conference reports

THANKS TO OUR SUMMARIZERS

NSDI '07: 4th USENIX Symposium on Networked Systems Design & Implementation

Cambridge, MA
April 11–13, 2007

KEYNOTE ADDRESS

Security of Voting Systems

Ronald L. Rivest, Viterbi Professor of Computer Science, Massachusetts Institute of Technology

Summarized by Soila Pertet (spertet@ece.cmu.edu)

Voting systems should provide end-to-end security where voters can verify that their vote was cast as intended and counted as cast. However, end-to-end security is complicated by the need to maintain voter privacy while ensuring verifiability.

Ronald Rivest presented several approaches for providing end-to-end security, even in the absence of trusted computing platforms. Each approach relied on voter receipts and public bulletin boards. Voters receive a receipt when they cast their vote; this receipt does not reveal how they voted. All ballots get posted to a public bulletin board, and voters can use their receipt to protest if their vote was not cast as intended.

Rivest described two cryptographic approaches for designing secure voting systems that preserved voter privacy: mixnets (Chaum) and public mixing (Adida). These approaches randomly permute encrypted ballots so that the ballots cannot be correlated back to the voters. Rivest also presented a three-ballot scheme that did not rely on cryptography. In this scheme, each voter casts three ballots with at least one but no more than two votes for each candidate. The voter keeps an arbitrary copy of one of the ballots as a receipt, and all three ballots are posted to the public bulletin board.

Rivest briefly discussed Internet voting. In his opinion, Internet voting is vote-by-mail made worse, because voters are denied an enforced moment of privacy to cast their vote free of coercion or bribery. This problem is compounded by the fact that about one in every four PCs belongs to a botnet.

Rivest also mentioned that security is not all about technology; election officials and poll workers are an important part of the process. He recommended that the audience members get more involved in the election process, for instance by volunteering to be poll workers.

Many attendees asked questions. How can ordinary users (who might not understand cryptography) gain confidence in voting systems? Rivest answered that they can gain confidence in the voting system through indirect verification, where an expert of their choice examines the publicly available source code and cryptographic algorithms for the voting system. Can you cast your vote using virtual technology, for example, have your vote initially cast on a USB key at home? To do this you would need a trusted electronic agent that can represent you (e.g., a cell phone). There is still a lot of research that needs to be done before we get to this point: Building trusted computing platforms is a hard problem. Another attendee pointed out that there is a big gap between the ideas proposed in academia and what is being done in practice, so how can we bridge this gap? Rivest answered that we can bridge this gap by getting involved (e.g., volunteering to be a poll worker) or working with the state representatives to raise the standard. What level of tamper protection do the cryptographic schemes provide? Do they simply detect security violations or can they recover from the violations once detected? It is possible to recover from security violations; for example, you can replace a cheating mix server if the proofs do not match, and then redo the computation.

CONTENT DELIVERY

Summarized by Anupama Biswas (anupamabiswas@gmail.com)

- ***Do Incentives Build Robustness in BitTorrent?***

Michael Piatek, Tomas Isdal, Thomas Anderson, and Arvind Krishnamurthy, University of Washington; Arun Venkataramani, University of Massachusetts

Awarded Best Student Paper!

Michael explained that the mechanism used by BitTorrent functions tit-for-tat for reciprocating with clients. He proved this statement by using a client BitTyrant that replaces the strategy used by BitTorrent. The BitTorrent strategy does not eventually lead to improvement in performance. BitTyrant responds to those clients that do not make altruistic contributions by degrading performance. Also, robustness in BitTorrent is not due to incentives. BitTorrent does not address the problem of performance degradation if the peers strategically manipulate the system.

The key idea for BitTyrant is to carefully select peers and contribution rates so as to maximize download rates per unit of upload bandwidth. The strategic behavior of BitTyrant is executed simply through policy modifications to existing clients without any change to the BitTorrent protocol. Michael showed the performance of BitTyrant, evaluated on real swarms, establishing that all peers, regardless of upload capacity, can significantly improve download performance while reducing upload contributions. Also, the performance is affected as peers individually benefit from BitTyrant's strategic behavior, irrespective of whether or not other peers are using BitTyrant. Peers not using BitTyrant can experience degraded performance owing to the absence of altruistic contributions. Taken together, these results suggest that incentives do not build robustness in BitTorrent. In addition to the primary contribution, BitTyrant, the efforts to measure and model altruism in BitTorrent are independently noteworthy. First, the model used is simpler and is still sufficient to capture the correlation between upload and download rates for real swarms. Second, existing studies recognizing altruism in BitTorrent consider small simulated settings or few swarms that poorly capture the diversity of deployed BitTorrent clients, peer capacities, churn, and network conditions. One of the questions asked concerned the possibility that a hacker might just give the impression of having a higher upload time. This might lead to it being assigned more bandwidth than the other peers. Michael said that this situation has been handled successfully by BitTyrant.

- ***Exploiting Similarity for Multi-Source Downloads Using File Handprints***

Himabindu Pucha, Purdue University; David G. Andersen, Carnegie Mellon University; Michael Kaminsky, Intel Research Pittsburgh

Bindu Pucha presented a new approach for downloading similar data from multiple sources using a technique called File Handprints. The approach presented for downloading a specific file is a mix of the existing approaches. Currently approaches such as BitTorrent try to locate an exact copy of the object the user is looking for but do not look for the desired object in similar sources. Another approach is looking for chunks of the data object in multiple sources, which involves performing a number of lookups. This leads to limiting the scalability of the system. The approach presented locates similar objects using a constant number of lookups and inserting a constant number of mappings per object. The handprinting is similar to shingling, fingerprinting, and deterministic sampling. It uses the technique of exploitable similarity and not document resemblance. When performance was checked against the performance with BitTorrent, it was found to exceed the performance of BitTorrent in locating a file from multiple resources. The download time was faster for a given P2P connection.

- ***Cobra: Content-based Filtering and Aggregation of Blogs and RSS Feeds***

Ian Rose, Rohan Murty, Peter Pietzuch, Jonathan Ledlie, Mema Roussopoulos, and Matt Welsh, Harvard University

Blogs and RSS feeds are becoming increasingly popular. However, the problem lies in finding and tracking interesting content in blogs, which currently is a cumbersome process. A solution such as providing the users with the ability to perform content-based filtering and aggregation across the millions of available Web feeds, obtaining a personalized feed containing the articles of a user's interest, will be of real value. Also providing real-time updates of articles of interest helps the user to avoid having to keep tabs on a multitude of interesting sites. Ian provided such a solution in the paper. The blog search sites available do not present a clear idea as to how well these sites scale to handle large number of feeds and users and also provide low time delay for the searches. The contents of the various blogs keeps on changing, and it is not known how fast the current Web searching techniques can search such blogs with minimum time delay. Similar content posted on various blogs, which requires the user to search through multiple blogs, can be aggregated using SharpReader and FeedDemon, both of which collect stories from multiple sites along thematic lines (e.g., news or sports). A single RSS feed looks into a individual blog. It does not aggregate the data with similar content.

He presented Cobra (Content-Based RSS Aggregator), a distributed scalable system that provides personalized views of articles to users taken from potentially millions of RSS feeds. The information is collected by the system that crawls, filters, and aggregates vast numbers of RSS feeds. It delivers to each user a personalized feed based on the user's interests. Cobra consists of a three-tiered network of crawlers, filters, and reflectors. Crawlers scan the Web feeds. Filters match the crawled Web feeds to user subscriptions, and reflectors provide recently matching articles on each subscription as an RSS feed, which can be browsed using a standard RSS reader. This system is capable of handling a large number of source feeds and users, keeping the latency time low.

OVERLAYS AND MULTICAST

Summarized by Murtaza Motiwala (murtaza@cc.gatech.edu)

- ***Information Slicing: Anonymity Using Unreliable Overlays***

Sachin Katti, Jeff Cohen, and Dina Katabi, Massachusetts Institute of Technology

Sachin Katti presented a new technique for distributing content anonymously in overlays without keys. Although Freenet allows anonymous distribution of content, it has very few users, since it relies on exchange of keys. Overlays are ideal for anonymous content distribution; however, they cannot be used as is since they don't involve any public keys and are not reliable owing to the large degree of node churn (nodes joining and leaving the overlay).

Information slicing achieves each of these objectives by splitting the original message and sending the slices on disjoint paths between the source and the destination. The technique presented has the key feature that only the destination gets all the pieces and is able to decode the original message, while none of the intermediate nodes gets the complete message. The anonymity of the sender and receiver is achieved by letting each node know only its next hop. Also, information slicing is able to deal with the node churn in overlays by adding redundancy using network coding to deal with loss of data resulting from the loss of a node on the path.

The authors evaluated their technique using simulation as well as on PlanetLab. Sachin presented the evaluation of anonymity, churn resiliency, and throughput performance in the talk.

Anonymity was evaluated by using an overlay of 10,000 nodes in which attackers can control nodes, snoop traffic, and collude. Entropy was used as the metric to determine the amount of information leaked to the attackers. The anonymity of the scheme was found to be comparable to Chaum mix, despite its having no keys.

In order to measure the resiliency of information slicing to node churn in the overlays, the authors compared it to onion routing with source coding. The results showed that information slicing had a much better resiliency to churn compared to onion routing. The throughput achieved by information slicing was also found to be much better than onion routing, because information slicing tries to use parallel paths to the destination. The results from the evaluation on PlanetLab were found to match the results from simulation.

There was a question on how link disjointedness was achieved, to which Sachin replied that the scheme required the source to be smart about picking paths in the overlay to get disjoint paths, for example by choosing nodes in different ASes. To a comment about anonymity decreasing with churn, Sachin noted that there was a tradeoff between increasing redundancy to protect against higher churn and having lesser anonymity. There was also a question on what the authors considered to be the throughput to which the response was that the throughput was the one observed at the destination. Also, on the subject of how information slicing can be incorporated in current P2P applications, the answer was to create a list of users who are interested in using information slicing and then choose paths using those nodes.

- ***SAAR: A Shared Control Plane for Overlay Multicast***

Animesh Nandi, Rice University and Max Planck Institute for Software Systems; Aditya Ganjam, Carnegie Mellon University; Peter Druschel, Max Planck Institute for Software Systems; T.S. Eugene Ng, Rice University; Ion Stoica, University of California, Berkeley; Hui Zhang, Carnegie Mellon University; Bobby Bhattacharjee, University of Maryland

Animesh used the example of an overlay that is optimized for data dissemination and in which a control overlay is used to build and repair the overlay and a gossip protocol is used to disseminate membership information and nodes use probes to select a parent based on the metrics delay and available bandwidth. He quickly noted that such an approach will not scale in the case of large groups and high membership churn. He then proposed the idea of using separate control and data overlays, where the control overlay can be shared among different data overlays.

The control overlay in their architecture uses the anycast primitive for selecting overlays, which Animesh noted is a key factor in building efficient overlays. The nodes also keep aggregated information of metrics such as spare bandwidth capacity and depth. This helps in quickly pruning the trees without doing an in-depth first search when an anycast request comes in.

SAAR was evaluated on Modelnet using about 350 nodes with different available bandwidths. The evaluation showed that SAAR enables low join delays. The experiments showed that the initial SAAR delay is high; also, in the case of a single tree 99% of the delay was less than 2 seconds. The evaluation also showed that SAAR provided good streaming quality as compared to ESM, whose quality was poor. SAAR performed much better for the single- and multiple-tree cases.

To the question of whether SAAR introduced a single point of failure and how secure it was, the authors responded that they had not addressed freeloading and malicious behavior in the control overlay; however, mechanisms in structured overlays may be applied here to counter those.

- ***Ricochet: Lateral Error Correction for Time-Critical Multicast***

  Mahesh Balakrishnan and Ken Birman, Cornell University; Amar Phanishayee, Carnegie Mellon University; Stefan Pleisch, Cornell University

Mahesh Balakrishnan presented Ricochet, which is used to provide a time-critical reliable multicast in data centers. In a data center, it is common for a node to subscribe to multiple multicast groups. This could lead to situations in which there is a high data rate at some nodes, leading to overload and eventual dropping of packets. The challenge in such systems is to recover packets in real time and the solution must scale in the number of receivers, senders, and multicast groups. After observing several existing multicast solutions for data centers the authors observed that the latency (i.e., the time taken to recover from lost packets) is inversely proportional to the data rate.

Although Forward Error Correction (FEC) can be used to provide reliability guarantees with no retransmissions, the node has to wait for “r” data packets before generating the FEC and hence again the latency is inversely proportional to the data rate. The authors propose Lateral Error Correction (LEC), a new reliability mechanism to allow packet recovery latency to be independent of per-group data rate. In LEC, a node exchanges XORs of incoming data packets with c randomly chosen receivers. The protocol scales, as it is gossip style and has tunable per-group overhead.

Mahesh noted that the bandwidth overhead of Ricochet is proportional to the additional number of packets used for error correction. Also, the computation overhead is small, since XORs are very fast to compute (150 to 300 microseconds/packet). Also, the number of intersections between the various multicast groups is not exponential, as it is limited by the actual number of nodes in the system.

The authors evaluated Ricochet on a 64-node cluster using three packet-loss models (uniform, burst, and Markov). The evaluation showed that most lost packets were recovered in 50 milliseconds. Also, Ricochet was found to scale to hundreds of multicast groups and was about 400 times faster than SRM. The evaluation showed that Ricochet was resilient to bursty losses as well and could handle short bursts of 5 to 10 packets well. To handle even higher bursts, the authors used staggering in Ricochet. Amazingly, with a stagger of 6, Ricochet can recover 90% of packets from a burst loss of 100 packets.

There was a question on how Ricochet might work in wide area networks and not clusters. Mahesh replied that for a wide area network the losses might not be independent and the nodes might have to use an intelligent way to pick the people to talk to. Ricochet is available for download from http://www.cs.cornell.edu/projects/quicksilver/Ricochet.html.

WIRELESS

Summarized by Murtaza Motiwala (murtaza@cc.gatech.edu)

- ***WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks***

  Rabin Patra and Sergiu Nedevschi, University of California, Berkeley, and Intel Research, Berkeley; Sonesh Surana, University of California, Berkeley; Anmol Sheth, University of Colorado, Boulder; Lakshminarayanan Subramanian, New York University; Eric Brewer, University of California, Berkeley, and Intel Research, Berkeley

Rabin Patra presented the motivation, design, and implementation of WiFi-based Long Distance (WiLD) networks in developing regions. WiLD uses 802.11 radios because they are low-cost, have no spectrum costs, and have good data rates. At present, WiLD has been deployed in a number of places in the developing regions, including in India at Arvind Hospital (12 clinics approximately 15 km apart) and in Ghana. From their experience, they found that the point-to-point performance of WiFi over long distances is poor; for example, on a 60-km link the performance of TCP is 0.6 Mbps vs. 6 Mbps for UDP.

The design of WiLD is focused on fixing 802.11 by replacing CSMA with TDMA and enforcing synchronization on multiple links to avoid collision losses. The design has the constraints of not involving any hardware changes and not permitting any modification of end hosts through modification of WiLD routers. Also, the routers are inexpensive and thus have low processing power. Rabin explained that the problems with using 802.11 over long distances are with ACKs, as they are inefficient over long links and also the ACK timeouts are very short compared to the delay over a long link (~110 km). Also, higher propagation delay increases the likelihood of collisions at the receiver end. The authors thus made the choice of using sliding window flow control at the MAC layer and disabled 802.11 MAC ACKs completely. They also enabled simultaneous sends and simultaneous receives by providing a 12-dB isolation. For recovering from losses, WiLD uses bulk ACKs or adaptive FEC. For using either of the techniques there is a tradeoff between bandwidth and delay; thus for bandwidth-sensitive protocols they used bulk ACKs, whereas for delay-sensitive ones they used FEC.

The authors implemented WiLD by modifying the Atheros madwifi driver, and they used the Click router to perform FEC encoding and decoding. Their evaluation showed that for a single-hop case, WiLDNet's performance increases with increase in distance. Also, in the multihop case, WiLDNet is more spectrum-efficient than traditional 802.11. Rabin noted that their future work was to look into remote network management and planning for WiLDNet.

There was a question regarding what kind of traffic WiLDNet was intending to optimize. Rabin responded that they were looking at a mix of applications including images, video conferencing, and Web browsing. As to whether the authors had evaluated WiLDNet against available wireless solutions such as WiMAX, Rabin replied that they had not done so, since they had a strict cost factor in mind.

- ***S4: Small State and Small Stretch Routing Protocol for Large Wireless Sensor Networks***

  Yun Mao, University of Pennsylvania; Feng Wang, Lili Qiu, and Simon S. Lam, The University of Texas at Austin; Jonathan M. Smith, University of Pennsylvania

Yun Mao presented S4, a routing protocol for large wireless sensor networks that achieves small state and small stretch. Yun noted that there are numerous challenges in coming up with a point-to-point routing protocol for wireless sensor networks, owing to limited resources and to the RF phenomenon. Also, there is a huge debate going in the community on whether the routing protocol should have a small state or small stretch (path length compared to the optimal path length). Yun noted that for wireless sensor networks, state and stretch were related, as routing protocols, which aim at providing low stretch, invariably require more state. Yun noted that shortest path routing gives optimal stretch but requires $O(n)$ state whereas hierarchical routing, which requires only O[square_root(n)] state, gives paths with large stretch. Other proposals, such as geographical routing and virtual coordinate routing, are also unable to avoid the state vs. stretch tradeoff.

S4 uses the theoretical ideas from the compact routing algorithm to achieve a small state (i.e., of O[square_root(n)]) with a constant bound on the worst-case stretch of 3. S4 uses two types of nodes: beacon nodes and regular nodes. The beacon nodes are O[square_root(n)] in number and know how to route to the regular nodes close to it (cluster), whereas each of the regular nodes knows how to reach the beacons. S4 uses two rules to route in the network: Inside a cluster, route using the shortest path; outside a cluster, route to the beacon closest to the destination node. The challenges facing S4 are to ensure that no flooding takes place and that each node maintains its routing state for reaching the beacons and to provide resiliency to link and node failures.

Yun presented the evaluation of S4 using high-level simulations with no loss and reliable nodes, TOSSIM packet-level simulations using lossy links, and the mica2 testbed of 42 nodes. The authors used the Beacon Vector Routing (BVR) protocol from NSDI '05 as the benchmark for comparison with S4. The results of the simulation and evaluation showed that S4 had smaller average stretch and variation compared to BVR. S4 also achieves smaller state and is unaffected by obstacles in the paths.

There was a question regarding the complexity of the code and the amount of memory required by the code itself. Yun replied that the complexity of the code for S4 was similar to that for BVR; however, they didn't have any actual numbers. There was also an inquiry on how the beacons were placed and if there was an intelligent method for placing them. As to whether there were any real applications that used 4,000 or more sensor nodes, Yun replied that there were no such applications at present, but he said they might be seen in the near future.

The software can be found at http://www.cs.utexas.edu/~lili/projects/s4.htm.

- ***A Location-Based Management System for Enterprise Wireless LANs***

  Ranveer Chandra, Jitendra Padhye, Alec Wolman, and Brian Zill, Microsoft Research

Jitendra Padhye presented the design and evaluation of a management system for wireless LANs. The system has been deployed and is in use on one of the floors in the Microsoft building. Although there has been lot of work in the area of managing wireless LANs, they each have their shortcomings: Some cannot cover the area properly, whereas others are too expensive to deploy or are not scalable.

The wireless management system presented was deployed on the DAIR platform, which consists of attaching air monitors to ordinary desktops in offices. These air monitors are used to collect wireless data and send it to a central database. DAIR has several advantages: Since desktops are ubiquitous in practically every office, this leads to a dense deployment with no extra cost. It is also robust, as desktops are always up. Also, such a scheme can use very simple algorithms for determining the location of the wireless nodes, measuring the quality of the wireless signal, etc. Furthermore, storing the data collected from the sensors in a central place allows historical analysis of data. In the talk, Jitendra presented the use of DAIR in estimating the transmission rate obtained by wireless clients at various locations on the office floor.

The aim of DAIR is to locate the clients at the granularity of an office on the floor and, since the air monitors were deployed on the desktops in the offices, it was a simple task to determine to which office the air monitor belonged. Also, to pinpoint the location of the client at the granularity of an office only required the use of simple algorithms, such as choosing the air monitor receiving the strongest signal or using the spring-and-ball method or the strongest AM method. The air monitors also intercept the packets from the clients to the nearest AP. For each conversation between the client and the AP, the system simply chooses for analysis the data from the air monitor that received the highest number of packets for that conversation. The authors were able to detect an AP that was not functioning properly using this system and were able to alert the network operators and get the AP fixed. The system also detected an area of poor coverage in the office and reported it to the network operators.

Finally, Jitendra noted that the attachment of air monitors on the desktops causes an additional load of about 2% to 3% on the desktop and contributes additional traffic of less than 10 kbps. To the question of whether the authors evaluated any other technique besides choosing the data from the air monitor that received the highest number of data packets, Jitendra replied that they did not, since they didn't see much packet loss with their technique of using the one that saw the highest number of packets. There was also a question regarding interaction between floors, to which Jitendra replied that they did investigate that possibility and that it was easy to detect with their scheme if the client was connecting to an AP on another floor, although their evaluation was concentrated on a single floor. Hari Balakrishnan asked how their techniques compared to the ones used by cellular companies. Jitendra explained that in a cellular network, if the cell phones themselves are used as the monitors, there is no way of telling where the cell phone (mobile client) is located.

POSTER SESSION

Summarized by Eric Eide eeide@cs.utah.edu

- ***Distributed Data Management for Storage-centric Sensor Networks***

Devesh Agrawal, Gal Niv, Gaurav Mathur, Tingxin Yan, Deepak Ganesan, Prashant Shenoy, and Yanlei Diao, University of Massachusetts, Amherst

Devesh summarized StonesDB, a database system for wireless sensor networks. Because emerging sensor network devices have increasingly high-capacity and energy-efficient flash memories, StonesDB stores collected data on the nodes of a sensor network. The complete StonesDB system has two levels. The lower layer consists of the sensor network nodes, each of which runs a local database. The upper layer, which receives and routes user queries to appropriate sensor nodes, consists of resource-rich proxies that implement distributed data management services atop the sensor network. The StonesDB architecture is intended to optimize energy efficiency at its sensor nodes, and this requires careful and novel implementations of many database components. A second challenge lies in designing StonesDB to support a variety of sensor network platforms with varying resource constraints, which requires a corresponding variety of design tradeoffs and optimization points.

- ***Lightweight OS Support for a Scalable and Robust Virtual Network Infrastructure***

Sapan Bhatia, Marc Fiuczynski, Andy Bavier, and Larry Peterson, Princeton University

Sapan presented recent work on VINI, a virtual network infrastructure designed for evaluating new protocols and services. VINI seeks to offer a high-performance network testbed, one that supports many concurrent experiments and that also provides each experiment with fine control over issues such as routing and traffic shaping. Such control requires OS support that is not found in testbeds such as PlanetLab, which isolates experiments from each other via Linux VServers. Sapan's poster focused on two areas of work that allow VINI to overcome the VServer "virtualization barrier." The first was more complete virtualization of the Linux network stack and the initial deployment of this OS work onto VINI. The second was a secure bootstrap system for the VINI control plane, which helps to address security concerns that arise from VINI's virtualized network stacks.

- ***Real Time-Sharing in Emulab through Preemption and Stateful Swapout***

Anton Burtsev, Prashanth Radhakrishnan, Mike Hibler, and Jay Lepreau, University of Utah

Anton and Prashanth presented work that allows experiments within the Emulab network testbed to be "swapped out" without losing state. An Emulab experiment is analogous to a UNIX process: Its resources include a set of testbed-managed hosts, and its state includes the contents of the memory and disks on those hosts. Currently, when the resources for an experiment are released, the contents of node-local memories and disks are lost. Thus, experiments are normally swapped out only when they are complete. Anton and Prashanth are working to preserve node-local state, however, which will allow Emulab experiments to be swapped out (and back in) more freely. Their goal is for such transitions to be transparent to the software that executes within the experiment and, except for scheduling delays, transparent to users of the testbed as well. The poster outlined the many challenges of stateful swapout as well as their current solutions.

- *The Case for Conditional Link Metrics and Routing*

  Saumitra M. Das, Purdue University; Yunnan Wu and Ranveer Chandra, Microsoft Research, Redmond; Y. Charlie Hu, Purdue University

Saumitra presented this poster, which described the need for and potential benefits of "conditional link metrics." A conditional metric is one that varies according to context: For example, the cost of sending a packet over a given link may depend on the set of links that the packet has already traversed. Such metrics provide ways to express interdependencies within a network. Among other examples, Saumitra proposed using a conditional metric to capture the dependency between network coding and routing. Network coding is a link-layer technique that can reduce wireless transmissions, but the opportunities for network coding depend on traffic patterns and hence depend on routing decisions. Saumitra and his colleagues have implemented two systems that utilize conditional metrics to help practical network coding and multiradio networks, with promising results.

- *Residential Broadband Networks: Characteristics and Implications*

  Marcel Dischinger, Andreas Haeberlen, and Krishna P. Gummadi, Max Planck Institute for Software Systems; Stefan Saroiu, University of Toronto

Marcel presented current work in the measurement of residential broadband networks. Such networks are increasingly important for emerging Internet applications, such as VoIP and P2P systems, but there is little data that characterizes these networks at scale. Marcel and his colleagues obtained measurements from 1,500 residential broadband hosts spread over 11 major cable and DSL ISPs. These measurements were acquired through probe trains and required no cooperation from the measured hosts. The poster summarized the collected data, which shows that residential networks are often quite different from academic networks. For instance, the data shows that some ISPs allow short bursts of traffic—perhaps to speed up Web page downloads—but significantly rate-limits large flows.

- *Characterizing and Replaying Proprietary Workloads*

  Archana Ganapathi, Armando Fox, and David Patterson, University of California, Berkeley

Archana described the problems faced by developers who must predict or test the behavior of a networked system. For example, the maintainers of a commercial Web service may need accurate models of their system to estimate future resource demands, but generally they have insufficient tools and test resources. Academic researchers, in contrast, are often interested in analyzing production systems but are unable to obtain actual application traces from companies. Archana described an approach for solving both problems: Use machine-learning techniques to generate artificial but realistic workloads that drive systems into desired behaviors. The poster summarized two tools in development: AWE-Gen, which generates synthetic workloads from actual trace data, and AWE-Sim, which replays the workload against the target system, thereby creating the desired system conditions.

- *A Deductive Framework for Programming Sensor Networks*

  Himanshu Gupta and Xianjin Zhu, Stony Brook University

Xianjin's poster described a novel programming methodology for wireless sensor networks, one based on logic programming. Many sensor network applications are designed to collect facts about the world and process queries against those facts. This design closely matches the main structuring principles of logic programming languages, suggesting that logic programming can be a good fit for sensor network computing. A user-written logic program specifies behaviors in a declarative and high-level fashion. Ideally, these programs can be automatically compiled into distributed and resource-efficient code for the nodes within a sensor network. The goal of Xianjin's research is to design and implement the framework that makes this vision a reality. In particular, this work requires new and energy-efficient techniques for evaluating logical "joins": the process of searching for data that is needed in order to satisfy one or more queries.

- *Network Troubleshooting: An In-band Approach*

  Murtaza Motiwala, Georgia Institute of Technology; Andy Bavier, Princeton University; Nick Feamster, Georgia Institute of Technology

Murtaza presented Orchid, an in-band network path diagnosis system for locating faults in a packet-based network. Most network diagnosis tools produce their own network traffic to locate faults. This traffic is out-of-band with respect to normal application traffic, and, as a result, it can fail to detect a variety of faults that affect application traffic. Orchid, however, inserts a diagnostic header into the packets that carry application data. When a flow begins, Orchid transmits a probe packet to record the addresses of routers along the flow's network path. After this, routers along the path use the Orchid header within data packets to record faults. Orchid-enabled routers need only a small amount of state (a single counter) per active flow. Experiments show that the current Orchid prototype, implemented with Click and deployed on PL-VINI, can accurately diagnose many faults with only small network overhead.

- ***Amazon S3 for Science Grids: A Viable Solution?***

  Mayur Palankar, Ayodele Onibokun, and Adriana Iamnitchi, University of South Florida; Matei Ripeanu, University of British Columbia

Amazon's Simple Storage Service (S3) offers pay-as-you-go online storage, and as such, it provides an alternative to in-house mass storage. In this poster, Mayur and his colleagues evaluated S3 as a storage facility for the DZero Experiment, an international high-energy physics collaboration. Traces from the DZero community over 27 months show that 560 users worldwide transferred 5.2 PB through DZero. Mayur and his colleagues characterized S3: They observed availability and data access performance, and they evaluated the feasibility, performance, and costs of a hypothetical S3-supported DZero collaboration. They concluded that S3 could be a viable storage system for DZero in terms of availability and performance, but that it would be expensive—in excess of $1.1 million per year. Costs could be reduced by using BitTorrent along with S3, exploiting data usage and application characteristics to improve performance. Finally, Mayur noted that S3's current security architecture is inadequate for science collaborations such as DZero in terms of access control, support for delegation and auditing, and built-in assumptions of trust.

- ***XMon-BGP: Securing BGP Using External Security Monitors***

  Patrick Reynolds, Oliver Kenney, Emin Gün Sirer, and Fred B. Schneider, Cornell University

Patrick presented a low-cost and incrementally deployable way of securing the Border Gateway Protocol (BGP). BGP connects autonomous systems within the Internet: It constitutes critical infrastructure but has well-known security problems. Previous attempts to secure BGP, entailing new routers or extensive modifications to router operation, have not been widely deployed. XMon-BGP proposes to secure legacy routers that employ (unsecured) BGP by deploying external security monitors (XMons). An XMon examines traffic to and from a legacy device and checks it for conformance against a security specification. It can thus protect against compromised routers, misconfigurations, and even insider attacks. Multiple XMons can communicate via an overlay to compensate for autonomous systems that have not deployed an XMon. The XMon software runs on a trustworthy computing platform (Nexus, the subject of a companion poster) that can vouch for the correctness of the XMon outputs. Experiments showed that XMon-BGP works well, that it has no trouble keeping up with BGP traffic, and that a relatively small deployment of XMon-BGP could secure a majority of all Internet routes.

- ***Nexus: A New Operating System for Building Trustworthy Applications***

  Alan Shieh, Dan Williams, Kevin Walsh, Oliver Kennedy, Patrick Reynolds, Emin Gün Sirer, and Fred B. Schneider, Cornell University

Dan described the design and implementation of Nexus, a new operating system for trustworthy computing. Traditional operating systems lack abstractions and mechanisms for using increasingly available secure coprocessor hardware. In contrast, Nexus leverages such hardware to support trustworthy applications that have strong behavioral guarantees, without restricting users to particular software applications. One of the new mechanisms in Nexus is "active attestation," which securely captures properties of software components. This can be used for both local and remote access control. Dan complemented his poster with a live demonstration of applications on Nexus. A media server checked active attestation labels on requests to ensure that the media players (i.e., clients) would not leak the content to disk. Active attestation labels also distinguished user-keyboarded messages from script-generated spam. Finally, Dan demonstrated Nexus's support for legacy applications by running Linux, X Windows, and various applications such as Firefox and Thunderbird on Linux atop Nexus.

- ***The SPINDLE Disruption Tolerant Networking Project***

  Christopher Small, Rajesh Krishnan, and the members of the SPINDLE team, BBN Technologies

Christopher presented the SPINDLE project, which is developing new technologies for disruption-tolerant networks (DTNs). Commonplace networks, such as those based on TCP/IP, require stable end-to-end paths in order to operate. To relay messages, there must be a complete path from source to destination (and back). In contrast, a disruption-tolerant network supports reliable communication in more hostile and poorly connected environments. The SPINDLE project is developing new routing algorithms for DTNs that take disconnection and link discovery into account. In addition, they are developing techniques that use caching, distributed indexing, and content-based data retrieval to improve access to data in the face of network disconnections. An application can use a declarative specification to describe the routing, resource management, and other policies that a DTN should use in handling its data.

- ***Scaling Full-Mesh Overlay Routing***

  David Sontag, Massachusetts Institute of Technology; Amar Phanishayee and David Andersen, Carnegie Mellon University; David Karger, Massachusetts Institute of Technology

David Sontag described recent work that improves the scalability of routing in one-hop overlay networks. Routing in existing overlay networks, such as RON, scales poorly because every node communicates with every other node.

David presented a new algorithm that scales much better while still supporting best one-hop routing over the complete network. In the new algorithm, every node measures the paths to all of its neighbors, but it sends that information only to a subset of the other nodes in the network. The trick is that these subsets are chosen so that, for any two nodes A and B in the network, there is at least one node C that receives the neighbor data for both A and B. Thus, the common node C can tell A and B about the best path between A and B. David and his colleagues are now evaluating the effectiveness of their new algorithm in the RON testbed. Among other tasks, they are investigating the resilience of the new algorithm to node and link failures.

- ***Efficient Cooperative Backup on Social Networks***

Dinh Nguyen Tran and Jinyang Li, New York University

Dinh presented BlockParty, an online backup system for cooperating groups of friends. The benefits of online backups are well known, and P2P networks provide convenient, online, and physically distributed storage. However, implementing a backup system within a traditional P2P network is difficult because of node churn, misaligned incentives, and ill-suited models of trust. BlockParty therefore allows each user to specify the other users with which he or she will cooperate. In practice, users choose to cooperate if they are also real-world friends. In comparison to other P2P systems, BlockParty has limited choices for storing data. Therefore, a primary concern of BlockParty is efficient utilization of disk space. Dinh described their scheme for coding data blocks, which saves spaces on BlockParty hosts. Finally, although BlockParty users trust each other not to deny service, each node periodically (and efficiently) verifies that its neighbors hold the expected backup data. If loss is detected, BlockParty undertakes repairs. The project software is available at http://www.news.cs.nyu.edu/friendstore/.

TOLERATING FAULTS AND MISBEHAVIOR

Summarized by Yun Mao (maoy@cis.upenn.edu)

- ***Beyond One-Third Faulty Replicas in Byzantine Fault Tolerant Systems***

Jinyuan Li, VMware, Inc.; David Mazières, Stanford University

Jinyuan Li stated that a Byzantine fault tolerant (BFT) system will behave correctly when no more than f out of $3f + 1$ replicas fail. In particular, BFT aspires to two properties: consistency (or safety), meaning all operations execute as if they are sequentially conducted on replicated state machines, and liveness, meaning protocols can make progress even with malicious replicas. When an attacker controls more than f failures, in a traditional BFT system, the system behavior is totally unexpected. Jinyuan argued that there is a large space between complete correctness and arbitrary failures. He first introduced the fork consistency, which is a relaxation of the linear consistency. He used a card-swipe access control service as the application to demonstrate that fork consistency is useful because it leaves replicas or clients with "forked views," and the misbehavior will eventually be revealed via out-of-band communication. The misbehavior that is nonerasable can be used as proof of attack. Jinyuan then showed that it is possible to still achieve fork consistency when no more than $2f$ failures happen.

The BFT2F protocol is based on the PBFT protocol. The intuition is that each replica keeps its execution history and the client waits for $2f + 1$ matching replies instead of $f + 1$ in PBFT. The problem of achieving fork consistency is that the protocol has to be a two-round protocol, and the liveness property is sacrificed if clients crash between rounds. The communication overhead is also not negligible. To avoid using a two-round protocol, Jinyuan said it is necessary to further relax the consistency guarantee to a fork * consistency. In a fork * consistency, it is possible for an honest replica to execute an operation out of order, but at least any future request from the same client will make the attack evident. Later, the optimization of the BFT2F protocol to achieve fork * consistency was discussed, and the performance penalty was studied. Finally, he showed that it is possible to generalize BFT2F to BFTx. This makes it possible for the system designer to tune the tradeoffs among consistency, liveness, and failure handling.

Someone from MIT asked how much correctness is sacrificed in the deployment. The answer is that the correctness is exactly guaranteed as either the fork or fork * consistency model specifies. Petros Maniatis from Intel Research at Berkeley asked about whether the two different weaker consistency models make a difference in the application. Jinyuan answered that the main differences are in performance and liveness. In an application like the card-swiping example, or other typical access control applications, these properties might be desirable.

- ***Ensuring Content Integrity for Untrusted Peer-to-Peer Content Distribution Networks***

Nikolaos Michalakis, Robert Soulé, and Robert Grimm, New York University

Nikolaos described a security problem from the successful P2P CDN systems: What if a malicious peer changes the content in an arbitrary way and sends it to the client? What if you want to see a sweet, good-looking Britney Spears but the peer gives you a bald, crazy one? The goal of this paper is to detect (but not prevent) bad replicas for both static and dynamic contents. However, this goal is not as easy as it appears initially. Nikolaos tried to go through the entire design space that includes the existing solutions and found some problems: Client verification could be quite expensive for small devices; if the client downloads multiple copies and accept the majority, the load of CDN is at least tripled and can only tolerate less than 50% misbehaving replicas; using other peers as spies could put them into a spy list by attackers easily and diminish the effectiveness; if volunteers forward bad content to a verifier, it is hard to draw a conclusion as to who corrupted it. All these lessons suggested that attestation is necessary. However, if only a few trusted verifiers are selected, the system doesn't scale well because the load on the verifiers is as much as the load of the entire CDN. In sum, the lessons learned are that the solutions that preserve existing servers and clients are not sufficient. Servers must sign their responses and clients must verify them. Replicas must produce attestation records for accountability. Sampled forwarding from clients is desirable to reduce network traffic.

Then Nikolaos presented the idea of the paper, "Repeat and Compare." Repeat essentially simulates the response-generation process. The requirement is to get rid of all nondeterminism so that, with the same external inputs and parameters, the identical results can be repeated. He argued that nondeterminism is possible to achieve in Web-related applications. In fact, in many cases, given the random seed, the pseudo random generator works just fine. The Compare part consists of two stages: forwarding attestation records to verifiers and then detecting misbehaving replicas. The attestation records are forwarded to verifiers via clients with probability p to a randomly selected verifier. Checking the freshness of an object is also a little tricky, requiring a trusted global synchronized clock to make timestamps to detect misbehavior. Eventually, the bad replicas are published based on a punishment policy to reduce the incentive to cheat.

Ryan Peterson from Cornell University asked about how many parties in the CDN system need to be changed in order to support Repeat and Compare. Nikolaos said that changes made at the client, verifier, and server are needed. The hard part is at the client side. However, it is inevitable that the client must be changed, because it has to be able to reject the misbehaving replica.

- ***TightLip: Keeping Applications from Spilling the Beans***

  Aydan R. Yumerefendi, Benjamin Mickle, and Landon P. Cox, Duke University

Confidentiality is harder to achieve than you might think! Aydan made two points at the outset of his talk. First, access control misconfigurations are widespread. A Kazaa usability study found that many users share their entire hard drive with the rest of the Internet. Second, even if you have perfect security and configuration, your privacy will only be as secure as your least-competent confidant who shares the information with you. Unfortunately, cryptography, secure communication channels, and intrusion detection systems do not prevent these problems.

Aydan proposed a new approach to prevent information leaks: a privacy management system called TightLip. TightLip takes a different path from conventional security software, in that it allows users to define what data is important and who is trusted regardless of the software that accesses them. There are three key challenges to TightLip: first, how to identify sensitive files and trusted hosts and protect that metadata; second, how to track the flow of the sensitive data through an OS and identify potential leaks; third, how to develop policies for dealing with the leaks. TightLip differs from most of the related work because it requires no change to the applications and hardware, and only minor modifications to the operating system.

The key concept in TightLip is a new OS object: doppelgänger processes. Doppelgängers are copy processes that inherit most of the state of an original process. They are spawned when a process tries to read sensitive data. The kernel returns sensitive data to the original process and scrubbed data to the doppelgänger. These two processes run in parallel. As long as the outputs for the two processes are the same, the original's output does not depend on the sensitive input with very high probability. The input/output are monitored by the system call arguments and result. When a difference is found, TightLip invokes a policy module, which can direct the OS to fail the output, ignore the alert, or even swap in the doppelgänger process. The scrubbing process depends on the data format. By default, it replaces each character from the sensitive data source with "x." Finally, by running several conventional benchmarks, Aydan showed that TightLip prototype overhead is quite modest.

Someone from Georgia asked about whether you can still forward emails when the email is marked sensitive. Aydan responded that it depends on the policy module. Then the concern from the questioner was that the configuration of the policy module could be as complicated as the applications.

Amin Vahdat from UCSD expressed some concern that as the data flows inside the application, more and more output might depend on the sensitive data so the false-positive rate could be high. Aydan said it's a common problem for all information flow analysis. However, for a subset of applications such as Web and P2P clients, the flows are quite simple and the false-positive rate is low.

Emin Gün Sirer from Cornell University asked what factors make it harder or easier for TightLip to scrub data. Aydan said it totally depends on the application. For example, scrubbing an email text is fairly easy, but to scrub a binary matrix might be very hard.

NETWORK MEASUREMENT

Summarized by Prashanth Radhakrishnan (shanth@cs.utah.edu)

■ *Peering Through the Shroud: The Effect of Edge Opacity on IP-Based Client Identification*

Martin Casado and Michael J. Freedman, Stanford University

Martin Casado, who presented this talk, started by speaking about the dependence of IP addresses on client identification in today's Internet. He noted that edge technologies such as NATs, proxies, and DHCP obscure the client's identity and thus motivated eliminating the effects of these on server identification of clients.

Their approach was to use the Web as a measuring platform to measure the effect of edge opacity, perform analysis on the results, and develop methods for servers to eliminate these effects.

To measure the Internet edge, they use active content execution at the clients. Clients are made to execute the active content in two ways: by "bugging" existing Web pages and by redirecting a percentage of CoralCDN's requests through measurement servers. From their measurements, about 60% of the clients were behind NATs and most NAT sizes were quite small. Also, IP deallocation resulting from DHCP was slow. Moreover, 15% of the clients were behind proxies, which were generally larger than NATs. Martin concluded that proxies pose a major problem for IP-based client identification and then discussed techniques for real-time proxy detection for servers.

During the Q&A session, Amin Vahdat (UCSD) asked if they had made a comparison of network characteristics (RTT, bandwidth, loss-rate, etc.) between clients behind proxies and NATs versus those directly connected. Martin said that the data was generally a bit messy for such analysis, but he noted that RTTs through proxies were longer and that NATs didn't seem to affect the RTT much.

Justin Cappos (University of Arizona) pointed out that Martin had mentioned DNS blacklisting as one of the motivations for this work, so he asked if Martin had any idea on the number of SMTP mail servers that use proxies. Martin replied that their measurements were just for the Web and do not include mail servers. John Agosta (Intel) observed that an adversary could potentially use this system to discover IP addresses behind a proxy to create a hitlist. Martin acknowledged that possibility but noted that in their system clients had to explicitly talk to the servers (thus reducing the probability). Tom Anderson (University of Washington) asked for Martin's comment on the ethics of gathering information by surreptitiously running "spyware" on clients. Martin answered that they have stayed well within the security model. He added that there was an implicit contract that when you go to a Web site you could execute things on the Web site and he further cited the example of Google Analytics, which gathers statistics by similar means.

■ *A Systematic Framework for Unearthing the Missing Links: Measurements and Impact*

Yihua He, Georgos Siganos, Michalis Faloutsos, and Srikanth Krishnamurthy, University of California, Riverside

Yihua He gave this talk on finding the missing links in current Internet topology at the AS level. He explained the need for an Internet topology and notied that the topologies derived using the current state-of-the-art techniques are incomplete because they underestimate the peer-to-peer links between ASes.

In this work, they collect data about AS edges using multiple methods such as existing BGP routing table dumps, exploring Internet routing registry, and inferring Internet Exchange Point (IXP) participants. All the links are validated by reverse traceroute. As a result, they found 40% more AS links and 300% more peer-to-peer AS links, most of which are at IXPs. Yihua noted that as a result of these "new" peer-to-peer links, the ASes could avoid using their providers to reach many destinations, lowering ISPs' costs and increasing revenue.

During the Q&A session, Vytautas Valancius (UIUC) asked about the total number of distinct edges collected. Yihua said that it was roughly 50,000.

Nick Feamster (Georgia Tech) noted that their conclusion about most of the peer-to-peer links being at exchange points was interesting. He then asked whether Yihua had an idea of how close the inter-AS connections are at specific exchanges. Yihua said that the measurements for this are inaccurate because the traceroutes are done only from selected points. Nick also asked why they did not consider routes collected at exchange points, especially since the missing edges are at the exchange points. Yihua acknowledged that it was a good idea to try out.

EMULATION AND VIRTUALIZATION

Summarized by Prashanth Radhakrishnan (shanth@cs.utah.edu)

■ *The Flexlab Approach to Realistic Evaluation of Networked Systems*

Robert Ricci, Jonathon Duerig, Pramod Sanaga, Daniel Gebhardt, Mike Hibler, Kevin Atkinson, Junxing Zhang, Sneha Kasera, and Jay Lepreau, University of Utah

Robert Ricci presented the talk on Flexlab. He contrasted two popular methods for evaluating networked systems, namely, "emulators" that provide control and reproducibility but lack realism and "overlay testbeds" that provide realism but lack control and reproducibility. Rob introduced Flexlab as a hybrid method that combined the merits of emulators and overlay testbeds, while eliminating the demerits.

Rob then described the Flexlab architecture. An application runs inside the emulator hosts along with a monitor that reports the application's network operations. The application's traffic passes through the path emulator, controlled by a pluggable network model. The network model may take its input from a measurement repository or may be driven in real time by the application's behavior reported by the monitor.

Given that accurate modeling of the Internet is still an open problem, they explore the approach of modeling the Internet, in real time, from the application's perspective. The application's behavior running inside the emulator (Emulab) is used to generate traffic in the overlay testbed (PlanetLab) and collect Internet measurements. The network conditions experienced by the PlanetLab traffic is applied to the application's path emulator, giving the impression that the Emulab hosts communicate across the Internet.

The evaluation results, from running microbenchmark iPerf, indicated that FlexLab could accurately emulate Internet traffic conditions. Through a case study with BitTorrent, they showed that FlexLab was able to remove some of the artifacts of PlanetLab host conditions (namely, CPU availability) that hurt BitTorrent throughput.

During the Q&A session, Dave Marwood (Google) asked whether users of the application-centric Internet modeling platform were limited to the network effects considered in FlexLab. Rob answered that users were limited to those, but he noted that the other network effects will be seen on latency, bandwidth, and packet-loss measurements. He said that as future work they plan on adding run-time validations to the system. Mark Chiarini (Tufts University) had a question on the throughput spikes of BitTorrent seen only in FlexLab, but not on PlanetLab, early on in the experiment. Rob said that it may be due to BitTorrent trying to ramp up its download rates. He noted that those spikes depended on the amount of CPU available and postulated that such spikes don't happen in PlanetLab because CPU is scarce.

Peter Druschel (MPI-SWS) asked whether, to use FlexLab, PlanetLab is always required to be online. Rob answered that their paper talks about two offline models that do not require PlanetLab. He further added that in future work they plan to record the application's Internet measurements from a run and replay it during its future runs.

- ***An Experimentation Workbench for Replayable Networking Research***

  Eric Eide, Leigh Stoller, and Jay Lepreau, University of Utah

Eric Eide began by saying that experiment testbeds such as Emulab provide resource management, but there still is a need for managing the experiment workflow. Workbench helps researchers manage their activities, software artifacts, data, and analyses and enables them to navigate through experiment history and replay or branch from any point. Effectively, Workbench is fundamental for repeatable research. They have evolved the Emulab testbed management software to be the basis for experimentation with Workbench.

Eric gave an overview of the current experiment lifecycle of Emulab: Users create persistent experiment definitions containing logical resources, "swap-in" experiments to allocate physical resources, and "swap-out" experiments to deallocate physical resources. He noted that in its current state Emulab confuses experiment definition with instance.

Workbench breaks experiments into multiple abstractions. A "template" is now the experiment definition. It is a versioned repository that stores the topology, parameters, and software. Creating a template "instance" involves assigning values to the parameters and allocating physical resources. A "run" is a context for doing a unit of work. An "activity" is a collection of processes, workflows, etc., that execute within a run. Finally, a "record" is a flight recorder of a run that saves all the things produced within a run in the database. "Record" is used for replaying experiments.

Given that this is a user tool, Eric discussed a couple of user case studies. He mentioned that the experimenters who used Workbench found the transition to Workbench relatively easy and the abstractions intuitive. He also noted that Workbench served as a useful platform for communication of results. Finally, Eric discussed a couple of problems they were facing with storage and node failures in PlanetLab that they plan to address in future work.

During the Q&A session, Pankaj Thakar (VMware) asked whether they had considered use of virtual machines for the Workbench activities. Eric agreed that it would be a good idea to consider. Mark Chiarini (Tufts) commented that Workbench is an excellent way of capturing scientific process and making research reproducible. Peter Druschel (MPI-SWS) observed that similar to Workbench's event capturing mechanism or experiment replay, low-level events at finer granularity need to be captured for debugging purposes. He asked if Eric viewed these as related or as separate concerns. Eric replied that they represent a continuum and that the latter is something that would best be implemented as part of the testbed infrastructure that Workbench could leverage. He also mentioned that this is currently a work in progress.

- ***Black-box and Gray-box Strategies for Virtual Machine Migration***

  Timothy Wood, Prashant Shenoy, and Arun Venkataramani, University of Massachusetts Amherst; Mazin Yousif, Intel, Portland

Timothy Wood presented this talk on Sandpiper, a system targeted at virtualized data centers. Sandpiper monitors resource usage and automatically detects and removes hotspots by leveraging virtual machine (VM) migration and dynamic resource allocation.

Timothy gave an overview of Sandpiper's architecture. A per-physical-machine agent monitors VM (CPU, memory, and network) resource usage and reports to the central control plane. They explore two approaches to gather VM resource usage information, namely, the (application- and OS-independent) black-box techniques and the (application- or OS-specific) gray-box techniques. The central server has a hotspot detector to decide when to migrate VMs using a combination of resource thresholds and historical data trends. It also includes a profiling engine to decide on the amount of resources to allocate to VMs, again based on historical data. Finally, the migration manager decides where to migrate VMs to mitigate hotspots. This decision is based on heuristics that take into account the physical machines' percentage resource utilization and the cost of migration measured in terms of the VM memory size.

Timothy then spoke about their evaluation results. The results demonstrated the effectiveness of migration and also showed a scenario (detection of memory hotspots) where black-box techniques were insufficient and gray-box techniques had to be used. Timothy concluded with brief descriptions of related and future work.

During the Q&A session, Diwaker Gupta (UCSD) noted that they seem to treat all resources as equal, but in reality applications may be more dependent on some specific resource. Timothy agreed that this was true, because they do not include application-specific knowledge in the decision. Diwaker also pointed out that gray-box techniques could be employed from outside VMs. Timothy said that they were planning on doing that in the future.

Pankaj Thakkar from VMware observed that in Sandpiper a VM could potentially keep gaining memory. Timothy acknowledged that in a real deployment they would need some cap on the allocations. Ryan Peterson (Cornell) asked whether they take into account factors such as frequency of flash crowds. Timothy answered that flash crowds were assumed to be rare and that their profiling takes recent flash crowds into account.

DEBUGGING AND DIAGNOSIS

Summarized by Prashanth Radhakrishnan (shanth@cs.utah.edu)

■ *Life, Death, and the Critical Transition: Finding Liveness Bugs in Systems Code*

Charles Killian, James W. Anderson, Ranjit Jhala, and Amin Vahdat, University of California, San Diego

Awarded Best Paper!

Charles Killian gave this talk on finding bugs in distributed systems by using model checking techniques. He started with a brief background on model checking, up to the state of the art where model checking is used on unmodified systems code to detect bugs that violate safety properties. With an illustration of the Pastry system, he argued that liveness properties (i.e., conditions that should always "eventually" be true) are richer and more natural for expressing errors in distributed systems.

Since a distributed system's state space explodes exponentially, exhaustive search techniques are insufficient to find violations of liveness, which are not expressed with short or even bounded executions. Charles introduced the notion of dead state space from which liveness can never be achieved, and this state space corresponds to errors. To find the dead state space, they combine the exploration of existing model checkers with random executions from every state encountered. An execution that ended in a dead state may have early transient states, which can lead to live states. To help identify the liveness error, they automatically find the critical transition that pushed the system into the dead state using a binary search between an identified transient and a dead state. Charles then detailed how they employed this technique to find a bug in the Pastry system.

Charles briefly spoke about the implementation of their software model checker, MaceMC, which is built over MACE (a language for defining event state systems). The user needs to set up the system in MACE by defining the system states, events, and transitions. Charles concluded with the lessons they learned, including the insight that it is possible to learn new safety properties from violations of liveness properties and the kinds of bugs distributed systems were prone to.

During the Q&A session, Ken Birman (Cornell) pointed out that it may be difficult to find liveness properties in large distributed systems. For instance, large DHT-based systems may undergo continuous churn and may "eventually" never satisfy the liveness property. Charles said that there could be other liveness properties for which this technique could be effective, but they have not studied such large systems yet. Mark Chiarini (Tufts University) observed that dead state identification is a reformulation of the halting problem and confirmed that it was determined experimentally.

■ *WiDS Checker: Combating Bugs in Distributed Systems*

Xuezheng Liu, Wei Lin, Aimin Pan, and Zheng Zhang, Microsoft Research Asia

Xuezheng Liu presented this talk on WiDS checker, a unified framework to check distributed systems through simulation and reproduced runs from real deployment. Their approach was to use library-based deterministic replay coupled with predicate checking. Xuezheng started with an example illustrating the complexity of bugs in distributed systems.

Their system is implemented in the middle layer between the distributed application and the OS, which provides the ability to intercept OS API, inject failures, simulation, and replay capabilities. All nondeterminism is logged and Lamport clocks are used for consistent group replay. During replay, WiDS interprets the entire distributed system as a sequence of events ordered by the "happens-before" relation. The entire system is replayed in a simulated process. Predicate checking happens at event (message receive, timer expiration) boundaries, includes liveness properties, and is decoupled from the replay (i.e., predicate checking happens on separate state copies). They found 12 bugs in four well-studied distributed systems (Paxos protocol, Boxwood, BitVault, and Chord), including a specification bug in Paxos.

Xuezheng noted that the downside of their work is that applications need to be ported to the WiDS middle layer. He concluded with a comparison of WiDS to other related systems. During the Q&A session, Mike Dahlin (University of Texas, Austin) asked about the runtime overheads involved. Xuezheng explained that their API interception layer is lightweight and thus their overheads were minimal.

- ***X-Trace: A Pervasive Network Tracing Framework***

  Rodrigo Fonseca, George Porter, Randy H. Katz, Scott Shenker, and Ion Stoica, University of California at Berkeley

Rodrigo Fonseca presented the talk on X-Trace, a pervasive network tracing framework. X-Trace gathers end-to-end execution traces, including various applications and the network stack, across administrative domains in the wide area. The goal is to capture the causal structure of a task, which is a specific activity that includes many operations at different abstraction levels, components, and administrative domains. Task ID and operation ID are propagated along the edges and nodes report the operations to a reporting infrastructure. Rodrigo noted that there are no layering violations in X-Trace.

X-Trace requires network support for opaque extension headers and device support for metadata propagation and reporting. The cost of X-Trace is minimal given the limited tracing metadata and asynchronous reporting. Also, nodes inside an administrative domain could send the reports to a domain-local repository for desensitizing information. Since the tracing functionality is independent across multiple layers, X-Trace supports partial and incremental deployment.

Rodrigo illustrated the working of X-Trace in a multilayered system with multiple node failures. The X-Trace software, APIs, and a public reporting service are available for general use.

During the Q&A session, there was a question on the overheads involved for normal operation. Rodrigo answered that tracing could be selectively enabled at different layers. Mark Chiarini (Tufts University) asked about sending the metadata in-band, in the case of UDP, for example. Rodrigo replied that they may not want to do it for fear of introducing extra bytes in the application stream if care is not taken when stripping the metadata at the other end. Mike Dahlin (University of Texas, Austin) asked about the information reported by the nodes. Rodrigo said that recording the edges is fundamental to X-Trace and other information would be application specific. Pankaj Thakkar (VMware) asked about the benefits of capturing traces externally in a nonintrusive manner. Rodrigo pointed out that there is a continuum on the amount of intrusiveness with varying tradeoffs and that they chose to explore this design point.

- ***Friday: Global Comprehension for Distributed Replay***

  Dennis Geels, Google, Inc.; Gautam Altekar, University of California at Berkeley; Petros Maniatis, Intel Research Berkeley; Timothy Roscoe, ETH Zürich; Ion Stoica, University of California at Berkeley

Gautam Altekar presented the talk on Friday, a system for debugging distributed applications through a combination of deterministic replay, symbolic debugging, and a language for expressing distributed conditions and actions. Gautam laid out the important features of Friday, namely, global comprehension, cyclic debugging, a familiar programming and debugging environment, usability in PlanetLab, and its support for legacy C/C++ applications. He compared Friday to other related tools (including WiDS, presented earlier in the same session) and noted that only Friday supported all these features.

Their general approach is similar to that of WiDS: continuous logging, consistent distributed snapshots, deterministic replay, and cyclic debugging. Unlike WiDS, logging is done in an application-transparent manner through libcall interpositioning. Friday provides users with the ability to specify global predicates in extended Python. The global predicates use the distributed breakpoint and watchpoint primitives that are implemented locally by leveraging GDB. Gautam also presented the predicate checking code for a couple of case studies, including a bug in a secure routing protocol.

Gautam noted that their approach is limited by the ability to specify relevant predicates and is susceptible to the quirks of the systems they leverage, namely, GDB and Python. As future work, they planned to improve the language support for debugging and also to explore ways to make root-cause isolation easier.

During the Q&A session, Mike Dahlin (University of Texas, Austin) enquired about the learning curve involved in using Friday's predicates. Gautam said that they have found specifying C predicates in Python to be complicated and that they were thinking about using domain-specific languages for improving the ease of predicate specification. James Anderson (UCSD) asked how they log discrete events; the answer was that each system call event is annotated with a Lamport clock. Then James brought up the case where multiple writes are merged into a single read with TCP. Gautam acknowledged that the scenario is not currently handled.

NETWORK LOCALIZATION

Summarized by Yun Mao (maoy@cis.upenn.edu)

■ *Network Coordinates in the Wild*

Jonathan Ledlie, Harvard University; Paul Gardner, Aelitis; Margo Seltzer, Harvard University

Locality is important in P2P file-sharing systems such as Azureus, or maybe almost all distributed systems. A typical way of exploiting locality is to use network coordinates (NCs) to predict the network distances and to choose peers based on the prediction. However, in the "wild world," developers from Azureus found that the coordinates are inaccurate and unstable to use, which motivates this research work. Jonathan gave a little background on the BitTorrent design, in particular, the peer discovery process after a new node joins the network. He argued that locality-biased swarms have improved bandwidth among peers and reduced inter-ISP bandwidth.

Jonathan next gave a short tutorial on how Vivaldi, the state-of-the-art NC system, works and their methodology to study and refine such a system. Their goal was to observe and understand the causes of inaccurate prediction of Vivaldi, test new techniques in simulation and real environments, and release their results to the latest Azureus version. To collect data, they had instrumented Azureus clients run on PlanetLab, logging every update to collect a detailed picture with a PL-to-non-PL latency matrix. They also summarize the coordinates' behavior in the software, such as instant errors and stability. One caveat of the measurement is that it is impossible to force all clients to run the latest version of Azureus, so different versions of the algorithms or parameters might be mixed. Although the NC maintenance messages are all piggybacked on existing control packets so that no additional messages are required, the authors discovered that the view of the network to the routing tables is limited. That is, there is a local bias in the communication pattern that leads to damage of the NC accuracy. Moreover, when a node receives an update from a node far away, the coordinates tend to be very unstable. Their insight is that the NC optimization should be against the whole network, with recent updates being more decisive. They proposed the idea called neighbor decay by maintaining a recent neighbor set, and they scale the force of each neighbor by its age, which limits the impact of high-frequency (near) neighbors and extends that of low-frequency (far) ones.

In the evaluation, Jonathan demonstrated that the coordinates are more stable than Vivaldi, and they improved the performance of the actual application to speed up DHT lookups. Detailed information is available at http://pyxida.sourceforge.net/.

Eugene Ng from Rice University was wondering how much benefit the underlying system can get from the gain of the new NC system on top of Vivaldi. Jonathan gave a positive answer, especially when the destination is only one hop away.

Petros Maniatis from Intel Research at Berkeley asked whether the actual Azureus users were happier after the coordinates system was deployed. Jonathan said yes, based on the feedback that he got indirectly from Paul Gardner. They were also building tracker optimization to see whether the ISPs are happier to see more local traffic. It is a gradual process because not all users will update to the latest version at once.

Eugene Ng said that based on the figures, there was still some degree of coordinates drifting. He was concerned about how much staleness could affect the performance. The answer was that all stale information was cut off after 30 minutes. It was true that if some node A was experiencing huge network latency change, and another node B only talked to it briefly, then B might have a problem. But this would be a rare event.

■ *Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts*

Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer, Cornell University

Bernard discussed why geographic information about Internet hosts is useful. Some typical applications include location-aware content distribution, network monitoring and attack localization, and geography-based service discovery. However, commercial IP to ZIP code databases provide quite rough granularity and are prone to provide stale data. He presented the Octant system, which copes with the problem from a perspective of system constraints: The constraints are set from network latency measurement and other sources. He argued that Octant differs from other systems in three ways: first, it uses both positive and negative constraints; second, the system can give the users a confidence-factor-like number to reason about uncertainty; finally, Octant only needs the aid of a small number of landmarks to solve the system of constraints geometrically.

The simplest format of a constraint is described by a circle. A node is known to be inside the circle if the constraint is a positive constraint, and it is outside if the constraint is a negative constraint. Typically, a positive constraint is derived from the union of all positive circles, and a negative constraint is derived from the intersection of the negative circles. How does Octant derive constraints? By measuring the latency from the node to a landmark, we can derive both a positive and a negative constraint. To represent the complex, irregular constraint regions, Bezier curves are used because of their preciseness and conciseness.

The hard part of the work is in deciding the radius of the circles based on the latency measurement. Bernard said that as one can use the speed of light as a conservative bound, measurement results show a strong correlation between latency and distance. Furthermore, a more aggressive bound can be used when additional geographic information is used. However, sometimes those geographically nearby nodes are separated by large latencies as a result of either long, indirect routes or high inelastic delays. Octant iteratively localizes intermediate nodes as additional landmarks to deal with indirect routes, and it models the high inelastic delays as height to cope with the latter problem. In the evaluation section, Octant was compared against GeoLim, GeoPing, and GeoTrack in terms of accuracy when using PlanetLab nodes as landmarks. The median error was 22 miles, and Octant outperformed other systems. He encouraged the audience to give a try at http://www.cs.cornell.edu/~bwong/octant/query.html

Someone from Harvard University asked whether there are some commonalities in the cases where errors of Octant are high, and how the system would perform outside the United States. The answer was that usually in those cases the hosts have very high latencies to all landmarks. Since PlanetLab doesn't have many nodes outside the United States, Octant doesn't work very well outside the United States. But as the node number increases, Octant is expected to do well too.

Tom Anderson from the University of Washington asked why in the evaluation Octant is not compared to the latest system, Topology-based Geolocation (TBG), published in IMC '06. Bernard responded that at the time of paper submission TBG's code was not available.

INTERNET INFRASTRUCTURE

Summarized by Anupama Biswas (anupamabiswas@gmail.com)

■ *dFence: Transparent Network-based Denial of Service Mitigation*

Ajay Mahimkar, Jasraj Dange, Vitaly Shmatikov, Harrick Vin, and Yin Zhang, The University of Texas at Austin

Denial of Service (DoS) attacks are a common problem affecting the availability of Internet services. Ajay presented a novel approach in mitigating DoS attacks. It is a network-based defense system called dFence. DoS has received a lot of attention but no apt solution has been provided to date. One of the reasons is that most of the defense mechanisms require software modification on either the routers or the customer ends or both, which reduces the transparency of the networks to the ISPs and customers. Another reason is that the solutions are not compatible with the existing TCP/IP implementations. Hence there might be deployment issues. Yet another reason relates to SYN cookies, which are backward-compatible but are not used by many users, because they are set off by default. They are provided with standard Linux and FreeBSD distributions. Finally, users cannot wait until the Internet is reengineered so that DoS attacks can be prevented in a better way. An immediate, easily deployable solution is required, and if there are no DoS attacks the solution should not affect network performance.

Ajay explained that dFence in a way provides a transparent solution to the existing Internet infrastructure as it does not require any software modifications at either routers or the end hosts. It dynamically introduces special-purpose middleboxes on the path of the hosts under attack. It intercepts the IP traffic in both the forward and backward directions and applies stateful defense policies that mitigate a broad range of spoofed or unspoofed attacks.

One of the questions asked was how this system differs from the Cisco CAR. Ajay answered that Cisco CAR only does inbound traffic interception and stateful mitigation whereas their approach does outbound traffic interception, which helps to enable more policies.

■ *RBGP: Staying Connected in a Connected World*

Nate Kushman, Srikanth Kandula, and Dina Katabi, Massachusetts Institute of Technology; Bruce M. Maggs, Carnegie Mellon University

It has been observed that BGP dynamics does not take care of packet loss when network links go down. This problem mainly occurs if there are multiple paths from the source domain to the destination domain. Through the paper Nate presented the idea that if the underlying network is still connected then the Internet domain remains connected. R-BGP is the solution that guarantees that a domain will remain connected to a destination as long as it has a policy-compliant path to that destination after convergence. The solutions provided to reduce data loss involve the complexities of the Internet and the BGP protocol. The solution provided through Resilient BGP (R-BGP) is much simpler. The data plane is isolated from any harmful effects that might occur while waiting for BGP to converge to the preferred route: The data plane is set to forward packets on precomputed failover paths. Hence packet forwarding continues unaffected throughout convergence and the routing table is not flooded with entries of all possible failover paths. This solution allows two challenges to be met: low overhead and continuous connectivity. To ensure low overhead, there is a single entry of the failover path for each neighbor. Continuous connectivity is guaranteed by providing a small amount of update information with each BGP update. Thus R-BGP works similarly to BGP except that it ensures connectivity in Internet domains as long as the underlying network is connected.

One question involved what happens if the link comes back. How does the router behave? Nate answered that in such a situation a nonfailover path will be chosen over the failover path.

- ***Mutually Controlled Routing with Independent ISPs***

  Ratul Mahajan, Microsoft Research; David Wetherall, University of Washington and Intel Research; Thomas Anderson, University of Washington

The Internet is made of ISPs that cooperate among themselves to carry traffic as well as compete with each other as business entities. No individual ISP can tune the traffic to flow through a particular route based on its self-interest; all must agree while each keeps its self-interests in mind. BGP, the most used routing protocol, provides some control, allowing ISPs to configure their outgoing traffic but giving no control over incoming traffic. This presents another problem. Suppose the incoming route fails; then the ISP cannot shift the traffic to another route as it is beyond its control. This problem is not new but can be mitigated through network engineering or by using newer routing protocols such as RCP.

The solution suggested by Ratul is the development of an interdomain routing protocol called Wiser. Wiser has the same overhead as BGP and it is complete and practical in all senses to run across multiple ISPs. There is no need for the ISPs to disclose any kind of sensitive information. Also, it allows ISPs to exert full control over the paths and make decisions based on their own interests and optimization criteria. He explained how Wiser builds the coordination mechanism on existent bilateral contracts that are already in place and is incrementally deployable across pairs of ISPs. Each of the downstream tags advertises routing with costs that are similar to BGP Multi-exit Discriminators (MEDs). Each of the upstream ISPs then selects the path with an amended process. This process considers the sum of its own costs and those reported by the downstream ISPs. Hence both the upstream and the downstream ISPs exert control on their route choices. This protocol has in-built mechanisms to discourage potential abuse.

- ***Tesseract: A 4D Network Control Plane***

  Hong Yan, Carnegie Mellon University; David A. Maltz, Microsoft Research; T.S. Eugene Ng, Rice University; Hemant Gogineni and Hui Zhang, Carnegie Mellon University; Zheng Cai, Rice University

Hong presented an experimental network, Tesseract, that provides direct control of a computer network. This computer network can be under a single administrative domain. In a typical IP network today, the desired control policy of an administrative domain is implemented via the synthesis of several indirect control mechanisms. The design that evolved from 4D architecture tries to overcome the problem. It promotes the idea of decomposing the network control plane in four different planes: decision, decomposition, discovery, and data. The network consists of something known as network decision elements. There are two abstract services to enable direct control: the dissemination service, which carries opaque control information from the network decision elements to the other nodes in the network, and the node configuration service, which provides an interface through which the decision elements command the nodes to carry out the desired control policies. The dissemination service enables plug-and-play bootstrapping in this network. The various distributed functions implemented on the switch nodes are neighbor discovery, dissemination, and node configuration services.

Tesseract reduces the need for manual code and enables a variety of different network policies to be implemented without making changes to the actual network. Hong's paper demonstrates the successful working of Tesseract with normal IP forwarding in an Ethernet network. Also, the paper evaluates its responsiveness and robustness when applied to different backbone and network technologies. It is seen that Tesseract is resilient to failures. Some questions about the scalability of the 4D network and the aftermath of a network failure were left unanswered.