In VINI Veritas: Realistic and Controlled Network Experimentation

Andy Bavier Nick Feamster Mark Huang Larry Peterson Jennifer Rexford

Public Review by Nick McKeown

The biggest reason for reading this paper is that it's timely. It elbows its way into the middle of a community-wide debate about the merits of a programmable research platform (GENI), and the paper tackles a bunch of interesting questions that GENI will face.

VINI's goal is to enable researchers to create experiments in which they deploy their own routing software, use real end-user traffic, and are subject to network events, such as link failures. It addresses the problem of how to multiplex both logical and physical resources so that multiple Internet scale experiments can run simultaneously and in isolation in realistic environments. Realism is the goal here, and the paper walks us through the trials and tribulations of providing a virtualized infrastructure, in which each experiment occupies slices of programmable nodes interconnected by virtual links, and connected to end-users. At a high-level, think of PlanetLab nodes in backbone POPs, interconnected by virtual links (rather than part of an overlay), and connected to end-users who want to opt in to an experiment or service. VINI pushes the state of the art (moving past PlanetLab, RON, XBONE etc.) by offering more latitude to researchers at the routing level, isolation, the ability to create real, complex topologies and the ability to inject exogenous events, such as link failures. The idea is that researchers will create short-term experiments, or provide long-term services - e.g. a service in which they offer a more reliable network.

To decide whether VINI is a good idea it helps to decide if we agree with the value of a national or global-scale programmable platform for network research. The pros and cons of such an infrastructure are many, and it would be naive to characterize it as black and white. Like PlanetLab before it, VINI tries (and GENI will try) to provide a more realistic alternative to simulation and emulation of proposed network architectures. For years, the community had to rely on simulators, which now seem a little dated, and it's not clear who was convinced to adopt anything new based on *ns2* simulations; to have lasting impact always required additional validation steps, such as XBONE or homegrown prototypes. It is therefore appealing to test an idea on a platform that is fast enough to interact with real end-applications and with real end users. Better still, if we can create any topology, choose the distance between nodes, the behavior of the links, and give it the look and feel of a real nationally operated network then we can have more confidence that our results will extend to the network as a whole.

We have to ask if a programmable platform distributed over the whole nation is actually more realistic than backhauling end-user traffic to a central platform, like Emulab. A platform in a lab can be virtualized, sliced and shared among multiple experiments, can provide short-lived and long-lived experiments; and by implementing delays between nodes, you can create most of the important characteristics of a large-scale network. It's also easier to support, evolve, and grow; and probably easier to protect against malicious users. We have to ask how much is gained from spreading the infrastructure around: the topology of *truly* long-haul physical links is limited to the location of the POPs the nodes happen to live in (who is to say for sure that future interesting networks will have their switching centers in the same location as today?), and the network is only the size of the operator's network; not nearly the size of the whole Internet, with all its interconnected providers. To make a VINI network feel like the whole network means emulating a much larger network of long haul links, just like a centralized platform would do. And because VINI (or GENI) is itself a non-commercial network, we will be tempted to delude ourselves into thinking that an experimental architecture

running on VINI is representative of the experience of running a real commercial network.

For me, the bottom line is this: I believe that the main benefit of a distributed platform is that the connections from the experimental network to the end-user (or service) can be made shorter - and hence more realistic - than if their traffic is back-hauled to a central location. Is this benefit worth all the extra cost and complexity of spreading the network over the country or world? Yes, I believe it's worth it. I think we should try to make the research platform as representative of the way we use networks as we can, while always remembering that the likeness is imperfect. Too often we hear that an idea will work because the theory or the simulations say so, even though we know the assumptions or models are quite unlike reality. Likewise, we shouldn't fall into the trap of automatically assuming that something that looks promising on a programmable platform will also work well when deployed more widely. The main reason for doing it is that it can be a heck of a lot more realistic than the tools we had available in the past. In the end, our ability to have impact - a lasting measurable effect - is based on our ability to convince ourselves and others that our ideas will fly, then launch the ideas on a path to being used (possibly a long path with many improvements). A more realistic, but imperfect, infrastructure will help.

VINI is a head-start program for GENI; a classroom for those who will build GENI. VINI is also the next step in network testbed evolution. The major technical contribution is getting the right synthesis of existing ideas in one place to attack the problem – the bringing together of XORP, Click, UML, OpenVPN and PlanetLab is interesting and described well. As one reviewer noted, the idea of slices is implemented in Planetlab, and was in X-kernel work in the 1990.s; machines running multiple operation systems going back as far as IBM VM/CMS in the late 60s. The VINI approach to virtualization is not particularly novel, and a reasonable networking engineer given the problem statement and some coffee would have come up with something similar. This includes interfacing with end hosts, and using UML for virtualizing networking components etc. The real contribution is articulating the requirements and properties of such a test-bed, then figuring out how to correctly synthesize existing tools.

Once VINI is available, it will be interesting to see how realistic the experiments are: How much real closed-loop traffic (rather than just mirrored traffic) will be routed over VINI and made available to experiments? What will the incentive be for users to direct their traffic over VINI if the network performs less well than a dedicated (possibly existing network)? The benchmarks in the paper suggest around 10-50% as fast, so it might be challenging to deploy services people will use. How precisely scheduled will the resources be in practice (e.g. CPU and link capacity) - particularly when traffic runs over shared links?

In the end, there is good reason - from the widespread use of PlanetLab - to believe that if you build it they will come. It is clear that the research community has benefited enormously from PlanetLab, and VINI has the potential to build on this momentum to provide an even more valuable platform to facilitate a new class of interesting research.

Revisiting IP Multicast

Sylvia Ratnasamy Andrey Ermolinskly Scott Shenker

Public Review by Bruce Davie

Multicast could be the poster child for the irrelevance of the networking research community. Few other technologies (quality of service springs to mind) have generated so many research papers while yielding so little real-world deployment. This paper tackles the failure of multicast to achieve its promise by proposing a radical rethinking of the mechanisms for delivering interdomain multicast at the scale of the global Internet.

Multicast has, from day one, been dogged by problems of scalability. This paper does not fix the scaling problems so much as squeeze the toothpaste tube, causing the scaling problems to move from one place to another. But the result could actually provide a much better set of tradeoffs, as the authors demonstrate that the scaling-related costs of their scheme could be quite tolerable with today's hardware. The scaling costs are also borne most heavily by sending domains, which happen to reap the most benefits from multicast.

Interdomain multicast faces other problems besides scalability. These include undesirable fate sharing among mutually distrustful ISPs, and an overabundance of protocols to address the first two problems. All of these issues are tackled in this paper.

Unlike Holbrook and Cheriton's '99 SIGCOMM paper that introduced source-specific multicast (SSM) as a solution to the above-mentioned challenges, this paper tries to preserve the "classic" multicast model in which any host can send to a group. The authors argue that there are enough applications that can benefit from the classic model to warrant a re-examination of the mechanisms needed to provide it.

The mechanisms that are proposed here, referred to as "Free Riding Multicast" (FRM), definitely deserve the label "outside-the-box thinking" (a term used favorably by more than one PC member). The first piece of mechanism is an extension to unicast BGP to carry group membership information (this is the free-riding part). For each prefix that is carried in BGP, the set of multicast groups with active receivers in that network is added to the BGP advertisement, encoded as a Bloom filter. Because Bloom filters produce false positives, an additional mechanism, which amounts to on-demand access-control lists, is needed to "prune" traffic that is mistakenly sent to networks with no active receivers.

With this "simple" extension to BGP, a border router in the domain of a sender to a group can construct an AS-level tree to reach all the domains with active receivers. Routers in downstream domains are unable to determine how much of the tree they are responsible for—just because I have a path to prefix X doesn't mean that I'm on the path from the source to prefix X. So the border router in the source domain *source-routes* the packet. The authors propose a fairly efficient way to encode the entire AS-level tree in a shim header.

Forwarding packets bears no resemblance to any sort of existing multicast (or unicast) forwarding. At the source domain, a router determines the appropriate AS-level tree by looking at the information it learned from BGP for this group. It the build the shim header that encodes the tree. It can cache this header, so only the first packet sent by this domain to the given group incurs the cost of the lookup and of building the header. At transit domains, routers examine only the shim header, and forward to neighboring domains based on the tree that is encoded in the shim. Interestingly, the amount of state that is stored in a transit router depends only on the number of AS neighbors it has—it is completely independent of the number of multicast groups or anything else related to the usage of multicast. This is perhaps the most striking advantage of the FRM approach.

Not only does the shim header consume bandwidth, but the design that makes it efficient to process and fixed in length means that it sometimes fails to capture the entire tree in one header. In this case, packets have to be sent more than once on some links, with different shim headers capturing different sub-trees. Thus some of the bandwidth gains of ideal multicast are lost.

A solid aspect of the paper is its evaluation of the costs of FRM. The evaluation makes assumptions about the number of groups, number of members, and rate at which members join and leave groups which would seem to be consistent with wild success of multicast. (Groups with 10 million members spread through every AS in the Internet, for example.) There are many costs to consider: route processor memory, line card memory, and bandwidth inefficiency are the main ones. The shim header, at 100 bytes in this evaluation, seems like a big deal if packets are short (e.g. for interactive voice). A striking result is that border routers would need on the order of a few gigabytes of extra route processor memory for the group membership information that is "free-riding" with BGP. Whether this is reasonable or excessive probably depends on one's perspective—it is in the range of what very high-end routers support today.

Finally, there is also an implementation of FRM, demonstrating that it actually can be made to work, followed by a useful discussion of how ISPs might use FRM to gain more control over multicast—another of the perennial problems. Since source domains bear much of the cost in this scheme, it is handy that FRM provides a means by which an ISP could control which customers get to send multicast and even to charge based on group size.

The big question raised by this paper in my mind is this: does the classic any-sender multicast model offer enough advantages over SSM to warrant all this complexity and cost? For that matter, can the segment of the application space that needs something other than SSM be adequately served by multicast overlays? We can also reasonably debate whether there is such a thing as a "simple" extension to BGP at this stage in the Internet's evolution.

Nevertheless, this paper is novel and well thought out, and uses both implementation and numerical evaluation to show that an entirely new approach to multicast could perhaps be realized. Even if the exact approach described here never sees deployment in the public Internet, the exploration of a new point in the design space is welcome, and suggests that inter-domain, network-layer multicast is not dead yet.

Designing DCCP: Congestion Control Without Reliability

Eddie Kohler Sally Floyd Mark Handley

Public Review by Steven Low

Sigcomm has been a primary venue for the publication of congestion control protocols for IP networks, starting from the seminal paper of Van Jacobson in 1988. Like Jacobson.s paper on TCP Tahoe, this paper is not a mere proposal, but describes a protocol, DCCP (Datagram Congestion Control Protocol), that is ready to be deployed . DCCP has already been approved for IETF standardization. Unlike most congestion control papers that typically focus on algorithms and implementations and their performance evaluation, however, this paper is about design choices and lessons learnt.

The goal of DCCP is simple enough. It is to provide congestion control without packet loss recovery for such applications as multimedia streaming that value timeliness over reliability. Depending on how one counts, numerous measurements have consistently shown that more than 90% of Internet traffic runs on TCP. Unfortunately, the congestion control algorithm in TCP is not ideal for multimedia applications, for two main reasons. First, TCP provides 100% reliability, i.e., every lost packet is retransmitted until it is correctly received at the receiver. This can be wasteful if the retransmission attempts delay the packet so much that it is out-of-date when it arrives safely. Second, the congestion control algorithm couples congestion control with loss recovery. This is a good feature when Jacobson developed the Tahoe/Reno algorithms in the late 1980s, but becomes a problem as wireless components increasingly become an integral part of the Internet. When packets are lost due to wireless effects, such as bit errors, handoffs, fast channel fading, as opposed to congestion (i.e., sharing of scarce resources among competing flows), the right response is to retransmit the lost packets, but TCP also cuts its rate unnecessarily, leading to inefficient use of channel and violation of application requirements.

Both of these problems . proliferation on the Internet of real-time applications and wireless infrastructure . become important only after the original design of TCP congestion control. Instead of evolving TCP congestion control to address these problems, there is a growing temptation to run real-time applications over UDP. This is however unsatisfactory, both because UDP does not exercise congestion control and can therefore be unhealthy for the Net and because UDP often has difficulty traversing firewalls and NATs. DCCP is developed to be the solution.

Decoupling congestion control and loss recovery turns out to be subtler than it might appear. Reliability in TCP provides a simple structure in which liveness, congestion control, flow control, and other management functions are seamlessly integrated. Through explaining how these functions are redesigned without the reliability semantics, the paper also provides a deeper understanding of some of TCP.s subtler features. For instance, an ack in TCP is cumulative, i.e., it acknowledges all packets that have been received except the packet with the current acknowledge number. Since DCCP is unreliable and does not retransmit lost packets, ack cannot be cumulative. This requires additional information, in the form of DCCP options, to communicate from the receiver to the sender which packets have been received, in a way analogous to SACK. It also necessitates periodic synchronization and additional communication between the sender and receiver to keep their window size in synchrony and to bound the receiver state. The authors have taken advantage of recent protocol advances to equip DCCP with other features from the outset, such as feature negotiation, support for mobility and multihoming, and robustness against denial-of-service attacks. It also provides a framework for experimentation and deployment of multiple and new congestion control algorithms. Currently, only

TCP-like algorithms and TFRC have been defined, both being loss-based. Interestingly, it also includes mechanisms to differentiate delays due to network and that due to receiver processing. This allows a better estimate at the sender of network congestion as measured by round-trip (network) delay, and can be useful for delay-based congestion control algorithms should they be included in DCCP in the future.

The paper gives a clear account of how these design choices have been made. Reading a protocol specification is often difficult with its myriads of details that must be meticulously followed. A logical and careful explanation of the underlying design rational will be a great help to engineers who want to implement the DCCP specifications. This paper does the job admirably and will be a useful reference for years to come.

.. at least for readers who have already decided to implement DCCP. For others who want to explore different options for transport protocols for multimedia applications, this paper offers limited help. As some of the reviewers pointed out, the (submitted version of the) paper lacks a serious performance evaluation of DCCP and a comparison with other approaches.

For instance, there have already been a few transport protocols in deployment such as RTP, RTSP, SCTP, UDP-Lite, and people have also been implementing congestion control in the application layer. Almost all of these approaches are based on UDP, and thus have their disadvantages. It is unclear however whether the pain is strong enough to propel a wide spread deployment of yet another new transport protocol, and the readers would surely have benefited from the deep insights of the DCCP designers on this debate. For the authors. analysis of these alternative protocols, the readers have to consult reference [17] of the current paper.

Another alternative which may seem far fetched at first sight is to evolve TCP into a form that is also suitable for multi-media applications. Recall that the main problems with TCP congestion control is the insistence on 100% reliability and the coupling of loss recovery and congestion control. One could argue that insistence on 0% reliability, as in DCCP and most UDP-based approaches, may be as undesirable as that on 100% reliability. Most streaming applications buffer for a few seconds before playing out in order to absorb network fluctuations. Since round-trip times are typically less than 500ms, most applications will benefit from a few retransmission attempts. Time-lined TCP attempts to add partial reliability to TCP, but as pointed out in the paper, application requirements are much more diverse than can be supported by a simple deadline. DCCP has chosen unreliability over a potentially more complex protocol that supports partial reliability. This is probably because of the important design goal of minimalism . it would have been beneficial to document the rationale of this major decision. The second problem of coupling loss recovery and congestion control can be easily solved by adopting a delay-based congestion control algorithm.[1] It will also provide a smoother throughput than TCP-like congestion control, which is often beneficial to streaming applications. A more detailed discussion on these high-level design choices, as opposed to implementation choices, would have also been useful for protocol researchers and engineers.

[1] Incidentally, DCCP argues that one should respond to reverse-path congestion. This is hard to do without additional communication between the sender and the receiver because of the cumulative nature of acks. A delay-based algorithm naturally reacts to reverse-path congestion.

Y. Cheng J. Bellardo P. Benko A. C. Snoeren G. M. Voelker S. Savage

Public Review by Jitendra Padhye

802.11-based wireless LANs (WLANs) are deployed by many business and universities. These networks are known to suffer from problems such as hidden terminals and interference from external sources of noise. The performance of these networks is further affected by factors such as the placement of the access points (APs), the channel assignment, and the way the radio signals propagate through the building. While the impact of many of these factors has been studied individually, their combined impact and their interactions in deployed WLAN networks have received much less attention.

The biggest obstacle in studying the behavior of deployed WLANs is the difficulty of creating a comprehensive view of events taking place in the network. Due to the nature of the wireless medium, and the specifics of the 802.11 protocol, it is usually necessary to combine observations gathered from multiple observation points to generate a complete picture of the events in the network.

Consider, for example, a wireless client communicating with an AP. Imagine that a monitor placed near the client observes the client retransmitting a data frame. To determine the cause of the loss of the original transmission, we must distinguish between two possibilities: either the AP failed to receive the original transmission correctly or the client and the monitor failed to receive the acknowledgment generated by the AP. We can answer this question if we have another monitor deployed near the AP, and we combine the traces generated by the two monitors.

The authors this paper have built a system, Jigsaw, that monitors a WLAN using multiple monitors and combines the traces to generate a comprehensive view of events taking place in the network. By using this system, the authors study their WLAN network for a period of 24 hours, and present several interesting observations.

There are three key challenges involved in building a system like Jigsaw. The first challenge is determining the number monitors needed, and their placement. The authors know the location of every AP in their network. They deploy multi-radio monitors close to each one. Such dense deployment of monitors ensures comprehensive coverage, but results in the second challenge: managing the scalability problems involved in the data collection. The Jigsaw system consists of 39 monitors, each monitoring 4 channels. The monitors themselves do very minimal processing of the raw data. In effect, each monitor simply writes every frame heard on the air to a central data store. Since each frame sent on the wireless network can be heard by multiple monitors, the amount of data generated is quite large. The authors use compression to reduce the amount of data sent over the wired network. Still, the peak load generated by the monitors on the wired network can be as high as 80Mbps, and these bursts can last up to two minutes. The third challenge is devising techniques for merging the traces obtained from multiple monitors. There are several hurdles to overcome: time synchronization, ambiguities created by frame loss and non-standard behavior of certain hardware. In addition, the authors also want to complete the merging in near real-time.

To synchronize the time across traces gathered by multiple monitors, the authors leverage the fact that most frames are overheard by multiple (but not all) monitors. By identifying such frames, the authors find common reference point to synchronize groups of monitors. The sets are further synchronized by finding common members among various sets.

The next step after the clock synchronization is to merge the traces to create a unified view of the frame exchanges. Since each frame is observed by multiple monitors, the authors combine all observations of a given frame to to form a *jframe*.

The *jframes* are then grouped together into into *exchange records*. For example, the authors group together RTS/CTS exchanges along with transmission of the corresponding data frame, retransmission attempts (if any) and subsequent acknowledgment (if any) into a single record. For this purpose, the authors use a finite state machine to represent the 802.11 protocol.

However, due to the nature of the 802.11 protocol, it is sometimes not possible (even with multiple monitors) to conclusively decide if the data transmission was successful. The authors cleverly leverage information from transport layer (e.g TCP sequence numbers) to resolve these ambiguities whenever possible. The sequence of *exchange records* provides a comprehensive view of the network activity.

The authors monitored their department's WLAN network over a period of 24 hours using the Jigsaw system. They present several interesting observations. For example, they show that a few clients in their network suffered from interference from other simultaneous transmissions. This particular observation demonstrates the key strength of the Jigsaw system - the ability to combine observations from multiple vantage points to create a unified view. The authors have also presented some results related to the use of 802.11g protection mode, and TCP loss rates.

One of the concerns expressed by the reviewers was that the paper should have included more results of this type, instead of describing the nitty-gritty details of system.

My main concern is the scalability of the system, since it records every frame observed on the air. As discussed earlier, the system generates a lot of traffic on the *wired* network. While the Gigabit Ethernet network in the UCSD CS department is capable of easily handling this traffic volume, one wonders whether the monitors could have done a little bit more processing to substantially reduce the amount of data submitted to the central store. For example, if a monitor were to overhear a data frame, followed by the corresponding ACK, perhaps it could have simply written an exchange record to the central store, instead of writing out the full frame. Full frames would be written only if the monitor can not resolve all the ambiguities.

I am also concerned that transport layer information will not be available if WLAN traffic is encrypted. Corporate WLAN traffic is usually encrypted. I wonder how often the authors had to use transport-layer information to resolve link-layer ambiguities.

Despite these concerns, I like this paper a lot. With the measurement infrastructure in place, the authors should be able to study their WLAN in great detail. They will doubtless come up with several interesting observations, and I look forward to reading the follow-up papers. I also hope that this paper will encourage other researchers to build similar systems, and study the WLAN networks that they have access to.

I would like to note that the WLAN studied in this paper is a a stable, well-managed network: the positions of the APs and their channel assignments are well-known and do not change. Also, the clients are generally present only within a well-defined area. Mobility does not seem to be much of a concern. It would be interesting to think about how the Jigsaw approach can be extended to study other types of WLAN networks such as public hotspots, or temporary networks such as those set up to serve large conferences.

Measurement-Based Models of Delivery and Interference in Static Wireless Networks

C. Reis R. Mahajan M. Rodrig D. Wetherall J. Zahorjan

Public Review by Ant Rowstron

In the last few years we have seen a trend in wireless networking research moving away from using simulation based experiments to running experiments on real wireless testbeds. The cynical would say that we have moved from a world where researchers tuned their wireless protocols to work well on a particular simulator with a wide range of workloads, to a world where researchers tune their wireless protocols to work well on a single fixed testbed with an (often very) small set of workloads. This shift has made comparison of results difficult, repeatability nearly impossible and the exploration of a wide set of scenarios onerous.

This paper begins to address the issue by developing a better physical layer model. A primary use of such models is in simulators, but they can also be used for testbed experiments and in designing interference-aware protocols. The paper claims that currently "RF propagation in realistic environments is sufficiently complex that the only existing feasible method for estimating packet delivery between two nodes is to measure it". The paper addresses this issue by demonstrating that it is possible to gather a set of (simple) measurements from a real wireless testbed, and then use these measurements to seed a physical layer model that captures effects of interference etc in the real testbed. The proposed model is cross validated, using a simple MAC and traffic model. The basic approach is very simple; a single node transmits fixed size packets and the other nodes in the network monitor both the RSSI and received packet counts. This is repeated for all nodes in the network, resulting in N^2 measurements, generating an RF profile capturing the relationship between the measured RSSI values and delivery counts. The paper examines how these are affected across time, the effect of external interference (such as that generated by a microwave oven) and how these vary over time.

The paper derives models that capture physical layer properties of the wireless network. These are based on the standard signal-to-interference-plus-noise ratio (SINR) model but are recast to use the measurements of RSSI and packet deliver ratio (the RF profile) plus some hardware specific fixed parameters. Two models are used to capture the behavior of the physical layer: one of them predicts the probability that a receiver correctly decodes a packet when other nodes are transmitting and the other model captures the deferral behavior, where a node defers transmitting because it senses that a channel is busy.

In order to cross validate the physical layer model, a simple MAC and traffic model are used. The MAC captures the key essence of the MAC used for 802.11 broadcasts and a traffic model is of two concurrent nodes competing to transmit. The cross validation compares the derived model to two other models, and shows that the proposed model performs well. Of course the cross validation confirms that the proposed model performs well!

The paper is interesting, and from the perspective of someone who has fairly recently started working in the wireless networking space the long term goals of the research are laudable. The question mark in my mind is exactly how general the current approach really is. The RF profile is gathered using a particular bit-rate (the broadcast bit-rate) and with a fixed packet size of 1084 bytes. The paper claims that in a deployed network the RF profile could be gathered using application traffic (so presumably with different packet sizes). However, it remains unclear to me exactly how this would work. Following on, the cross validation is performed using a very simple traffic model of two

nodes attempting to concurrently broadcast packets. Given the long term goal is to predict the effect of using different MAC or routing protocols this goal seems a long way from demonstrating modeling the effect of two nodes competing to transmit concurrently. It would also have been interesting to see the cross validation include a second wireless network testbed. Finally, given the statements in the introduction stating that the only existing feasible method for estimating packet delivery ratio is to measure it, the validation would have been strengthened if the authors compared the performance of their physical layer model against one used in a wireless simulator such as QualNet. While the history model used in the validation clearly provides an approximation of the best case it would be interesting to understand how close the current simulators get!

However, the general significance of this paper is that it demonstrates it is actually possible to generate models, parameterized by actual measurements of a wireless network, which provide credible predictions of the expected performance. This paper should act as a catalyst for further work in the area. Overall I would recommend others to read this paper, I think that it is an interesting paper which should provoke discussion on this important subject.

Interference-Aware Fair Rate Control in Wireless Sensor Networks

Sumit Rangwala Ramakrishna Gummadi Ramesh Govindan Konstantinos Psounis

Public Review by Robert Morris

Congestion control in wireless networks is a tough problem, probably harder than in wired networks. Bottlenecks are hard to observe directly: instead of a queue feeding a slow link, a bottleneck is typically a physical region of spectrum contested by multiple senders, who may not be able to directly communicate even though they interfere. Congestion collapse could occur even in the absence of long queues or delays, since congestion may cause corrupted transmissions due to interference rather than discards due to queue overflow. One might hope for an ideal MAC that would make wireless compatible with existing wired congestion control schemes, but in practice such MACs don't exist.

This paper proposes IFRC, a congestion control algorithm for wireless sensor networks. IFRC uses the sensor net's constrained communication pattern, in which all data flows down a tree to a base station, to help each node share congestion state with just the other nodes in the tree likely to interfere with it. Each node maintains an estimate of a reasonable per-flow rate for the flows that it forwards (and the flow that it originates). A node cuts that rate in half when its queue gets long; gradually increases the rate over time; adopts the minimum rate over the nodes in the tree that could interfere with it; and adopts the minimum of the rates between it and the tree root. The effect is that nodes generate sensor values (flows) at lower and lower rates until the tree between them and the root becomes un-congested. No node has to explicitly reason about dividing the available capacity among the number of flows, since neither of these values are easy to observe directly; instead the division happens implicitly, much as in CSFQ.

The paper's core contributions lie in two areas. First, the analysis of which nodes in the tree might interfere. These nodes are sharing a bottleneck, and need to exchange and agree on a common per-flow rate. The paper's careful analysis of this set is necessary because a node's set extends beyond its immediate neighbors. The second area is a thorough analysis and evaluation of parameter choices, often the Achilles heel of congestion algorithms.

The authors evaluate an implementation on a large sensor test-bed, itself a significant contribution. The evaluation focuses on fairness, showing that rate allocations are fair across sensors and that the rates are about as high as are achievable assuming that all nodes use the same rate.

This paper points the way to a couple of avenues for continued work. First, the focus on fairness might profitably be relaxed. It seems likely that the most pressing problem in sensor nets' use of wireless lies in efficiency and utilization, with fairness a second-order concern. Second, the paper never establishes the existence or magnitude of the problem it is solving; it shows that IFRC works well without showing that sensor nets without IFRC do not work well. A comparison with existing sensor net protocols would be valuable. Third, the paper assumes that interfering stations efficiently take turns, rather than consistently destroying each others' transmissions, since in the latter case source rate reductions are not an effective way to avoid congestion losses. The implication that scheduling the use of the medium is outside the scope of a wireless congestion control protocol is itself a significant claim which deserves explicit discussion.

Analyzing the MAC-level Behavior of Wireless Networks in the Wild

Ratul Mahajan Maya Rodrig David Wetherall John Zahorjan

Public Review by John W. Byers

In contemplating this paper, the first question that may spring to the reader's mind is: why doesn't there already exist a good tool to infer the MAC-level behavior of operational wireless networks from the incomplete information gathered at a set of passive monitors? Surely such a tool would be invaluable, both to operators and to network researchers, but we don't seem to have it in hand. In fairness, there do exist useful building blocks that the authors leverage in designing their Wit tool, notably the work by Yeo et al to merge views observed by multiple wireless vantage points (ref [27] of this paper), but none to date has closed the loop to reconstruct valid wireless conversations consistent with all of the individual views. Ultimately, this paper provides a healthy dose of insight into why this is a challenging problem, as building such a tool requires quite a bit more algorithmic and systems savvy than a naive reader might first expect. This paper also goes beyond tool-building, since, as several of the reviewers mentioned (I'm paraphrasing rather liberally), the demonstration of the tool's utility is only as convincing as its ability to shed insight on concrete applications where other methods are inadequate. One such measure that the authors focus upon is channel contention, i.e. estimating the time-series of the number of active contenders for the shared wireless medium over the duration of the experiment. I'll have more to say on this shortly.

Returning the composition of the Wit tool, I was particularly interested in the proposed technique to reconstruct a wireless conversation from the partial information collected at the distributed set of participants. After the views are time-synchronized and merged (technically difficult in its own right), the premise is that the set of valid conversations can be described by a finite state machine (FSM), in which messages correspond to edges and the accept state corresponds to completion of a valid conversation. With full information, it is easy to see which FSM path was traversed in a given conversation, but when messages are not observed by any monitor in the transcript, the challenge is to identify the path or paths through the FSM that traverse the edges corresponding to observed messages, and fill in the blanks with appropriate non-observed messages. By attributing weights or probabilities to each way to fill in a blank, the authors demonstrate that it is tractable to pull out the most probable conversation given a set of observations. While this is certainly appropriate for dealing with a single conversation, a concern I have here is that this can lead to systemic bias when dealing with aggregates. Drawing an analogy to flipping a coin that is biased towards heads, the authors' methods would always guess heads when a coin flip happens in a conversation, the outcome of the flip is not observed, and the rest of the transcript is consistent with either heads or tails. To give the less probable outcomes their due, and to provide aggregate statistics that are unbiased in the expectation, it seems better to sample from the possible paths according to their relative probabilities, and not just reflexively choose the most probable path with certainty. While the validation results that the authors present indicate that this is not a big deal, it would be interesting to verify that this does not materially affect the authors' conclusions, and to watch out for this potential pitfall when the fraction of missed messages grows large.

Moving on to experimental results, the authors have evaluated their tool in simulation, where the ground truth is known, but also on a substantial live trace that they collected at Sigcomm '04. One insight that is obvious in hindsight

from their data collection is that it is crucial to place the vantage points in such a way as to strike the right balance that ensures sufficient overlap between observations to facilitate merging of traces while also achieves good overall coverage. In their experiments, one vantage point turned out to have insufficient overlap with the others, and its data could not be used as a result.

The authors' emphasis in the latter portion of the paper is inference of network-wide activity that could only be gleaned otherwise by a full-on instrumentation of all nodes in the network. Representative measures that fall into this category are offered load, the number of active contenders for the channel, and throughput. To estimate these measures, the authors have to make quite a number of assumptions, all of which seem reasonable, but all of which likely contribute to error terms in the findings. A surprise they find when evaluating these network-wide measures in the Sigcomm '04 dataset is frequent inefficient utilization where the most likely culprit based on their measurements is an unhappy situation where there are relatively few contenders who are waiting unnecessarily long intervals in backoff. It would be very useful to validate these findings, root cause the source of the problem in the MAC protocol, and correct it. Such an activity exemplifies one of the largest potential upsides of Wit.

In conclusion, while the reviewers thought that this was likely to be a useful tool, several of them speculated that there might be simpler methods out there that could be just as, or more effective than those employed in Wit. They also wondered how broad Wit's applicability would be. Sounds like they're throwing down the gauntlet for Sigcomm '07!

Capacity Overprovisioning for Networks with Resilience Requirements

Michael Menth Rudiger Martin Joachim Charzinski

Public Review by Anja Feldmann

This paper tries to resolve the question of overprovisioning (CO) vs. per-flow admission control (AC) in packet networks in the pretense of traffic shifts either caused by hot spots or link failures. The basis of the extensive evaluation is the assumption that the network is dimensioned to provide an uniformly high quality of service (minimal congestion) to almost all admitted traffic, without degradation (for example) if single links are failed. Traffic engineering or route optimization strategies are not investigated.

After reviewing related work the paper outlines the scenario considered in this paper: real-time traffic flows with exponential interarrival times and identically independently distributed holding times chosen from three bandwidth classes. The reviewers find that the advantage of this scenario is that quality can be measured in terms of loss probability and that it leads itself to a nice mathematical treatment of both the admission control as well as the overprovisioning approach for the single link situation. Yet, the disadvantage is that (a) if much of the traffic is made of data transfers (which can tolerate, and will adjust to, short-term overloads) then the analysis may be moot, as there will be plenty of leftover bandwidth for the real-time traffic when traffic is rerouted and (b) that traffic does not adhere to these assumptions.

The paper shows that in a single link setting AC can beat CO, if the figure of merit is good service for the nominal (not hot spot traffic). The paper then moves on to more general networks, and the impact of resilience to link failures and to hot spots, in network dimensioning, for both CO and AC. A synthetic reference topology, termed labnet03 of 20 nodes and 36 links is used in the evaluation, as is shortest path routing. The paper is very careful about finding appropriate metrics for comparing CO vs. AC given multiple sets of possible failures and hotspots in complex networks. For CO the authors first derive the aggregate load on all links for a given traffic scenario and then apply the one link model. They then take the maximum over all traffic scenarios. A related procedure is used for AC, with the traffic scenarios taking into account failures but not hot spots. Among the many results is the finding that when failures are taken into account CO and AC have similar costs in capacity. Yet, unfortunately the paper does not consider inter-domain routing aspects (e.g., policy routing) and intra-domain routing aspects such as the setting of appropriate link weights.

I personally find it disappointing that the paper ignores more realistic traffic dynamics ranging from time of day variations, to alternative arrival processes, to Zipf's like distribution of elephants and mice, to shifts in the traffic demands. Furthermore, the paper does not consider that network dimensioning is limited by the available interface speeds. Therefore links bandwidth is usually only possible in fixed, discrete chunks. In addition, as one of the reviewers points out, path diversity is needed to recover from network failure and is therefore another important design criteria that should have been discussed.

COPE: Traffic Engineering in Dynamic Networks

Hao Wang Haiyon Xie Lili Qiu Yang Richard Yang Yin Zhang Albert Greenberg

Public Review by Jennifer Rexford

During the past few years, traffic engineering in IP networks has been a popular topic with researchers and practitioners alike. Traffic engineering involves adjusting the routing of traffic to the prevailing demands, for better user performance and more efficient use of network resources. In contrast to load-sensitive routing, where routers adapt to changes in load automatically, traffic engineering involves centralized optimization of the routing configuration based on a network-wide view of the traffic. This paper contributes to the large body of work on intradomain traffic engineering, where the central ingredients are the traffic matrix (the offered load between pairs of edge routers) and the network topology. The goal is to select the routing configuration (such as the link weights used in shortest-path routing protocols) that minimizes some objective function, such as the maximum utilization over all links.

Traffic engineering naturally maps into an optimization problem, and early work focused on the simplest incarnation—optimizing intradomain routing for a single topology and traffic matrix, for a simple objective function. Over time, researchers focused on identifying routing configurations that perform well for multiple traffic matrices (corresponding to different times of the day, or compensating for uncertainty in traffic measurements) and multiple topologies (corresponding to different failure scenarios), based on predictions of the kinds of variations that might arise in practice. Some work went to the extreme of oblivious routing, where the routing configuration is optimized to handle any possible traffic matrix. These two approaches optimize for the common case (without regard for the worst case) or for the worst case (without regard for the common case), respectively.

Each approach has its limitations. Optimizing for the common case can leave the network in a horrible state when the traffic patterns suddenly shift. Optimizing for the worst case can lead to poorer performance the vast majority of the time, when the traffic matrix matches the operators' expectations. This paper proposes a hybrid approach to robust optimization: optimizing for the common case, subject to a bound on worst-case performance. That is, the authors consider both a set of predicted traffic matrices and a maximum acceptable penalty to endure for the remaining traffic matrices. The resulting scheme is called Common-case Optimization with Penalty Envelope, or COPE for short. The main contributions of the paper are formulating and solving the resulting optimization problem, and comparing COPE to traditional common-case and worst-case approaches through an extensive evaluation on traffic and topology data from a tier-1 ISP and the Abilene Internet2 backbone.

The paper also describes some extensions of COPE for the underexplored problem of interdomain traffic engineering, where an ISP network may have multiple egress points that can reach each remote destination. The proposed solution seems to force all traffic for a destination prefix to a single egress point, which is fairly restrictive, but the initial performance results are promising. Extending the treatment of the interdomain case to more realistic local routing policies, and evaluating under real changes in interdomain routes, would be an exciting area for future work, especially since interdomain routing changes may very well be the cause of most of the large, unexpected shifts in the traffic matrix.

In summary, the paper is interesting for drawing attention to the need for robust optimization, with the nice twist of optimizing for the common case while still considering the worst-case performance. This approach may be germane to other aspects of data networking, beyond routing. On the other hand, the paper is arguably a bit narrow, compared

to other SIGCOMM papers, in that it proposes a hybrid of two existing solutions to a fairly specific problem (i.e., robust intradomain traffic engineering). Still, as the Internet becomes an increasingly important part of the world's communication infrastructure, we certainly need ways to make IP networks more robust, and this paper represents an important step in that direction.

Realistic and Responsive Network Traffic Generation

Kashi Venkatesh Vishwanath Amin Vahdat

Public Review by Ratul Mahajan

Consider the task of evaluating a system whose performance depends on the nature of traffic. Examples of such systems are packet classifiers, routers, and tools to infer properties of network paths. In my own work, I have faced this task while designing router-based mechanisms for defending against denial-of-service attacks and for active queue management. Ideally, you would be able to deploy your system and study its performance, but of course that is often intractable.

To compensate, you will probably resort to studying the system against realistic traffic and hope that you can find enough real packet traces to satisfy your needs. But the odds are that you will find very few. You now have the challenge of leveraging these limited number of traces to explore the space of current and future traffic scenarios. Projecting a trace to a different scenario requires the ability to produce realistic variations. For instance, you probably do not want to double the amount of traffic by simply duplicating each packet.

"Realistic and Responsive Network Traffic Generation" aims to help researchers with the challenge above. It presents *Swing*, a tool that takes as input a packet trace from both directions of a link and can synthesize traffic that is statistically similar or differs in a semantically meaningful way. Previous work has achieved this for aggregate characteristics, such as average bandwidth, or for individual applications; Swing advances the state of the art by reproducing packet arrival burstiness at a range of timescales and for all applications.

The cornerstone of Swing is a structural model of traffic whose goal is to capture enough detail to reproduce arrival burstiness. In this model, user and protocol characteristics are captured by breaking the trace into applications, clients, sessions, and so forth. Network characteristics are captured using the delay, loss rate, and capacity between clients and the traced link. Swing generates traces by first inferring the values of model parameters from the input trace. In this step, it borrows heavily from the network measurement literature. It then uses the extracted values to instantiate the model in an emulator. Emulating different parameter values will generate variants of the input traffic.

The key contribution of the work is neither the model nor the parameter inference techniques; these individual components are fairly straightforward or use known techniques. Rather, it is showing that the combination can synthesize traces with realistic properties, including packet arrival burstiness at sub-RTT timescales. I find this surprising in view of the many simplifying assumptions that make Swing feasible. The primary lesson from the paper is that both end-host behavior and network characteristics must be emulated to reproduce the burstiness observed in real traffic.

The trace generation methodology of Swing is very appealing. Because it is based on a high-level, structural model of traffic, one can use it to generate semantically meaningful variants. The paper provides some evidence that varying parameter values changes the traffic in expected ways. Further, because Swing emulates real clients, it generates not merely a static trace but "responsive traffic." This makes it applicable to studying systems that modify traffic,

for instance, by selectively dropping packets. The framework also enables future research in understanding traffic behavior in the Internet. For instance, it can be co-opted to more deeply investigate the burstiness of Internet traffic and isolate the various underlying factors.

Ultimately, only time will tell how useful Swing proves to be for researchers. The current version may not be appropriate for all kinds of evaluations. For instance, Swing does not reproduce the original IP address and port distributions, which complicates using it for systems such as packet classifiers and both examples from my work mentioned earlier. Another limitation is that it treats various model parameters, such as path capacity and loss rate, as independent. But realistic synthesis of certain kinds of variations may require that the relationship between various parameters be captured. Unfortunately, the authors provide little guidance as to for which tasks Swing is or is not appropriate.

The area of generating synthetic traffic for experiments and testing is an important one and the paper makes a valuable contribution to this growing body of work. Swing is a tool that many researchers will likely find useful, and future research is bound to improve it further.

A Basic Stochastic Network Calculus

Yuming Jiang
Public Review by Vishal Misra

Promise. Since its introduction in the early 90s, network calculus has tantalized the research community with hopes of an elegant theoretical framework for tractable analysis. Central to the theory of network calculus is the use of alternate algebras (e.g., min-plus), to transform non-linear systems into classical linear systems. With the right transformation, a lot of the machinery of linear system theory carries over to network calculus, providing powerful tools for the analysis of networks.

Development of network calculus has progressed along two tracks - deterministic and stochastic. Deterministic network calculus has been successfully employed in the design and provision of networks to provide *deterministic* service guarantees for regulated flows. However, service guarantees are typically required by multimedia flows, which can withstand some amount of loss or (excess) delay. For such flows, *stochastic* service guarantees are more important and that is where stochastic network calculus makes an appearance. Additionally, the very nature of modern networks makes deterministic analysis very limited in its applicability. On the other hand, the theory of deterministic network calculus is considerably more advanced than its stochastic counterpart. That had placed the area in the classical resting place of many theories - tractable and practical looking at each other longingly across a divide. This paper takes an important step in bridging that gap.

This paper proposed two new definitions: maximum-(virtual)-backlog-centric (m.b.c) stochastic arrival curve and stochastic service curve for stochastic network calculus. These two definitions are general, and many definitions in previous work for stochastic versions of arrival curve and service curve are special cases of these two new definitions. By employing these two new definitions, stochastic network calculus is unified with its deterministic counterpart such that five important operational properties of deterministic network calculus can be inherited. This brings the two branches of network calculus under one unified umbrella. The paper also defines a *stochastic strict server*, employing two stochastic processes to characterize a server. Armed with this new definition, the paper provides independent case analysis for the 5 properties, providing improved results. The independent case (i.e., where different flows are not correlated with each other, a likely case for open loop multimedia streams) is potentially an important scenario and the paper provides the first full analysis. The reviewers were unanimous in agreement that the technical results of this paper are very strong and make a very important contribution to the area of (stochastic) network calculus.

That being said, the reviewers were also in violent agreement that the paper is an extraordinarily dry read and is extremely technical. It is not for the faint of heart, if Greek symbols bother you - run! The paper is not a classical Sigcomm paper in that there is no validation, no relation to practical systems and nary a trace of an application. The theory rests on numerous assumptions, both explicit as well as implicit that are questionable. If you are looking for instant gratification, this paper is not for you. Understanding and interpreting the results presented in this paper requires extensive domain knowledge and patience. However, good or great research need not provide immediate applications, and the paper provides important tools that will add to our ability to gain insights on the behavior of complex networks. The paper takes an important step forward in advancing the state of the art to more realistic models and analysis, and will hopefully provide an important foundation to further build the theory. Promise.

Systematic Topology Analysis and Generation Using Degree Correlation

Priya Mahadevan Dmitri Krioukov Kevin Fall Amin Vahdat

Public Review by Aditya Akella

The problem of topology characterization and modeling has received much attention from researchers both in networking as well as other areas such as Physics and Biology. In general, research on this topic has centered around the following three issues: (1) Using empirical techniques to tabulate existing interconnections and obtaining topology maps (e.g. Rocketfuel, Skitter in networking); (2) Characterizing key observable properties of the topology. These could include simple properties, such as the average degree or the degree distribution (e.g. Faloutsos power-laws paper), or more complex properties such as graph clustering, path length distribution, likelihood and spectrum (Li et al's HOT paper). (3) Developing algorithms to generate synthetic graphs that mimic one or more, and preferably all, of the above observed properties (e.g. degree-based and structural generators).

Of these issues, developing effective mechanisms for synthetic topology generation (#3 above) poses tricky challenges. State of the art approaches can effectively mimic "local" properties (e.g. the degree distribution) of observed topologies. However these local approaches often fail to reproduce more complex, global graph properties such as clustering and graph spectrum. The global metrics cannot be simply ignored, since they directly influence several desirable properties of any topology, such as robustness to attacks, effect of node/edge removal and routing efficiency. One might wonder if it is possible to directly generate synthetic graphs that match specific global properties. While this is possible for a few of the properties, there are no known algorithms for several others. Moreover, techniques that mimic a given global property are not guaranteed to mimic future metrics that may be of interest.

This paper introduces a new series of graph metrics each of which characterizes a given graph using degree correlations. In the limit, any given graph can be completely described according to any given property using this series of metrics. The series of properties, called "dK series", essentially describes the correlations among the degrees of subgraphs of size d, for $d = 2, 3, ...$ For $d = 0, 1$, OK captures the average degree of the graph, while $1K$ captures the degree distribution. It is easy to see that each metric in this series constrains the graph to a finer level of detail. The definition of this series is the central contribution of this paper.

Two interesting aspects of this series make it useful for topology analysis and synthetic graph generation: (1) The metrics are "inclusive": if a graph satisfies a more constraining property in the series (e.g. dK), it will also satisfy any of the less constraining properties preceding it (e.g. $d'K$ for $d' < d$). (2) The metrics are "constructible". The authors outline several algorithmic approaches for constructing random graphs that can reproduce a given metric in the series.

Another interesting result is that in practical situations involving observed AS-level and router-level graphs, the synthetic topologies generated according to the dK-series *quickly* converge to the empirically observed graphs. A small d, such as $d = 3$ is shown to be sufficient. This is an important observation because generating dK graphs for high values of d requires significantly higher computation as well as exponentially larger number of input parameters. Finally, the paper presents a simple methodology by which network practitioners can determine the smallest value of d that captures all the essential characteristics of a given input graph.

This paper was highly rated by the Sigcomm reviewers. It was generally agreed that the ability to generate random

graphs with increasing levels of fidelity, and with the added guarantee of quick and accurate convergence to a given input graph, is theoretically significant. This result may also be relevant to areas outside of core networking (e.g., dynamic engineered systems, graph theory, biology etc.) and is likely to generate substantial follow-up work.

The most glaring concern with this work was about its *usefulness to network practitioners*. Three separate reviewers expressed reservations in this regard. The rest of this review will focus on this issue.

How can Internet practitioners use this result? As with any topology generation mechanism, several possibilities can arise:

- Synthetic topologies for simulation: As it stands, the proposed approach takes *a given graph as input* (say a Skitter or a HOT topology) and tries to synthetically generate random clones that faithfully mimic several key properties of the input graph. This in itself is not very useful: why would one want to use a synthetic clone for simulation when the real graph is available on hand? This is not to say that there are no applications at all, but rather that the applications are few and relatively uninteresting.

- Generating Internet-like graphs based on degree structure: The paper has several interesting algorithms for constructing dK-graphs, for $d \geq 2$. It is possible that practitioners may find these useful for generating "Internet-like graphs". However, there are two potential road-blocks: (1) Note that we first need to perform a measurement study of Internet topologies and derive the necessary distributions for P_d. This is not impossible, but definitely not easy for a large d and a large enough input graph. Also, for $d \geq 3$, the P_d distributions are high dimensional structures. Having to provide such complex structures as input to the graph generation algorithm could be cumbersome for users and also prone to errors. (2) Even if the above process were made more user-friendly, one can only use it to accurately reproduce current topologies. In contrast, what is more useful (e.g. for studying scalability, robustness etc.) is the ability to scale up or down to arbitrary graph sizes. It is conceivable that various measurements conducted over time show that P_d is invariant with graph size for some large enough d, say $d = 3$. If this is indeed true – and this is an interesting open question – the proposed graph generation algorithms will come in very handy.

- Application to evolutionary models for the Internet: There is no direct way in which the proposed approach applies to evolutionary models for the Internet (a popular example of such models is "Preferential Attachment") – the authors admit this as well. However, the approach can be used to *validate* any existing model (see Section 6 for an elaborate explanation). To me, this is a very important contribution. In general, the "dK-series" is the best available set of metrics to understand if two graphs are identical, and if they are not, to what extent do they differ.

- Other applications: The paper is full of neat tricks for solving important sub-problems. First, there is the approach for limiting dK-space exploration and the related idea of checking for convergence to the input graph. Then, there is the idea of "dk-targeting $d'k$-preserving rewiring" to generate dk-random graphs. Finally, there is the evaluation methodology, which is extremely thorough and well thought-out. To me, these tricks/approaches are among the paper's biggest wins: my guess is that networking practitioners will find one or more of them extremely useful in their work.

Additional comments on the above issues can be found on the discussion blog for this paper.

Note that some of the above drawbacks can be viewed as challenging open issues. I strongly believe that there is bound to be a lot of follow-up work addressing one or more of these. There are still other, less obvious issues that the paper leaves open. The most important among these pertains to convergence: Why is $d = 3$ sufficient to reproduce existing Internet topologies? Is it possible to theoretically demonstrate the relationship between the "convergence speed" and the complex engineering decisions that go into building a network?

I want to conclude this review with a comment on what this paper means for the future of topology generation research. A couple of reviewers felt that the paper may have come quite close to bringing down the curtains on this area. I would argue that this is far from being true. Instead I would argue that this paper opens up fresh avenues for both empirical and theoretical research in this area. If anything, this paper may have just extended the lifetime of the topology modeling debate by a few extra years (and papers).

Feel free to disagree with anything I have said.

Minimizing Churn in Distributed Systems

P. Brighten Godfrey Scott Shenker Ion Stoica

Public Review by Emin Gün Sirer

Churn, caused by the departure and arrival of nodes, is the bane of modern distributed systems. At best, churn necessitates additional bandwidth and coordination to transfer service functionality from failed nodes to others. Often, it leads to service interruption, with user-visible side-effects. And there are many existing distributed services where the churn rate, if over a given threshold, would overtake the system's ability to recover and thus render the service completely inoperable.

And it's not just the large-scale systems that are affected by the adverse effects of churn. To be sure, peer-to-peer systems need to pay careful attention, since their sheer size can exacerbate the overheads associated with even modest churn rates. But churn is also an issue for small systems, where making a bad choice can have a significant impact on the perceived quality of service.

Clearly, a principled approach to minimize churn, one that might subsume the various heuristics that have been proposed in past literature, is called for. Luckily, churn can be managed through the judicious selection of nodes which are available for admission into the system; selecting the nodes that are likely to remain up for a long time will reduce the churn rate and improve service. The question is: how do we find these nodes?

The current paper presents a quantitative guide to churn resulting from different node selection strategies, analyzes the churn incurred by some classes of strategies, and embodies a surprising result with practical implications.

The paper categorizes strategies for selecting nodes along two axes. *Predictive* strategies base their selection on node characteristics such as past uptime, whereas *agnostic* strategies do not take such measures into account. *Replacement* strategies perform dynamic replacement of failed nodes, whereas *fixed* strategies select the nodes they will use prior to deployment and stick to that set for their entire operation.

Three points among the spectrum of node selection strategies examined by this paper will resonate with practitioners. A predictive replacement strategy, embodied in many systems, is to prefer nodes with the longest current uptime. Another approach, an agnostic replacement strategy, is to select nodes according to a preference list based on a measure, like a hash, that does not intend to optimize churn. A final approach, also an agnostic replacement strategy, is to simply select replacement nodes at random.

The paper provides a precise definition of churn and quantitatively examines the churn incurred by various strategies. It examines the performance of different strategies under a synthetic trace as well as an extensive set of real-world traces that span peer-to-peer networks, corporate workstations, and public web servers.

The result is quite surprising: random replacement, despite its lack of smarts, performs better than preference list strategies, and is quite close to the best strategy under most scenarios. The paper then analytically models the performance of the random replacement strategy, and shows that it approximates selection based on longest uptime. There are, however, a few key differences with longest uptime that make random replacement more compelling: (i) it is very simple to implement (ii) it does not rely on truthful reporting of uptime by a peer, and (iii) it avoids the problems of detecting and properly classifying node outages caused by network failures.

The paper then examines the implications of this analysis on system design in the context of an anycast service, a DHT-based lookup service, a multicast tree maintenance algorithm and two replica placement strategies for DHT-

based storage services. The last scenario extends the quantitative analysis to a setting where nodes carry long-term state. For each scenario, the paper examines end-to-end performance metrics, and discusses how randomization can be used to transform commonly used strategies that, in essence, equate to a preference list into node selection criteria that reduces churn by approximating random selection. The changes required are straightforward and small, yet have an appreciable impact on the performance of the systems.

An open issue in the paper is the impact of these selection criteria on fairness. Some nodes may be utilized much more frequently than others; an issue the paper defers to future work.

It is rarely the case that system designers can be told to do less to improve performance. This paper shows that less is, in this context, better.

Source Selectable Path Diversity via Routing Deflections

Xiaowei Yang David Wetherall

Public Review by John Chuang

One of the key architectural debates facing the Internet today is the tussle between network control and endpoint control of packet routing. While source routing has been clearly spelled out since the original Internet Protocol specification, the network operators have been reluctant to accept source route requests. They argue that source routes could lead to security vulnerabilities, unpredictable network performance and traffic engineering nightmares. It is quite understandable why the network operators would be loathe to cede control over how packets are routed in their networks.

From the endpoint's perspective, however, there is tremendous value in being able to over-ride default routes. They can either avoid congested nodes and links, or indirectly, encourage network competition and innovation with a credible threat of taking the business elsewhere. Overlay routing has been proposed to enable endpoints to select non-default routes without the cooperation, or even the knowledge, of the networks. However, this technique is fraught with its own set of problems, starting with the challenge of scalability.

In this timely and well executed paper, Yang and Wetherall attempt to find a sweet spot between network and endpoint control of packet routing. As the paper title clearly explains, their design employs route deflections to generate diverse path options that can be selected by the source. At the same time, the design allows the networks to retain control for the most part, since they get to specify the deflection rules which determine the deflection set.

As noted by the authors (and a number of reviewers), route deflection is not a new idea in and of itself. Network router implementations of fast reroute (FRR) already allow packets to be deflected off the shortest path in response to failures. The crucial insight made in this paper is to render the route deflections visible to and selectable by the endpoint. This way, the endpoints are empowered to select from a small but hopefully diverse set of routes, and the networks do not have to surrender control over routing.

The reviewers noted a number of strengths in this paper. One reviewer commended the authors on the completeness of the work, which is "fully developed from idea to architecture to mechanisms with proofs to implementation options to evaluation." Another reviewer found that the investigations provided interesting statistics on the path diversity in backbone networks and the application of route deflections. A well-respected senior researcher gave the ultimate thumbs-up to this paper, saying "I wish I'd written this paper, so how could I possibly recommend rejection? Very nice job!"

At the same time, several reviewers voiced concern about the practicality of the proposed scheme, and raised a number of questions that could lead to interesting follow-up work. First, as in any dynamic routing scheme, it is important to investigate any potential instability that may arise due to the route deflections. Is the time-scale of route discovery (probing of different deflections) likely to be faster than that of network re-convergence? Could any synchronization of the probes lead to undesirable routing oscillations?

Second, this scheme requires buy-in from the network operators, and therefore it is worth asking the same set of questions as was posed for IP source route – what are the security and trust implications; what are the implications to traffic engineering and performance predictability?

Third, the reviewers identified a number of alternatives that this scheme should be compared against. For route

deflection, how do the proposed deflection rules compare against random deflections or the currently deployed solution of using an alternate equal-cost path first? For endpoint route selection, how does this deflection-based scheme compare against the use of multi-homing or a third-party route service provider? In addition to achievable path diversity and locus of control, there might be other factors such as cost of deployment that should be taken into account. It is worth noting that there is another paper in the SIGCOMM 2006 program, by Xu and Rexford, that takes a different approach to this problem. This suggests that the design space may be quite large and worthy of further exploration by the community.

Finding the right balance in the route control tussle space is as important as it is difficult. While it may take the community a few iterations to converge to the right solution, this paper provides a good example of *how* we should go about finding it.

MIRO: Multi-path Interdomain ROuting

Wen Xu Jennifer Rexford

Public Review by Yin Zhang

It has long been recognized that the default Internet routing does not always satisfy the diverse performance, reliability, and security requirements of end users and applications. This has also been clearly demonstrated in several measurement studies (*e.g.*, the Detour work from UW). Recent research has considered several schemes that allow end users to specify their own routes through the use of either source routing or routing overlays. A major hurdle to the adoption of these schemes is that they leave transit domains very little control over the traffic traversing their networks. Such control is essential for ISPs to ensure smooth network operations and maximize revenue. The scalability of these schemes is another major concern.

This paper presents a multi-path interdomain routing protocol called MIRO that tries to simultaneously (i) offer considerable routing flexibility, (ii) give transit domains control over the flow of traffic through their infrastructure, and (iii) avoid state explosion in disseminating reachability information. Conceptually, MIRO can be viewed as (loose) source routing at the AS level (as opposed to at the router level or between overlay nodes). With a number of small extensions to BGP, arbitrary pairs of domains can negotiate the use of alternate paths in addition to the single path that BGP would normally provide. Tunneling is then used to send traffic along the selected alternate paths. During the negotiation, the responding AS can use its local policy to determine whether or not to reveal any additional routes to the requesting AS and thereby control the traffic traversing their networks.

As noted by the the reviewers, MIRO has many advantages, including its simplicity, backward compatibility with BGP, practicality in its balance between route selection and ISP control, and incremental deployability. The paper analyzes the the effectiveness of MIRO in terms of the ability to find alternate paths and the incremental state cost. The experimental evaluation is quite solid and clearly demonstrates the benefits of MIRO even when it is deployed by only a small fraction of ISPs.

The reviewers also noted some ways in which the work could be improved. First, additional information is needed in order to understand whether the design decision of being backward compatible with BGP is better than designing a new routing protocol from scratch (esp. given the many known problems of BGP). Due to space limit, the paper has to omit certain protocol details (*e.g.*, the soft-state protocol required for maintaining tunnels). It is not entirely clear after fleshing out all these details whether the complexity of the final protocol will become comparable to designing a new protocol. It would be useful to see a more formal protocol design with detailed information such as the message types, state diagrams, etc. A real prototype implementation of MIRO in XORP will also help significantly to clarify the design, expose unforeseen complexities, and quantify protocol overheads.

Second, it would be useful to see a more systematic and thorough exploration of the route selection policies an AS might wish for and whether and how MIRO could effectively support them. Currently, the paper mainly considers policies of the form ”avoid AS X”, which may or may not be the most useful policy in practice.

Third, it would be useful to see more details on how MIRO decides to start or stop exploratory negotiations for alternate paths and the algorithm that BGP speakers use to select the appropriate paths. Intuitively, MIRO can be viewed as a way of performing source routing or overlay routing at the AS level (as opposed to between individual routers or overlay nodes). So some form of probing seems necessary to assess the quality of different paths. The route

selection algorithm may also need to take into account economic considerations such as customer payment options, which can make the problem quite complicated.

Finally, just as for source routing and overlay routing, MIRO needs to answer the question of how its use by multiple end points and intermediate ISPs would affect the stability of end-to-end routes. The route decisions made by ISPs directly involved in the negotiation may also interfere with traffic engineering goals of intermediate ISPs on the route between the requesting ISP and the responding ISP. Such issues are interesting avenues of future work.

In summary, the paper outlines a promising approach that achieves a good balance between routing flexibility with ISP control and is incrementally deployable in today's Internet. It represents an important step towards the eventual development and adoption of a practical multi-path interdomain routing protocol.

Network Monitors and Contracting Systems: Competition and Innovation

Paul Laskowski John Chuang

Public Review by Ramesh Johari

Over the past decade, the Internet's commercial landscape has undergone tremendous change, while at the same time the network's status as essential public infrastructure has been cemented. As current discourse suggests, the future economic structure of the Internet is a subject of vigorous debate. Recent headlines about "network neutrality" speak to the deep uncertainty about the future regulatory infrastructure (if any) that should govern the Internet. In part, this debate is fueled by the fact that according to common wisdom, no national backbone provider makes a profit on their data network services, and that to date incentives for investment in upgrading access networks have been woefully inadequate. Nevertheless, the value generated by the network far exceeds the cost of provision and maintenance: today the Internet is vital to the economies of most of the industrialized world.

It is in this context that the paper by Laskowski and Chuang should be considered. The paper suggests that there is a fundamental link between *innovation* (or a lack thereof) by ISPs, and *accountability* in the network. "Innovation" is broadly interpreted as any action by an ISP that improves "quality" (again broadly interpreted) along at least one end-to-end data path. "Accountability" refers to the network's support for contracts on quality that can be enforced.

The authors consider a highly stylized model, where accountability is ensured through the presence of *contractible monitors*: essentially black boxes that provide verification of network behavior, and can serve as vehicles for contract formation. They begin by arguing that the current Internet has a fundamental lack of accountability that precludes contracts for anything other than best-effort service; and in turn, this situation has led to commoditization of the ISPs and a pursuant lack of innovation. They then consider the introduction of more sophisticated contractible monitors into the network, and in turn study the requirements on such monitors necessary to ensure sufficient accountability to support innovation. Their main conclusion is that monitors which provide verifiable proof of behavior along the remainder of a path from a node to a destination are in some sense the "minimal" class that can provide sufficient accountability.

The paper is to be commended for its ambitious take on a rather difficult collection of problems; at the same time, there are several criticisms that are relevant. The authors formulate their results in a simultaneously quite abstract as well as simplified model of ISP interaction; as a result, the reader will harbor some doubts about the generality and practical implications of their results. At times it is difficult to separate assertion from technical proof, but that is perhaps a consequence of the modeling task they have undertaken.

We outline some of the shortcomings of this approach:

1. *Timing*. The authors consider a three stage game model of competition, where firms innovate at the first stage (determining network quality), price at the second stage, and then route traffic at the third stage. This sequence of decisions makes some very significant assumptions regarding the behavior of ISPs. First, it assume that ISPs commit to a single quality level over the *entire lifetime* of the routing game; thus nodes cannot "innovate" once the game is underway. Further, the very threat of renegotiation of contracts and attendant quality levels

in the future (e.g., depeering, or termination of transit agreements) is often critical to the dynamics of contracts between ISPs; these cannot be captured by the current model.

2. *Single source-destination pair.* The authors only consider a model consisting of a single source-destination pair. This is forgivable when the authors frame their main result—that "rest-of-path" contractible monitors are in some sense the minimal vehicles for accountability that can lead to innovation. Conditions on contractible monitors would likely only become more stringent in a more complex model. However, the choice of a single source-destination pair calls into question the power of the *positive* results in the paper, and in particular the practical value of the model formulation, since these insights may break down in a multiple source-multiple destination model. It is also worth noting that from a mathematical standpoint, the proofs make heavy use of the single source-destination pair assumption, and so cannot be generalized easily.

3. *Single ISP innovations.* The model of innovation considered requires individual ISPs to determine whether they choose to innovate; however, this rules out many innovations that require *end-to-end* cooperation between ISPs to implement—for example, packet marking strategies for DDoS detection. It would appear that the primary innovations being considered here are those where a single ISP can substantively alter the quality of the path; these are typically likely to be performance upgrades (e.g., investment in capacity), rather than end-to-end protocol innovations.

These are a sampling of some of the issues raised by the paper; throughout, assumptions raise as many questions as they answer. Perhaps the biggest question about the paper is whether it accurately describes the failings of the modern Internet's contractual system. After all, the network neutrality debate is framed around the question of whether service providers can ever recover the value they generate through an individual investment or innovation, without simultaneously stifling application diversity. The present paper provides insight into the accountability necessary for contracting systems that can successfully sustain investment, but it is unclear whether, for example, the results can be used to interpret the reluctance of access telcos to upgrade their networks to fiber.

Ultimately, the greatest value of this paper is precisely the myriad questions that are raised. By making a philosophical case for their modeling approach, they simultaneously force the reader to pause and examine the Internet's economic ecosystem from a macroscopic perspective. Given that the Internet currently faces a very uncertain economic future, this introspection is likely to be a very useful thing for the networking community.

Building an AS-topology model that captures route diversity

W. Muehlbauer Olaf Maennel Steve Uhlig Anja Feldmann Matthew Roughan

Public Review by Christophe Diot

Any step toward the understanding of the Internet as a complex system deserves publication and this paper makes such an important initial step. It is both important for the scientific community to understand better the structure of the Internet, and for network operators to manage their network.

The problem addressed here is to infer an AS level topology of the Internet that captures AS path diversity. The task is ambitious. While there is a large body of work on inferring AS graphs, AS interaction, or predicting AS paths the resulting AS topologies generally lack accuracy for one or more of the following reasons:

- Simplifying assumptions are made, such as representing an AS as a single router. - Lack of BGP data. - Try to mimic inter-AS policies.

This is to my knowledge the first paper that (1) takes an orthogonal approach based on both heuristics and simulation and (2) uses a large amount of experimental data to validate its methodology. The percentage of correctly inferred AS paths is in itself important, but it is not what I would remember from this paper.

The authors of this paper build a model that uses multiple observation points (i.e. Routeviews, RIPE, etc.) to infer by multiple iteration starting from a naive one-router-per-AS model. A BGP simulator is used to build AS path. An iterative process corrects false path inference. The cutest idea of the paper is to define the notion of quasi-router, together with the observation that AS relationships are not defined per AS but based on commercial relationships among AS operators.

The modeling methodology is evaluated quite thoroughly and success is higher than previously proposed models. The prediction of AS paths not used to built by the model is also discussed.

The following section is about what the reviewers liked and did not like in the paper, and that I think is still relevant in the final version of the paper.

What reviewers liked: the results showing the limitation of single router AS based models; the discussion of path diversity at the AS level; the use of a simulator and of multiple routers per AS to model AS paths.

What reviewers did not like: the motivation of the accuracy metrics (which remains pretty mysterious to me too); the absence of discussion of the impact of IGP failures on the modeling tool; and that the paper seems to oversell some results and contributions. But what can one ask for in 12 pages:-)

In fact, the paper does not really describes a model to build an inter-AS topology, but a heuristic that refines an inter AS topology by iterative steps that improve AS path inference. However, what I find important is that the authors do not try to reverse engineer routing policies. Instead they try to infer AS paths from observation. This is an important methodological change in this area where most people try to mimic BGP behavior and ISP policies to model inter-AS topologies: a model does not need to match reality! and this paper proves that by not trying to match reality, you can get better results.

The success of this kind of approach is directly related to the amount and quality of information that you inject in the system. My personal feeling is that the more data you use, the more you know. It would have been nice to discuss how much can be predicted based on how much data is used by the modeling tool proposed. I am afraid that the nature of BGP will make prediction difficult.

I list below multiple other questions for which I would love to have an answer and that complements the list of future works identified by the authors:

- Would using more input data improve accuracy? - Is the method robust to errors in the training set? how about injecting errors in the training data set? - Related to the previous question, what is a stable view of the routing? and when do you know that you have such a stable view? Do you have to re-train your modeling tool on a periodic basis, or after major routing events? Should you study over time and use the prevalent routes (or stable routes?) - How would prefix aggregation impact the model? - One of the strengths of this work is that authors do not try to reverse engineer what really happens to build their model. So, would this same model be as successful to answer other question about inter-As topology?

Last comment, this work would be most useful if the tool and the annotated data set were available to the community. I suspect it will. And I encourage other researchers to reuse the tool and the data set in order to really compare the efficiency of new methods, and possibly to validate the current results.

And sorry for inaccuracies in this public review. This review was written based on Sigcomm reviews and on discussion with colleagues at Thomson (with special thanks to Constantinos Dovrolis and Renata Teixeira). I suspect it would have been much better if written by someone who was involved in the discussion at the TPC meeting, which I was not part of.

The Impact and Implications of the Growth in Residential User-to-User Traffic

Kenjiro Cho Kensuke Fukuda Hiroshi Esaki Akira Kato

Public Review by Hui Zhang

Understanding network traffic is critical to network, protocol, and even financial engineering. As an example for the last point, the telecom bubble of late 90s and early 2000 was partially justified by the assertion that Internet traffic was doubling every 100 days. And for a long period of time, with the exception of Odlyzko's analysis, the research community could not verify the assertion by technical means.

It is hard to synthesize a birds-eye view of the Internet traffic due to Internet's distributed ownership structure. Previous traffic studies were based on either end-to-end measurement from sparsely distributed Internet end points or measurement within a single backbone. It is usually difficult to verify or generalize the results.

This paper is the first of its kind to characterize the Internet traffic of an entire country, taking the traffic study to an unprecedented scale. All reviewers of the paper agree that this is the most important contribution of this study. In addition to the technical observations, the study demonstrates that it is possible to coordinate multiple service providers and obtain a macro-level understanding on the Internet traffic at the scale of a country. Along this line, it would be helpful if the authors document the security/anonimization measures they took to convince the service providers to share the data.

In addition to the scale, I think the study is interesting also because it is about the networks in Japan. Japan is not just any country, but a country with a large population, a high peneration rate of residential fiber access, and a history of aggressively adoptiong new applications. If one wants to pick a country to gain insight into the future of the Internet, it is hard to find a more appropriate country than Japan. In the context of understanding future networks, one needs to look at both application and technology trends. The paper has done a good job of studying and highlighting several application traffic trends: the significance of P2P traffic, symmetic nature of the application demand, and the lack of crisp models to distinguish types of users.

While it might be too much to ask for a single paper, one area that should be further studied is the inter-play between traffic demand, network engineering, and economics. For example, how is the growth and changing pattern of residential broadband traffic affecting service provider networks? Where are the bottlenecks in today's service provider networks? (access vs. metro vs. backbone? transmission vs. switches vs. routers?) How are the bottlenecks likely to change assuming we are on the existing technology evolution path? Are today's flat-pricing scheme economically sustainable for access or backbone service providers? Would a usage-based pricing scheme be more favored by service providers and if so what consequences would it introduce?

It seems logical that the majority of the peer-to-peer traffic is video (file download or streaming). Given my personal interests in clean slate design of future networks and scalable distribution of Internet video, I would like to make an observation and pose a question. Multiple technology trends (higher fixed and mobile broadband access speeds, wider adoption of fixed and mobile video capturing/playback devices, higher resolution of video) point to the direction that video traffic will dwarf any other traffic over the Internet. The research community has studied the problem of video distribution and explored several architectural choices: adding QoS and multicast support to routers,

deploying caches/content distribution servers throughout the network, and leveraging the upstream capacity of end systems. The question is then: assuming that video is going to be the dominant traffic (with the understanding that video is not a single "type" and download and streaming have very different requirements) and we have a choice to design the network from scratch, what should be our design? In addition, is this possible to have such a design without significantly limiting the network's flexibility to support future applications?

Would a network traffic study help such a clean slate design question? Personally, I do think it would. I will leave the readers to connect the dots and look forward to seeing more future research to build on top of this study.

Towards Deterministic Network Diagnosis

Yao Zhao Yan Chen David Bindel

Public Review by Mark Crovella

There is a lively interplay between link-level measurements and path-level measurements of networks. The nature of networks, and the Internet in particular, means that oftentimes we have one kind of measurement, when what we really want is the other. For example, in some situations we can easily obtain measurements of traffic flowing over links, but we would like to know how traffic is flowing over the various network paths: this is the problem of ”traffic matrix estimation.” In other situations we can measure metrics like delay or packet loss over network paths, but what we want is to know corresponding properties of links. In that case, we have the problem of ”network tomography,” which is the subject of this paper.

Often we can get considerable leverage out of the existence (or assumption) of an additive relationship between path and link measurements. Work in network tomography is concerned with link properties that are additive over paths, such as packet delay and (log-transformed) packet loss rate. For such properties, the network acts like a linear system, in the sense that the measure of any path is a linear function of the link measures.

If we assume that link measures are stable over the measurement period, and that network topology and routing are likewise stable, then the network’s routing – how paths are mapped to links – defines a linear operator relating link and path measurements. The linear algebraic view of link and path measures has played an important role in a variety of research threads, starting with the paper by Shavitt et al. (citation [20] in this paper). There has been progress on using the linear algebraic formulation to guide inference of link metrics from end-to-end metrics, as well as to select the minimal set of end-to-end paths that need be measured in a network.

The current paper is concerned with clarifying and demonstrating what network-internal measures are possible from end-to-end measurements. To that end, it introduces the concept of the minimal identifiable link sequence (MILS) as a natural and useful concept. The MILS represents the smallest sequence of links for which a measure may be unambiguously computed. The paper then describes an algorithm to efficiently find MILS measures when the network is treated as an undirected graph, and describes a heuristic to compute MILS measures in directed graphs. Finally, the paper evaluates the quality of MILS loss rate estimation in simulation and live Internet experiments.

There are a number of strengths of this paper, as noted by reviewers. One of the weaknesses of some previous work in this area has been to model the Internet topology as an undirected graph. Under such an assumption, path measurements from A to B are the same as from B to A. Since Internet paths are often asymmetric, and since path measures from A to B are usually different than from B to A, the use of undirected graphs to model the Internet is problematic. This paper recognizes the need to work with directed graph models and spends the majority of its effort on those models. In fact, the paper exposes the dramatic difference between the two ways of modeling the network: in undirected graphs, it is usually possible to identify a number of MILSes along each path; for directed graphs (without loops), each path is exactly one MILS.

Another strength of this paper is the way that linear algebraic concepts are used to explain problems and solutions. For example, section 4.1 reformulates the results of Shavitt et al. ([20]) in a way that unifies those results with the contributions in this paper. Likewise the discussion of MILSes in Section 3 clearly relates graph properties with their corresponding matrix properties.

Reviewers also noted some ways in which this paper could be improved, mostly concerned with practical aspects of the methodology. First of all, the key approach used to resolve the problem that arises in undirected graphs is to note that if a path has nearly zero loss, then so do all of its constituent links. Since many paths in the Internet have very low loss rates, this can yield loss measures for quite a few links. However, it is not clear whether this approach is applicable to metrics other than loss rate.

Second, the stability of metrics like loss over measurement timescales is unclear. Does a ”bad” link stay ”bad” for long enough to make these methods practical and useful?

Finally, reviewers noted that the average average MILS length seemed to comprise about 3 or 4 physical links (with a few very long MILSes). Reviewers wondered whether this level of granularity would be useful in practice.

Notwithstanding these points, the paper does a good job of defining a valuable concept for tomography work (the MILS) and showing what methods are practical for analyzing network measurements at the MILS level. The paper also demonstrates the power of the linear algebraic approach to network analysis, which will likely form the basis for further improvements and advances.

The Role of PASTA in Network Measurement

François Baccelli Sridhar Machiraju Darryl Veitch Jean Bolot

Public Review by Constantine Dovrolis

PASTA (Poisson Arrivals See Time Averages) is one of those acronyms that most people in our community know, but probably few understand. Of course there is an obvious connection between measuring a network path and sampling a stochastic system, and so PASTA has been established as very important for network experimenters because it promises unbiased estimation. Even better, PASTA is a very general property as it only requires the quite reasonable "Lack of Anticipation Assumption" (LAA), i.e., that the measured path cannot somehow anticipate when it will be sampled again. PASTA's importance has been emphasized so heavily in several previous papers that it gradually became a doctrine that measurements which do not use Poisson probing are probably incorrect.

This paper will probably change dramatically the way you think about PASTA. It illustrates in an engaging manner that Poisson probing is certainly not the only probing process that results in unbiased estimation in the "non-intrusive" case, i.e., when the probing packets are so small and rare that they do not affect the measured path. A certain bias may be present in the intrusive case when using non-Poisson (e.g., uniform or periodic) probing. The point that the paper emphasizes, however, is that with a finite number of probing packets we should not focus on unbiased estimation, but instead, on minimizing the Mean-Square-Error (MSE). MSE depends on both the estimation bias and variance, and is certainly a more meaningful metric to minimize than the bias alone. In terms of MSE, Poisson probing is not necessarily optimal however. Another key point that the paper makes is that in several network estimation problems we try to estimate a certain function of the observed variables. For instance, we use (intrusive) delay measurements to estimate jitter or even to estimate the mean delay in the non-intrusive case. When this is the case, PASTA is again a sub-optimal policy because it does not do anything to avoid or reduce the involved "inversion bias".

At the more fundamental level, the paper shows that in the non-intrusive case, the lack of bias property of Poisson probes is actually shared by a much larger class of probing processes: first, non-intrusive jointly ergodic arrivals see time averages (NIJEASTA), and second, non-intrusive mixing arrivals see time averages (NIMASTA). These generalizations are interesting, if not useful, in that they are sometimes easier to check than the LAA assumption of the original PASTA. In the intrusive case, Poisson probing is provably unbiased but it can suffer from the inversion basis. The paper does not provide a solution to the hard problem of minimizing the inversion bias or the MSE. The solution to that problem, if it exists, would depend on the particular estimation problem at hand. The paper does recommend a Probe Pattern Separation Rule, however, which guarantees that successive probing packets cannot be "too close" in time. The objective of such probing is to decrease the intrusiveness of probing and thus the inversion bias.

The paper received nine (!) reviews and it generated a lot of discussion. As it usually happens with such a large number of reviews, some reviews were mostly positive while others were more negative. It should be noted however that none of the reviewers identified an important flaw in the paper. On the positive side, the reviewers liked the fact that the paper reconsiders at a fundamental level a widely believed, but rarely questioned, conventional wisdom in network measurement. One reviewer wrote that the paper "nicely cuts against the grain of a cherished principle in our community." Another reviewer wrote that "the paper goes a long way to clarify some issues that have been implicit in network measurement for a long time but have never been properly analyzed. This paper presents a foundation, both

theoretical and practical, for much better informed network measurement."

On the negative side, the paper was viewed by some reviewers as not so significant, given that it focuses more on the mathematical technicalities behind PASTA instead of proposing specific recommendations on how to improve the state-of-the-art in network measurement. One reviewer wrote that "this is PASTA for mathematicians instead of networking folks." Another reviewer wrote "Poisson probes may not always have the optimal MSE, but how far can they be in the worst case? Is there any practical scenario in which using something other than Poisson would be a significant improvement?" A third reviewer found the paper "low on novelty" in the sense that the paper mostly formalizes some observations that had been previously discussed more empirically or informally (e.g., the existence of an inversion bias is known, but what can we do about it in practice?). A fourth reviewer would like the authors to consider that in practice we only have a short measurement interval and a small number of probes before the underlying path shows signs of non-stationarity. How can we determine the probing process so that we minimize the MSE for some known classes of realistic traffic models in that case?

I would like to close this public review noting that the paper focuses mostly on delay measurements. The problem of packet loss rate estimation is much harder, and it still remains unclear how to correctly apply the PASTA property in that case. Sending Poisson probing packets (say using *zing* instead of *ping* does not measure the packet loss rate at a network path where most of the traffic is contributed by bursty TCP sources and where buffers use the Tail-Drop policy. It would be great to see some more results for that measurement problem.

XORs In The Air: Practical Wireless Network Coding

Sachin Katti Hariharan Rahul Wenjun Hu Dina Katabi Muriel Medard Jon Crowcroft

Public Review by Scott Shenker

Any sufficiently advanced technology is indistinguishable from magic.

Arthur C. Clarke

The computer revolution has left us so jaded that we take the near-miraculous for granted. We think nothing of measuring our computational power in gigaflops, our communication rates in gigabits-per-second, or our storage in terabytes. Moreover, we hardly blink before the miniature wizardry of nanotechnology. To some extent we are the victims of our own success; because of the staggering rate at which we are making advances, there are few technologies in computer science that meet Clarke's definition of "sufficiently advanced".

Readers of this paper are in for a special treat because network coding is one of these "magical" exceptions. We are all very familiar with unicast – a single packet being sent to a single destination – and multicast – a single packet being sent to multiple destinations. Network coding transcends these categories by allowing one, in a wireless setting, to do the logical equivalent of sending two (or more) different packets to two (or more) destinations *with a single packet transmission*. In doing so, network coding accomplishes the seemingly impossible feat of *increasing* the information content of each packet transmission.

The network coding concept has been floating around the theory literature since the paper by Ahlswede, Cai, Li, and Yeung on "Network Information Flow" in 2000, and much is known about the theoretical benefits of network coding. However, as anyone who has deployed wireless networks knows, there is a huge gap between theory and reality when it comes to wireless networking protocols. Most research papers tend to stay safely on one side or the other of that chasm. The beauty of this paper is that it takes the reader on the perilous journey across this divide by describing the difficult and detailed engineering required to put network coding into practice.

Equipped with a practical implementation of network coding, this paper does an extensive analysis of the benefits one might see from network coding in practice. It turns out that the performance improvement depends greatly on the particular setting. For instance, the improvement for TCP flows in a particular mesh network considered in the paper is only a few percent, but for UDP in the same network it can be as large as 300%, and for TCP in a different mesh network the performance improvement can be on the order of 40%. Similarly, when looking at UDP flows in an access network, the performance improvement varies between roughly 5% to 70% depending on the ratio of uplink to downlink traffic. The paper gives clear and satisfying explanations for these performance variations, but it isn't clear which results are more indicative of what you might see in a real deployment.

As a result, after reading this paper one is left wondering whether network coding will be the "next big thing" or another interesting idea that didn't make it out of the lab. That verdict can only be rendered by real usage, and won't be immediately forthcoming. In the meantime, we can celebrate this paper for shedding valuable light on an important new idea, and for doing so in a way that reminds us what great research looks like; a radical new idea, careful engineering, and extensive testing. This paper sets a standard to which we should all aspire.

Growth Codes: Maximizing Sensor Network Data Persistence

Abhinav Kamra Vishal Misra Jon Feldman Dan Rubenstein

Public Review by Sylvia Ratnasamy

The question of how to best retrieve sensed data from within a sensor network has, by now, been quite extensively explored. Much of this work, however, has been in the context of environmental monitoring where a key goal is to extend the lifetime of the network and, consequently, the focus has been on power efficiency and devising methods that minimize the number of packets transmitted (and hence power consumed). In this paper, the authors direct our attention to a somewhat different application. The scene is that of a sensor network deployed in an emergency situation – fires, floods, earthquakes and the like. Here the need for sensed data is immediate and the sensors themselves are at high risk of being destroyed. There is thus little point in conserving energy, and the focus instead is on maximizing the amount of sensed data than can be retrieved from a rapidly failing network.

One option might be to build a tree rooted at the sink node and relay sensed data up the tree. This would be efficient but – the authors argue – would incur delays as the tree is constructed and repaired. Moreover, naively implemented, tree-based collection could result in the permanent loss of data when nodes in the tree fail (particularly problematic if congestion causes data to accummulate at a small number of nodes along the tree). A second option the authors raise is to have each node replicate the data it discovers to a random neighbor until all data is replicated at all nodes, including the sink. This is robust and avoids having to discover the location of the sink. The downside is the sink will see a fair number of duplicates go by before it retrieves all the data.

The authors' proposal is thus to use coding to improve the rate of retrieval of the above replicate-at-random option. The coding scheme used for such a task faces two main challenges. First, because the data is spread over multiple sources in a network, the encoding technique must be amenable to distribution. Second, the encoding scheme must be robust to a failing network in the sense that, since we cannot predict what fraction of encoded data will ultimately arrive at the sink, the sink should be capable of recovering as much original data as possible from whatever encoded data it does receive. This *partial recovery* capability is important because it avoids situations in which the sink can do nothing with the limited amount of encoded data it received before the network failed. Existing coding techniques do not simultaneously meet this dual challenge and hence the authors design a new code they call a *growth* code.

Growth codes are designed to maximize the amount of information that can be recovered at a sink node at *any* point in time. This is achieved by the clever idea of growing the complexity of codewords (*i.e.*, the number of data items encoded per codeword) over time – initially, a codeword represents just a single data item, but over time, a codeword is the XOR of increasing numbers of data items. As a codeword traverses the network, intermediate nodes add data to it to build a codeword with the desired complexity for that time. The authors derive the optimal rate at which the complexity of codewords should grow to maximize the likelihood the sink will receive codewords that it can immediately decode to recover new data. They then design a distributed protocol that implements this growth and evaluate their protocol through analysis, simulation and implementation.

The reviewers were unanimous in thinking growth codes a novel and clever idea and an elegant fit for networked environments. The implementation of growth codes adds little complexity to a network application – every node independently figures out when to transition to higher complexity codewords without requiring any non-local information or difficult assumptions such as synchronized time or phases. Another nice property is that all nodes – not just the sink

– eventually discover all sensed data.

Perhaps the biggest concern among the reviewers was whether the regime in which growth codes offer a compelling performance advantage might prove somewhat limited. Relative to random replication, trees offer a more direct and efficient path out to the sink and it isn't clear that one could not build and maintain trees in a timely fashion nor that simple caching along the tree would be insufficient replication for the survivability of data. Hence on the one hand, the case for growth codes appears limited to the scenarios in which tree-based collection proves unworkable. On the other hand is the original option of simply replicating data around with no coding. Upto 50% of retrieved data, growth codes are identical to no coding, the performance gap remains fairly small upto 60-70% of retrieved data and the difference is really marked only as one gets close to retrieving all sensed data. As data in sensor networks is typically expected to be amenable to aggregation and exhibit some degree of redundancy, it isn't clear that optimizing to squeeze every last drop out is important. Nonetheless, the authors contribute an interesting twist on the problem of data collection in sensor networks and a valuable candidate design.

Ultimately, the reviewers all believed that growth codes are a novel, simple and general technique. While some papers offer the perfect solution to an immediate problem others provide us with valuable tools and algorithms that outlive the application context in which they're introduced. If you have nodes that need to replicate their data throughout a network in a manner that is speedy and robust while requiring remarkably little routing support, growth codes could be just what you're looking for.

SybilGuard: Defending Against Sybil Attacks via Social Networks

Haifeng Yu Michael Kaminsky Phillip Gibbons Abraham Flaxman

Public Review by Thomas Anderson, University of Washington

"I am not a number!" – #6, The Prisoner

What is an identity? In the social world, identity is more or less straightforward: a person, identified by family and friends, with fingerprints and DNA, represented by a birth certificate, passport, driver's license, voter registration card, or other identifying documents. In the cyberworld, identity is much more elusive, often by design. We could, after all, stamp a unique, unchangeable ID into every computation device, and insist that the device ID be used with public key cryptography to validate every action taken online. This of course would come with a loss of privacy: complete attributability of everything we might say or do online. Whether by design or by historical accident, Internet identities are not this strict. IP addresses can be easily spoofed, arbitrary numbers of email or chat room accounts set up, and multiple public keys created to represent different facets of my online self. For many reasons, I suspect we wouldn't have it any other way.

The ability to cheaply create online alter egos comes at a cost. Many distributed system and network designs that would work with a one-to-one correspondence between people and online identities, no longer work when identities can be created at will. Virtually every peer to peer system assumes some practical limit on the number of malicious users, but if attackers can easily create multiple identities, that assumption becomes invalid. In the networking arena, effective solutions against denial of service attacks also require some strong notion of identity. Exploiting multiple virtual identities is called a Sybil attack, and despite this type of attack being well-known for decades, no practical and effective solution has been proposed, until now. One might think widespread use of captchas would be able to prevent an explosion of virtual identities, but even there, the going rate for answering a captcha in real-time is a few pennies each.

This paper takes a uniquely creative approach to addressing this long-standing problem. SybilGuard leverages the graph properties of social networks to detect when people (even colluding attackers) are using multiple fake identities to undermine a distributed system. Suppose every person exchanges keys with a limited number of well-known trusted friends, and the sum of these social networks is small-world in the technical sense of the term (e.g., the graph has small diameter). The paper shows that this social network can be leveraged to prevent online Sybil attacks by observing that an attacker will have a limited number of friends in the real social network, and therefore Sybils of the attacker will also have a limited number of real friends. The detection procedure is bit involved, and so I do not attempt to reproduce it here; instead, read the paper! Or try to come up with it on your own: like most great ideas, the procedure is clever, but obvious once you read it.

There are some practical hurdles to be overcome if the approach is to stand the test of time. One is bootstrapping: While SIGCOMM is both a social and a virtual organization and so would map well onto SybilGuard, many online communities are purely virtual. There is considerable evidence that humans are particularly poor at identifying who to trust based on their virtual personas. Initial deployments of SybilGuard will need to be for applications where well-connected social networks already exist.

Second, whenever an honest user's node is corrupted, the attacker can potentially leverage that user's friends to create a number of new identities which SybilGuard would declare legitimate. The number of Sybils that can be created this way is on the order of square root of the number of participants, for each corrupted node. This becomes a serious problem for SybilGuard whenever attackers can compromise large numbers of machines. Recent studies have shown that a large fraction of all home machines are infected with spyware, and current botnets number in the hundreds of thousands of nodes. Thus, the size of botnets is roughly on par with the square root of the total number of machines on the Internet, allowing a determined attacker to create roughly as many Sybils as participants. Until this ratio significantly improves (in favor of more secure host operating systems!), the SybilGuard approach may be better thought of as significantly raising the bar, but not providing a foolproof approach.

My final concern is more fundamental to the approach and yet, strangely, more easily addressed. Judith Kleinfeld, in "Could it Be a Big World After All?" (http://www.uaf.edu/northern/big_world.html), argues that the evidence for social networks being small world is remarkably thin. On what basis do we know this "fact"? Or is it just an urban legend propagated by our desire to feel connected? Clearly, we all know who the President of the United States is, and he could locate any specific person on the planet in a small number of hops. The small world property, and SybilGuard, requires much more, however: that the network remains mixed even without such universal authorities. There are reasons to be skeptical: the average person on the planet has never been more than a small number of miles from where they were born. (Of course, the average person doesn't have a computer either.) Definitively answering whether in fact we live in a small world, or a world of small societies as Kleinfeld argues, would be a highly significant research contribution in itself. Oddly, all of this may not matter to SybilGuard; it may be sufficient for the system if its users simply favored creating a few long distance trust relationships, for example, to their favorite college professor. We all know that academic research is a very small world indeed.

Modeling Adoptability of Secure BGP Protocols

Haowen Chan Debabrata Dash Adrian Perrig Hui Zhang

Public Review by Alex C. Snoeren

This paper presents a methodical, well-reasoned study of a problem that should be of great concern to all of us, namely "Why are most protocols published at SIGCOMM doomed to become irrelevant?" (OK, so they focused on Secure BGP protocols in particular; I'd argue the lesson seems broadly applicable.) I balked at many of the sweeping assumptions the authors made along the way, but in most cases I don't have a better alternative to offer. In the end, I'm not sure how confident I am in any of the authors' conclusions with respect to the adoptability of the particular protocols they studied, but I don't believe that really matters: they equip the interested reader with enough tools and insight to conduct her own investigation.

The authors start from the assumption that an autonomous system (AS) will adopt a new secure BGP protocol if and only if the the immediate benefits of adoption are greater than some 'switching threshold' (think activation energy). For simplicity, they further assume that all ASes have the same switching threshold. From there, the paper develops a metric of 'adoptability' which indicates the highest switching threshold ASes can have that will result in widespread deployment of the protocol. The paper's key contribution is that all the recent secure BGP protocols have a very sharp transition point: for global switching thresholds at or below their adoptability value, almost complete adoption is assured. Conversely, adoption stalls out almost immediately for thresholds even slightly above the adoptability value,

Using this observation, the authors set about evaluating the adoptability of various secure BGP variants. While deterministic bounded-rational greedy AS behavior is straightforward to simulate, the tricky part is in quantifying both an AS' switching threshold and the immediate benefits it would receive. The authors sidestep the first issue by focusing on providing an ordering on the protocols, arguing that a protocol with a lower adoptability value is more likely to be deployed than another; they do not attempt to answer the question of whether any particular AS would deploy a given protocol. Even ignoring the complications of determining a switching threshold—which surely varies from provider to provider, the authors' assumption notwithstanding—determining the immediate benefits of deploying a protocol is quite tricky as they often depend on which other ASes have deployed the protocol. The bulk of the paper focuses on developing an iterative simulation methodology that starts with a set of initial adopters and computes instantaneous switching benefits for the remaining ASes.

Despite numerous simplifying assumptions, the analysis seems clever and well executed. The paper loses a bit of steam as it winds down, however. The final protocol ordering is unsurprising—except perhaps for the lack of differentiation between SPV and S-BGP—and the discussion section hits on the right issues but provides little in the way of additional insight. The notion of staged deployment in Figure 13 is particularly striking in this regard: the goal is laudable, but it isn't immediately clear why a minor protocol change has a lower switching cost than a major change (a software patch is a software patch from the point of view of a network operator), or, specifically, how one might break up any of the protocols analyzed here to stage their deployment.

The discussion section does, however, anticipate readers' obvious concerns that adoption is a process involving many factors that seem difficult to summarize in a single value. For example, ASes could be motivated to adopt strictly for competitive reasons (the protocol may have minimal added value from a technical perspective, but customers' perception is likely more important), or be pressured into adopting by political mandate or required vendor upgrades. Each of these would likely result in a different adoption process. Here again the paper would benefit from additional specificity. For example, one of my students points out it would have been nice to see a simulation accounting for real-world scenarios such as a simultaneous roll-out of two competing protocols by different vendors. It's also unclear whether the model is robust to economic externalities: Would this approach have explained the growth of Akamai?

In sum, I really enjoyed this paper despite the overly—but likely necessary—simplifying assumptions, and hope someone will pick up the mantle the authors throw down in their future work section and refine the technique to produce a realistic yet quantifiable metric of adoptability that future protocol designers can use to evaluate the practicality of their SIGCOMM submissions.

Understanding the Network-Level Behavior of Spammers

Anirudh Ramachandran Nick Feamster

Public Review by Mark Handley

Spam is a problem we all face, but at first glance it does not seem to be a hard problem to solve. The devil, of course, lies in the detail. Current spam filters do a fairly good job, but are far from a complete solution and there is inevitably an arms race between spammers and anti-spam software vendors. Thus having a good understanding of how spam as an industry actually works is crucial to understanding what approaches to take in the future. This paper provides a fascinating insight into the techniques spammers use to transmit spam, viewed from the network point of view. Inevitably it is a snapshot in time (albeit a 17-month long snapshot), and so there is fair chance that it may date quickly, but it will be interesting to see how the spammers chance their tactics in future compared to the picture Ramachandran and Feamster paint for us today.

The authors look at spam from the point of view of the IP addresses used to relay spam messages, and they analyze from three main points of view:

- the distribution of IP addresses of spam relays as compared to the distribution non-spam mail senders.
- the behavior of individual bots in a live botnet used to send spam.
- the prevalence of briefly announcing new (hijacked) BGP routes to send spam.

To perform these analyses the authors have used several datasets - a large dataset of spam from a single domain, the mail logs of non-spam email relays at a large email provider, a log of all the IP addresses of the bots in a botnet used by spammers, and a trace of all BGP route changes seen by the domain receiving the spam.

For me, the comparison of where spam comes from versus where legitimate email comes from is the least satisfying part of this paper. The main reason for this is that, although it may perhaps be reasonable to assume that spam is sent to domains in a relatively untargetted manner, it's not reasonable to assume that legitimate email is similar. Even for a large email provider, it is reasonable to assume that email has some geographic locality. Thus the distribution of legitimate email across the address space is likely to vary depending on the vantage point. Thus it's hard to draw any strong general conclusions from such a comparison, and the authors are obviously wary of doing so. But unfortunately this leaves this part of the analysis feeling incomplete - there is a lot of interesting information, but it's hard to draw conclusions from this information. I'd like to have seen more logs of legitimate email used (especially from different countries), so that we can see what normal variation is in legitimate email. Without this it's hard to see if the spammers are doing something substantially different.

The analysis of botnet behavior is fascinating. The authors have managed to obtain a trace of all the IP addresses in a particular botnet at one moment in time, and use this to analyze the behavior of individual bots. Of course this is complicated by some of these bots using dynamic addressing, so again the conclusions have to be qualified. But the insights are nonetheless intriguing - most bots send spam to a domain only once, and each bot rarely sends very many messages to that domain. Presumably each bot sends to many domains though, which indicates that collaborative anti-spam techniques are likely to be needed rather than looking at the behavior of individual bots. But even so, the results indicate that current IP address based blacklisting is of limited effectiveness.

Finally the paper looks at the use of "BGP Spectrum Agility", whereby a hijacked IP address range is briefly advertised via BGP and used to send spam. This is not the most prevalent technique observed, but it is perhaps the

most sophisticated. The results are a surprise - large /8 prefixes were seen to be hijacked, and IP addresses widely distributed within the prefix were used to originate spam. This has clear implications for the need to secure the routing infrastructure.

Overall, this paper cannot do what I wish it would, which is to reach some form of ground truth for how spamming functions today. Although the datasets used are substantial, they are not sufficiently diverse to draw such strong conclusions. The authors should not be faulted for this though - this paper represents a great deal of work, and the insights provided are indeed intriguing and enlightening. To get closer to ground truth would require a large international collaborative effort, many more datasets, and a huge amount of analysis. I hope this paper will serve as a call-to-arms to others to continue this work in a collaborative manner.

DDoS Defense by Offense

Michael Walfish Mythili Vutukuru Hari Balakrishnan David Karger Scott Shenker

Public Review by Adrian Perrig

The defense against Distributed Denial-of-Service (DDoS) attacks continues to be an important research challenge. With the proliferation of large networks of compromised hosts (a.k.a. botnets), DDoS attacks pose a significant threat to on-line services. This paper considers DDoS attacks that exhaust server resources such as computation.

The paper proposes the *speak-up* approach, where clients are automatically encouraged to obtain their fair share of server resources by increasing the magnitude of bandwidth consumed by a request. The technique follows the line of DDoS defense research where clients are required to pay a virtual currency to obtain service. In approaches with computational puzzles that currency is client computation, in speak-up it is client upload bandwidth. Similar to other virtual currency schemes, a client in speak-up obtains a share of server resources proportional to the amount of virtual currency it can afford.

In more detail, the speak-up approach works as follows. A high-capacity server, called the *thinner*, handles all client requests and decides which requests to forward to the server. In case of server overload, the thinner starts to throttle client requests it forwards to the server. Furthermore, while the server is overloaded (e.g., during a DDoS attack), the thinner encourages all clients (good and bad ones) to send large quantities of congestion-controlled traffic towards it and keeps track of the amount of traffic received from each client. To decide which requests to forward, the thinner engages in a continuous virtual currency-based auction. Whichever client sent the largest amount of traffic to the thinner wins and its request is forwarded to the server.

This approach works well if all of the following conditions hold.

- The thinner's access links are not congested, even if all clients and attackers send traffic at maximum speed. If the links were congested, legitimate clients would back off due to congestion control, while attackers could ignore congestion control and send at higher capacity.
- Clients have spare upload capacity to send drastically higher amounts of traffic towards server, otherwise they could not increase their chance to obtain service by much.
- Attackers already send at high capacity and cannot send at much higher capacity even if they are asked to speak up.
- Clients will send at maximum rate until they obtain a connection, which could potentially take a long time under a strong attack.

The main advantage of this approach is that the network elements do not need to be changed. Many prior approaches require router changes, some even client changes. Speak-up only requires server changes, the addition of a thinner with high-capacity network access, and a client that can support Javascript (or another language that can be instructed to launch massive amounts of traffic towards a server).

Speak-up's main shortcomings include the following. A good client would need to flood the server under attack, which would make it more difficult to detect malicious clients since now even good clients flood. Despite the

congestion-controlled nature of good client's flooding traffic, especially edge networks would still impacted by increased traffic volumes, potentially harming other flows. If the access links towards the thinner become congested, the properties of the speak-up approach are likely to deteriorate (not evaluated in paper).

In summary, speak-up represents an innovative angle to addressing DDoS attacks. As the evaluations demonstrates, the approach works well under the tested assumptions and conditions. The paper is very well written. Future work will need to evaluate the approach under severe DDoS attacks and compare it against other alternative approaches to further demonstrate speak-up's viability.

Beyond Bloom Filters: From Approximate Membership Checks to Approximate State Machines

Flavio Bonomi, Michael Mitzenmacher, Rina Panigrahy, Sushil Singh, George Varghese

Public Review by Ellen Zegura

There once was a finite state machine,
Whose size made router developers quite green.
Said Michael to Flavio,
"We can make this compactio,
All we need is a Bloom filter supreme."

(With apologies to Rina, Sushil and George – you'll go in the next limerick.)

It would be difficult to point to a data structure that has seen more innovative use in networking over the last ten years than the Bloom filter. (Thanks, Burton.) The clever ability of a Bloom filter to represent the membership of a set approximately and compactly has turned out to be useful for network applications ranging from efficient caching to network monitoring. By their nature, Bloom filters are especially useful when the set is very large and the application can tolerate false positives.

As the title advertises, this paper goes beyond Bloom filters to approximate finite state machines. One can view the motivation for the work in this paper from both a theoretical perspective and a practical perspective. First, as a pure issue of data structures and algorithms, one might ask how to go beyond the approximate set membership supported by Bloom filters to represent other approximate information. Concurrent membership in a finite state machine is a nice and natural step beyond set membership. It isn't a small step – state machines are subject to ill-behavior on the part of flows that requires clever mechanisms to cope.

The authors begin with the "direct" use of a Bloom filter, where the set consists of (flow-id, current state) pairs. That is, the state of each flow corresponds to an element in the set. State is modified by set lookup followed by set deletion followed by set insertion. The authors then develop two additional data structures that improve on the direct Bloom filter approach. The key innovation is to allow a "don't know" state in the Bloom filter to represent set membership (rather than just a "yes" or "no") – a sort of politician's Bloom filter. That is, a query to check if flow X is in state 3 could return yes, no or don't know. The insight is that "don't know" may be far more useful to an application than a false positive or false negative. There are some fun and important details related to ill-behaved systems, for example, invalid state transitions, as well as the need to clean out the data structure to remove non-terminating or ill-behaved flow state.

From a practical perspective – this is SIGCOMM after all -- the authors motivate interest in approximate finite state machines by observing an increasing trend to put transport and application-layer information in network devices for better performance, security, etc. Network devices at the core (and at some edges) must track many flows at a high data rate, hence compact data structures that can fit in fast memory are appealing. Two specific applications are used as examples: active queue management for MPEG video flows and, less developed in this paper, real-time detection of peer-to-peer traffic. The authors admit that the investigation of applications is preliminary, and I would agree.

It seems to me that the proof of the practical value of this paper will emerge over time, as the creative minds of network application developers add ACSM's to their toolbox. Go forth and approximate.

Detecting Evasion Attacks at High Speeds without Reassembly

George Varghese J. Andrew Fingerhut F.Bonomi

Public Review by Jon Crowcroft

”Let no one else’s work evade your eyes, Remember why the good Lord made your eyes, So don’t shade your eyes,” Nikolai Ivanovich Lobachevsky[1]

This paper reshapes the agenda set in the work in the seminal paper on traffic normalisation by Christian Kreibich et al:) basically, they say they can split the string they are trying to match into small pieces, and then run a match algorithm on the pieces of packets and pieces of search string they are trying to (exact) match, and even if the sender tries lots of evasion (interspersing long ”chaff” packets with overlapping sequence numbers or transmitting the chunks out of order, they can still do a simple fast path detector, then switch to a slow path - one major problem in the work that they admit to is is that they require a ”small” modification to endnodes (receiver) TCP processing to reset [RST, 2] sources that send mismatching data in segments that overlap pre- or post-existing segments sequence space. Of course (as they to point out, and discuss the limitations of) that such a change would lay open any TCP to a single packet take down by anyone able to spoof the sender’s source address simply sending random data in a guessed current part of the sequence number space (there are ways to guess that too); oh, and bad guys ignore resets [2].

Anyhow, thats not the main point of the paper!

Another shortcoming in the work is that they only do exact matches and not regex or any approx matching. They do, however, demonstrate they can go very fast (10 times faster than existing systems, e.g .easily at 10Gbps).

Plus points of the paper are 1/ it does a nice job of hinting that there MIGHT be such ways to do things like this without normalisation. 2/ the paper is beautifully written (mainly) in the style of some of Dave Clark’s early RFCs - especially the nice list of ”folk theorems” and their downfalls (e.g. they cite one of his papers on Upcalls, in their discussion other myth/theorems such as why layering is bad:) 3/ 4 byte chunk might be a problem - c.f. polygraph indicates that the invariant part of polymorphic worms might be as small as 1 or 2 bytes only[3].

A couple of thoughts spring to mind: a) there may be a principle here about well behaved/designed protocols and the complexity of normalisation versus detection of anomalies in un-normalised flows - for example, if TCP were sensible and packet based rather than byte based, many of these attacks would go away (and same argument has been oft made about why IP fragmentation is a bad idea). b) Could one use homomorphic hash functions to build the match engine on pieces? This would make the matching process order invariant.

Conclusions: a nice old-style computer science algorithm paper worth reading as an exercise for the mind, but also, a step towards something that might make a day in the life on an old tired IDS a bit less hard work.

[1] ”Index I copy from old Vladivostok telephone directory.” Tom Lehrer, 1959, TW3, and Harvard.

[2] ”Ignoring the Great Firewall of China” Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson http://www.cl.cam.ac.uk/ rnc1/talks/060628-Ignoring.pdf

[3] Newsome, J., Karp, B., and Song, D., Polygraph: Automatically Generating Signatures for Polymorphic Worms, in the Proceedings of the IEEE Symposium on Security and Privacy (Oakland 2005), Oakland, CA, May, 2005.

Algorithms to Accelerate Multiple Regular Expression Matching for Deep Packet Inspection

Sailesh Kumar Sarang Dharmapurikar Patrick Crowley Jonathan Turner Fang Yu

Public Review by George Varghese

A key issue often debated by the SIGCOMM PC is whether a particular paper belongs to ”core networking”: does the paper deal, however admirably, with a topic that will appeal to only a small segment of the SIGCOMM audience? Of course, if proven this has the pleasant side effect of allowing a paper to be rejected without a deep review.

At first glance, the topic of analyzing packet streams for a match with one of a thousand possible regular expressions at Gigabit speeds appears to be a terrific candidate for cutting using the ”core networking” razor. Several PC members understandably argued on this basis (but to their credit added insightful reviews as well).

While this paper may not immediately appeal to the SIGCOMM audience that enthuses over P2P and large scale measurement research (and I like that kind of research as well), a mandate for any SIGCOMM PC is to introduce new areas for future researchers. It is this contention — that research into wire speed regular expression processing is worthwhile as a continuing research saga — that I will argue for.

The context comes from security. Early Intrusion Detection/Prevention Systems (IDS/IPS) often detected an attack using an exploit signature — a string of bits which if detected can cause the IPS to drop the packet. But a specific exploit (such as the Code Red Worm) takes advantage of an underlying vulnerability/software bug (e.g., the IIS Buffer overflow). An exploit signature can be foiled by a simple change of exploit — e.g., by doing the same buffer overflow with a different set of strings and code. Thus, security vendors (e.g., ISS, Intruvert, NetScreen, Cisco) have realized for years that it is best to use more abstract vulnerability signatures that cover all exploits that use a vulnerability. For instance, for a buffer overflow one wants to detect attempts to send data to a particular buffer name that is longer than a specific length. This is often described using a regular expression, often using PCRE (Perl Compatible Regular Expressions) syntax. Thus the popular Open Source IDS Snort has morphed from having very few rules with reg-exes to having a large fraction with reg-exes in recent years. In fact, on a certain Tuesday of every month when Microsoft releases their recent vulnerabilities it is very hard to contact IPS signature writers as they are busy writing vulnerability signatures (hopefully) before an attacker can construct an exploit

Despite some questions about the effectiveness of IPS, they are deployed widely in the Fortune 500 enterprises and and so are approaching a billion dollar market. Now the further twist is that the core networking vendors have eyed this large security industry with great interest, and would like to steal the security vendor’s lunch by integrating security with switches and routers. Thus all the enterprise switch vendors (need I name them) are working on integrating security into their switches. At first, this is being done by separate service cards, but the emphasis is on putting such IDS security functions in every switch line card. Putting IPS/firewall functions in every line card allows a much finer granularity notion of security than was possible when IPSs were only deployed on the periphery of every network. Instead, IPS (and firewalls) could segment every local area network in an enterprise preventing attacks from crossing regions of an enterprise and also thwarting internal attacks (caused, say, by bringing in infected locktops. A colleague of mine calls it providing a ”lock for every door”.

But putting IPS in every line card implies doing IDS functionality cheaply at 20 - 40 Gbps because that is the

speed of most enterprise switches today. If you put this together with the fact that most IPS's today have thousands of regular expression, the problem of doing wire speed RegEX at 1-10 Gbps is very important to security and router vendors today.

Should the SIGCOMM audience care about what may be a hot topic for router vendors. Well, consider fair queueing (FQ) as a precedent. FQ became a consideable cottage industry in SIGCOMM and there are hundreds of papers on the efficiency of fair queuing. The SIGCOMM industry has grown to love (and has contributed enormously to) FQ mechanisms within routers. As with FQ, regular expression research is also freighted with algorithmic questions. Also, there is a natural progression: from wire speed prefix lookups to wire speed classification to wire speed exact and RegEx matches. The earlier points in the series have been well represented in SIGCOMM, so why not wire speed reg-ex?

Could wire speed reg-ex research be obsoleted by changes to systems or protocols? I do not think that clean slate approaches (for example, strong authentication) by themselves will obviate the need for packet inspection. If hosts are modified to store signatures (as in the Microsoft Shield work), there will still be need for reg-ex processing at the hosts, albeit at somewhat lesser speeds. An interesting paradigm shift (advocated by Vern Paxson among others) is to use parsing so that one is not looking for strings without context. But it is not clear that parsing is any easier at high speeds.

There was also debate in the PC about whether everything about RegEx matches had been said earlier and better in the compiler community. But earlier results are not encouraging. The standard approaches are to either use an NFA (hard to simulate on a uniprocessor, parallel NFA logic for NFAs is often unscalable) or to use a DFA (but DFAs blow up in storage for reg-exes with wildcards such as $A_i * B_i$ because of the need to keep state for "cross-product" states in which the DFA has encountered A_i and A_j). More fundamentally, previous networking solutions to even standard algorithmic problems like prefix lookups show us that sheer speed (10-100 memory references) and limited fast memory makes the networking solutions at least somewhat different from earlier software counterparts. One has to exploit hardware, make some assumptions about the data, and so on.

Ah, what of the paper? Well, you really must read it for yourself. I will say that this paper uses DFAs and suggests a method to compress redundant edges in the DFA graph. Earlier work suggests ways of getting rid of egregious nodes in the DFA graph, but this paper removes repeated edges by substituting a "default edge". But default edges create longer paths in the DFA graphs. This creates a storage-time tradeoff that the authors explore very nicely. While this idea is suggested in earlier Compiler work, the development seems different and the hardware context needs new ideas to avoid memory bandwidth bottlenecks.

In more detail, the authors introduce a data structure called D(2)FA. This is an extension of a standard DFA, where if two states have common transitions, then one of the states is made to "point" to the other state, by having a default transition without consuming an input character. It is possible that the D2FA takes very long to traverse because for an "n" input string one could be following a default transition for every input character, and this might take too much time to process the string. The authors propose an algorithm for coming up with a bounded time D2FA by not compressing all edges if they violate the time bound. They then also come up with a hardware architecture to map the D2FA onto an ASIC with multiple memory blocks and logic. Parallelism is a hard problem because the shared states in regular expressions becomes a bottleneck. The authors present both randomized (intuitively easy) and deterministic methods (harder) to tackle this problem.

This is nice work and a great start, but I suspect that even the authors do not feel that the full story on wire speed Reg-Exes has been written. There is more to do. This is a neat paper in a high profile conference that could serve to inspire some future research team, sufficiently well versed in algorithms, programing languages, and hardware, who will tackle the RegEx problem (using the standard data sets used in this paper for comparisons) to find the next break though.

Virtual Ring Routing: Network Routing Inspired by DHTs

Matthew Caesar Miguel Castro Edmund B. Nightingale Greg O'Shea Antony Rowstron

Public Review by Brad Karp

If a friend summarized a networking paper to you as "DHTs meet ad hoc routing," you might cynically think you were being challenged to a duel of disdainable paper topics, in which the one who proposes the paper that is the most excruciating juxtaposition of two tired areas wins.[1] ("Oh yeah? QoS meets multicast!" you might competitively respond.) How pleasing, then, to read a paper like this one that demonstrates—somewhat surprisingly, even to a non-cynic—that algorithmic structures from the well studied area of DHTs can be adapted to create a promising, novel solution to the well studied problem of ad hoc routing (and perhaps more broadly, network-layer routing in other settings).

In the broad literature on overlay-based DHT routing systems and layer-3 routing systems alike, a central theme has been the trade-off between the quantity of state held at each node and the stretch of routing (*i.e.,* the ratio between the length of the path found by the routing system and the length of the shortest path). Intuitively, holding progressively more state at each node (describing more and more of the network's topology) can allow routing with progressively less stretch. On a dynamic network, in which links and nodes may fail, however, there is another important cost beyond the storage at each node: the corresponding bandwidth burned keeping each node's state up to date when the topology changes. In sum, the metrics typically used when evaluating any routing system, overlay or layer-3, are stretch (path length), state per node, routing protocol message cost (including the message cost incurred to correct state after a topological change), and robustness under topological change (*e.g.,* on a dynamic topology, the fraction of users' packets delivered successfully).

In Virtual Ring Routing (VRR), the authors' creative insight lies in how they adapt a Pastry-like ring structure, previously used in a DHT overlay above layer 3, to instead provide layer-3, any-to-any routing *itself.* As in Pastry, each network node in VRR is identified by a unique, random, flat ID, and nodes are ordered in a ring by this ID. Each node maintains a routing table whose entries point toward that node's $r/2$ immediate predecessors and $r/2$ immediate successors on the ring. The key difference between a DHT overlay and VRR concerns how a node routes to its predecessors and successors. A node in a DHT overlay may simply invoke layer-3 routing to reach its predecessor or successor. But because VRR sits directly above the link layer, VRR nodes must set up forwarding entries to their successors and predecessors along what are typically multi-hop physical paths (through other VRR nodes). As a result, a VRR node contains not only routing table entries that point toward its nearest neighbors in ID space, but also entries for the paths between other pairs of VRR nodes on which it sits. VRR greedily forwards a packet toward the node in the routing table whose ID is closest to the destination ID in the packet.

The twist is that routing around VRR's ring requires fewer hops than you might think, because of a state size-stretch trade-off. The authors show by way of back-of-the-envelope analysis that a VRR node will keep routes to an average of $O(\sqrt{n})$ nodes in an n-node network. So a VRR node knows about *many* nodes besides its predecessors and successors, and has accordingly higher probability of "shortcutting" across the ring, rather than walking it linearly.[2]

I find VRR a novel, promising approach to layer-3 routing, and the anonymous reviewers (whose comments I incorporate in all that follows) unanimously agree. The breadth of the real-world evaluation included in the paper is impressive, as well: the authors built VRR implementations for both sensornet (mote) and 802.11a (PC) platforms, and include measurements of VRR on testbeds for both platforms that show VRR performing comparably to BVR and MR-LQSR in some experiments and outperforming them in others. The real-world evaluation is particularly heartening against the backdrop of the dearth of real-world validation in the ad hoc wireless community, and the inadequacy of simulation for modeling radio behavior accurately.

My quibbles with the paper reflect a desire to understand more about the details of VRR: the exact design goals

[1] In the case of those two particular areas, your public reviewer tars himself with that broad brush.

[2] One might liken this behavior to Chord with a finger table of size $\sqrt{n}$, populated with nodes selected *randomly* from around the ring.

for the protocol, precisely why the protocol performs as it does, and the extent to which the original design goals are responsible for the protocol's performance.

It is somewhat surprising that the authors do not set out narrow, specific, *positive* scaling goals for VRR's design, even though scaling is the crux of routing protocol performance. The only goals stated in the introduction are to build a DHT-inspired routing protocol (hardly tangible), avoiding flooding, as used by some prior routing protocols, and avoiding location-dependent addresses, as used by other prior routing protocols. Harking back to the above discussion of trade-offs in prior routing systems, how might one set scaling goals? First, consider the particular network on which you need to route (*e.g.,* number of nodes, diameter, density of topology, rate of change of topology, memory available per node, *&c.*), and next, aim for scaling properties that fit it best. For example, on current sensor network nodes, such as on the various mote platforms, per-node storage is severely limited; in such a network, minimizing per-node state may be important. What are the big-O scaling goals for VRR for metrics such as per-node state, stretch, and messaging cost to maintain up-to-date state, and how do other routing protocols big-O scale by those metrics?

I long for more nuanced explanations of *why* VRR performs as it does. The evaluation presents a wide range of experiments, in ns-2 simulations and on two testbeds, that show VRR often outperforms other protocols, and sometimes performs comparably to them. One gets the sense at moments that the evaluation does more to show that VRR *wins* than to *explain* its and the other algorithms' performance, and carefully ascribe behavior to particular algorithmic details.

For example, what is the *distribution* of state per node across the network? While the authors give a back-of-the-envelope prediction that the mean state per node should be $O(\sqrt{n})$, the evaluation does not present any measurements. It seems likely that the distribution will be skewed, as nodes toward the center of the network will be on more paths than those at the edges (given that predecessor/successor relationships are between randomly selected nodes). Does the state in a node at the core of the network grow faster than $O(\sqrt{n})$? Moreover, if some nodes hold more state than $O(\sqrt{n})$, do they play a disproportionate role in finding alternate paths after nodes die?

I am puzzled by the authors' explanation for why VRR finds paths with low delay: that VRR can recover from path failures without discovering new routes, and repairs broken routes efficiently. Many of the other routing protocols the authors simulated and ran in deployment share the former property (*e.g.,* DSR will often have alternate routes cached when one route fails, BVR can simply forward greedily via a different neighbor when a neighbor dies, *&c.*), so it does not seem to be distinguishing in and of itself. What specifically about the local repair scheme is better than these other protocols' analogous schemes? It's difficult to assess from the evaluation in the paper whether VRR repairs broken routes efficiently—there may be sufficient path redundancy to mask individual paths that remain broken for some period. Moreover, it seems the operative question is whether VRR repairs paths *more* efficiently than other protocols. Without the relevant direct comparisons, *e.g.,* moving or killing a set of nodes, measuring the count of packet transmissions incurred by VRR to repair the broken paths, and repeating the same measurement for other routing protocols, it's hard to answer that question.

The authors seem to suggest that VRR outperforms MR-LQSR in their first 802.11a experiment simply because MR-LQSR uses longer per-packet headers than VRR. If so, this result does not seem to have much bearing on VRR's inherent scaling properties.

Finally, it is difficult to compare the performance implications of protocols' loss-rate awareness in a network of diameter 4 (as was the case for the 802.11a testbed). Measurements taken on the Roofnet outdoor 802.11b testbed indicate that (a) Paths that incorporate relatively high-loss links may offer greater throughput than paths consisting of lower-loss links, and (b) Wireless links' packet loss rates are distributed across the entire range of $[0, 1]$, so there is no obvious loss rate threshold above which to exclude links.

Greedy-style routing algorithms, including BVR, GPSR, and VRR, tend to use loss-rate thresholding to filter which physical links are available to the routing protocol. VRR uses the ETT generalization of this technique; it uses both loss rate and link bandwidth to exclude low-throughput physical links from routing. Because the authors' 802.11a testbed is four hops in diameter, VRR's two-hop-limited comparison of paths' throughputs spans half the longest route in the testbed (assuming that the routing system finds shortest paths, or nearly so). There is reason to wonder whether in a network of greater diameter, comparing ETTs for end-to-end paths without *any* link thresholding, as done by Roofnet's Srcr, might offer greater throughput than eliminating lossy links by rigid thresholding and comparing ETTs only at a two-hop horizon, as done by VRR. In a wireless network of sufficient diameter, if there are points that feature highly localized loss, one could imagine that a greedy forwarding choice using knowledge only of loss within two

hops might walk down an end-to-end path that traverses only high-loss links later.

VRR is a fresh, subtle, and promising design. The notion that DHT routing techniques may bring benefits to layer-3 routing is compelling, and one that deserves further exploration. Given that the authors have built real implementations of VRR for motes and a PC-based networking stack, I hope they will release their source code, to help foster such further exploration.

ROFL: Routing on Flat Labels

Matthew Caesar Tyson Condie Jayanthkumar Kannan
Karthik Lakshminarayanan Ion Stoica Scott Shenker

Public Review by Arun Venkataramani

Among all the ideas in Sigcomm 2006, ROFL is easily the most disruptive one that, if ever adopted, would completely change the face of the Internet.

The authors set out to address the ills of mixing location and identity in Internet addressing. Today, an IP address serves both to identify a host and encode information about where the host is attached to the network. Indeed, the ability to group nearby addresses into prefixes is key to scalable Internet routing. However, mixing location and identity creates engineering headaches when a host can have multiple locations (mobility) or when a host has multiple identities (multihoming). A slew of proposals such as Mobile IP, IPv6 with multihoming, and follow-on works address these as special cases, usually, by introducing a level of addressing indirection.

But, the authors appear to be architectural purists. They take the idea of architectural uniformity to the extreme, and propose to cleanly separate location from identity by getting rid of the notion of location altogether from Internet addressing. Instead, ROFL routes directly on *flat* identities devoid of any structure (intuitively, like routing directly to MAC addresses). Even as one wonders about the practical use of such a radical change, the authors reveal that their real motivation is to question popular wisdom that structured addressing is necessary for scalable routing. This audacity and the engaging architectural exercise of running far enough with it to convince a reader that ROFL, although far from perfect, is not unthinkable found favor with the reviewers, making it one of the most highly rated papers.

The technical ideas behind ROFL borrow heavily from prior work in distributed hash tables (DHTs). Intradomain ROFL is essentially a DHT implemented directly over the link layer with no other underlying connectivity. In this respect, ROFL is similar to VRR or Virtual Ring Routing, a DHT over the link layer targeted for wireless networks, also published in this conference. Like in Chord, routing reachability is ensured by maintaining successor and predecessor pointers along a virtual ring of identifiers. However, as there is no underlying network layer, routing to next-hop neighbors on the virtual ring is implemented via source routing — each node maintains a source route to its successor and predecessor along the ring. Furthermore, intermediate routers cache routes passing through them. Routing is greedy, i.e., each router forwards a packet to the physical next-hop router along the route to an ID that is closest to the destination, but does not exceed it. Joins and failure recovery work by adjusting predecessor and successor pointers and tearing down cached pointers. To avoid failures from partitioning a ring into two disjoint rings despite underlying physical connectivity, routers periodically flood the smallest ID in their ring that will ensure that the rings are merged.

Interdomain ROFL works by merging intradomain rings up along the hierarchy of provider ASes. Thus, each node has successors and predecessors corresponding to each level of the AS hierarchy. In this respect, ROFL is similar to Canon, a DHT that supports hierarchies of traditional DHTs. A useful invariant that ROFL maintains, known as the *isolation property*, is that the path between any two IDs lies physically within the hierarchy created by the least com-

mon ancestor AS up along the provider hierarchy. ROFL reduces stretch, i.e., the inflation compared to the shortest underlying path, by maintaining *proximity-based fingers* that, similar to Pastry, are digit-correcting IDs belonging to the smallest ancestor hierarchy. For applying policy such as preferring peers over providers, ASes have no easy way of determining which IDs belong to which ASes. The authors suggest two alternatives: the first to introduce a virtual provider AS for any clique of peering ASes and rely on the isolation property for choosing peers over providers, and the second is to exchange large bloom filters to keep track of whether an ID belongs to the hierarchy of a peering AS or not. In some cases, this hack might result in a packet traversing a peering link only to be returned back along the link upon a false positive. The authors outline ways to perform limited interdomain routing control, multihoming, anycast, multicast, and enable security capabilities. The objective here appears to be to illustrate that such primitives are conceivable in ROFL, not to actually design them in any convincing detail. The lack of sufficient detail appears to be endemic, but perhaps it is unavoidable given the breadth of the paper.

The strength of the paper is the definition of the problem itself, a preliminary design, and an evaluation over real topologies that suggests that, with sufficient router memory, routing on flat labels may not be as impractical as it sounds. It is commendable that the paper touches upon practically every aspect of network layer design in 14 pages. But the paper also leaves a reader wanting for more. For example, it is unclear what the asymptotic scaling properties of state in a router and average path lengths are. The control overhead of a host join and failure appear to be high and can cause ripples in distant ASes; the isolation property is only for data paths. There is basically no support for traditional traffic engineering other than breaking up a large AS into smaller ones and relying on interdomain policy. Although clean support for mobility and architectural uniformity are stated goals, the design ends up classifying hosts into two categories, *ephemeral* and *stable*, anyway to reduce control overhead. The experiments show that, with sufficient router memory, stretch can be made close to one. I believe the stretch refers to average stretch, but says little about the worst case or the 90'th percentile stretch. The protocols reveal much more information about AS relationships than is easily inferable today. The lack of any structure in identifiers hurts scalability of Internet-scale measurement and performance prediction techniques.

These shortcomings are besides the point of the paper, which essentially is an architectural exploration. They can be viewed as a contribution in that they help in concretely identifying challenges that must be addressed to make routing over flat names practical. The authors are under no pretense either, "*The results are close enough to tempt, but not enough to satisfy*", they say. The spirit of the paper is perhaps best captured by the following remark, "*The art of architecture is gracefully maneuvering within the boundaries of the possible. Our goal here is to investigate whether those boundaries can be expanded, not to seek grace*". Whether the expanded architecture can enable any useful functionality not achievable in today's Internet, only time will tell.

Policy-based Routing with Non-strict Preferences

Chi-ken Chau

Public Review by Walter Willinger

In recent years, Sigcomm has published a number of high-quality routing papers that can be viewed as stepping stones towards a general theory of network routing (e.g., Griffin and Wilfong, Sigcomm'02; Sobrinho, Sigcomm'03; Griffin et al., Sigcomm'03; Griffin and Sobrinho, Sigcomm'05). Given the importance of routing for the health of the Internet, such a general theory that treats in a common framework the various routing strategies currently implemented in the real world is highly desirable. The reviewers felt that the paper by C.-K. Chau on "Policy-based Routing with Non-strict Preferences" fills an important gap in the existing theory. The author studies policy-based routing where nodes may have paths that are either incomparable or equally good (referred to as "non-strict preferences" – more on this unfortunate notation later). The paper is largely an extension of previous work that developed an abstract algebraic formalism to deal with dynamic, distributed network routing, but does a good job motivating the need to account for the so-called non-strict preferences and illuminates the problems in the context of a number of simple examples. The main results consist of sufficient conditions for the existence, optimality, and asynchronous convergence of stable routings when nodes may have to select among sets of incomparable or equally-good paths.

That said, there are some definite impediments that make this paper difficult to read, comprehend, and appreciate. The reviewers all agreed that, like many of the previous papers in this area, the presentation of the technical results is heavy on notation and short on intuition. One reviewer confessed "experiencing bad gadget/dispute wheel-fatigue", expressing a common concern that there must be more to a general theory of routing than bad gadgets and/or dispute wheels. More specifically, this paper's use of terminology is often confusing and at times obfuscates the key issues. Despite some reviewers' pleas for clarification, the paper's title suggests that the central issue is "strict" vs. "non-strict" preferences, but this is not the case. For one, the term "non-strict preferences" is explained in Section 1 and simply refers to "non-deterministic, multi-path routing" where nodes are faced with either incomparable paths or with sets of equally-good paths. Moreover, the algebraic framework that was introduced by Sobrinho and that is exploited in this paper is based on preference orders that are total orders, but the explanation provided in Section 3.1 of how the different notions of order relations can capture strict vs. non-strict preferences is not coherent.

This leaves the reader scrambling as to the "real" problem that this paper is trying to solve. In a public comment posted on 8/15/06, Tim Griffin suggests a possible formulation of the "real" problem: assume a total pre-order on routes, and imagine defining, for any (deterministic and equal-cost) multi-path solution, a compatible set of (non-deterministic) uni-path solutions; i.e., each such uni-path solution is contained in the multi-path solution. A natural question is under what conditions these two different notions amount to "the same thing," and the paper shows that the sufficient conditions of [4] are directly applicable. Assuming Griffin's formulation captures the essence of the problem that this paper attempts to study, the author missed an opportunity to clearly define the paper's main objective and present the results in a coherent and consistent manner. This is rather unfortunate, because at the heart, the results of Section 4 are interesting, with potential for important future work on, for example, decomposing multi-path solutions via suitable non-deterministic tie-breaking into compatible sets of uni-path solutions.

Quantifying Skype User Satisfaction

Kuan-Ta Chen Chun-Ying Huang Polly Huang Chin-Luang Lei

Public Review by Flavio Bonomi

This paper discusses a wildly successful peer-to-peer application, Skype, which is in many ways bringing about a revolution in voice communications, with its creative use of Voice over IP, and which is driving Service Providers crazy in their effort to identify Skype calls using their facilities for free. My mother, 78 years old, a loving Italian grandmother, is an avid user of this technology!

The key issue addressed here is the quantification of Skype user satisfaction. This is a challenging goal, when Skype calls are tricky to identify, when there is no easy access to the end-to-end voice signals, and when the measurement is supposed to be non-intrusive.

The authors take on this challenge armed with creativity and a solid statistical toolbox.

First, they define a simple but sufficiently effective filtering method to identify potential Skype calls, based on monitoring requests to a well know server, catching the ensuing communication, and identifying the dynamic port numbers used by the end hosts. This approach provides already a useful contribution.

Next, they collect measurements on calls that are considered .active., i.e., satisfy a set of requirements on their moving average packet rates and packet sizes. .Relayed. calls are also identified. The key characteristics of Skype calls they monitor, either directly, or via creative and non-intrusive measurements on the call path are: call duration, call bit rate, jitter and round-trip times.

Based on a solid statistical study, including survival analysis, correlations, regression analysis and hypothesis testing, they conclude that:

1) Call duration is a key measure of user satisfaction (short calls are usually bad, since the cost is not an issue here!).

2) The most important QoS measures impacting the quality of a Skype call, i.e., its duration, are bit rate, jitter and round-trip times, and, impressively, they come up with their relative impact: 46

3) They define and statistically validate a User Satisfaction Index, a precise function of the three QoS measures above In their thorough study, the authors pursue one more step: the validation of the quantitative User Satisfaction Index with correlating it to a number of independent metrics that try to quantify interactivity and smoothness of the conversation.

Once more, it is very challenging to infer interactivity and smoothness of a conversation when such conversation travels within encrypted packets!

Undeterred by this new challenge, the authors find new and creative ways to identify such characteristics of a call from external study of the traces, applying wavelet de-noising, and a soft thresholding technique to detect call activity, or talk bursts.

The final cross-validation of the User Satisfaction Index with these independent metrics is perhaps somewhat less satisfying than the previous epic journey. While the results are encouraging, and the Index seems to work appropriately, the correlations used for this validation seem a bit weak. This may not be a problem of the newly defined Index, but a limit in the difficult validation process.

In summary, I found this paper well written, thorough in its investigation approach and rich in contributions. I was particularly impressed by the mix of data collection techniques and statistical inference displayed by the authors.

At a time when networks are trying to embed capabilities that may help to enhance the delivery of services and applications, this work offers an innovative perspective on which network metrics are relevant to the behavior of an emblematic application, such as Skype.

I enjoyed this paper, and learned from this reading. Thanks!

Enabling Contribution Awareness in an Overlay Broadcasting System

Yu-Wei Sung Michael Bishop Sanjay Rao

Public Review by Srinivasan Seshan

Enabling efficient multicast communication on the Internet has preoccupied the networking research community for the past two decades, if not longer. Through the late 1980's and the 1990's, the community published numerous papers on addressing the challenges of making IP Multicast practical. Unfortunately, satisfactory solutions for several problems were never developed and, as a result, IP Multicast was never widely deployed. By the late 90's, it certainly seemed that multicast was doomed to become a footnote in Internet history.

Starting around 2000, a number of groups began developing systems (including the predecessor of the ESM system described in the paper) for supporting multicast efficiently at the application layer. These application-layer overlay systems renewed interest in solving the efficient multicast problem. As a result, we once again saw numerous papers published on the topic of multicast. Unfortunately, as with IP Multicast, we have yet to see little if any use of these designs or results in widely deployed systems.

Most of the effort over the past five or so years has focused on organizing the participating hosts into delivery trees without incurring too much overhead. Some of the key problems that have been explored include ensuring that the delivery tree is efficient and that the system tolerates the arrival/departure of clients. An important problem that has been largely ignored and is critical in real deployments is that the wide range of client capabilities. The core contribution of this paper is a design that nicely combines several previously proposed ideas, such as taxation, multiple delivery trees and layered codecs, into a single system that accommodates client heterogeneity. The design also incorporates some careful engineering to ensure stability, reasonable startup behavior, effective resource monitoring, and fair sharing of excess resources. The system design seems simple and straightforward which I view as an asset in a topic area that easily accommodates overly complex designs.

More than anything else, this paper highlights that we are still addressing relatively basic problems in making overlay multicast practical and widely used. The rate of progress in solving or even beginning to address some of these basic problems is certainly disconcerting. One factor that has certainly contributed to this issue is the lack of useable implementations of overlay multicast. Almost every review of this paper commented on how refreshing it was to see a real implementation and reasonably realistic deployment. This was certainly a key factor in the acceptance of this paper.

While I liked this paper overall, it does have a number of weaknesses. First, the design relies on layered coding of the source stream, making it largely only relevant for video data. In today's world of TiVo and video-on-demand, it is unclear to me how critical it is to support live video streaming efficiently. However, there are many other applications that may benefit from overlay multicast – they just aren't well supported by this system. Second, the paper also seems hampered by considering current uses of video on the Internet in other ways. For example, the paper only considers a 400kbps source rate. However, I suspect this commonly used source rate may well be a result of the fact that existing systems don't support heterogeneity well (e.g., layered codecs are rarely used) and that server bandwidth is often constrained. The proposed design certainly eliminates both these issues and I wished the authors had considered

more futuristic scenarios. Third, while the deployment was “reasonably” realistic, it certainly was not real and it was unclear what the PlanetLab part of the deployment got them other than some headaches. For example, the trace playback certainly could have had some hiccups, as evidenced by comments such as “several clients are limited by the bandwidth near them... causing them to under-perform”.

In summary, while this paper makes some important progress towards making overlay multicast practical, we still have a ways to go. An interesting question to ponder is whether overlay multicast will succeed in the end or will we see papers titled “Revisiting Overlay Multicast” in the future.

Planet Scale Software Updates

Christos Gkantsidis T. Karagiannis P. Rodriguez M. Vojnovic

Public Review by Arnaud Legout

In 1974, Leonard Kleinrock defined in the article entitled “Research Areas in Computer Communication” large computer communication networks as networks with thousands of nodes. Today, the Internet enables communications among several hundreds of millions of machines. This incredible scalability is the result of the Internet protocols design. However, this scalability has a cost: it is today impossible to get a global view of the Internet. As a consequence, it is increasingly hard to understand the current protocols dynamics and to design new ones that have good performance on real networks.

A fundamental issue is that, due to its complexity, it is not possible to model or simulate the Internet. It is not possible either to perform experimentations on realistic scenarios before a protocol is deployed, because it is hard to foresee the behavior of users and the interaction with real background traffic. For these reasons, one needs to make assumptions and simplifications on the evaluated scenarios.

To perform relevant simplifications, a good understanding of specific metrics (relevant to the considered protocol) at the scale of the Internet is important. But, as previously noted, it is not possible to get a global view of the Internet. However, it is still possible, but hard, to get a view of a large subset on specific metrics.

The article “Planet Scale Software Updates” characterizes a software update system at the Internet scale. The authors have analyzed 4 traces from “Windows Update”, which is probably the largest software update system in the world.

In a first part, the authors evaluate specific properties of the updates. In particular, they study the correlation of the requests among updates. They show that updates are clustered, i.e., most of the users request updates per group. The authors also characterize the users requesting updates. In particular, they show that: most of the users request the updates within the first day after the update is made available, only 20% of the machines are always online, and that most of the users have all the updates when they use an automated update system.

A first look at the paper may lead to the following conclusion: the presented results are not surprising considering the design of the windows update system. However, “not surprising results” should not be underestimated. When one looks at a system with hundred of millions of users, there is no trivial conclusion. The value of this paper comes from the unique data sets considered that come from an Internet scale real update system, one trace consisting of queries of 300 million different computers. This strength was acknowledged by all the reviewers along with the relevance to study a global update system.

A frustration shared by several reviewers is that much more could have been done with this data set. Whereas it is true that one would have expected more exciting results considering the unique data sets evaluated, the presented results are important (even if not surprising) and will probably be the basis for further studies. The presented results are consistent with the focus of the paper and are relevant. One solution to avoid such a frustration would be to make the data sets publicly available. Reproducibility is fundamental, and availability of the data improves the confidence in the results. However, there are often legal and technical issues when one makes publicly available real data sets. The authors do not plan to make the data sets publicly available due to legal issues. They are working on anonymized traces, but they also face legal and technical issues.

In a second part, the authors evaluate how to improve the update system using caches or peer-to-peer delivery. The authors show that both full cache deployment and peer-to-peer delivery can dramatically reduce the load on the server. They also show that peer-to-peer locality can significantly reduce inter-ISP load. Whereas the presented results are interesting and promising, they are very preliminary. Indeed, practical issues are not discussed at all. However, such issues can significantly impact the efficiency of an update system based on caches or peer-to-peer delivery. For instance, which level of security can be achieved using a peer-to-peer system? Can a peer infer the current security patch missing on a particular computer? How to enforce peers to reciprocate? What will be the willingness of peers to reciprocate? What is the impact of the limited number of connections (outdegree) a peer can open?

In conclusion, despite several weaknesses, this paper will undoubtedly have impact, because the community needs such studies based on the evaluation of a real “planet scale” system.

Drafting Behind Akamai (Travelocity-Based Detouring)

A. Su D. Choffnes A. Kuzmanovic F. E. Bustamante

Public Review by Z. Morley Mao

Due to policy constraints and lack of load-sensitive routing, today's IP routing often cannot satisfy performance and robustness requirements of real-time applications such as Voice over IP and many other important applications such as financial transactions. To deal with path inflation and degraded performance of existing IP paths, overlay routing has been proposed in an attempt to bypass bottlenecks in real time. Extensive performance monitoring across potential overlay nodes is required to effectively identify preferred overlay paths with high probability of overcoming the performance bottlenecks affecting an existing IP path. Although existing work showed that randomly picking an intermediate node to perform one-hop source routing yields high success rate, to ensure the selection of the overlay node provides high guarantees of good performance, network monitoring often requiring active probing appears inevitable. With increasingly larger scales of overlay networks, such performance monitoring becomes more challenging.

The main contribution of this paper is to propose a novel use of existing Content Distribution Networks (CDNs) such as Akamai by identifying low-latency Internet paths suggested by the CDNs for the purpose of improving overlay routing. Certainly other applications can also make use of discovered low-latency path with no additional active probing. The paper focuses specifically on Akamai given its large size and more importantly its open redirection mechanism easily queried for the purpose of discovering "nearby" servers from a given client's perspective. Although the redirection algorithm is proprietary, the authors using extensive measurements showed that latency is a primary metric and thus can be taken advantage of for overlay node selection. The novel insight of making use of an existing system's collected performance information for other applications requiring similar information in an non-intrusive and light-weight fashion is refreshing. As the Internet keeps growing with more applications relying on dynamic adaptation based on discovered network performance very likely obtained through active probing, it is important to identify ways to more scalably identify dynamic network behavior. Using information already available from an existing application without requiring cooperation from the application is one such approach to reduce probing overhead.

Besides this novel insight, the paper however has several shortcomings which may offer potential avenues for future work. The authors did a thorough job in the measurement analysis for verifying that Akamai's server redirections strongly correlate with network conditions measured by latency of the path between clients and servers. However, when it comes to the evaluation to support using discovered preferred servers for the purpose of one-hop source overlay routing, there are still a number of questions unanswered. The measurement performed is quite limited and the obtained performance improvement is based on comparing within the limited set of returned Akamai servers rather than "all" Akamai edge servers. Arguably, this part of the work requires additional investigation with a larger set of measurement data to better understand how information obtained from Akamai can be used effectively for applications such as overlay routing and the potential limitations. For example, it is useful to quantify under what type scenario information from Akamai-based can be beneficial. For smaller overlay networks or networks that do not share many network locations with Akamai servers, the proposed system appears less applicable. One limitation that has been under0emphasized is the potential implication of the proposed system if it becomes wildly successful, i.e., a large number of overlay networks start to latch onto Akamai. The discovered paths may no longer be optimal if too many

users start to shift their traffic to them. This may in turn result in more frequent probing of the Akamai DNS system and unnecessary overhead.

Finally, this work encourages us to think of other novel ways to make use information collected by existing systems for our own applications. If such systems attempt to hide such information, it may be useful to understand how we can create sufficient incentives to share these information. For certain systems such as Akamai, it appears that it may not be easy to completely deter the proposed work in this paper, as long as its redirection mechanism is public. Instead of inferring properties from existing systems, it also behooves us to think about designing common infrastructures such as “network weather service” to allow sharing of commonly used network performance related information.

A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance

Feng Wang Zhuoqing Morley Mao Jia Wang Lixin Gao Randy Bush

Public Review by Dina Katabi

For those who are not BGP experts and want to learn about BGP badness and why it occurs, this paper is a good starting point. The paper highlights the negative impact of BGP's dynamics on the data plane. Recent BGP work has focused on control plane pathologies, reported the long duration of BGP convergence, and discussed the protocol's instability. In contrast, this paper looks at what happens to data packets as BGP tries to converge to new routes. It notes that data packets suffer loss, delay, and reordering. Even when a new route becomes available –which should be a good thing– BGP causes packet losses. Similar results have been reported in the work of Labovitz, but it is interesting to see that after six years, BGP still suffers from the same problems.

The main contribution of the paper, however, stems from its explanation that the interaction of routing policies, iBGP, and the MRAI timer may lead to transient disconnectivity. When a route changes, the policies governing its advertisement may change. For example, the old route to the destination may be through a customer while the new route is through a provider. In this case, the domain does not want to advertise the new route to its peers. Though the peers may eventually learn an alternative route to the destination, they can see intermittent losses between the withdrawal of the old route and the advertisement of the alternative one. A long MRAI timeout may exacerbate the situation because it adds an extra delay before the alternate route can be advertised. iBGP can cause some routers to experience intermittent disconnectivity to the destination when other routers in the same domain have operational routes. Operators and BGP experts can probably predict the existence of these pathological scenarios. The paper however is the first to report these scenarios to the larger research community and support its argument with careful measurements.

The exact numbers in the results must be taken with a grain of salt however, because all of the results in the paper are for a single Beacon prefix. How representative is this Beacon prefix, which has two tier-1 providers? The answer is unclear. Thus, the causes reported in the paper for packet loss caused by BGP updates are neither the only ones nor necessarily the most representative. For example, other scenarios in which iBGP plays no role may exist. A study with more destination prefixes is needed to discover all factors that cause packet-loss during BGP's convergence and quantify their contributions.

This paper is worth reading. Don't let the plethora of numbers and figures turn you away. They don't add much beyond illustrating the fact that BGP dynamics can create intermittent bursts of packet loss. The interesting content is in sections 4.3, 4.4, 4.5, and 5.3, where the paper explains the root cause of the observed BGP pathologies. Be sure to get to the second half of the paper where most of the action happens.