is a diverse set of multicast applications in use. Somewhat surprising: 50% of the sessions have only one Provider Edge (PE) receiver.

Discussion:

Praveen Yalagandula (HP Labs): What are you going to do with the observations? Answer: The next step is to conduct a more detailed analysis. We can understand what are these applications and why they are using multicast. There might be some opportunities to fine tune the parameters.

Jeff Mogul (HP Labs): You mentioned deep packet inspection in future work. As a provider, what are your obligations to your customers about what you can do and what you can't? Answer: we can do studies to provide better service to our customers.

Yanpei Chen (UC Berkeley): Will a subset of the 5 features better explain the data? Have you considered eliminating a feature to see if better clustering results? Answer: We did not try that. We have only 5 features, so we used everything we have.

Understanding Data Center Traffic Characteristics

Authors: Theophilus A Benson, Ashok Anand, Aditya Akella (University Of Wisconsin); Ming Zhang (Microsoft Research)
Presenter: Theophilus A Benson

Summary:

Very little is known about data center networks, making them hard to evaluate. This is a study of data center network traffic, including the description of a mechanism for estimating fine-grained behavior from course-grained measurements. They have SNMP data from the entire network, but packet traces from only a few

links; they found a way to estimate the fine-grained behavior of the other links.

Some of the overall findings were that few data center links experience loss, many links were unutilized, traffic exhibits on-off characteristics, and that the arrival rate is log-normal. They created a traffic generator based on their measurements of real data center traffic.

Discussion:

Srikanth Kandula (Microsoft) [or was this someone from MIT?]: Where are packets getting dropped? Answer: Buffer overflow in the switches.

Huan Liu (Accenture): Did you correlate the counters on both sides? Answer: Not yet.

Srikanth Kandula (Microsoft): Would you get qualitatively different results for different data centers? Answer: The traces are from one data center now. We are trying to get traces from other data centers. The result is consistent across links for the available data set.

Albert Greenberg (Microsoft): You would expect that since there is more bandwidth at the top of the rack, that those would have more traffic on them, so when I take the log normal, those would be the heavy hitters. Did you see that? Answer: We just measured utilization, not bytes.

Changhoon Kim (Microsoft): What applications do they run in the data center? Answer: Messaging, search, video, email.

Kashi Vishwanath (Microsoft): How responsive are the models? What faith do we have that, if we changed the architecture, we would still see the corresponding traffic generated by your models, that are not simple extrapolations.? Answer: By changing parameters in our model we should be able to generate appropriate traffic.

Understanding TCP Incast Throughput Collapse in Datacenter Networks

Authors: Rean Griffith, Yanpei Chen, Junda Liu, Anthony Joseph, Randy Katz (RAD Lab, EECS Dept. UC Berkeley)
Presenter: Yanpei Chen

Summary:

This is a follow-on to the CMU work on incast; the initial intuitively derived fixes have limited applicability. This study shows that the incast problem is not solved. Incast is real, but can be masked by application inefficiencies; it is not observed everywhere. It is most visible in single-hop topologies with single bottleneck, e.g., DFS [distributed file systems?]. It affects key applications that have synchronization boundaries.

Non-TCP workarounds are possible, but they are problematic. CMU suggested reducing TCP's RTO-Min and using high resolution timers, but authors of this paper are convinced that these fixes are limited. They used N-to-1 transfer as a simple workload, with a fluid flow model with synchronized boundary considerations. They need to use real machines, because ns-2 models are not realistic enough (w.r.t. switch buffering) to generate the problem. They isolated the incast problem by creating a custom workload generator.

They confirmed previous work showing that one needs a high-resolution RTO, not just a small one. Different networks have different results, but when the SRTT is the same, they got the same results. They found that two data centers with the same setup have different results, due to different hardware versions of the switches. Turning off delayed ACKs gives more aggressive behavior, which is helpful for small SRTT, but harmful for large SRTT.

Randomizing RTO does not help, except in ns-2. They have not figured out why randomization works in ns-2 and not in a physical network.

An analytical model is hard due to overlapping effects. A model based solely on the effect of RTOs reproduces the shape of the curves they see. They have a systematic over-prediction because they we don't account for secondary effects, such as inter-packet wait time. A better TCP needs to deal with different networks, switches, and background traffic. We are working on adaptive CWND management.

Discussion:

[Janardhan Iyengar (Franklin and Marshall College)?] What do you mean by adaptive CWND management? Answer: [missing]

Albert Greenberg: Why does Microsoft not see incast? Have you found incast in real data centers? Answer: Yes.

Jad Naous (Stanford): Why are non-TCP implementations problematic? Do you mean things that don't exhibit TCP behavior? Answer: For example, you can use Ethernet flow control, but that breaks down beyond a single hop topology. Application scheduling is scary, too clumsy. Large switch buffers are expensive.

Srikanth Kandula (Microsoft): How big was your block size? Answer: 256KB fixed, 200us RTT [or 1 ms RTT? notes vary on this]. Variable is 1MB total across all senders.

Srikanth Kandula (Microsoft): With 100MB block, maximum of 10 concurrent senders with multihop, as at Microsoft, would we see incast? Answer: If your RTT is low across multihops, then you won't see incast.

Jeff: I've heard that Google believes it's a problem too.

Yanpei Chen (UC Berkeley) What about delay-sensitive TCPs like Vegas? Answer: Vegas also suffers from this problem, and there is a deployment issue with Vegas. There is so much engineering that has gone into optimizing New Reno and SACK, then when we go back to Vegas, everything is unoptimized.

Changhoon Kim (Microsoft): Since the problem is synchronization, won't 10GigE solves this? Answer: The problem will be worse. Once you get a large bandwidth delay product, the synchronization doesn't matter.

5. SESSION 4: DIAGNOSIS AND TESTING

Session chair: Albert Greenberg, Microsoft Research
Session scribe: Nathan Farrington

Change is hard: Adapting Dependency Graph Models For Unified Diagnosis in Wired/Wireless Networks

Authors: Lenin Ravindranath (MIT); Victor Bahl, Ranveer Chandra, David A. Maltz, Jitendra Padhye, Parveen Patel (Microsoft)
Presenter: Parveen Patel

Summary:

Jitendra Padhye presented a system called MnM that extends the Sherlock network fault diagnosis system (SIGCOMM 2007) to include both wired and wireless networks. The primary problem with existing fault diagnosis systems is that they are specialized for either wired networks or wireless networks, and the combination of both is insufficient to correctly diagnose problems. MnM works by running an MnM Agent program on each notebook computer, which monitors things such as network connectivity, and sends reports to a single MnM Inference Engine application. The inference engine analyzes these reports and infers both the locations of the notebooks as well as the likely causes of any network faults they are experiencing.

Discussion:

[missing]: How can you tell how much your system needs to be automated? Answer: We don't know which application, just that you had network access.

Jeff Mogul (HP Labs): You showed the CDF of the location accuracy, but what distance was good enough? Answer: It depends on the granularity of your recovery. You need accuracy in order to get priors for the system.

Peter Druschel (MPI): If these location priors turn out to be important, then it's an indication that the network folks should come and fix it. Answer: Yes, but you still need the priors because things don't stay static.

[Ramanchanra?]: How often do you not return the root cause? Answer: The system will always blame something, but it might be wrong.

Remote Network Labs: An On-Demand Network Cloud for Configuration Testing

Authors: Huan Liu, Dan Orban (Accenture Technology Labs)
Presenter: Huan Liu

Summary:

Huan Liu described a system called Remote Network Labs that is similar to Amazon EC2, but with routers instead of virtual machines. The goal is to eliminate the need to construct one's own dedicated and expensive network testbed, by leasing access to real physical routers distributed across the Internet. This allows expensive network equipment to be time-division multiplexed across users.

Discussion:

Nathan Farrington (UCSD): Does bandwidth cost more than the routers? Answer: We don't know.

Jad Naous (Stanford): Are there any race conditions due to the WAN separation? Answer: We can add artificial delays.

[??] (Washington Univ. St. Louis): How do you do sharing? Answer: Strict reservations. Is it easy to reboot the router to a clean state? Answer: Yes, you have console access.

Yanpei Chen (UC Berkeley): How fine-grained is the monitoring capability? Answer: Every packet is tunneled through the Internet. Clarify: The user can add additional equipment.

Jad Naous (Stanford): Why is it distributed? Answer: We couldn't put all the equipment in the same location.

Can you do power management? Can you take measurements? Answer: You could probably add a power meter and expose it as an interface.

Is there virtual router support? Answer: Juniper and Cisco routers support this.

When customers add their own equipment, do they share that with other customers? Answer: Not right now.

6. SESSION 5: DATA CENTER NETWORKS

Session chair: Aditya Akella, University of Wisconsin
Session scribes: Nathan Farrington

Diverter: A New Approach to Networking Within Virtualized Infrastructures

Authors: Aled Edwards, Anna Fischer, Antonio Lain (HP Laboratories, Bristol)
Presenter: Anna Fischer

Summary:

"Cloud computing: it's all about sharing." Anna Fischer presented a system called Diverter that allows users of cloud computing infrastructure to configure their own virtual networks and provides isolation between different users.

Discussion:

Jad Naous (Stanford): Is this an extension of trusted virtual domains? Answer: Yes. That work was more focused on security. That was L2 network virtualization.

Chuanxiong Guo (Microsoft): How do you map a virtual MAC to a physical MAC? Answer: It's based on the virtual IP addresses. It's discovered by using ARP, and the machine replies with its physical MAC.

Aditya Akella (Wisconsin): Did you look at the requirements from commercial cloud offerings? Answer: Microsoft Azure is doing similar stuff. Amazon is L2 and doesn't provide much isolation or bandwidth guarantees.

Sunay Tripathi (Sun): You have a large L2 network and want it to be flat, do you see problems with broadcasts? Answer: We only use broadcast for a very few cases, and we can leverage technologies such as PortLand.

Changhoon Kim (Microsoft): You mentioned something about routing VMs, why do you want your own routing VMs? Answer: To allow users to configure them. Physical routers are hard to configure.

Why should we integrate services, servers, and networking in a Data Center?

Authors: Paolo Costa, Thomas Zahn, Antony Rowstron, Greg O'Shea (Microsoft Research Cambridge); Simon Schubert (EPFL)
Presenter: Paolo Costa

Summary:

Paolo Costa presented a new data center network architecture, called Borg, that uses a k-ary 3-cube topology, in which each server connects directly to its 6 closest neighbors (N,E,S,W,Up,Down). Since this is a direct network, there are no dedicated switches. The goal of Borg is to determine if some applications can run better on a direct topology where the servers are also responsible for routing, than on a traditional indirect topology such as a fat tree, where the switches do not have knowledge of the actual applications or traffic patterns.

Discussion:

Chuanxiong Guo (Microsoft): Are you saying the networking and the distributed services should be combined seamlessly? Could the services be separated from the networking part? Answer: [missing[

Nathan Farrington (UCSD): Could we run this on a fat tree? Answer: Some aspects are topology specific.

Jad Naous (Stanford): Can you comment on the topology? Answer: We run our software in user mode and do software routing. But software routing may not be the way to go. We are thinking about how to offload the routing onto the NIC. MSR Asia was able to use the NetFPGA for this.

Jeff Mogul (HP Labs): It's hard to divide up a square or cubical mesh among multiple applications. Is that going to be a problem when you try to do multi-tenancy on a very large installation? Answer: We are still exploring this.

Aditya Akella (Wisconsin): If you had to turn off servers, then what would happen? Answer: You can power down the server and leave the NIC running.

Jad Naous (Stanford): So you will have a dedicated switch on each server just to do routing? Isn't this the same as putting a switch outside the server? Why not use a Fat Tree or BCube? Answer: Cube has simpler cabling complexity and more fault tolerance.