

# From the Reviews

---

What follows is a lightly edited selection of interesting, supportive, and contrary tidbits from the program committee's reviews of the papers selected for HotNets V. The first, italicized paragraph summarizes the paper. The editing has conflated comments made by different program committee members, so "I" may refer to a melded PC hive mind rather than an individual. Of course, reviews reference the *submitted* versions of the accepted papers. The authors have addressed some, but not all, of our comments in their final copies; it's interesting to see which comments led to revisions. We hope you enjoy this look behind the curtain.

## (R)EVOLUTIONARY BOOTSTRAPPING OF A GLOBAL PKI FOR SECURING BGP

---

*Develops a scheme by which ISPs can develop a global PKI for authenticating routing updates in an incremental fashion. The key trick is based on ISPs first issuing self-signed certificates which other ISPs honor based on earliest-seen-cert. Thus, an attacker can undermine an ISP's certificate only by issuing a (misleading) cert prior to the ISP issuing a cert. This both gives ISPs an incentive to issue certs and requires attackers to act quickly as the rollout proceeds.*

You might face a problem in convincing the global players to actually agree on a grassroots certificate authority. If the routers can work with multiple roots of trust, why would ISPs pressure the emerging roots to combine into a single root? I understand that less trusted roots will want to group under a trusted single root, but the most popular/trusted CA roots might have limited willingness to do so. Still, even the global players are likely to prefer a grassroots certificate to no certificate. Indeed it is a "nice" feature that you can force this by finding someone evil. It's a nice gimmick that evil attacks can be resolved by increasing the security level!

The approach depends on the premise that attackers will be ultimately detected, which is worrisome, particularly if attackers confine their activities to a limited window of time. But even though "first cert is legit" is brittle, particularly as attackers become aware of it, it also has the property of mapping to how things are often deployed (i.e., legit use predominates initially).

Do prefixes get hijacked intentionally, or inadvertently, via misconfiguration? If the latter, the security of private keys would seem less of an issue.

## DON'T SECURE ROUTING PROTOCOLS, SECURE DATA DELIVERY

---

*Argues that connectivity (or availability) is a much simpler problem for routing to solve than path integrity (i.e., that the connectivity is achieved via the correct set of routers), since endpoints can verify connectivity. Therefore, routers should*

*be focused on providing as many connectivity options to a destination as possible ("Availability Centric Routing", or ACR) rather than trying to achieve an optimal path or trying to ensure that malicious actors cannot manipulate the path. A combination of ISP-assisted source routing, end host-based measurements of path characteristics, and cryptography could secure the current Internet against route hijackings without the deployment of any flavor of secure BGP (or other changes to routers) as long as the tier-1 ISPs are trustworthy.*

An interesting thought experiment that directly frames my own wondering about secure routing.

ACR introduces high costs for end systems. There's the overhead of locating the route; shouldn't that be the task of the ISP's? It seems a bit strange that you seem to imply that it is easier for security-unconscious end users to secure *all* end systems, than it would be to secure a major part of the network infrastructure that is actually administered by folks that should know at least the basics of security.

If we had secure routing then we could be more lax about end-to-end integrity checks; the paper should acknowledge that this is something significant we give up (though admittedly relying on routing to give us address-based authentication is brittle).

This paper makes perfect sense and it is totally contrary to conventional wisdom (and on top of it all it has a reasonable path to adoption). Looks like the perfect HotNets paper!

On the other hand, regarding conventional wisdom, the paper could more explicitly bring out the connection between the premise and the end-to-end principle.

## A TECHNICAL APPROACH TO NET NEUTRALITY

---

*Proposes a technical solution to the network neutrality problem. ISPs favoring net neutrality would deploy in their networks neutralizers that would have the role of obscuring from discriminating ISPs the addresses of content providers that send traffic through discriminating ISPs.*

Creative, topical, and thought provoking, but I can't fig-

ure out if it is hasty or thoughtful. Either way, it opens up an interesting line of research.

The central assumption that some ISPs will actively participate in your proposed scheme seems a little suspect—it’s one thing for an ISP to speak out in favor of network neutrality but quite another for it to actively assist its users in circumventing the techniques put in place by other ISPs (some of which it might have business relationships with!). Seems like this might justify ISPs retaliating with attacks—some would say defenses—aimed at the neutralizer mechanisms.

While the approach is different than that for anonymizers, it’s similar enough to them, and to VPNs. Basically, it’s pretty much the solution one pictures arriving at given the formulation of the problem, with the mildly innovative aspects being the use of statelessness and then leveraging this to use anycast. I don’t buy that the ISP can no longer discriminate. It can if it knows who the neutralizer’s customers are and is okay with discriminating against all of them en masse. Note that this means it’s difficult for anyone to run a neutralizer except on a large scale with a lot of customers.

Presumably only at-risk traffic uses the neutralizer service. In that case, why wouldn’t just discriminating against all neutralized traffic be fairly effective? If market forces could indeed be relied on to discourage ISPs from such discrimination, then wouldn’t network neutrality be less of an issue?

You take a narrow view of net neutrality and discriminatory behavior, in which an ISP wants to disfavor one particular content provider in the hope of extorting money from it. There are other goals and discriminatory behaviors that you don’t try to prevent. For instance, a discriminatory broadband ISP may favor one content provider, maybe because it pays extra money; in a world where this content provider has very few competitors, this has the same effect as disfavoring the remaining, which is something that you were trying to avoid. Other sophisticated “attacks” against your proposal are also possible, such as examining the nature of the traffic generated by an application. For instance, it is probably trivial to identify VoIP traffic if I wanted to discriminate against it. Trying to prevent such attacks, for instance by normalizing traffic of all applications, would significantly increase the cost of the solution (and may hurt some applications). This is a form of packet discrimination that you do not consider.

Hoping that consumers are in a strong position to make changes may be true in the long run, but that is the long run.

Wouldn’t regulation, when it finally comes about, trump any technical solution?

## AN AXIOMATIC BASIS FOR COMMUNICATION

*Argues that the field would benefit from a formal framework by which to formulate and reason about networking tasks. It therefore develops a framework for formalizing network communication similar to that of denotational semantics for*

*formalizing program execution. The framework includes an interesting set of definitions and primitives that capture the operation of a forwarding element. These are then composed to reason about activity along a multi-hop path.*

The high-level goal here is great. The value is in considering that there might be an apt, formal set of principles, although this particular approach might not have the right set of axioms. Very flawed, but provocative and in an interesting direction.

The formulation focuses mainly on simple message delivery. Of networking tasks, this seems like the most easily amenable to this form of analysis: routing is a standard algorithms concept with well-defined state variables, and one can reason about overall correctness in terms of the consistency of just this state. That is, one can reason about deliverability in terms of just routing entries that effectively abstract away all the computation—BGP, policy, intra-domain, etc.—that went into computing that state. Many networking tasks—congestion control, reliability, security, latency, etc.—seem to lack this property. How useful will this approach/formulation be for such tasks? And given that the very simple NAT example is > 10 lines, how complicated will this get as we try to express even slightly more realistic tasks (e.g., fetching a web page that involves client, proxies, routers, L7 switches, load balancers, DNS resolvers/servers, web caches, server clusters, etc.)? The PODC/DISC/theory community offers many analysis tools and approaches. We don’t seem to use them in part (I suspect) because things get too hard when we try to capture the zillion dependencies and interactions in real systems.

The network is (approximately) the one object we consider in “CS” which is not assumed to be fail stop. Additionally, it has phase delays (though we worry about that in large scale hardware designs): state changes take place, if at all, at different times and in different orders.

I like where I think this is trying to go but can’t understand the paper.

## PROTOCOL DESIGN BEYOND GRAPH-BASED MODELS

*Wireless protocols frequently model inter-node communication (or interference) by means of graphs—an edge is placed between two nodes that can directly communicate (or interfere). Graph-based models are widely acknowledged to be wrong, but everyone uses them anyway. This paper adds urgency to the search for better models by showing through measurements and protocol designs that non-graph-based protocols can perform much better than graph-based models in practice.*

The call for improved modeling of wireless communication is a very useful one and the empirical results are eye-opening. However, it feels a bit like one camp of theoreticians talking to another camp of theoreticians. Systems have moved beyond graph-based models already; per-

haps the work is a bit late. Nevertheless, people *do* continue to use graph-based models, for example in sensor networks, and theoreticians are people too, and the paper is admirably clearly written—and decorated with useful experiments! Furthermore, the suggested protocols differ somewhat from the prior work I know of. Presenting this work could make a difference.

It could be greatly enhanced if the authors proposed a protocol based on their understanding of wireless networks through physical models.

The paper implies that most protocols are *designed* to fit a graph model, whereas I believe that it is far more common to see protocols being *evaluated* (not designed) using graph models. Are there examples of in-use protocols that globally schedule packet transmissions based on graph-based models? Graph-based models are incredibly simple to use for evaluation purposes. It would be good to include some discussion on what parameter-setting complexity is required in the SINR-based models. Sounds like there is: the relation between power and distance, the value of alpha, the interference at the receiver, noise at the receiver and beta?

Though the proofs about capacity are interesting, the result is underwhelming: the distinction between  $<$  and  $\leq$ .

## SOME IMPLICATIONS OF LOW POWER WIRELESS TO IP NETWORKING

---

*Describes a study of the performance of a personal-area-network link layer, namely 802.15.4, and considers the resulting implications for IPv6 routing.*

This paper is well-written, readable, and presents the problems in much more interesting depth than most of the other papers in its class. Reading it, one gets the impression that there are interesting research problems in this space, and that the techniques used by systems like Roofnet are not immediately applicable to this domain. The authors also give some good insights into the tradeoffs in code complexity, network reliability, and power/memory consumption in this environment, which is a much different set of tradeoffs than the wired world. Overall, it's a good eye-opener, and it has the data to support its claims.

The IPv6 connection seems tenuous. Many of the Implications apply to technologies other than IPv6. The two directions of an IP route may need to differ, but even if 802.15.4 nodes have IP addresses, will they route via IP routing? Probably not. And asymmetric routing is common in IP networks anyway.

The nugget I got out of the paper was about acknowledgement losses causing problems for fragmentation reassembly and the ETX metric. Is the lack of intermediate links in your testbed mostly due to its being indoors and anchored? Do you have any evidence that the link technology would cause different observations here? Do you have an 802.11 testbed for comparison?

## NETWORK SYSTEM CHALLENGES IN SELECTIVE SHARING AND VERIFICATION FOR PERSONAL, SOCIAL, AND URBAN-SCALE SENSING APPLICATIONS

---

*Articulates privacy and accuracy concerns that should be addressed by future applications that rely on sharing sensor data in personal, social and urban settings. A distinguishing feature of these applications is that the sensing devices, ranging from tiny motes to expensive video cameras, are owned and operated by individuals.*

The application space the authors highlight is one that has been receiving plenty of attention, and deployment, in certain communities (HCI/ubiquitous computing, wireless), and we're probably overdue for a networking paper on the topic. While the paper offers little new information, it does a nice job in pulling the discussion together and identifying the challenges, and could serve as a useful starting point for a discussion on the network implications of personal and urban computing.

While the straw-man architecture is useful, it's a bit of letdown in that it's mostly a formalization of how these applications are built today—i.e., a sensor network relays data to a server through a few proxies and clients query the server, again through a bunch of proxies. This proxy-centered architecture is traditional and somewhat limited; scenarios of a large number of mobile, autonomous, and yet related sensors (e.g., camera cellphones in the same city block, building) are at least as important and probably more challenging. Much of the novelty in your architecture seems to be in getting the selective sharing and context verification right, but it isn't clear how much of this is a networking issue, and the paper doesn't explore this in much depth other than telling us where in the infrastructure such functionality would be implemented. It would be nice to see a straw-man of data naming schemes, query language and pub-sub interfaces that might support some sample sharing and verification policies.

Are there organization/provider boundaries that need to be respected? Is this easily done with a DNS-like infrastructure that assumes an administrative hierarchy? What is the business relationship between client/sensors and their mediators, are there charging/accounting requirements? You say mediators are like firewalls in various ways, but if I am behind a firewall, I know who runs the firewall, and can hire and fire that person. I don't get the sense that mediators have that level of administrative "closeness" to their users.

"Citizen" is an odd word, but I have no alternative.

## RETHINKING WIRELESS IN THE DEVELOPING WORLD

---

*Describes efforts to connect rural communities in Africa and India via long-range wireless links.*

I enjoyed reading this paper. While folks have been working in this area for the past few years, this is the first time I have seen anything close to a research agenda laid out, and the discussion about density and WiLD (wireless long distance) vs. mesh is also very interesting.

Remote upgrade and management seems key to running these networks at a low cost, but your story on that front seems pretty weak so far. If there are existing satellite and GSM connections that you can use, why bother deploying WiLD? Presumably, the deployment cost for such networks has already been paid by someone, and just using them (more or less) shouldn't be that expensive.

High-gain directional antennae have not taken off because of the management problems they bring about in the face getting disoriented (e.g., during high wind conditions). Do you face this problem? Electronically steerable antennae, which you also propose for fault tolerance, can mitigate this. How expensive are such antennae?

The authors argue for long-range wireless (instead of cell, satellite, etc.) based mainly on the lower cost of deploying commodity wireless equipment, then argue that WiFi is a better choice than WiMax due to the latter's cost. This is an interesting argument and I would like to see it better explored: it's a bit surprising that a technology designed for connecting users in the same room via a base station is a better way to connect users over a wide geographic area than a technology designed to, well, connect users over a wide geographic area. How much does the spectrum necessary to run WiMax cost in Ghana? How would WiMax or microwave relay perform at 50 km? Also, cost is repeatedly cited as an overriding design decision; it would be nice to know how low the budget actually is. The project isn't well-funded enough to upgrade a 256 MHz CPU?

The proposed changes to the 802.11 MAC aren't very carefully examined or justified; data from the real deployment or even a mention of how well the real deployment actually performs would be helpful. How many of these problems are the result of choosing WiFi over other technologies? Does WiMax's MAC address many of the MAC issues described here, such as stop-and-wait?

## SERVICE PORTABILITY

*Use HTTP redirection to find what you really want! The prototype Permafind system enables service portability, e.g., for email, blogs, web pages, and so forth, using the standard mechanisms of redirection, indirection, relaying, and proxying. It is intended as an incremental solution that is immediately deployable.*

This paper amusingly describes Permafind as "technically boring", and yes, it doesn't hold any huge technical innovations. How can you not like a paper that says right in the abstract that it has no technical innovation? However, the pragmatic combination of simple mechanisms to solve a pressing problem is a major strength. It is refreshing to en-

counter a well-thought-out solution that seems to actually achieve its own goals. This may actually end up being deployed on a large scale, although perhaps not in the form the authors envisage, because it is simple, useful and dare I say it, obvious. Although I don't like the solution in the long term, in the short term, and to inform the long term, it's nice, simple, and cute.

It seems to me that Permafind is itself a provider, so in solving the problem of provider changes, you add a new one.

I didn't see much of a discussion of problems with this approach, such as how to handle bookmarks, or the problem with search engines giving the ephemeral pathnames instead of those findable through the redirection service.

## THE END OF INTERNET ARCHITECTURE

*The main argument is that the basic notion of considering possible network architectures for a future Internet is detrimental, as it presupposes that there should be a distinction between nodes that use the network vs. nodes that facilitate the network's operation. That is, the idea of a "network architecture" imposes an artificial distinction between "networking" and "distributed systems" that needn't be fundamental and erroneously presupposes the nature of how a future global networking infrastructure should be structured. In summary, network architecture is pernicious and should be stamped out.*

The author's emphatic decrying of the damage done by imposing a separation between networking and distributed systems doesn't resonate for me. I don't see that separation as particularly manifest, and to the degree in which I do see it (as evidenced in the network systems papers that appear in SOSP vs. SIGCOMM, say) I don't see it as rooted in the notion of network architecture so much as in how networked systems emerged from multiple parent disciplines (operating systems, digital communication). Furthermore, to me, the term "network architecture" means "a set of networked communication abstractions and the relationships between them", whereas I eventually gleaned that for the author it more specifically means the layered, hosts-vs.-routers structure that *today's* Internet manifests. Surely it's clear that a future architecture needn't impose that style of structure, although it *will* need a set of abstractions along with (coherent) relationships between them.

My model of systems, whether architectures or large software systems, is that once you deploy it (assuming it gets used a lot) it starts to accrete barnacles. After some period of time, you re-do the system from scratch (hopefully), incorporating as many of the barnacles as seems right into the new design (so now they become layers, or compartments, or functions, ...), and deploy it. At which point, it starts to accrete barnacles.

Suppose we do as you propose: what is the guarantee that the result of this organic growth will be a desirable state?

Sometimes a paper I agree with makes me question my own beliefs. This is one such paper. For the most part, I agree that network architecture is another name for software, and it's nice to see that point expressed. But in practice there is one important difference. Network architecture specifies packet headers. These well-specified headers, and the behavior behind them, add to the ways in which the network may be programmed. Only a tiny fraction of the libraries in existence (C? C++? MFC? POSIX?) are specified anywhere nearly as compactly and effectively as the Internet protocols. Those clean specifications of *data*—what's on the wire—have helped us add new functionality without breaking existing networks. Thinking about headers is different enough from thinking about software that it might deserve a different name. Not maybe what conventionally people mean by network architecture, but still.

While I think this paper will provoke discussion, I don't think it'll be particularly illuminating discussion. It'll be more like a discussion between parties who spend most of the time arguing until they figure out that they mean different things by the same terms.

The paper feels as if you think it's shocking, but in practice I think most researchers don't care that much about network architecture.

Cut Figure 1. It's so bad it transcends bad.\*

## DECONGESTION CONTROL

*The paper turns our viewpoint on managing congestion upside down: rather than design mechanisms that worry about picking a conservative transmission rate to avoid packet drops, the authors propose that hosts simply transmit greedily at a high rate and routers instead drop packets fairly across active flows. To tolerate drops, the authors propose that endhosts erasure code their data streams.*

An intriguing idea, and definitely novel, but the authors could have gone further in trying to convince us that it might actually be sensible.

The impact of dead packets sounds like a possible show-stopper. If it turns out that congestion is not just limited to the edges of the network (inter-continental links?), then should we just give up on the idea of decongestion control? Is there some fix one could come with, or if not, how bad might the wastage be?

You say that “end-host congestion control is typically suboptimal and, critically, relies on the goodwill of end hosts for success.” Well, any practical control protocol for something as complex as the Internet is going to be “sub-optimal”, at least according to some metric. Second, is this problem with nice/evil hosts a problem in *practice*, or just in theory?

The TCP receive window doesn't *ensure* that the sending rate doesn't exceed the receiver's ability to consume data,

it just makes it very very likely. Secondly, since Van Jacobson's 1988 paper, it hasn't been clear that we actually need window updates. Why do we “privilege” end hosts over routers (who have not so much control over the load dumped on them)?

What exactly is the difference between current TCP and the proposed protocol? Eventually, there are two proclaimed differences: no retransmissions of lost packets due to the use of erasure codes, and novel greedy congestion control. But you have suggested a schema where the caravan size and type can be adopted. How does this differ from adopting the cwnd? Isn't it possible to game the proposed system by ignoring the feedback? Isn't it possible for an aggressive sender to just send with a huge redundancy and thus succeed?

Is this more or less incrementally deployable than (say) XCP? And given current router capabilities, is the per-flow state due to fair queuing really an issue?

## A SIMPLE APPROACH TO DNS DOS MITIGATION

*A simple approach to improving the client-perceived reliability of the DNS system: DNS servers are allowed to aggressively cache records past the records' TTLs in a “stale cache.” A record in the stale cache can be used only when the authoritative servers for the record's domain are not available; the stale cache allows queries to be answered despite server failure.*

This paper is great: a simple solution that appears to do the trick. I agree that the new DNS architectures are likely an overkill, given that DNS seems to work the vast majority of the time. Your arguments about your approach making the right trade-offs are convincing. Your easily deployable change to DNS resolvers that can make a real difference in some cases. The weakness is that the proposal is so simple and obvious (in hindsight) that it will probably not generate much discussion.

A true DNS DoS defense would allow the DNS server to serve updated information about the domain(s) it is authoritative for. Something along the lines of “mitigating the effects of DoS attacks on DNS servers” might be more fair.

I don't like not knowing when (any) state in network will be purged. I used to work for a CEO who said he would sign (almost) any contract, as long as he could get out of it in a finite period of time.

I suspect this will be effective in practice, but it is hard to say because of the underlying Zipf popularity distributions. On the positive side: the resolvers are likely to have in their stale caches information about popular zones; the less popular zones may be missing but they are also less likely to be attacked (?). On the negative side: if servers higher up in the hierarchy, e.g., the root servers, are attacked, many unpopular zones under them will be unresolvable because they will not be present in the stale cache; the heavy tail may imply

\*In the final copy, Figure 1 remains.

that a significant fraction of queries are for zones that have not been resolved in the past.

Does your scheme interact poorly with diagnosing local DNS-related problems? Consider a situation where I can reach my local resolver which cannot reach other servers due to some problems. Today, I will discover this quickly because I'll not be able to resolve most domains. But with your approach, I will continue to resolve (to stale information) many of the names without realizing that something is amiss.

## SPACE: SECURE PROTOCOL FOR ADDRESS BOOK BASED CONNECTION ESTABLISHMENT

---

*A conceptually simple but nice insight regarding a way to leverage easily-discovered evidence of likely trust alignment, coupled with modest use of an associated, out-of-band secure channel. In the realization of this insight, two users who want their PDAs to associate first engage in a protocol that allows them to verify whether the other user is already in the address book stored in their PDA (i.e., telephone numbers, perhaps street addresses, etc.). If so, then they use the contact information already in the address book to perform a key exchange, for example by sending a text message to the cell phone number in the address book. This second step means that an attacker who lies about their identity won't be able to complete the protocol unless they have also compromised someone's out-of-band access.*

The paper includes an extensive security and privacy analysis, but doesn't particularly explore the issues of (1) latency in establishing an association, (2) how often will one want to establish an association with someone not in one's address book, (3) the requirement that address books have a canonical form to enable correct matching.

I understand the general problem that this work is trying to solve, but by restricting it to the set of people who have each other in their address books, it makes the problem either a lot simpler or trivial, depending on how you look at it. In considering the problem, it also makes sense to consider the alternatives. I guess one could install the necessary software on both ends to allow this kind of key exchange. One could also just include public key information in vcards and include them in address books.

The described protocol is unnecessarily complicated considering the bottom line: it's only as secure as sending an SMS message. For example, what does the first phase (where contact information is exchanged) provide besides the phone number we'll use to send the SMS containing a public key? Are there simpler possibilities that are equally secure? Since Alice and Bob exchanged contact info at some point in the past, they could have exchanged a public key; or, if users aren't good about maintaining keys, why doesn't Bob just SMS his public key to Alice immediately? If the key comes from a phone number in Alice's phone book she'll associate that key with the entry. Neither SPACE

nor these straw-man protocols are very satisfying considering that the security "bottoms out" in SMS. In contrast, existing Bluetooth authentication techniques, though currently cumbersome, have a refreshing basis in physical reality. For example, when inputting a PIN to bind my Bluetooth keyboard to my desktop the chain of trust is short and ends with me trusting my OS and display, not an unknown telecom network's SMS system. A system that was more automatic, but with the same easy to grasp security guarantees, would be more compelling.

One attack not discussed is theft of a PDA which is also the user's cell phone, enabling the attacker to complete the out-of-band part of the protocol.

The idea is fairly modest, but I liked the basic insight.

## EXPLOITING SOCIAL NETWORKS FOR INTERNET SEARCH

---

*Presents a search engine that indexes locally cached content. This lets it take advantage of local context and annotate Google results with locally relevant results. Essentially, users of a social group help enhance Google's web search results by sharing each others' indexed browsing history.*

Does a good job of showing a new direction where systems researchers can make progress on the challenging and important problem of better web search. I found the paper non-obvious, quite intriguing, and promising in terms of the possible performance gains.

You might think a bit about  $k$ -anonymity. Hits in Google don't tell me much; hits in a local cache tell me, for example, that a male colleague is looking to date women in San Diego. How to allow a user to share useful information, without exposing themselves to leaking personal or private information, is a challenge.

Don't we all know how to refine our search by adding terms to find the relevant page?

## FREE RIDING IN BITTORRENT IS CHEAP

---

*Discusses a way to build a free-riding BitTorrent client, and shows by actual experiments that this client can maintain healthy downloads without uploading anything or uploading faulty items. Since there is a general impression that BitTorrent is good at hindering free-riding and is robust to attacks, the paper shows an interesting contradiction.*

You show that having many open TCP connections does not harm performance, as commonly believed in the BitTorrent community. Maybe you could delve a little bit more into why that is the case, given the fact that BitTorrent has chosen to maintain only a few open connections, supposedly for better performance. You also show that, even when downloading only from leechers, the performance of a selfish peer is still acceptable, a remarkable result.

The idea that such free-riding attacks can be used by corporations for fighting uncontrollable distribution of copy-

righted material is indeed a new one, and deserves more attention.

It comes as no surprise that the BitTorrent protocol can be gamed. Not only have selfish attacks been demonstrated before, but intuitively one would expect the protocol to be vulnerable. The incentives built in to BitTorrent are not ironclad, they are not designed to be. If a freeloader downloads slower than a non-attacker, who would want to be a freeloader? BitTorrent incentives provide “economic” *encouragement* to upload. They are not meant to *strictly require* uploading; there is no security consequence if a freeloader can download without uploading. If you could download *faster* without uploading, that would be a big deal, and might cause BitTorrent to fail eventually, but it seems from your results that no one would freeload in practice, yes? If this isn’t true, then why not? So it’s not clear whether “lack of punishment . . . raises concerns about the future of peer-to-peer file sharing”.

That a “sharing community” is so much easier and more advantageous to attack is intuitively obvious, but only after reading the paper.

### CAPTURING COMPLEXITY IN NETWORKED SYSTEMS DESIGN: THE CASE FOR IMPROVED METRICS

---

*Proposes a quantitative measure of the complexity of network algorithms that would capture the intuitive qualities of “cleanness” and “elegance”. In summary, the paper attempts to quantify networked system design taste.*

I found this proposal highly stimulating. I started out thinking that the problem was obviously intractable, but came away with a sense that the idea really could go somewhere. This was the most unexpected paper in my set, yet it has the feeling “why hasn’t anyone done this before?”. It seems both intuitive and the correct approach, and tackles a problem that is both very important and has received little rigorous attention so far.

How might metrics be used to capture notions like “robustness”? Can they illuminate the tradeoffs of hard vs. soft state (briefly alluded to at the end), or benefits from adhering to the end-to-end principle?

Thinking about complexity is important and interesting. However, I’m not sure this paper doesn’t just substitute subjective aesthetic judgments about metric formulae and parameter values for current subjective aesthetic judgments about protocols themselves.

There is one deep way in which the intuition codified in the paper differs hugely from my own, which is that “complexity decreases as the network offers more delivery options”. I guess *your* intuition is, if there’s only one path, that path must be kept up to date; if there are two, either of them can fail. Or maybe your intuition is based on flooding, which is simple. Although flooding should have low complexity, intermediate choices between single-hop and flood-

ing seem to have *more* complexity than the extremes. For example, maintaining a RON network or a DHT is quite complex, more so than BitTorrent for example. I think your metric seriously underestimates the implementation complexity of DHTs, and allows protocols to “cheat” to lower their complexity. Consider a protocol that wants to transfer a local value from  $x$  over a long path with  $d$  transport dependencies. The complexity of this is  $d$ . If instead of generating one local value at  $x$ , the protocol generates  $m$  local values, all of which can be used to compute the desired state at the remote node, the complexity of getting the state would go down to  $d/m$ . This feels wrong. What if you were to split “any” state derivations into categories? For example, you could distinguish “any of  $m$  *carefully chosen* inputs” from “any of  $m$  *arbitrary* inputs”.

I didn’t understand why a wireless node that blindly broadcasts doesn’t count for transport dependency; if it fails, the system still fails, right?

### DISCOVERING DEPENDENCIES FOR NETWORK MANAGEMENT

---

*Knowledge of the inter-dependencies in a distributed system (between hosts, applications, in-network boxes, etc.) would be useful for diagnostics, managements and so forth but is also hard to figure out. The paper proposes to apply machine-learning techniques to measured network traffic to discover this graph of dependencies.*

It’s nice to see some attempt to model the underlying communication relationships explicitly and causally, with the goal of using it to find problems. I realize that similar systems have been tried with more invasiveness, but this one seems to be happy with just network traffic, so I think that’s a good step forward. It would be useful if the authors gave some examples of the kinds of problems that they found, particularly the ones that were more transient, and which would be harder to find with other debugging approaches.

This is a neat idea, and it will be very interesting to see how it works out, but is it possible that a discussion on this topic is most useful when accompanied by comprehensive evaluation and results?

Can you dig deeper in discussing what types of dependencies the proposed approach can infer? For example, it doesn’t seem like this could extract dependencies that aren’t necessarily linked in time or by protocol: e.g., you download a piece of software from a (bad) server, and that software later initiates a (on-the-wire) seemingly unrelated exchange.

Is it possible that the proposed approach might just make management/diagnostics more complex? Imagine the plight of a sysadmin trying to figure out why his diagnostic system raised an alarm when the alarm is the consequence of a complex machine learning algorithm with seemingly obscure thresholds and parameters all over the place.

I believe Bob Braden has mentioned that Jon Postel wor-

ried that the Internet had gotten (or would get) to the point where it couldn't be rebooted because of circular dependencies.

### FLEXLAB: A REALISTIC, CONTROLLED, AND FRIENDLY ENVIRONMENT FOR EVALUATING NETWORKED SYSTEMS

---

*Proposes a hybrid emulation/live-deployment scheme for assessing network systems in a controlled-but-realistic fashion. The core idea is to associate with emulated nodes with actual live nodes as counterparts, and incorporate the network conditions experienced by the live node back into the emulation for greater fidelity.*

The problem of finding a good platform for network experimentation is an important, and, importantly, open one. Having used (and abused) PlanetLab quite a bit I couldn't agree more that it is overloaded to the point of uselessness. This paper takes a step towards addressing the problem of determining the characteristics of the emulated links: it's hard to argue that link characteristics derived from measurements being made in real time of real links with (near) identical application load won't be accurate. As long as the correct metrics are being measured, that is.

The main benefit of the system proposed in this paper would seem to be psychological for the experimenters, a feeling of "currency", of "running on the real Internet", etc.

Far too much weight is put on the system being "realistic". Emulation cannot in general gather results that are "statistically significant" for the Internet. We simply don't understand how micro-properties like the delays between two adjacent packets affect higher-level measurable properties. You cannot possibly remove "any artifacts that might be introduced by special measurement traffic", and trying to emulate all of the "fascinating" details of the Internet seems impossible. For instance, I never would have thought to model a gateway that drops all fragmented IP packets silently. Or to connect a node in China to only a few nodes in the US.

There's a fundamental issue regarding *lag* in sending reports of the network dynamics a live node experiences back to the emulation. Since the live nodes can be deployed at distant locations, this latency will impose fundamental limits on the sort of network dynamics that the emulation environment can incorporate. How serious are those limits? What sort of constraints do they place on the nature of high-fidelity emulation? What classes of studies can the hybrid support, and what classes are beyond it?

Using maximum one-way delay as an approximation of bottleneck queue size seems quite risky. Often, extreme outliers are due to surprising causes/pathologies. And why is it reasonable to assume that congestion mostly happens along the forward direction of a path? Surely this is only more likely if the load induced by the flow itself is likely to tip the scales in terms of congestion.

It's a complicated system. If the goal is to provide nodes with PlanetLab-like link characteristics but lightly-loaded CPUs (à la Emulab), couldn't we just put more machines at the ends of the existing PlanetLab links? If this system were in use and there were 10 times as many Emulab nodes as PlanetLab nodes, wouldn't that be equivalent to having an extra 10 machines at each PlanetLab institution (as long as playback features were still available)? After all, this system doesn't reduce load on PlanetLab links, it only changes CPU utilization.

### INTERCONNECTION DISCRIMINATION: A TWO-SIDED MARKETS PERSPECTIVE

---

*Describes ISP price setting as a two-sided market: the content provider market on one side and the eyeball market on the other. The authors use this two-sided-market model to show that it is rational behavior for ISPs to subsidize one market at the expense of the other in an attempt to increase their overall profit. The paper also explains through two-sided-market arguments why ISPs are subsidizing residential access at the expense of the (presumably more competitive) content provider market, which has multihoming. Or alternately, it explores how a particular branch of economic theory, Industrial Organization, might inform engineering design for the Internet.*

This is an interesting paper and I believe the networking community could benefit from hearing your view of the pricing strategies of ISPs. My main complaint is that you do not clearly explain the assumptions about the cost model for the ISPs. The most intense discussions may be about the assumptions you base your arguments on, not about the internal logic of your argument.

The intent of this paper seems to be to convince the networking research community that economic considerations are as important as technological ones when designing protocols or architectures for the Internet. In this, the paper fails. The claims that this theory should impact Internet research are never justified—the ideas seem more relevant to engineers employed by ISPs—and the paper isn't terribly accessible to the non-expert. It's a current and interesting topic, especially given the ongoing network neutrality debate, but not appropriate material for HotNets.

### ACHIEVING GOOD END-TO-END SERVICE USING BILL-PAY

---

*Presents a mechanism to provide good end-to-end service between arbitrary endpoints by adding billing information to individual packets. Each ISP is supposed to retain parts of the so-called "nanopayments" associated with a packet: The better its service is, the more "nanodollars" it should receive. The paper argues that such an approach provides better end-to-end service quality, helps to defend against network floods, and discourages spam.*

I'm not entirely convinced but it's certainly creative and thought-provoking.

I would think the *granularity* of Bill-Pay, which is per packet, would be wrong, and one would want granularity at the level of an e-mail message. Also, if you did this, isn't there a danger of confusing the email nanopayment with the nanopayment for the transport (TCP/IP)?

I am skeptical that it will fly. Better end-to-end service can probably be bought using such a mechanism, but my concern is that users will be unwilling to use Bill-Pay since it expects them to pay extra money without knowing in advance what kind of performance improvement they are likely to experience. It's only after they have sent some (non-trivial) amount of traffic along a path that they discover what kind of performance they get. This may mean that not many users will use nanopayments, which means that the investment into Bill-Pay infrastructure will go to waste.

The digital cartographer and secretary are central, but is a digital cartographer feasible? It appears that there are just too many paths to keep track of. For each destination of interest, there will be an exponential (in the number of ISPs along the path) number of paths. Furthermore, is a digital secretary feasible? The general problem of detecting malicious activity from normal user behavior is very hard. Even in the limited setting of DoS attacks and spam, people have been talking about it for a while but nothing convincing has materialized. Similarly, your suggestions on preventing man-in-the-middle attacks significantly drives up the cost of deployment.

Some concerns: The question of route stability is a central one but isn't addressed at all here. Didn't people give up on load-based routing because of stability problems? What is the impact of strategic behavior by ISPs? For example, a large ISP that is willing to operate at a loss for some period of time in order to put some smaller competitor out of business? How do you secure the user and ISP OADs from bad receivers and ISPs. What happens if the leftover nanopayment is insufficient to cover the remainder of the path? And the proposed DoS defense seems to imply that attackers can make legitimate clients pay more to access the server. This doesn't seem like a good idea at all. How can you force email senders to include a payment with their messages? How do you introduce the approach? How does it work for UDP? With TCP you can do "piggybacking", but with UDP, is extra signaling required?

## FIGHTING COORDINATED ATTACKERS WITH CROSS-ORGANIZATIONAL INFORMATION SHARING

*Outlines the design of a system to allow a small number of sophisticated network monitors ("detectives") to make use of observations made by large numbers of other machines ("witnesses"). The network monitors use the observations of witness machines to aid the discovery of bad actors in*

*the network (e.g., a bot net). The query mechanism ensures that private information isn't revealed to witnesses, and that witness replies are believable, via a combination of hashing and encryption.*

This paper is well written and describes an interesting vision. The high-level concept sounds great: it's an excellent idea to draw on observations from multiple places taken with "simple and generic traffic monitoring devices", and the scheme for sharing information seems very cunning.

The architecture of the described system is clear, but its potential benefits are only alluded to. Testing the ability of witnesses to aid the detection of bots via control traffic would be a great addition.

Another deterrent for the detectives who would consider "fishing" for private data at the witnesses is post facto auditing. If the logs at the witness show that a detective was engaging in impermissible fishing, that detective might be excluded from the system. As it is probably hard to get in, this would be a serious disincentive. Relying more on this disincentive could allow more query flexibility.

This is nice work that will most certainly move forward the efforts to put together a network-wide defense against many classes of computer hijacking techniques. The biggest problem I have with this paper is that the entire solution was pretty predictable, and the problem statement itself had nothing surprising either.

The paper leaves a lot of questions unanswered. Witnesses "log the facts", but what does this actually entail? How long are records kept for? And with how much detail? If a single witness can reveal "a wealth" of information about which hosts have downloaded the code, then witnesses are expected to be in the network, i.e., routers. If witnesses might be highly resource-constrained then it's even more important to think about the storage and processing costs of being part of this architecture. How do detectives locate witnesses? Does a witness somehow advertise itself? Since witnesses must run the software to answer queries from detectives, there's an upfront commitment to participating, so presumably this information could be stored centrally. Does every detective need to know about every witness? How much coverage of the Internet by witnesses would be required for the system to be effective? Are there timeliness constraints on queries (surely there need to be)? Also are witnesses aware of the identity of detectives? Is the encryption of returned tuples primarily intended to hide the extra records produced by collisions from the detective? or to hide the information from third-party observers? or to verify the witness's statement (in which case a cryptographic MAC could have been used instead of Kerberos-style abuse of encryption)? Maybe it's all three? More explicit mechanisms would help disambiguate this question (e.g. using MAC and encryption would make it clear that both privacy and verification were desired).

So full of holes it will probably generate plenty of discussion.

## BLACK BOX ANOMALY DETECTION: IS IT UTOPIAN?

---

*Proposes an anomaly detection framework for network data that separates the anomaly detection module from the data manipulation module. Transformations can be applied to a wide variety of network measurement data until they are brought into the form of real valued constant-spaced time series, which can then be fed to the anomaly detector.*

The ability of the proposed framework to detect a very wide range of network anomalies based on very different sources of network data using the same “black box” anomaly detection procedure is a real strength. The paper is well written, timely and shows clear concepts. There are currently no surprising or outraging novelties, but the approach is important for the research community and has a lot of potential. I’d like to see this being used on real data to see where the limitations are and how much can be added if you combine this approach with some domain knowledge. Still the fact that this is in principle domain-knowledge-agnostic is a very strong aspect of this approach.

What kinds of anomalies are most promisingly detectable? We note a recent trend towards stealthy attacks, such as Shrew and RoQ attacks. The discussion should be steered towards the need of new definitions of what “anomalies” are. Flash crowds are anomalies as well, but not malicious.

The framework, as is sadly typical for things that claim to be “frameworks”, attempts to encompass any possible implementation. This is a bad property: if everything is allowed, nothing is defined. Much of the framework is pretty obvious as well. This paper is short on specifics.

To what extent do you assume time synchronization? Do you really need “a hierarchy of assumptions”? What would this mean? Sounds complex.

## GLAVLIT: PREVENTING EXFILTRATION AT WIRE SPEED

---

*How do you ensure no private/protected data leaves your network? The Glavlit approach centers on a whitelist specifying what content may leave the network. This whitelisting*

*is done out-of-band; a verification box sits at the boundary of the network and checks that data leaving the network is on the whitelist. While fairly straightforward in the abstract, things get tricky when dealing with dynamic content or preventing covert channels. The paper discusses these issues and proposes seemingly workable solutions.*

An interesting and perhaps increasingly important problem, but one gets the feeling that the proposed system is going to get very hairy quite quickly (dynamic content, unusual but legitimate protocol use, etc.). Is there something protocol designers could be doing differently to make this an easier problem?

Seems a heavyweight mechanism, and not that interesting.

This paper acknowledges that preventing exfiltration is impossible and actually gives an example of how it can be done while evading Glavlit detection. The system can slow down the information leakage to maybe thousands of times, but probably not millions of times, below the link speed. But the system is geared towards high-speed networks, so even if the spy needs to increase the amount of traffic by a factor of 100,000, it takes only 100 MB to leak a 1000-byte confidential memo in minutes on a fast-Ethernet link (even if there is significant competing traffic). The possibility of the spies discovering more efficient exfiltration channels is also menacing to the proposed system.

I don’t understand the motivation behind the solution. You are basically saying that you have an internal network with a mix of public and private content, and you only want the public content to leave the network. Why wouldn’t you do something simpler? Such as make sure that the web server contains only public content, and only the web server is allowed to send data outside; or trust the web server itself to deliver only the public content (just like you trust the warden)? You wouldn’t need to worry about covert channels, which, as you say, can only be mitigated (not eliminated). Did you consider and reject such simpler options for certain reasons? Could you elaborate on those?

Your discussion of protocol channel mitigation is unclear. A more clear of description of threats, mitigation techniques and their effectiveness would be helpful. Tell us how to think about the threat model.