Session Reports for SIGCOMM 2010

Shailesh Agrawal, Kavitha Athota, Pramod Bhatotia, Piyush Goyal, Phani Krisha,
Kirtika Ruchandan, Nishanth Sastry, Gurmeet Singh, Sujesha Sudevalayam,
Immanuel Ilavarasan Thomas, Arun Vishwanath, Tianyin Xu, and Fang Yu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

Abstract

This document collects together reports of the sessions from the 2010 ACM SIGCOMM Conference, the annual conference of the ACM Special Interest Group on Data Communication (SIGCOMM) on the applications, technologies, architectures, and protocols for computer communication.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Data communications*

General Terms

Algorithms, Design, Theory

Keywords

Conference Session Reports

Introduction

This document collects together reports of the sessions from the 2010 ACM SIGCOMM Conference, the annual conference of the ACM Special Interest Group on Data Communication (SIGCOMM) on the applications, technologies, architectures, and protocols for computer communication. The reports provide high-level notes of the paper presentations, and document the question and answer discussion at the end of each talk.

The authors of the reports were students attending the conference. We sought student volunteers among the those who had received travel grants, not as a condition of the grants, but out of their generosity in being more engaged in the conference.

— Aaditeshwar Seth, K.K. Ramakrishnan, Geoffrey M. Voelker

Keynote Session

Protocol Design for Effective Communication among Silicon or Carbon-based Nodes

Radia Perlman (Intel)

Report by Kirtika Ruchandan, IIT Madras (rkirti@cse.iitm.ac.in)

Some perspectives:

- On current trends: We do not need more user training to adapt to software/systems. Today's technology is full of random warnings that annoy or confuse users. People come up with horribly complicated software, put up all these mysterious pop-up boxes and then blame the users when things don't go right. Examples — "Do you want to view only the webpage content that was delivered securely?"
- On networking: Networking today is taught tightly coupled with TCP/IP, as if it were perfect and the only thing that existed. People just repeat things and nobody questions them anymore. There's a lot in there that's just wrong. For instance, the ISO failed because it had too many layers or the belief that if everything were encoded in XML it would all be interoperable or that security problems will go away once you have IPv6. Students should be shown various other protocols so that they can compare and contrast instead of teaching them a "corporate-checklist" of things to know. She even teaches Appletalk!
- On conferences and paper selections: We need to accept that paper selection is a somewhat random process. Professors should make students read really good papers that got rejected and also the bad ones that got accepted to drive this point further.

On protocol design and her work

- She worked on designing the layer 3 in DECnet.
- Back in grad school, she worked on how to get networks working even with malicious components. As long as there is one honest path between points A and B, one should be able to make it through.
- She made the link-state algorithm in ARPANET self-stabilizing.
- Thoughts on good design: It has to be "zero configuration". However, do give the user some knobs to play around with. Make sure that no knob tweaking worsens the performance.
- Analogies in the carbon-based world: Protocols for communication between humans!

On IS-IS: It was the obvious routing algorithm to choose to replace the spanning tree stuff, while working on TRILL (elaborated on later). How long a node waits here is dependant on what its neighbour advertises. OSPF copied this but made a blunder — which is seen in group handling. Note: More information about IS-IS vs. OSPF from her is available at http://www.ietf.org/mail-archive/web/ospf/current/msg00620.html

On her previous and current work

- Layer 2 is about point-to-point links and layer 3 uses source, destination addresses. Why have forwarding at both the layers?
- Why have a layer "2 and a half"?
- Ethernet: the routing algorithm here is O(n^2) in the number of links in general. She came up with the idea of pseudo-nodes to handle this.
- History of the naive bridge idea: Failures in loops due to the absence of hop counts. She was asked to make the memory overhead of the solution constant.
- "If you have k ports, remember the best message on each port", which makes the memory overhead (message length * # of ports).
- We should rewrite network stacks to put layer 3 in.

- Examples of the spanning tree did sub-optimal stuff: using just 2 Ethernets and 1 bridge.
- Problems with bridges: "If you cannot keep up with the wire speeds, you are not allowed to be a bridge".
- Views on why IP is suboptimal ("why not replace bridges with IP routers?") — one should check DECnet and CLNP for this.

The Boston hospital example: the company providing the switches used one huge bridged network. Bridging was never intended to do that: it was kind of a hack because people at the time were all confused about what Layer 3 was and they thought Ethernet was a competitor to DECnet. With bridges we did such a good job and it was so plug-and-play that you didn't have to think about them, so people are still taking large networks and doing bridges.

On TRILL — Transparent Interconnection of Lots of Links

- Concept of RBridges: devices that implement the TRILL protocol.
- Gist given in the version 2 of Algorhyme

> A network where RBridges can Route packets to their
> target LAN. The paths they find, to our elation, Are
> least cost paths to destination! With packet hop counts
> we now see, The network need not be loop-free!

About computing trees deterministically. With respect to supporting multiple interfaces (as questioned by an audience member), TRILL has problems with computing and choosing multiple trees that she wants students to solve.

Session 1: Wireless and Measurement

Report by Piyush Goyal, IIT Delhi (piyushcsiitd@gmail.com)

Efficient Error Estimating Coding: Feasibility and Applications

Binbin Chen (National University of Singapore), Ziling Zhou (National University of Singapore), Yuda Zhao (National University of Singapore), Haifeng Yu (National University of Singapore)

Motivation: Current applications leverage partially correct packets because some of these are recoverable into useful information; however some of them are corrupted to extent that they are not recoverable, which results in wastage.

Key idea: The usefulness of such partially correct packets depend on the number of error being below a threshold such that useful information is recoverable, an Error Estimation can help in judging if packet should be forwarded or retransmitted.

Related Work:

- Based on packet loss ratio
  - — Coarse grained info
  - — Need multiple packets to observe properly
- Based on signal-to-noise ratio
  - — Indirect measure and needs training
- SoftRate [Vutukuru et al., SIGCOMM'09]: Modify physical layer to obtain BER info
  - — Not supported by today's commercial hardware
  - — Better if per bit confidence needed

Evaluation:

- EEC-Rate implemented in MadWifi 0.9.4.
  - — Use per-packet BER to guide rate adaptation
- EEC-Rate consistently outperforms state-of-art schemes based on packet loss ratio or SNR
  - — In all experimental settings (e.g., indoor/walking/ outdoor, with/without interference)
  - — Up to 50% higher goodput in walking scenario
  - — Up to 130% higher goodput in outdoor scenario

Q: Can EEC distinguish b/w bit interference & failing?
A: It's possible to extend the algorithm. Apply EEC on two different parts and observe it.

Q: Did you look at EEC over long stream packets?
A: No.

Q: For evaluation you disregarded retransmitting, as no bakeoffs are there, should it affect the study?
A: EEC is for systems that leverage partial packets, so it should not.

Design and Implementation of an "Approximate" Communication System for Wireless Media Applications

Sayandeep Sen (University of Wisconsin Madison), Syed Gilani (University of Wisconsin Madison), Shreesha Srinath (University of Wisconsin Madison), Steve Schmitt (University of Wisconsin Madison), Suman Banerjee (University of Wisconsin Madison)

Motivation: Practical wireless communication is prone to errors, but different bits have different level of importance, so they require unequal protection.

Key idea: At symbol, wireless errors have a well-defined structure; it is more likely that the decoded symbol is close to the actual symbol compared to randomly chosen symbol; this can be exploited to provide unequal protection.

Related Work:

- PHY-MAC approaches:
  - — Do not exploit the unique approximation property of wireless errors.
- Network layer approaches:
  - — Disregard symbols received in error

Evaluation: Results on WARP SDR platform show that system provides improvement of the order of 5 dB to 20 dB in different scenarios.

Q: Was validation on external scenario where interference/fading exist, does QOM pattern hold there also?
A: No, we did in internal environment only.

Q: If you change constellation, how do you convey?
A: Indexing mechanism, PLLP header passes this info. If your error is far away, implies choice of modulation is bad so you should change that choice instead.

Not All Microseconds Are Equal: Enabling Per-Flow Measurements With Reference Latency Interpolation

Myungjin Lee (Purdue University), Nick G. Duffield (AT&T Labs – Research), Ramana Rao Kompella (Purdue University)

Motivation: Many new applications like algorithmic trading and high performance computing require extremely low latency. Even a microsecond of delay can translate into significant revenue loss in Data Centers. Existing solution provide solutions for average measurements, however different flows may exhibit different characteristics even when travelling over same link, so per-flow measurements are needed.

Key idea: Packets belonging to different flows that are closely spaced to each other have similar delays properties.

Related Work:

- SNMP and NetFlow
  - — No latency measurements
- Active probes
  - — Typically end-to-end, do not localize the root cause
- Expensive high-fidelity measurement box
  - — Corvil boxes (£90,000): used by London Stock Exchange
  - — Cannot place these boxes ubiquitously
- Lossy Difference Aggregator (LDA) [Kompella, SIGCOMM'09]
  - — Provides average latency and variance at high-fidelity within a switch
  - — Provides a good start but may not be sufficient to diagnose flow-specific problems

Evaluation: Evaluated on real router traces with synthetic workloads, real backbone traces with synthetic queuing.

- Achieves a median relative error of 10–12%
- Shows 1–2 orders of magnitude lower relative error compared to existing solutions
- Measurements are obtained directly at the egress side

Q: Did you ask vendors to implement your idea?
A: We tried to persuade Cisco, but were unsuccessful.

Q: If packet loss rate is high then accuracy of your framework will drop!
A: If the packet drops between In and Out of the node being analyzed, only then.

Session 2: Data Center Networks

(*No session reports because of scribe illness.*)

Generic and Automatic Address Configuration for Data Center Networks

Kai Chen (Northwestern University), Chuanxiong Guo (Microsoft Research Asia), Haitao Wu (Microsoft Research Asia), Jing Yuan (Tsinghua University), Zhenqian Feng (Microsoft Research Asia), Yan Chen (Northwestern University), Songwu Lu (UCLA), Wenfei Wu (Microsoft Research Asia)

Symbiotic Routing in Future Data Centers

Hussam Abu-Libdeh (Cornell University), Paolo Costa (Microsoft Research Cambridge), Antony Rowstron (Microsoft Research Cambridge), Greg O'Shea (Microsoft Research Cambridge), Austin Donnelly (Microsoft Research Cambridge)

Data Center TCP (DCTCP)

Mohammad Alizadeh (Stanford University), Albert Greenberg (Microsoft Research), David Maltz (Microsoft Research), Jitu Padhye (Microsoft Research), Parveen Patel (Microsoft Research), Balaji Prabhakar (Stanford University), Sudipta Sengupta (Microsoft Research), Murari Sridharan (Microsoft Research)

Session 3: Inter-Domain Routing and Addressing

Report by Arun Vishwanath, University of New South Wales (arunv@student.unsw.edu.au)

Internet Inter-Domain Traffic

Craig Labovitz (Arbor Networks), Scott Iekel-Johnson (Arbor Networks), Danny McPherson (Arbor Networks), Jon Oberheide (University of Michigan), Farnam Jahanian (University of Michigan)

Measuring the Internet is hard. There is limited "ground-truth" on inter-domain traffic.

Objective: Seek to examine the changes in Internet inter-domain traffic demands and interconnection policies.

Methodology: Leverage widely deployed Internet monitoring infrastructure and coax carriers into participation. Analysed more than 200 ExB of commercial Internet traffic over a two year period.

Key insight: Significant changes in inter-AS traffic patterns. Majority of inter-domain traffic by volume between large content providers, data centre/CDNs and consumer networks.

Q: It seems a little bit like classic television. Do you see any similarities with this?
A: I think it is a little bit different. Sure there are some similarities with other shared markets but the correlation is not unique. Very large networks have market power. It depends on who pays who. Cable operators bickering over funding model.

Q: Wondering if you can speak a bit more about the economic implications and other business/transit models for peering arrangements?
A: Many people are thinking about this and it is in the consideration/early stages. It depends on pricing, and power perspectives in terms of who has an upper hand in the negotiations.

Q: What is your take on whether or not you might see contracts specific to certain types of applications?
A: This gets into a long discussion. Clearly in wireless case you totally can, whether this looks better in fixed line is another story.

Q: I interpret CDNs as a brokerage service between content providers and thousands of ASs. If the number of top players decreases by an order of magnitute to say 150, then do you see likely a lesser role of CDNs in the Internet — do you have any data with regards to the role of CDNs?
A: Certainly a lot of thinks that we can talk about. Measuring CDNs is hard. Hard to figure out which CDN. Though, in general I think that the growth has led into the enterprise.

How Secure are Secure Interdomain Routing Protocols?

Sharon Goldberg (Microsoft Research, New England), Michael Schapira (Yale University), Peter Hummon (Princeton University), Jennifer Rexford (Princeton University)

There is no clear consensus on BGP security protocols. BGP security variants exist to prevent the propagation of bogus routing information.

Objective: Quantify the ability of the main protocols to blunt traffic-attraction attacks. To inform discussions about which secure protocol should be deployed in the Internet.

Methodology: Simulate attacks on each protocol on an empirically-measured AS-level topology and determine the percentage of ASes that forward traffic to the manipulator.

Key insight: Secure routing protocols only deal with one half of the problem. A clever export policy can attract as much traffic as a bogus announcement.

Q: How robust are the statements you make?
A: We ran the experiments twice on two different data sets, including UCLA's Cyclops data sets. Trends from UCLA and CAIDA datasets are the same.

Q: Have you compared the effectiveness of the various models versus their deployability? Now that you know which ones are effective, and at least almost, which ones are deployable, where do we go from here?
A: We completely ignored deployablity issues and implementability issues and so on. Currently working on the deployability aspects.

Understanding Block-level Address Usage in the Visible Internet

Xue Cai (USC/Information Sciences Institute), John Heidemann (USC/Information Sciences Institute)

What can simple observations about the Internet say?

Objective: Existing work provides little insight into the edge of the Internet and the use of the IPV4 address space.

Methodology: Use active probing, pattern analysis, clustering and classification.

Key insight: Provide information about how effectively network address blocks appear to be used, provide new measurements about dynamically managed address space, and show that low-bit rate last hops are often underutilized.

Q: Have you done any work using IPV6 data? You seem to have only IPV4?
A: No, we have IPV4 results only; probably in the future we will look into V6.

Q: How reliable is it to utilize ping response for detection? Will it work?
A: We didn't do anything about it here.

Session 4: Privacy

Report by Fang Yu, Ohio State (yufa@cse.ohio-state.edu)

Privacy-Preserving P2P Data Sharing with OneSwarm

Tomas Isdal (University of Washington), Michael Piatek (University of Washington), Arvind Krishnamurthy (University of Washington), Thomas Anderson (University of Washington)

Different types of data is being shared: public, private, public but without attribution. This talk focuses on "public but without attribution" data.

P2P network is good for content sharing. It is decentralized with no central authority to change the rules or interfere with the transfers. Their previous work has shown that current P2P systems are very easy to monitor.

Why isn't Tor good enough?

- Needs clients to be public
- Performance issue: high latency

Why isn't Freenet good enough?

- Performance is even worse than Tor

OneSwarm:

- Had thousands of downloads
- Uses social network to establish friendship (P2P) relationship, e.g., using Gmail account.
- Peers forward request and content over multiple hops, so you are never sure if the neighboring node is the true content requester or server.

Threat model: Attacker cannot observe traffic that she doesn't know. Attacker cannot inject traffic to arbitrary location in the network.

Finding peers:

- Peers can be looked up in a DHT.
- Small number of friends makes it difficult to get service, get good performance and peer can be vulnerable to attack. Solution to this is to add community service so that P2P relationships can be formed between untrusted nodes.

Resilience to timing attacks: This is dealt with by having same behavior to the repeated request.

Search content:

- Search is flooded and delayed at each hop. Cancel message is sent to cancel the search message once content is found.
- But there is no guaranteed every request is satisfied.
- Two types of searches: Hash (160 bits) search or content search (plain text).

To deal with long paths, the solution is to use multi-paths. This also provides resilience to failure on the path.

To transfer data, OneSwarm uses a modified version of BitTorrent. Also swarming download mode is used so that transfers for the same content is grouped.

Timing attack: Attacker monitors the delay for search and response. If the delay is small, the next-hop neighbor is suspected. The results show that even with an extremely large number of attackers, the attackers are still not able to narrow down to a single sender.

Performance:

- 20MB file transferred between 120 PlanetLab nodes. OneSwarm is much faster than Tor.
- Multipath has a huge impact on improving the transfer time.

Q: Is OneSwarm similar to Gnutella? Due to multi-hop, the performance will be really bad?
A: Multi-hop is over a social overlay. Performance is worse.

Q: What is relied on for not knowing the source?
A: Through delay and multi-hops.

Q: Then this can be modeled using random variables? With longer path, there can be leak of information?
A: It might be possible to estimate how far the source is.

Q: Is 20 MB too small? Why not try larger content?
A: Run this over Tor, not big of burden on the network. Same size for different protocols. Have problem to get complete 20MB transferred. 5 MBs then extrapolated.

Q: All nodes are stable?
A: Yes, all nodes are like a flash crowds.

Q: Did you experiment with node churn?
A: No.

Differentially-Private Network Trace Analysis

Frank McSherry (Microsoft Research), Ratul Mahajan (Microsoft Research)

Objective: Is it possible to do network analysis while achieving "differential privacy" guarantee?

They selected a set of sample traces to analyze. They produced a toolkit that people can use for doing this analysis. The complication is the tension between utility and privacy.

Related work for retaining privacy:

- Trace anonymization. Problem: hard to figure out what to remove.
- Code to data
- Secure multi-party computation

Differential privacy (DP):

- Used for checking whether a randomized computation depends on one particular record.
- Definition: Any computation result is equally likely to happen with and without a single record.
- Very useful with realistic applications.
- Simple example of DP is Count + Noise.

PINQ: A programming language ensures DP, so it is safer. DP normally involves introducing noise into the computation. As more analysis is run, privacy level degrades. Examples running analysis using LINQ: E.g., Worm fingerprinting in LNQ

Building analysis tools:

- CDF function using PING. As you keep running the queries, the cost accumulates. So you need to lower the accuracy requirement.
- String: Find frequently appeared strings in the data.
- Other tools and analysis: Frequent itemset mining
- High accuracy even with high privacy requirements.

Open questions:

- Is DP good enough? Can data be made so DP is good enough?
- Can we conduct research rather than reproduce research?
- With PINQ in mind, improve privacy or accuracy?
- Can we augment PINQ with network-specific functionality?

Q: Possible extension to PINQ to look for isolated events? E.g., network intrusion events.
A: You can imagine running such a filter. Second problem is using this filter on the data coming in real-time.

Q: Someone needs to hold on the raw packets?
A: You can do trace anonymization first. There hasn't been a magical solution today.

Q: Each person consumes epsilon for privacy. How do you set the budget for accuracy? What if budges runs out?
A: Privacy degrades with more tries. No good ways of coming up with the right budget yet.

Q: With correlation in the data, is it harder to guarantee privacy?
A: Yes, that is correct. Normally, there is less correlation.

Encrypting the Internet

Michael Kounavis (Intel Corp.), Xiaozhu Kang (Intel Corp.), Ken Grewal (Intel Corp.), Mathew Eszenyi (Intel Corp.), Shay Gueron (Intel Corp.), David Durham (Intel Corp.)

Motivated by the fact that only 600,000 out of 50,000,000 Web pages online use SSL/TLS. Can we change the infrastructure so all online transactions are protected? One problem with not being able to do so is protocol freedom, e.g., security protocols not allowed to be exported to countries outside of US.

Objective: Cryptographic operations are expensive, require millions of clock cycles. Goal is to speedup encryption and also enable authentication.

Contributions:

- Capability increased in CPU.
- Speed up symmetric encryption 4–12x
- Asymmetric encryption, RSA, 40% speedup.

Background introduction on AES. Encryption done though confusion and diffusion. One main contribution is AES-NI implemented in combinatorial logic. Implement different parts of AES in the processors. Encryption is done in parallel.

To speed up RSA: Reduce monolithic montgomery implementation by using 1.5 multiplications.

Tools are implemented in TLS 1.2 Evaluation results for comparing AES in different modes (CBC, CTR, ECB, GCM).

Future work: RSA 2048/3072 acceleration, SHA-3 winning algorithm, public trials.

To answer "Can we encrypt the Internet?", they believe this is possible.

Q: At some point, you will need a good source for random number generation. How many good number of random bits is needed per second?
A: We didn't address this. There are existing techniques.

Q: Invert Galois Field?
A: Isolate specific instruction for inversion.

Q: How general is your optimization? Can be easily applied to other algorithms?
A: You can implement in a variety of cryptographic algorithms.

Session 5: Wireless LANs

Report by Shailesh Agrawal, IIT Kanpur (sagrawal@cse.iitk.ac.in)

Fine-grained Channel Access in Wireless LAN

Kun Tan (Microsoft Research Asia), Ji Fang (Microsoft Research Asia), Yuanyang Zhang (Microsoft Research Asia), Shouyuan Chen (Microsoft Research Asia), Lixin Shi (Microsoft Research Asia), Jiansong Zhang (Microsoft Research Asia), Yongguang Zhang (Microsoft Research Asia)

Physical layer data rates in WLANs are increasing:

- Currently 802.11n standard has boosted data rates to 600Mbps
- Would increase to over 1 Gbps in future

This would degrade the data throughput effciency because of the overhead of the MAC.

Main problem: Current MAC protocol allocates the channel as a single resource at a time.

Solution proposed: FICA (Fine grained Channel Access)

- New PHY architechture based on OFDM
- Frequency-domain contention method using physical layer RTS/CTS signalling and frequency domain backoff for contending sub-channels

Evaluation using simulation and implementation on Sora software radio platform.

- FICA can improve efficiency ratio of WLANs by up to 400% as compared to existing 802.11

Q: You are using additional RTS/CTS messages. Isn't it increasing the overhead?

A: RTS/CTS are not packets. They are just messages. So overhead is not that great.

Predictable 802.11 Packet Delivery From Wireless Channel Measurements

Daniel Halperin (University of Washington), Wenjun Hu (University of Washington), Anmol Sheth (Intel Labs Seattle), David Wetherall (University of Washington)

WLANs based on 802.11 are

- Fast
- Reliable
- Ubiquitous

Goal

- Bridge theory and practice of performance evalutaion
- Accurately predict performance on real channels

Problem

- Performance on real channels is hard to predict
- SNR based on RSSI is the only option available
- Vary more than 10 db in real links

Solution: Effective SNR obtained by finding

- Channel state information
- SNR
- Bit error rate (BER)
- Average Effective BER
- Converting it bact to effective SNR

Evaluation: Implemented in Intel NIC
Result

- Measurement available in real NIC
- Predict packet delivery in real channel
- Matches good performance

Example Applications

- Rate/mimo/channel width selection: What is the fastest configuration for the link?
- Power consumption: Which antenna can i disable to save power?
- Spatial reuse: What is the lowest transmit power which i can use to support 100Mbps link?

Q: Is the debug mode publicly available?
A: Yes

Q: Error rate is dependant upon the processing speed, right?
A: Yes

SourceSync: A Distributed Wireless Architecture for Exploiting Sender Diversity

Hariharan Rahul (MIT CSAIL), Haitham Hassanieh (MIT CSAIL), Dina Katabi (MIT CSAIL)

Diversity is a fundamental property of WLAN

- Receiver Diversity
  — Sender broadcasts the packet
  — All opportunistic routing protocols
- Sender Diversity
  — Multiple senders connect directly to receivers
  — Can improve opportunistic routing

Sender Diversity hasn't been exploited much because

- Simultaneous transmissions don't strenghten each other
- Packets coming out of sync

How accurately we need to sync? Sync error less than equal to 20 ns.

Solution: SourceSync

- Enables concurrent senders to
  — synchronize their transmissions to symbol boundaries
  — cooperate to forward packets at higher data rates than they could have achieved by transmitting separately
- Allows all nodes that overhear a packet in a wireless mesh to simultaneously transmit it to their nex hop
  — This reduces bit errors and improves throughput
- Increases the throughput of 802.11 last hop diversity protocols

Implementation on FPGA of an 802.11-like radio platform.
Results

- SourceSync syncs distributed senders to within 20 ns
- Adds sender diversity gains to opportunistic routing
- Adds downlink diversity gains to WLANs

Q: A question about MAC scheme, is your sync header long enough to allow all nodes to join?
A: It's not the sync header, it's the gap that is long enough so that the nodes can join.

Q: Estimating delays, will the channel itself have phase response time?
A: No.

Q: You did take other components like channel access delay into account?
A: It is like regular carrier sense. They don't incur channel access delay. So, you dont need to measure them.

Session 6: Novel Implementations of Network Components

Report by Pramod Bhatotia, MPI-SWS (bhatotia@mpi-sws.org)

SwitchBlade: A Platform for Rapid Deployment of Network Protocols on Programmable Hardware

Muhammad Bilal Anwer (Georgia Institute of Technology), Murtaza Motiwala (Georgia Institute of Technology), Mukarram bin Tariq (Georgia Institute of Technology), Nick Feamster (Georgia Institute of Technology)

Motivation: Many new protocols require data-plane changes, such as OpenFlow. These protocols must forward packets at acceptable rates. Also, these protocols need to run in parallel with existing or alternative protocols. There are three approaches to solve the problem: develop custom software, develop modules in custom hardware, develop in programmable hardware.

Summary: Switchblade is a platform for rapidly deploying custom protocols on programmable hardware. It identifies and develops modular hardware building blocks. It enables and connects various building blocks in a hardware pipeline. It allows custom data planes to operate in parallel on the same hardware.

Switchblade offers three main features:

- Parallel custom data planes
- Rapid development and deployment
- Customizability and programmability

Q: Do you have any idea how to scale Switchblade for 10 Gbps?
A: It's the limitation of the FPGA card that we are using in the current implementation.

Q: What kind of hash function do you use?
A: Any hash function that allows larger bit string as input and compresses it into a 32 bit string.

Q: IBM released powerEN architecture for network processing that combines multi-threading and accelerator. Please look into it.
A: Okay, sure, that would be helpful.

Q: SwitchBlade has slow forwarding speed in software implementation. This shows that the approach in software doesn't work. However in RouteBricks and also, in PacketShader, the software implementation gives high forwarding speed.
A: Click and RouteBricks are standard PC implementations. Also, RouteBricks is limited with PCI bandwidth. I haven't looked into the PacketShader architecture yet, so cannot comment on it.

PacketShader: a GPU-Accelerated Software Router

Sangjin Han (KAIST), Keon Jang (KAIST), KyoungSoo Park (KAIST), Sue Moon (KAIST)

Motivation: PC-based software routers provide cost-effective packet processing with flexibility and programmability. However, a CPU-only implementation (like RouteBricks) doesn't scale for compute and memory intensive workloads like IPSec.

Summary: PacketShader is a high-performance software router for general packet processing with GPU acceleration. PacketShader exploits GPUs for highly compute-intensive workloads in packet processing for performance gains. On the other hand, the RouteBricks targets the CPU-only approach that leads to bottleneck for compute-intensive protocols like IPSec. PacketShader consists of two components: a Packet I/O engine that optimizes packet reception and transmission, and a GPU acceralation framework for offloading the compute-intensive part. They evaluated PacketShader on four protocols (IPv4, IPv6, IPsec, OpenFlow) to demonstrate the flexibility and performance advantages of PacketShader.

Q: Where do you implement the checksum?
A: We don't offload the checksum in GPU.

Q: Do you have cycle breakdown for the different stages? Which part is most important?
A: Details are in the paper.

Q: Why does the CPU-only approach have a faster speedup?
A: We made significant effort to optimize packet I/O path. Since IPv4 is more concentrated towards the I/O path as opposed to compute-intensive IPsec (relying on GPU for performance gains). Because of the many optimizations, we were able to get faster speedup for CPU-only as well.

Q: How do you think it will scale up in terms of power efficiency?
A: The GPUs reap more performance as compared to the extra power it consumes.

Q: Why do you think the era of network processors disappeared? Do you think that this approach has advantage over network processors in terms of programmability?
A: Network processor is not a commodity hardware, it does not provide general packet processing. On the other hand, CPU-based approach is more programmable.

EffiCuts: Optimizing Packet Classification for Memory and Throughput

Balajee Vamanan (Purdue), Gwendolyn Voskuilen (Purdue), T. N. Vijaykumar (Purdue)

Motivation: Packet classification is a key functionality in routers for determining the highest priority rule out of a set of rules to which a packet matches. Since the classifiers are continuously growing in size along with increase in the line speed, there is a need for packet classification at a high throughput with minimum memory footprint. The authors have tried to optimize the existing popular packet classifiers like HyperCuts and HiCuts which incur high memory overhead.

Summary: EffiCuts propose four novel ideas for efficient packet classification: separable trees, selective tree merging, equi-dense cuts, and node co-locations. The key insights of Efficuts is that many rules in a classifier overlaps vastly in size which causes replications. Also, since the rule-space density varies, the equi-size partitioning of the rule space leads to partitioning into sparse cuts.

Q: Do you separate rules based on large and small in each dimension? And how do you define small and large rules?
A: The rules spread across various database follows bimodal distribution. We define rules based on the fraction in the rule spaces.

Q: Are you comparing Efficuts with TCAMs that provide many optimizations for power consumption?
A: We are comparing with worst possible case compared to TCAMs, so essentially we take in account for all the optimizations.

Q: Do you think the packet classification could be done more efficiently on the GPUs?
A: Yes, that seems a nice problem.

Session 7: Cloud and Routing

Report by Phani Krisha, IIT Madras (p.phanikrishna@gmail.com)

Theory and New Primitives for Safely Connecting Routing Protocol Instances

Franck Le (Carnegie Mellon University), Geoffrey Xie (Naval Postgraduate School), Hui Zhang (Carnegie Mellon University)

Problem: Role of primitives for connecting multiple routing protocol instances. Existing primitives across multiple routing protocols are vulnerable to anomalies and are rigid.

Existing Solution: The existing primitives are route selection and route redistribution: Explained with example highlighting the requirement. For multiple instances of routing, new theory of primitives are proposed.

Proposed Solution: Algebraic approaches to routing are proposed. Routing across multiple routing protocol instances is proposed. Defined new conversion functions and illustrated with an example. A set of universal metrics are defined across multiple routing instances and are compared for route redistribution. The sufficient condition is proved with an example, mapping function does not require to be bijective. Example also discusses the choosing of route type and cost of the path chosen.

Summary:

- Prefer non BGP to BGP

- Prefer lowest cost
- Traffic engineering is adaptive for the new design primitives while it is static for old primitives.
- Network dimensioning is much more than getting protocols correct, hidden properties needs to be considered.

Q: The conversion functions need to be present across all routers?
A: Yes, it is required at each router for comparing the multiple routing instances. We plan to extend different definitions across routers.

Q: The functions are self contained. Should they be standardized?
A: We are discussing about this with the vendors.

Q: With different IGP's, how do you design the conversion functions across different areas and entities?
A: Each of them covers different bandwidth, distance and provides different reliability. Will I be able to convert the bandwidth into comparable terms may not be straight forward and it is left to the operators.

Q: What is the behavior of the network during convergence?
A: We have not looked into it and that could be the future work.

Q: How is consistency achieved when there is a global ordering?
A: Global ordering is not looked at and we need to enforce it.

Q: Does the conversion functions ensure correct and optimal routing?
A: Offering optimal routing depends on each routing instance. For example, we may have optimal functions for shortest path routing but not for OSPF.

DONAR: Decentralized Server Selection for Cloud Services

Patrick Wendell (Princeton University), Joe Wenjie Jiang (Princeton University), Michael J. Freedman (Princeton University), Jennifer Rexford (Princeton University)

Problem: How to handle the client requests specific to a particular location? Selecting the server for each request dynamically depending on the load at each server is also a problem.

Related work: Centralized coordination schemes and distributed heuristics schemes are proposed. Parameters used for the schemes are network latency, distance estimation using virtual coordinate system.

Existing solution: Donate the allocation process to a Content Delivery Network (CDN), also called outsourcing. However, proper outsourcing guidelines are to be framed for this solution.

Proposed solution: Configurable mapping policies are defined that consider client performance and server load.

Strengths of the proposed solution: Flexible mapping mechanisms like generating DNS responses dynamically, use HTTP proxies and HTTP redirections for requests. Secure registration using DONAR Update Protocol for any communication. Reliability through decentralization.

Implementation and Results: Deployed using a coral CDN. Trace-based simulation with measured important parameters such as network performance and load split weights.

Q: Different donor nodes exchange information among each other. How do you prove that this solution could converge to optimal?
A: Covered in the paper. One round of optimization is done by getting the local updates.

Q: End-to-end delay is considered important in many protocols. How does load balancing bring benefits?
A: Load balancing reduces the end to end delay in long term.

Q: Do you consider all servers to be equally functional? Any server-specific properties are required?
A: Yes, we consider all servers to be similar.

Q: Why end-to-end delay is not optimized in the protocol, as it is the only property the client measures for each request?
A: Global optimization like end-to-end delay does not cater to all the customers.

Q: Client requests are from DNS and not from the client itself. What is it about information truncation at the server?
A: Some reply based on the limitations of the DNS.

Q: What if a server fails or configurations are changed? How long does it take to resume?
A: Physically a server needs to be verified for a failure or some time expiry mechanism has to be implemented. This is not implemented in the current version. The authors are building support for supporting online.

Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud

Mohammad Hajjat (Purdue University), Xin Sun (Purdue University), Yu-Wei Eric Sung (Purdue University), David Maltz (Microsoft Research), Sanjay Rao (Purdue University), Kunwadee Sripanidkulchai (IBM Research), Mohit Tawarmalani (Purdue University)

Problem: In an enterprise network using cloud networking, what services are to be deployed in cloud and what services in the enterprise? The issues to be looked are:

- Data Privacy (National Privacy Law)
- Industry Specific Laws
- Performance Requirements (Service Level Agreements between cloud vendors and the enterprise)

First order challenges to hybrid clouds:

- Planning hybrid layouts
- Migrating security policies
- Security using access control lists
- Complexity of Enterprise Applications and the Data Center policies.

Usage of clouds: What services are to be deployed in cloud? and what services in the enterprise?

Proposed solution: Model the network as a graph in which a node is an application component and a link where interaction between two nodes is possible. Migration strategies are proposed modeling communication costs in Internet. Also migration reachability policies are defined and verified.

Q: How much time it took to understand the dependencies between the components?
A: Communicated directly with the operators on the interactions between the components and the dependency diagrams are taken from them. Leverage importance where the dependency is not clear at the operators. Extraction Dependency tools are also used.

Q: In a graph shown in presentation, requirement is $<10\%$ but the graph shows 17%.

A: It is statistical and determined using planet-lab. Something related to inaccuracies in coarse modeler large packets does not merge to 10%.

Q: Can you comment on rewriting the application completely with performance metric and maintaining legacy issues firmly?
A: Service Oriented Architecture, Modular structure can be implemented and we concentrate on this. However, order application may be a problem.

Q: Does the problem persist for newer applications also?
A: Configuration dependencies are available and can be used in designing new applications.

Q: Migrate an application into cloud. Similar data cannot be moved. How is it specified to reside in cloud or the enterprise?
A: You might have a constraint on the optimization model. Application architect will ensure this.

Q: Is the cost metric used CAPEX? Is it the 'RIGHT' metric?
A: The number of servers each enterprise must have is determined and yes, it is CAPEX and it is one time expenditure to enterprise. Yes, it is RIGHT metric.

Q: What about OPEX?
A: We have not looked on that aspect in this work.

Session 8: Network IDS

Report by Tianyin Xu, University of Göttingen (tianyin.xu@cs.uni-goettingen.de) and Immanuel Ilavarasan Thomas, IIT Madras (immanuel.ilavarasan@gmail.com)

NetFence: Preventing Internet Denial of Service from Inside Out

Xin Liu (Duke University), Xiaowei Yang (Duke University), Yong Xia (NEC Labs China)

Background: Denial of Service (DoS) attacks remain as a formidable threat to the Internet.

Existing Work: Receivers as victims — Denial of Edge Service (DoES) (e.g., AIP, AITF, CenterTrack, dFence) — cannot deal with new threat: Denial of Network Service (DoNS)

Challenge: How to design an open network architecture that is resistant to large-scale DoS attacks: combating both DoES and DoNS

Design Principle: inside-out, network-host, joint lines of defense

- Network controls its resource allocation
- End systems controls what they receive

Key Ideas: 1. Hierarchical design, 2. Secure congestion policing in the network, 3. Coupled with network capabilities

Goals:

- Scalable: no per-flow state in the core
- Robust: to compromistd routers and hosts
- Open: receiver explicitly authorizes desired traffic

Architecture:

- Two types of NetFence packets: Request & Regular
- Two types of feedback: nop (no attack) & mon (attack)
- Unforgeable Congestion Policing Feedback
- Congestion Feedback as Capability
- Fair Share Guarantee: achieves per-sender fairness for single bottleneck scenarios

A sender first sends a request packet and its access router stamps nop. A router under attack replaces nop with L↓. A receiver uses the feedback as capabilities. Sender sends regular packets with congestion policing feedback. Access router validates feedback: starts congestion policing and reset L↑. Bottleneck router establishes a congestion policing loop: if congested, replace L↑ with L↓.

Evaluation:

- NetFence is implemented in Linux and in ns-2 simulator
- Processing overhead is low: suitable for high-speed implementation
- Header overhead is just 20–28 bytes
- NetFence limits DoES attack with acceptable cost of scalability
- NetFence limits DoNS attack by providing fairness

Q: Can you distinguish good traffic and bad traffic?
A: No, NetFence cannot. NetFence is based on fairness to prevent Internet DoS. We use AIMD to control the rate limit.

Q: Can you separate flash crowds from malicious attacks?
A: No, we cannot. NetFence uses the same algorithm because both of them trigger the same signals. Presence of flash crowds also results in a failure of end-to-end congestion control.

ASTUTE: Detecting a Different Class of Traffic Anomalies

Fernando Silveira (Technicolor, UPMC Paris Universitas), Christophe Diot (Technicolor), Nina Taft (Intel Labs Berkeley), Ramesh Govindan (University of Southern California)

Background: Uncovering anomalies in large ISPs and enterprise networks is challenging because of the wide variety of such anomalies. Anomaly detection: monitoring traffic and mining unusual behavior.

Existing Work: It is hard to obtain a model of "normal" traffic

- current models must be trained from (normal) traffic data
- definition of an anomaly depends on the data
- in practice training isn't guaranteed to be anomaly-free

Problem Statement: Can we detect anomalies without having to learn what is normal?

Approach: A model of normal behavior based on empirical traffic properties. Advantages:

- no training → computationally simple and immune to data-poisoning
- accurately detects a well-defined class of traffic anomalies
- theoretical guarantees on the false positive rates

Limitation: Method is sensitive to changes in traffic characteristics

Empirical Traffic Properties: 1. Flow Independence; 2. Stationarity; 3. If flows satisfy properties above, they show equilibrium

ASTUTE-based Anomaly Detection:

- Given a detection threshold K' and a pair of consecutive time bins
- Measure: 1. Set of active flows; 2. Mean volume change; 3. Variance of volume changes
- Compute ASTUTE Assessment Value K(F) based on the above 3 measurements
- Flag an alarm if: $|\mathrm{K(F)}| > \mathrm{K'}$

Results:

- Small overlap between ASTUTE and other methods
- ASTUTE specializes in a different class of anomalies

Q: Is ASTUTE unable to detect large flows which buid up slowly and gradually?

A: Yes. There is a threshold as to what is detected as a stationary pattern.

Q: How does ASTUTE differentiate between flash crowds and DoS attacks?
A: We are not looking for badly behaved flows. The operator is aware of the existence of DoS attacks. Anomaly detection deals with detecting new problems which the network operator is not aware of. (Maybe due to bugs in the settings).

Q: Why is it that ASTUTE cannot find an anomaly involving a few large flows?
A: A small number of anomalous flows does not generate enough correlation to stand out from the large number of independent flows in the background traffic.

Q: With big elephant flows, if the number of flows is small, is the threshold too small?
A: We require a large number of flows for central limit theorem to hold.

Q: How can you differentiate types of anomalies, say DoS attacks from flash crowds?
A: We use a set of criteria (described in the paper) to classify the anomalies. In the case of DoS attacks we've shown here, they're all SYN flood attacks.

Q: You mentioned that part of your results are based on manual analysis of anomalies? How can you be confident of that analysis?
A: After working on ASTUTE, we have developed an automated method for root cause analysis (called URCA, published at INFOCOM'10). The results provided by URCA were identical to those of our manual analysis. The fact that we were able to obtain the same results through two independent analysis methodologies reinforces the confidence in these results.

Q: How did you obtain ground truth for DoS?
A: Large number of packets per flow implies DoS

NetShield: Massive Semantics-based Vulnerability Signature Matching for High-speed Networks

Zhichun Li (Northwestern University), Gao Xia (Tsinghua University), Hongyu Gao (Northwestern University), Yi Tang (Tsinghua Universi-ty), Yan Chen (Northwestern University), Bin Liu (Tsinghua Universi-ty), Junchen Jiang (Tsinghua University), Yuezhou Lv (Tsinghua Uni-versity)

Background: Accuracy and speed are the two most important metrics for Network Intrustion Detection/Prevention Systems.

Existing Work:

- Regex (regular expression) based approaches: high speed, poor accuracy
- Vulnerability signatures: better accuracy, low matching efficiency — sequential matching

Challenges:

- How to speed up vulnerability signature matching with large vulnerability rulesets
- How to parse the traffic and to recover the protocol semantic information fast enough for signature matching

Contributions:

- An efficient multiple signature matching scheme for a large number of vulnerability signatures
- Fast stream-fashioned lightweight parsing
- Evaluation and methodology

Efficient Matching:

- Tabular signature representation: convert the set of signatures to a two-dimensional table, named signature table, which is the key of simultaneously matching
- The candidate selection algorithm: to keep the per-flow state small

Automatic Lightweight Parsing: devise a parsing state machine to achieve stream parsing, which is sufficient for protocol parsing. Parsing Performance:

- Throughput: compared with the Opt BinPAC, NetShield speeds up binary protocols (DNS & WINRPC) about 12 times, and a text protocol (HTTP) about 3–4 times
- Memory: at most 16 bytes for all the three protocols

Parsing + Matching Performance:

- Candidate size: for all protocol and traces, the max size of candidate sets is no more than 8 with average size less than 1.5
- Throughput: speed up the matching by 8.8–11.7 times for HTTP, 2–4 times for WINRPC
- Scalablity: system throughput degrades gracefully when increasing the number of rules
- Memory consumption and breakdown: for 794 HTTP vulnerability signatures, we only need about 2.3 MB memory

Q: What's the memory overhead on representing the rules?
A: It depends on the rules in the memory. For mapping data structure, we only need 2.3MB for 794 HTTP vulnerability signitures, while DFA requires 5.29GB and XFA 1.08MB.

Q: Since the memory usage of XFA is better than that of NetShield, what is the advantage of of this method?
A: NetShield has higher accuracy and fewer false positives.

Q: Is it possible to apply NetShield in GPU?
A: I think that it is possible. The connections are mostly independent, so we can dispatch them into different GPU cores. With large number of cores, such as 240 cores from a GPU, the performance can be further improved.

Q: Can you go to slides 25. Here, in the graph, it shows that you can achieve 1 byte per cycle?
A: No, not 1 byte but 1 bit per cycle.

Q: I wonder whether it is possible to achieve multiple bytes per cycle?
A: With multiple cores, the performance can be further improved, but currently we do not achieve multiple byte per cycle.

Session 9: Network Architecture and Operations

Report by Kavitha Athota, Jawarhal Nehru University (athotakavitha@gmail.com)

R3: Resilient Routing Reconfiguration

Ye Wang (Yale), Hao Wang (Google), Ajay Mahimkar (UT Austin), Richard Alimi (Yale), Yin Zhang (UT Austin), Lili Qiu (UT Austin), Y. Richard Yang (Yale)

Motivation:

- Failures are common in operational IP networks
- Planned maintenence affects multiple network elements
- VOIP, Video Conferencing ,gaming etc. applications have stringent requirements on reliability

- All the above said need resiliency: network should recover quickly and smoothly from one or multiple overlapping failures.

Challenges:

- Number of failure scenarios quickly explodes
- Difficult to optimize routing to avoid congestion under all possible failure scenarios
- Difficult to install fast routing

Existing Approaches Limitations:

- Focus of recent studies is on reachability, i.e., minimizing the duration in which routes are not available to a set of destinations.
- Only consider a small subset of failures
- Online routing re-optimization after failure cannot support fast rerouting

It is also crucial to consider congestion and performance predictability when recovering from failures. Network Congestion is mainly due to traffic that has been rerouted due to link failures

Proposed Method R3

- equires no enumaration of failure scenarios
- Congestion free for all upto-F link failures
- Efficient with respect to router processing/storage overhead
- Flexible in supporting diverse practical requirements
- R3 optimizes protection routing for a set of traffic demands on the original topology

R3 provides congresion-free link-based protection using two phases of operation. During offline phase, optimize routing r and protection routing p together to route original demand plus rerouting virtual demand on the single original topology. The optimization tries to minimize congestion under failures. During the online phase, the routers apply reconfiguration on p upon each failure event, so that p can be used as fast rerouting.

Q: Have you looked at how you will deal with the traffic while a link is being repaired?
A: Similar as the online reconfiguration for link failure, we develop a simple online reconfiguration algorithm for link recovery. Think about a simple single-link failure case. When link l failed, the protection routing p_l for l is activated, and p is reconfigured to, say p^1, preparing for future failures. When l recovers, p_l will be immediately deactivated, and p^1 will be reconfigured back to the original p. Our online reconfiguration algorithm works if multiple links fail and recover in arbitrary order.

Q: During convergence (or reconfiguration) what will happen to messages generated like BGP updates?
A: The protection routing is fast rerouting, implemented by extended MPLS label stacking. No packets including updates in BGP and other routing protocols will be dropped after failure occurs. Online reconfiguration is only applied to prepare for future failures, so the convergence does not disrupt any current traffic rerouting. If multiple failures happen too close to each other, then online reconfiguration may be integrated with techniques such as Failure Carrying Packets to reduce instantaneous disruption. Note we assume that the base routing (such as OSPF) does not need to recoverge after failures, because R3 protection routing can handle any failure scenario. When R3 is carrying traffic, the operator can reoptimize the routing (with no rush) given the changed network topology, if failure recovery takes long time.

Q: If the number of failures partition the network, what will happen?
A: Generally, for a K-edge-connected network topology, we want to handle up-to-F link failures. If K is less than or equal to F, then there will always be some failure scenarios that can partition the network. For network partition scenarios, R3 cannot guarantee no congestion. Actually, even the reachability cannot be guaranteed for these cases. What happens in R3 is, the precomputation will not be able to find (r, p) that makes MLU less than or equal to 1. In other words, if F is too large, the optimal MLU under F link failures is still larger than 1. As per our theorem, R3 cannot guarantee congestion-free under failure scenarios involving up-to-F links.

Q: Comment on how you compare your system with oblivious routing scheme?
A: Oblivious routing is certainly an inspiring related work. Due to the challenge of topology uncertainty, they try to avoid congestion under a small set of failure scenarios. For example, only handling single-link failures. However, when multiple overlapping failures occur, the congestion is more likely to happen. We did survey with a list of large ISPs. At least one from Asia and one from Europe confirm that they see serious network congestions due to multiple network element failures. R3 is designed to handle these cases.

Q: How do you evaluate traffic protection priority?
A: We obtain traffic traces for different classes of traffic from the large US ISP. We apply different protection levels to different classes of traffic. The ISP may apply specific routing and forwarding policies to different classes of traffic, e.g., using DiffServ. In our evaluation, we assume the ISP apply a very simple policy: if VPN traffic has higher priority than general IP traffic, then after failure occurs, the VPN packets will be forwarded first; if the router needs to drop packets due to congestion, then it drops IP packets first. The evaluation results demonstrate that R3 with priority respect traffic priorities: by sacrificing IP traffic, it guarantees VPN traffic can be all rerouted.

Q: Can R3 be applied to inter-domain rerouting?
A: As long as we have sufficient control and knowledge of inter-domain rerouting, then R3 can be applied to inter-domain routing. R3 has no specific assumption on the network topology, it can handle multi-graph. Given the multi-domain network topology with link capacities, R3 can be used to protect multiple link failures. Note inter-domain rerouting may largely involve routing policies of different domains. We need to think about how to add constraints to R3 precomputation so that R3 can be extended to handle these policies. For example, if a transit domain wants to limit the resource used in rerouting, then we may need to add a set of constraints for this particular domain.

Detecting the Performance Impact of Upgrades in Large Operational Networks

Ajay Mahimkar (University of Texas at Austin), Han Hee Song (University of Texas at Austin), Zihui Ge (AT&T Labs – Research), Aman Shaikh (AT&T Labs – Research), Jia Wang (AT&T Labs – Research), Jennifer Yates (AT&T Labs – Research), Yin Zhang (University of Texas at Austin), Joanne Emmons (AT&T)

Motivation: Expected changes such as software upgrades lead to improvement in router CPU utilization. Unexpected changes such as bugs in new release of software may cause perfornamnce degradation. For Monitoring impact of upgrades, existing approaches are:

- Lab Testing
  - Cannot replicate scale and complexity of operational networks
  - Cannot enumberate all test-cases
- Monitoring upgrades in-field

- Critical issues are caught after a long time
- Operational challenge with large number of devices and performance event-series

Proposed system MERCURY detects the performance impact of upgrades in operational networks:

- Automated data mining to extract trends
- Scalable across a large number of measurements
- Flexible to work across a diverse set of data sources
- Ease of interpretation to network operations

Challenges addressed:

- How to extract upgrades? MERCURY identifies the triggers from large set of maintenance activities by caputuring two key metrics: rareness of activity and coverage of the activity
- Do upgrades induce behavior changes in performance? Detect persistent changes and distinguish from transient changes, MERCURY applies non-parametric rank-based behavior change detector.
- Is there commonality in configuration across devices? MERCURY applies static rule mining techniques to identify common attributes among all network elements that experience consistent behavior changes for a given trigger.
- Is the change observed network-wide? MERCURY first aligns KPI event series by trigger timestamp and aggregates them across multiple locations where no behavior change is detected at individual location, and then applies chage detection on the aggregated event-series.

Q: From high-level you have triggers. Have you ever thought of inverse where your triggers are consequences of alarms instead of upgrades and then correlate with upgrades?
A: Taking the other route — where you first run anomaly detection and then correlate with upgrades — can generate a lot of false alarms. Our focus is on persistent behavior changes rather than transient anomalies. Starting with upgrades gives hints about the triggers and lets the ISP to be more focused.

Q: Can you talk about time raising on attribute changes in the consequences of upgrades?
A: We need sufficient samples to ensure that we can trust the results with enough statistical confidence.

Q: Transient behavior also matters so what is your take on that?
A: Very good question. After interactions with the network operations team, they requested for near real-time (or transient) impact of upgrades — such as 4 to 6 hours immediately after the upgrades. We are currently looking into it.

Q: Can you comment on link upgrades on your work? Do you consider link statistics in the analysis?
A: Incorporating link-level statistics is challenging. We use a spatial proximity model when correlating the upgrades with the performance impact. OS upgrades tend to have router-wide impact rather than on individual interfaces in isolation. For example, when analyzing syslogs with respect to OS upgrades, we aggregated them into router-level by summing up the event counts across all the interfaces. We are now looking into applying change correlation at the spatial granularity level of per-interface.

Q: What time granularity do you use?
A: Right now, we operate on the time-scale of days. We construct a time-series on the aggregation level of per-day and then run change detection on top of it. We are now going to apply MERCURY on the granularity of hours.

California Fault Lines: Understanding the Causes and Impact of Network Failures

Daniel Turner (UC San Diego), Kirill Levchenko (UC San Diego), Alex C. Snoeren (UC San Diego), Stefan Savage (UC San Diego)

Motivation: Failure is a reality for a large network. Achieving high availability requires engineering the network to be robust to failure. Designing mechanisms to effectively mitigate failures requires deep understanding of real failures. Collecting comprehensive failure data is difficult. Many networks consider data proprietary so access to network data is limited. Some networks cannot invest time or capital. Network Failure History is a time series of Layer-3 failure events i.e., for each link a set of state transitions between up and down and annotate with "What caused the failure?", "What was the impact of failure?"

Existing Approaches and Limitations:

- Syslog
  - Describes interface state changes
  - Syslog messages are sent from routers to a central server using UDP and messages may lost
- Router Configuration Files
  - Maps interfaces to Links
  - Configuration files are logged intermittently and
  - Configuration files do not describe layer 2 topology
  - Operational announcements are written by humans so categorization is subjective

In these methods data is not intended for failure reconstruction. It is also confirmed that 1% of events mentioned in announcements are not in syslog.

Contribution: Methodology to reconstruct failure history of a network using only commonly avaialable data and no need for additional instrumentation. Apply to analysis of a production network.

Q: Have you looked at distribution of failures in the network? At the backbone or user, etc.?
A: Yes, there are several graphs in the paper devoted to answering this question and we have separated the failures into those on the backbone links, customer access links, and the high performance backbone.

Q: View from routing protocol in router estimate. How much more information do we need to be better for operations apart from syslog and from routing protocol? Can you elaborate on what issue has what impact on the data plane?
A: I don't know what extra data source would be more helpful in determining exactly when packets failed to reach their next hop. I would say that the routing protocol messages are not a bad source, they just underestimate the length of a failure.

Q: Clearly not all of your "failures" are due to maintenance and other planned activities. Have you looked at how your data set changes if you remove these planned failures?
A: The best indication of planned vs. non-planned failures would be looking at the scheduled tag from the emails. However, we

have not attempted to re-analyze the data removing these scheduled events.

Q: Are the failures mostly intra-domain failures?
A: All of the links we studied have both end points in CENIC controlled devices, so, yes, they are all intra-domain failures. Now in some cases the links may be a single point of failure between CENIC and a customer, but we do not look at links where only one endpoint is under CENIC's control.

Q: Are you seeing that upgrades are dominant cause of failures?
A: Upgrades are certainly one cause of failure, but I don't think it would be fair to call them the dominant cause.

Q: If you have the timeline of failures how often do you see concurrent failures?
A: We do see concurrent failures; often all of the concurrent failures will share a router. We have specific numbers in the paper.

Q: What source of information would be the most helpful in determining the impact of a failure?
A: I believe that link utilization information for the entire network would be very helpful in understanding how much impact a failure had on the end user. Currently when a link fails we can't tell the difference between it carrying no traffic and being fully saturated. Further, even if we could for just the link that failed, we would likely need to know the utilization of the remaining links to understand if the impact affected the end user.

Q: Are all the failures you considered inter-domain failures?
A: While we see failures that isolate a customer from CENIC, the individual link failures that caused the isolation would all be intradomain failures.

Session 10: Novel Technologies for Data Center Networks

Report by Sujesha Sudevalayam, IIT Bombay (sujesha@cse.iitb.ac.in)

c-Through: Part-time Optics in Data Centers

Guohui Wang (Rice University), David G. Andersen (Carnegie Mellon University), Michael Kaminsky (Intel Labs Pittsburgh), Konstantina Papagiannaki (Intel Labs Pittsburgh), T. S. Eugene Ng (Rice University), Michael Kozuch (Intel Labs Pittsburgh), Michael Ryan (Intel Labs Pittsburgh)

Problem:

- Traditional networks not enough, so FatTree, BCube but requires large number of devices and wires, hard to construct, hard to expand.
- Electrical packet switching versus optical circuit switching. Electrical packet switching network for low latency, and optical for high capacity transfer.

For example, MEMS optical switch using MEMS mirror array in different orientations for transfer, can switch in less than 10ms, but drawback is that it can't do packet granularity. However, this is promising despite the slow switching time, since packet granulartiy may not be necessary.

Design requirements/challenges:

- Control plane: switch needs to collect traffic demands, need to config based on it
- Data plane: traffic demultiplexing, plus optional optimization technique

c-Through:

- Central control to manage circuit configuration, no modification to applications and other switches, end nodes for traffic management
- Per-rack traffic demand vector for each server, this makes it transparent to applications, packets are buffered per-flow to avoid HOL blocking. Why buffers are useful – 2 uses – (i) traffic demand estimation & (ii) pre-batch data to improve utilization.
- Servers send vector to controller, who creates traffix matrix. Edmonds algorithm used to compute optimal configuration. Controller configures the switch consequently, so this is a centralized approach.

Evaluation:

- 16 servers with 1Gbps NIC, emulate a hybrid network
- Links bundled to form 4Gbps links, for each rack. But not always available. Controller emulates circuit switch behaviour.
- During reconfiguration period, 10ms, no host can use the optical paths.

Experiment with loosely synchronized all-to-all (MapReduce) and tightly (MPI-FFT) synchronized.

Observations:

- TCP can exploit dynamic BW quickly. 1000Mbps, ramps up from 100 to 200 within 10ms, and throughput stabilizes within 100ms.
- For mapreduce, overview: has many mapper and reducer stacks. 64MB block size for data loading, but during the data shuffling the blocks requests can be batched, not the others. Sort 10GB random data, used Hadoop for experiment. Full biseciton bandwidth is much faster than electrical, with reconfiguration interval 1 second, c-through achieves close to full bisection bandwidth for increasing block-sizes. When buffer fixed at 100MB and interval varied, c-through is good for lower intervals rather than higher, at higher just 25
- Yahoo gridmix benchmark: 3 runs of 100 mixed jobs web query, web scan and sortin g. 200GB of uncompressed data, 50 GB of compressed data. Even here, full bisection (optimal) is great, c-through is very close, so very promising for even applications wih all-to-all traffic.
- Future: power efficiency, scaling of hybrid data centers, making applications circuit aware.

Q: Surprised with claim of TCP works fine...? DCTCP would be better?
A: Faster rack TCP

Q: Do you think fibre cuts are possible, causing failures?
A: We haven't considered right now. Circuit will be recomputed, to connect it back.

Q: Control loop every 35 seconds fine for MapReduce, 100MB long transfers and 1 second, wouldn't you need to run the control loop a lot faster? Could you get bad behaviour due to oscillations, switching flows back and forth between electrical and optical?
A: We welcome MPI for smaller size, for that we need true traffic, we haven't seen such observations.

Q: What is the cost of optical technologies?
A: Cost issue is discussed in paper, higher than normal business

rates, larger scale switch doesn't increase per-port cost significantly. Cost has also been dropping very quickly.

Q: Is it a mature technology?
A: Yes, many companies are selling it today.

Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers

Nathan Farrington (UC San Diego), George Porter (UC San Diego), Sivasankar Radhakrishnan (UC San Diego), Hamid Hajabdolali Bazzaz (UC San Diego), Vikram Subramanya (UC San Diego), Yeshaiahu Fainman (UC San Diego), George Papen (UC San Diego), Amin Vahdat (UC San Diego)

Problem: Hybrid with optical circuit switch.

- Cost for electrical and optical: $500/port
- Rate for electrical versus optical: 10Gbps versus "rate free" (can support any data rate)
- Power for electrical versus optical: 12W versus 240mW
- Transceivers for electrical and optical: required versus NO
- Switching for elec versus optical: per-packet versus 12ms

Optical: by positioning the mirrors, it can send signal to different ports, can implement crossbar, does not decode packets, needs external scheduler, powerful with WDM. Wavelength Division multiplexing (WDM) gives each electrical transceiver a different code, sends the signal to WDM MUX, sends it to optical circuit via superlink (because it is super!). WDM DEMUX can demultiplex back to electrical switch. So, 8 electrical switches can be replaced by single optical switch. c-through used buffering to induce stability, here inherent stability is used.

Components in the system hierarchy: thread, process, server, rack, pod, data center. Higher up, more chance that they are communicating. Targets are modular data centers with N pods, k-ports, costing 6 million dollars, 100kW power, 6500 cables. Helios has 2.8x less million dollars, 6x less power, 4.7x less cables.

Solution/Approach:

- Electrical Pod Switch (EPS), Optical Control Switch (OCS)
- OCS has circuit switch manager, each communicating with every pod manager
- Control loop has to estimate traffic demand, hard problem, cant just ask the network, have to guess
- Compute optimal topology for maximum throughput
- Program the pod switch and circuit switch
- Will the flow become bigger if given bigger capacity, is it an elephant flow? Find out the bottlenecks in the network. Measurements are biased by current topology, so pretend that all hosts connected to an ideal crossbar switch and compute the max-min fair bandwidth fixpoint. This will give an estimate of how much each flow wants to talk to another.
- After getting pod to pod demand, edge weights represent interpod demand, algorithm runs iteratively for each circuit switch. Edmonds algorithm, to compute optimal topology.

Evaluation: Traditional network versus Helios network

- Network is vastly over-provisioned, so hosts couldn't saturate it, so created several traffic patterns. 190Gbps peak on traditional network. Some valleys are due to hash collisions.
- First time in Helios, got very bad performance, 160Gbps peak but only 43Gbps, packet switch had port debouncing, when packet arrived, makes note and sleeps and wakes many times, and after many times, thread enables layer 2 link, since makes sense to ensure that the link is really up. Since such feature is not needed, this was disabled and got better performance: 87Gbps.
- 600ms to initialize, so still not high enough even without debouncing. So, disabled EDC, 142Gbps was obtained. Traditional network still was better since same traffic cannot be routed through both circuit and packet switch.
- As stability in traffic patterns increases, throughput increases. Dividing level is 2 seconds, below this the throughput starts to drop off.

Q: How will traffic estimation work on different levels like data center, pod, rack, etc?
A: Measure at lowest level, and aggregate as we go up. Could lose information.

Q: At data center, maybe you have more aggregation so it is less of an issue, but lower in the stack, do you see the bias become more significant?
A: Talk offline.

Q: How many are optical and how many are electrical switches, depends on the traffic you get later. You have to predict how much traffic is stable and not?
A: Yes, traffic patterns keep changing, build hybrid electrical/optical switch, will be available in future.

Q: Quickly bottlenecked by I/O. What about about application limited flows?
A: Elephant flow classifier will handle that part. That is separate from the applications, many algorithms available for elephant flow classifiers.

Q: Slow RTTs, what will be impact on the ability of end-to-end flows to open the congestion window, given that you are not buffering on the path?
A: Not clear that our approach can give the performance, cannot work unmodified. We haven't applied it to the scenario of communicating between data-centers.

Scalable Flow-Based Networking with DIFANE

Minlan Yu (Princeton University), Jennifer Rexford (Princeton University), Michael J. Freedman (Princeton University), Jia Wang (AT&T Labs – Research)

Problem: Flow-based networking makes network easy to manage, support fine-grained policies but scalability is a challenge. Flexible policies: access control, customized routing, measurement for detailed HTTP traffic statistics. For this, we need flow-based switches which perform actions based on rules, actions like drop, forward, count, etc. Rules can be described in a flow space, like a matrix. Flow space can be 5D or even more — source address, source port, destination address, destination port, etc.

Challenges:

- High-level policies in management system, enforce low-level rules in switches
- Large number of hosts, switches and policies.
- Limited TCAM space, support host mobility, no hardware changes to commodity switches should be done.

How to handle: Pre-install the rules in the switches. Controller installs rules in switches, packets hit the rules in the switch, and are acted upon. When host moves, has to go to diff switches accordingly. Challenge: All rules cannot be stored in all switches (which is needed for mobility).

Alternate approach: Install rules on demand — initially switch has no rules, switch buffers the packet, and sends packet header to controller, controller installs the rules and then packet is acted upon. Challenge: Delay of going through the controller, switch complexity increases because FIFO buffer cannot be used since packets need to be continuously serviced, and misbehaving hosts can cause trouble.

Solution: Combine proactive and reactive approaches — DIFANE wants to keep all packets in data plane to get higher throughput. In first stage, proactively generate the rules. Centralized controller partitions and distributes the flow rules and sends to different authority switches and also sends the partition information. So, the appropriate authority switch for each packet can be found. Authority switch will cache the rules and keep packet in data plane. Other switches redirect to the authority switch, and send rules to the ingress switch also, for caching. So next time, the ingress switch can itself apply rules from the cache. If the caching algorithm is very slow, then packets could still get re-routed to the authority switch.

Improvement: Partition by wild-card rules, distributed directory service but not DHT, hashing doesn't work. Packet redirection and rule caching — if it doesn't match cache rules, then match with the partition rules and redirected to authority switch. In general, one switch can be ingress and authority switch at the same time, so it will have 3 sets of rules — cache rules, authority rules, and partition rules. The 3 sets of rules are in TCAM, cache rules have highest priority — packet can get processed directly, parition rules have least priority.

Developed Prototype: built with Open Flow switch, software modification needed only for authority switches — since it has authority rules and cache manager.

While caching wildcard rules, cannot simply cache matching rules R1 $>$ R2 $>$ R3, generate new rule covering the excess space and cache the new rule. Problem becomes harder when there are multiple authority switches in the network. To avoid cache conflicts, replace R1 with 2 independent rules, so after partition, we will generate more rules.

Challenges:

- How to partition to reduce number of rules?
- How are network dynamics handled? Policy changes at controller, causes timeout at cache, authority rules change, and no change to partition rules
- When topology changes at switches, no change in cache or authority rules, only parition rules need to change.
- Host mobility causes timeout in cache

Evaluation: kernel-level, click-based, open Flow switch. Compare delay and throughput — NOX vs. DIFANE.

- Delay evaluation: 4ms in DIFANE compared to 10ms
- Peak throughput: one authority switch, single-packet flow to see the key difference; ingress switch is bottleneck for NOX but not for DIFANE, because NOX centralized controller hits bottleneck

Q: How many authority switches do we need?
A: Depends on total number of rules and the TCAM space in these.

Q: Distributed or centralized?
A: DIFANE is in the middle, only controller is in center, and switches are the distributed directory of rules.

Q: Which switch has such small size like 160KB?
A: If it is smaller, then more switches needed.

Q: For IP forwarding, how would you choose the rules to put in the TCAM? How bad is your performance because of caching in the switch and too much dynamics?
A: When cache misses, longer path is still in data plane.

Q: With control plane, it can scale better, why stick to fast path?
A: Discuss offline.

Q: Experiment to find out the ideal cache replacement policy?
A: Did not do that, there is literature already.

Q: Good for handling static rules. What do you think about being able to handle more dynamic rules? Time of day? Secondly, how can it handle such dynamic rules?
A: What kind of dynamic rules, only traffic engineering uses dynamic rules to get updated state of network. Controller of network changes, and then push rules to authority switch. DIFANE won't have much gain from traffic engineering, all other rules like access control are static, even measurement rules can be accumulated later.

Q: Cost for authority switches, why not make all authority?
A: They are just normal switches, depending on memory size. Will need more redirection. Depends on how much to divide, too much division can cause problems.

Q: Is the cost same?
A: Yes, only thing is memory.

Q: DIFANE scales better, but makes Internet traffic more predictable?
A: Don't know which path the packet is actually taking. We can still know location of authority switch thus learn of possible paths.

Session 11: Social Networks

Report by Nishanth Sastry, Cambridge (nishanth.sastry@cl.cam.ac.uk) and Gurmeet Singh, IIT Guwahati (gurmeet.iit@gmail.com)

An Analysis of Social Network-Based Sybil Defenses

Bimal Viswanath (MPI-SWS, Germany), Ansley Post (MPI-SWS, Germany), Krishna P. Gummadi (MPI-SWS, Germany), Alan Mislove (Northeastern University, Boston)

Sybil attack: fundamental problem in distributed systems; especially a problem with online services such as webmail, social network, etc. Several observed instances of sybil attacks — digg, e.g., allows users to boost popularity, which has been exploited by creating sybils.

Current Sybil defense approaches limit using some resource:

- Resources 1: certificate from trusted authorities (e.g., passport). Only one network, cyborg uses this kind of identification. Users tend to resist this.
- Resource 2: crypto puzzles - resource constrainment
- Resource 3: link scarcity - difficult to maintain links to good users if you are sybil. attacker is limited by # links to non-sybil users.

All of these make use of small cuts in social networks sybilguard, sybillimit, etc., use it. Link scarcity is a promising approach but has many unanswered questions. All make same assumptions — for instance, they use only existing social network information — but schemes work using different mechanisms. Is there a common insight across all these schemes? Is there a common structural property these schemes rely on? Answering this would help address how these schemes work and also what the limitations are.

Propose a methodology for comparing schemes; find all of them work in similar manner. How to compare schemes? Treating each scheme as a black box is not useful because this only gives one point of evaluation & output is dependent on specific scheme.

Proposed approach: Take a trusted node, and look at social network in relation to that node. Internally schemes can be viewed as assigning a probability to nodes (different likelihood of being a sybil). Can view scheme as inducing a ranking on nodes, the higher the rank, the lower the probability of sybil.

How do the rankings compare (looking at different schemes)?

- Observation 1: ranks don't match.
- Observation 2: there tends to be a distinct cutoff. after this point, the rankings match — this point matches the boundary of the local community around the trusted node. e.g., comparing sybilguard and sybillimit, for partition similarity which peaks at same point as cut-off varies. Similarly the value of community strength is smallest => it detects communities in the network.

Common insight across schemes: nodes within local community are ranked higher; ranking within and outside community is in no particular order.

Implications: good and bad news

- good news: leveraging community detection (well studied topic) to detect sybils. taking an off-the-shelf scheme gives similar performance.
- bad news: this suggests dependence on graph structural properties (size location, characteristics of local community).

Explore two implications. (1) Are certain structures more vulnerable? Two topologies - one more enmeshed; other has two distinct communities. If you introduce a new network of sybils, then it could be (from the perspective of a node in one community) that nodes from other community have lower ranking than sybils! Hypothesis - community structure makes identifying sybils harder.

Testing community structure hypothesis: 8 real world networks. Simulated attack by consistently adding sybils. Measure accuracy using ranking (accuracy = probability that sybils are ranked lower than non-sybils). As community structure (modularity) increases accuracy drops below 0.5 (i.e., random). In practice, real social networks have many small local communities.

(2) Can attacker exploit this dependence? Attacker's goal is to be higher up in the rankings (increase likelihood of getting accepted). Changing attacker strength: In one case, attacker can randomly link to all community. To strengthen this, attacker can target links in the community of trusted node. Give more control over link placement by allowing attacker to place links to higher ranked nodes. Test by allowing attacker to randomly link to top N nodes. As more control is given, ranking of all schemes tested becomes worse than random.

Moving forward, we can still use sybil defense to whitelist nodes — higher ranked nodes we can also use information beyond graph structure.

Q: Question about parameters you use in your experiments. In most of your experiments the number of attack agents that you use is comparable or even larger than the number of sibyl nodes that you introduce. And in our sibyl limit work we actually proved a negative result showing that under these parameter values no scheme can possibly work well. So in that aspect your results are consistent to our negative result. But on other hand since these algorithms are designed for cases where the number of sibyl nodes are greater than attack agents, have you done evaluations under those settings and how do you think results and conclusion will change under those settings?
A: We did evaluation for topology where number of sibyl is much more than attackers. We had the same finding that if we obtain node ranking then all these schemes are finding local community of the cluster node on top of ranking.

Q: From our facebook study we found a mixture of compromised accounts and fake accounts when they launch attacks. How do you detect them? Do you have any insight about it?
A: If you are compromising an account and the compromised account has lots of friendship links, this would violate the assumption that the attacker has limited friendship in a good part of the network. And in those cases all these schemes won't work. It'll be very difficult to detect sibyls in that case. But in the other case when compromised account has limited number of links then all these work.

Q: In real life we observe that there are more fished accounts. In that case none of these schemes work. But when you talk about interactions...
A: If it was information from higher layers then it would definitely help in these cases.

Q: An extension of this, that sibyls can form not one community but multiple communities. Then this again becomes more difficult.
A: Yes, they can form multiple communities. So what these schemes are essentially trying to do is try to find a sparse set of links which is the boundary of the community. So no matter what sibyl topology is, we should be able to find it as long as there is limited number of attack links.

Q: (suggestion) If you insist that there be interactions (rather than just take every social edge that exists) then it can help bound whether some edge is an attack edge or a valid edge.

The Little Engine(s) that could: Scaling Online Social Networks

Josep M. Pujol (Telefonica Research), Vijay Erramilli (Telefonica Re-search), Georgos Siganos (Telefonica Research), Xiaoyuan Yang (Telefonica Research), Nikos Laoutaris (Telefonica Research), Parminder Chhabra (Telefonica Research), Pablo Rodriguez (Telefonica Research)

Scalability is a pain — designers have a dilemma: wasted complexity and long time to market vs short time to market but risk of death by success. We want transparent scalability: we get hardware scalability with cloud; but we don't have it in applications. We already have elastic resource location for the top 2 tiers. Elastic resource location for the data layer is the bottleneck.

For scaling the data backend, current options are:

- full replication (load of read requests decrease with number of servers; state does not decrease state with number of servers. it maintains data locality as well.
- horizontal partitioning or sharding: load of read requests decrease with number of servers; state decreases with # servers.

Maintains data locality as long as splits are disjoint. This is OK for e-commerce because users X and Y have separate data. But not good for social networks. In OSN, one users outbox is multiple people's inboxes, etc. This is an active area: hermes (SoCC), volley (NSDI) have looked at this kind of problem. This solution focuses on OSNs.

A bit of background, or OSN operations 101. Have all data from user distributed to friends of user. Operations:

- fetch inbox from server.
- fetch text where key in R [from server]

If all data is in one server, we have locality. If not in one server, then problematic — relational databases are not good for selects and joins across multiple shards of database. Key-value stores are more efficient (multi-get primitives transparently fetch data from multiple servers). But it is not a silver bullet — you lose SQL, abstraction from data operations; suffer from high traffic, eventually affecting performance (incast issue; multi-get hold; jitter; latency dominated by worst performing server).

Solution for maintaining data locality: random partition + replicate nodes who have friends in other places. This can end up with a full replication! If you split using small cuts in social net, you get advantage. Additional requirements:

- solution must be online [incremental] algorithm — dynamics of SN is important but there can be add/remove of server and other system dynamics. online partitioning is sensitive to initial conditions.
- fast [and simple]: and reactive to above changes
- stable (avoid cascades)
- effective — optimize for min-replica (i.e., NP-hard) — while maintaining a fixed number of replica per user

Algorithms try to minimize inter-partition edges, whereas we are trying to minimize number of replicas made across network — so can't use graph partition.

Evaluated on real OSN data (Twitter/Orkut/Facebook). Other algorithms tested include random partitioning, MO and METIS. SPAR has a lower replication overhead than the other algorithms, with only 22% overhead over the replication constraint.

Paper has evaluation details, only one point presented here: How many replicas do we generate? Others in paper, e.g., how many movements do we do, etc. Twitter: 16 partitions has 2.44 replicas per user (on average). Teplication with random is too costly. Using MO+ and METIS are not as bad but still worse.

SPAR architecture: Assume 3 tier application. We put spar middleware (data store specific) that intercepts messages. Also have a SPAR controller that manages partitions and directory service. Main point: data store and app are not aware they are running SPAR. Can implement it on mysql, memcache, cassandra, etc.

SPAR in the wild: take an application never designed to be scalable, designed to go on single servers. Take twitter data as of end of 2008, then put it on 16 commodity desktops and test SPAR on top of mysql and cassandra. SPAR on top of mysql: with full replications can serve 16 reqs/s. SPAR+mysql can come to 2500 req/s. SPAR allows for data to be partitioned, improving query ratios, etc. SPAR on top of cassandra: with vanilla cassandra 200 req/s; SPAR + cassandra = 800 req/s. Net bandwidth not an issue but network I/O and CPU I/O are.

Conclusion: SPAR provides means to achieve transparent scalability. Not necessarily for OSN, but applications using RDBMS.

Q: Let's say in Facebook I have different groups of friends. I want to show them some of my pictures, not all of them. So can I replicate a part of my pictures on one server and some on another server based on my groups of friends?
A: Yes. In the worst case you will have to replicate for all friends. Then if you want to set up a rule that all the set of friends are allowed to see the picture then you show that on the query itself. You have to guarantee this by defining the function of policies for that. Otherwise you will have to fetch the data from all the servers and that's what we want to avoid. Replication should be aware of that.

Q: You are doing partitioning based on linked structure. However, based on my observations about the way people live on the social networks, many links are basically non-interactive. For example, if I look at my Facebook wall there exist only a few friends whose updates come to the wall and so on. Have you looked at incorporating more dynamic interaction behaviours into your partitioning algorithm?
A: Yes, that is one option. But our approach is different. If a link is never used from replication perspective it does not matter.

Q: You could have a dynamic behaviour where you infer over time what is the expected chance of having an update from a friend and your overhead will be lesser than what you currently do. A follow up question, there is this open source distributed Facebook thing called Diaspora. Do you have any idea of what they are doing or if they are incorporating any of this, as this is relevant problem they must tackle?
A: Regarding Diaspora I only know from what I read in the news. I know it is completely distributed. I don't know if they have the idea of centralized or distributed system or not.

Q: I think the purpose of the partitioning algorithm may depend on the topology of the social network or degree of locality, and there is a lot of work to capture topology of networks. Have you considered applying that?
A: Yes, it does depend on the topology of the network. Replication depends both on the density of links and how close the links are. So more clusters will mean less replication is needed. The algorithm to be used has to be run on dataset of 41M users. Verification overhead is consistent in the network.

Q: A couple of observations. Firstly replication overhead would be much higher. Your data is of 2008 and social networks tend to grow more dense over time. Secondly in your formulation you have no notion of load per server. In social networks all nodes are not equally active. Their activity is power law distributed. This might end up in dedicating servers which are not used at all.
A: You are right in the consideration but in the paper you will see that we have a notion of load balancing using read and write patterns.

Q: think your comparison is on read fanout design. Guaranteed data locality is not always desirable for a lot of applications. I think you have not considered optimized approach where we do not replicate data on writes, which we call a write fanout design versus the read fanout design. That has not been compared.
A: Yes.

Q: When you partition the social structure, do you consider that there could be a rapid change in the topology?
A: We have an incremental solution so that only few replicas will help in case of such changes.

Crowdsourcing Service-Level Network Event Monitoring

David Choffnes (Northwestern University), Fabián E. Bustamante (Northwestern University), Zihui Ge (AT&T Labs – Research)

Internet is driven by services; user experience is a key benchmark, want to identify problems affecting e2e performance and do this transparently. Detecting network events — variety of existing detection approaches: internal lin/router monitoring; BGP monitoring, etc., but have no visibility into e2e performance, what the eyeballs are seeing.

Crowdsourcing event monitoring: Push monitoring to edge systems. If a large number of clients in single network see a problem then we have a problem. Use bittorrent client to do this. Our solution is NEWS. Has over 48k users.

Scalability: localization in time and space is what we want. To achieve scalability, passive monitoring and distributed detection.

Localization in time and space [online detection approach that isolates to network regions].

We have additional issues:

- privacy: don't want to reveal personally identifiable info of clients).
- reliability: end hosts are not controlled. need to gather reliabile information; potentially noisy info (use probability analysis to identify likely real events)
- adoption: needed because of crowdsourcing

Approach and architecture:

- local detection: passively monitor local perf info (signals), e.g., transfer rates, app specific (e.g., content availability in bittorrent)
- group corrobration: retrieve from distributed storage other events were published by neighbours. Likelihood ratio to distinguish network events from coincidence (this is inspired by use in medicine to test medicine effectiveness) Each user and any third party with access to the distributed storage, can see same info — in particular, an ISP can see this as well!

Evaluation challenges — participatory monitoring challenges — need large scale adoption; edge traces are rare. P2P apps are a natural fit. Used worldwide, generates diverse flows. BitTorrent consumes large bandwidth and vuze and mainline both allow extensibility and DHT.

Ono dataset for traces ($>$ 1 million BitTtorrent users) CEM (crowdsourcing event monitoring) case study. BT Yahoo provides confirmed events through a Web interface, so can confirm events. [there can be slight variation in published window] Local detection in BitTorrent. Peers monitor multiple performance signals — detect drops in throughput. Individual signals are noisy, uncontrolled, etc. If you use EWMA and averaging then you see some correlated trends.

Group corroboration: given locally detected events why would they occur at the same time?

- service specific problems (e.g., no seeder left); we can use app level information
- coincidence (e.g., noisy local detection); can use union probability
- problem isolated to one or more networks

Coincidence of network events: assumes that local events are independent. Intuitively for large n, coincidence becomes small.

Likelihood ratio: are detected network problems occuring more often than chance? compare coincidence (P_u) and network error ratio (P_e). Likelihood ratio = LR = P_e/P_u . LR $>$ ratio => network problem. Higher this ratio threshold, smaller detection probability.

In BT Yahoo, most events are no more likely than chance but some are transient spikes. Need to use a smoothing + likelihood threshold of 2.

Gold standard: false positives/negatives. Almost no ISPs publish this, so can;t get at ground truth. We have worked with ISPs like BT Yahoo, also work with ISPs under NDAs. It only works where we have coverage. With BT yahoo, 181 were detected by us. 68 events were reported. We detected 58 of them. North american ISP, subscribers $>$ 10K. Detected 50% of them. Robustness to parameter settings: robust to various detection settings, populations.

Service summary: NEWS implementation & deployment: 48k installs, 1k LOC for EWMA but lots more code for UI, user notifications. For group corroboration and localization, we use built-in DHT, etc.

Open issues: which network groupings are best (whois listings, topologies, ISP specific)? Where is ground truth? Crowdsourcing event labeling (newsight), can we apply these to other services like VoIP, video streaming, CDNs?

Q: I think your approach is really great because performance monitoring should be done at the application layer. I hope that you will extend that to other applications.
A: Thank you.

Q: Do you think for grouping it would be beneficial to make use of IP prefixing or geo-information location or you think these are too distributed?
A: I think it just depends on what network would be useful for isolating the problem. We are using BGP prefixes so I don't think that is a problem. Geo-location is possible. One option is ISP specific information, ISP tells what user support in the geo-location is and that would help to figure out where the problem is.

Q: This morning we had two talks on anomaly detection. Why is your technique different from their's?
A: The main difference is that we are at the edge of the network seeing the performance that the user is getting.

Q: But you are using moving averages as opposed to more sophisticated techniques we saw today.
A: The reason we are using moving averages is because one of the things we wanted to do is avoid the bunch of overhead on running the detection algorithm, moving averages does that. It is also well understood. We also tested other approaches. Generally we found a lot of them did not work because they assume there is a long continuous stream of information they could analyse. Where as in our case user session is not that long. It's long enough to use moving averages on them.

Q: One of the open questions that is probably missing is once you detect the event what do you do with them.
A: So now users detect the problem. Users and operators can see that there is a problem. The idea is that operators fix it. But we also need to do root cause analysis of the problem. We need to identify what the cause of the problem is so that we can go and fix it. That's the topic we don't cover in this work but it could be an important future work in this area.

Q: Maybe tie it with automated VoIP and making calls?
A: That would amplify the performance problems.