Lecture Outline

- Alternatives to IPv4 addressing architecture: security implications
- “Tussles” in architectures that affect multiple stakeholders
- Ethane: the good and the could-have-been-better
- Review of Diffie-Hellman key exchange
- Starting to look at Authentication

IPv4 Addressing Architecture

- High-level architecture of IPv4 addresses?
- Abstraction: addresses are both *locators* and *identifiers*
  - *Locators*: bits are topologically relevant
    - Includes: multicast, broadcast, private networks
  - *Identifiers*: addresses used to identify connection endpoints
    - Have global meaning
- Naming: addresses are associated with NICs rather than end systems or people

IPv4 Addressing: Mechanisms

- Addresses are represented with 32 bits
  - Limited room available for topological structure
  - Possible (today) to *fully enumerate*
  - Limited supply ⇒ *architectural stress* (NATs)
- Bit patterns have topological significance
  - Original design: class A/B/C networks
  - Current design: CIDR
- Packets carry source addresses
  - Which are set by sending system

IPv4 Addressing: Implications

- Addresses are locators
  - Routers can function without per-connection state
- Addresses are identifiers
  - Easy for end systems to associate incoming packets with existing connections/state
  - Mobility is tricky
  - Dynamic addresses are tricky
  - Migrating a connection from one system to another is (very) tricky
- End systems set source address ⇒ *spoofing*

Who Needs Source Addresses?

- Idea #1: build up return route in packet as it's forwarded
  - Each router adds its "address" to a list in header
  - "Address" might just be interface tag
    - E.g. return route of "I3, I6, I3, I2, I9" means "first router sends out on its Interface 3, then receiving router forwards to its Interface 6, then *that* receiving router forward to its Interface 3, …"
- Properties?
  - Spoofing requires infrastructure compromise
  - But: less flexibility for return paths
  - And: more header space requires

Who Needs Source Addresses?

- Idea #2: address is “routable public key”
- All messages are signed by source
- All messages are encrypted for destination
- Strengths?
  - No more spoofing!
  - No more sniffing!
  - Wide-ranging portability
  - Plenty of addresses (no need for DHCP)

Who Needs Source Addresses?

- Weaknesses?
  - Messes up prefix-based routing (HUGE)
  - Large addresses ⇒ spatial overhead
  - Generating addresses tricky for devices with little entropy available

- Suppose we can solve these issues: would the architecture be viable in practice?

Tussle

- "All messages are encrypted" ⇒ *tussle* between end users and site security monitors

- Architecture pre-supposes policy (e.g., 100% network privacy) because it shapes what is *expressible*

Tussles: Scope

- Tussles exist *across domains*
- Different *stakeholders* ⇒ different interests
  - Each vies for their own concerns (~ *adversarial*)
- Examples of stakeholders?
  - Commercial ISPs
  - Enterprise operators
  - Government (enforce laws; protect consumers; regulate commerce; restrict information)
  - Content providers
  - Intellectual property rights holders
  - Individual users

Architecting for Tussle: Avoid Overloading

- IP addresses having topological significance
  ⇒ difficult for sites to renumber
  ⇒ adds friction for sites to switch providers
  ⇒ architecture inadvertently undermines competition between ISPs

- Alternative?
  - Have locators distinct from identifiers

Tussles: Avoid Overloading, con't

- DNS provides both names-independent-of-location *and* human-visible branding
  ⇒ leads to land grabs / typo-squatting

- Alternative?
  - Opaque identifiers *plus* separate "directory service" for users to find sites
  - Today, in practice this latter is search engines

Tussles: Implications

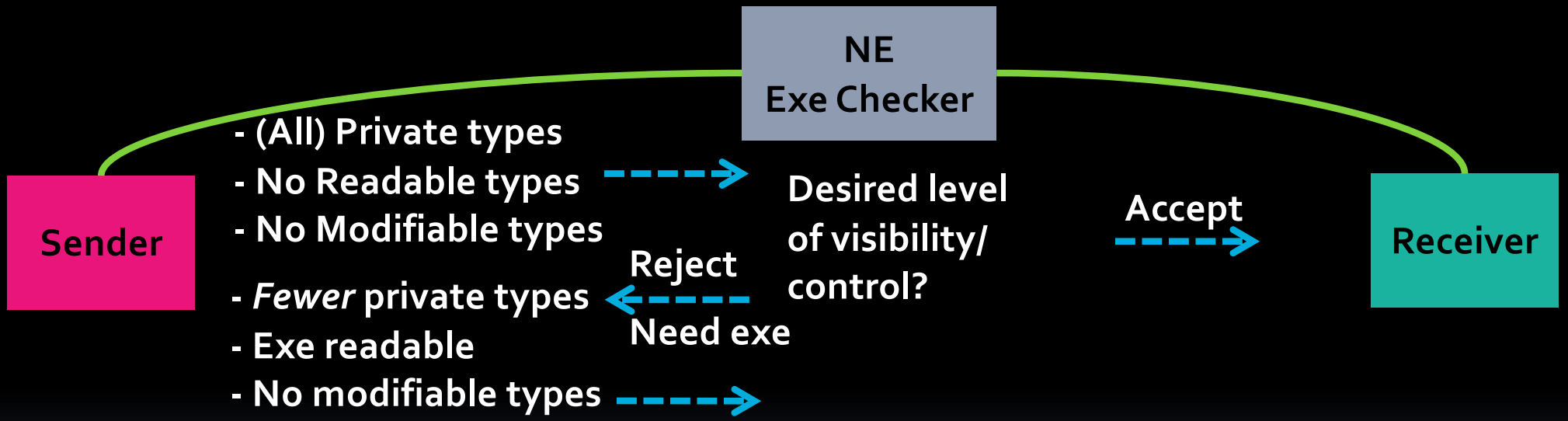
- For architecture, can design to presume tussle resolution (e.g., “communication is always encrypted”) …
  - Either **works great** or **fails hugely**
- *Or:* provide *choice* at tussle “boundaries”
- Choice requires *visibility* into the different opportunities
  - For our IPv4 alternative addressing example: maybe decouple *encryption* key from *routing* key
- Note: game theory can provide insights

Architecting for Tussle

- Today's middleboxes impose a *narrow dialog* between end systems and the network
  - Often, middleboxes are "invisible" to end systems
  - Often, a middlebox can only make a **best effort guess** as to nature of end-system activity
- Alternative architecture:
  - End systems describe high-level nature of traffic
  - Middleboxes signal whether acceptable or not
  - End systems choose alternative path depending on importance of maintaining privacy/integrity
- Consider architecting for this using **typing**

Dialog

Typing paves way for *dialog* to negotiate communication properties

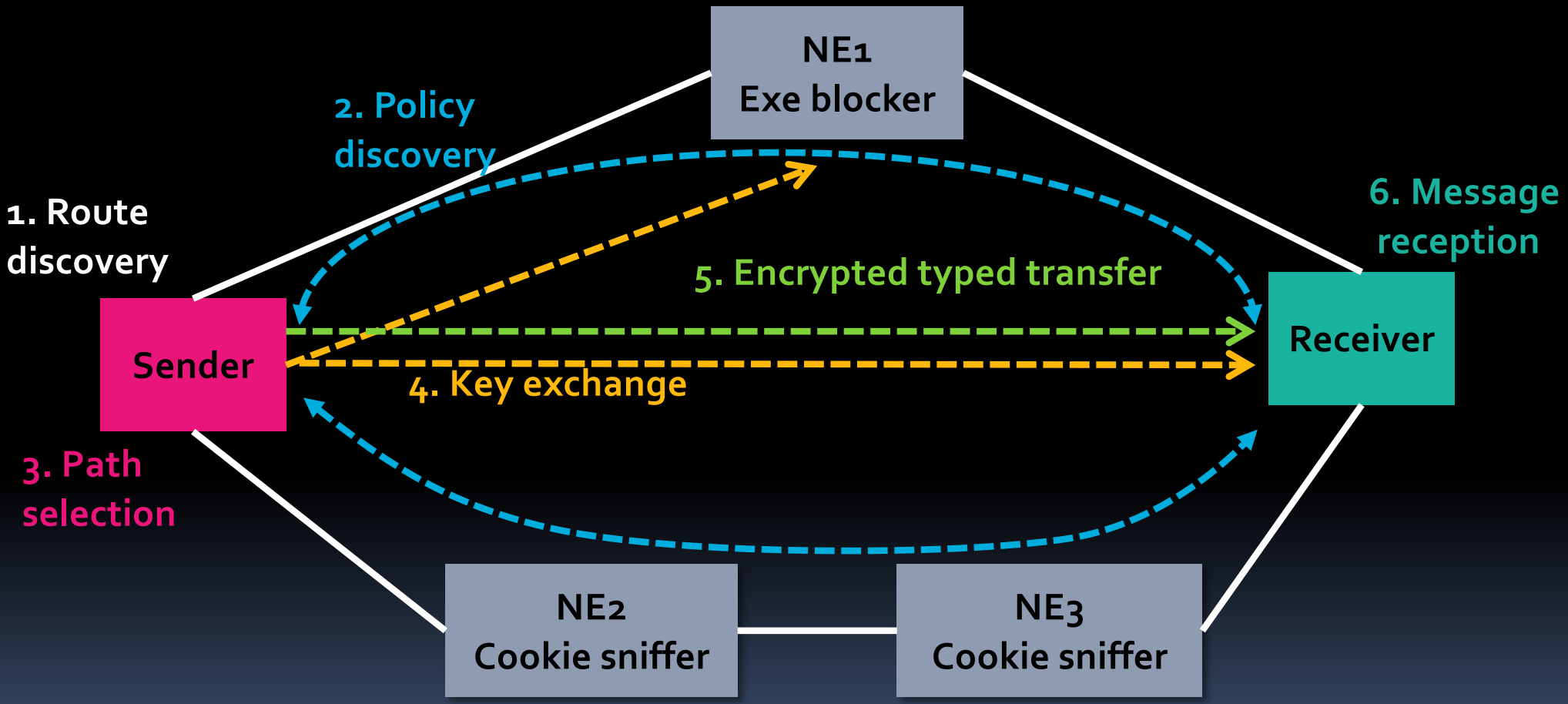


Pre-connection or in-band

Sender may choose an alternate path.
Fail if no such path → *reason in full view*

Network has upper hand, but visibility limits collateral damage

Progression of Communication



Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

ABSTRACT

This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows. While radical, this design is backwards-compatible with existing hosts and switches.

We have implemented Ethane in both hardware and software, supporting both wired and wireless hosts. Our operational Ethane network has supported over 300 hosts for the past four months in in Stanford University's network, and this deployment experience has significantly affected Ethane's design.

downtime in multi-vendor networks comes from human-error and that 80% of IT budgets is spent on maintenance and operations [16].

There have been many attempts to make networks more manageable and more secure. One approach introduces proprietary middleboxes that can exert their control effectively only if placed at network choke-points. If traffic accidentally flows (or is maliciously diverted) around the middlebox, the network is no longer managed nor secure [25]. Another approach is to add functionality to existing networks—to provide tools for diagnosis, to offer controls for VLANs, access-control lists, and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to allow auditing. This can be done by adding a new layer of protocols, scripts, and applications [1, 10] that help automate configuration

Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
ty

Scott Shenker
U.C. Berkeley and ICSI

Mechanism
separate from policy

ABSTRACT

This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows. While radical, this design is backwards-compatible with existing hosts and switches.

We have implemented Ethane in both hardware and software, supporting both wired and wireless hosts. Our operational Ethane network has supported over 300 hosts for the past four months in in Stanford University's network, and this deployment experience has significantly affected Ethane's design.

downtime in multi-vendor networks comes from human-error and that 80% of IT budgets is spent on maintenance and operations [16].

There have been many attempts to make networks more manageable and more secure. One approach introduces proprietary middleboxes that can exert their control effectively only if placed at network choke-points. If traffic accidentally flows (or is maliciously diverted) around the middlebox, the network is no longer managed nor secure [25]. Another approach is to add functionality to existing networks—to provide tools for diagnosis, to offer controls for VLANs, access-control lists, and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to allow auditing. This can be done by adding a new layer of protocols, scripts, and applications [1, 10] that help automate configuration

Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

ABSTRACT

This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows. While radical, this design is backwards-compatible with existing hosts and switches.

We have implemented Ethane in both hardware and software, supporting both wired and wireless hosts. Our operational Ethane network has supported over 300 hosts for the past four months in in Stanford University's network, and this deployment experience has significantly affected Ethane's design.

downtime in multi-vendor networks comes from human-error and that 80% of IT budgets is spent on maintenance and operations [16].

There have been many attempts to make networks more manageable and more secure. One approach introduces proprietary middleboxes that can ex if placed at network choke-point or is maliciously diverted) around t longer managed nor secure [25]. ctionality to existing networks—to provide tools for diagnosis, to offer controls for VLANs, access-control lists, and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to allow auditing. This can be done by adding a new layer of protocols, scripts, and applications [1, 10] that help automate configuration

Ease of management

There are currently approximately 300 hosts on this Ethane network, with an average of 120 hosts active in a 5-minute window. We created a network policy to closely match—and in most cases exceed—the connectivity control already in place. We pieced together the existing policy by looking at the use of VLANs, end-host firewall configurations, NATs and router ACLs. We fou nfiguration files contained rules no long e of the network, in which case they we e policy.

This sort of lack of coherent overall policy is typical in enterprises

There are currently approximately 300 hosts on this Ethane network, with an average of 120 hosts active in a 5-minute window. We created a network policy to closely match—and in most cases exceed—the connectivity control already in place. We pieced together the existing policy by looking at the use of VLANs, end-host firewall configurations, NATs and router ACLs. We found that often the existing configuration files contained rules no longer relevant to the current state of the network, in which case they were not included in the Ethane policy.

… as is having lots of *stale* policy

Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

ABSTRACT

This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then e
ples extremely simple flow-based E
ized controller that manages the a
While radical, this design is back
hosts and switches.

We have implemented Ethane in both hardware and software, supporting both wired and wireless hosts. Our operational Ethane network has supported over 300 hosts for the past four months in in Stanford University's network, and this deployment experience has significantly affected Ethane's design.

Proof-of-principle deployment

downtime in multi-vendor networks comes from human-error and that 80% of IT budgets is spent on maintenance and operations [16].

There have been many attempts to make networks more manage-
able and more secure. One approach introduces proprietary middle-
oxes that can exert their control effectively only if placed at net-
work choke-points. If traffic accidentally flows (or is maliciously
iverted) around the middlebox, the network is no longer managed
or secure [25]. Another approach is to add functionality to ex-
isting networks—to provide tools for diagnosis, to offer controls for VLANs, access-control lists, and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to allow auditing. This can be done by adding a new layer of protocols, scripts, and applications [1, 10] that help automate configuration

Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

ABSTRACT

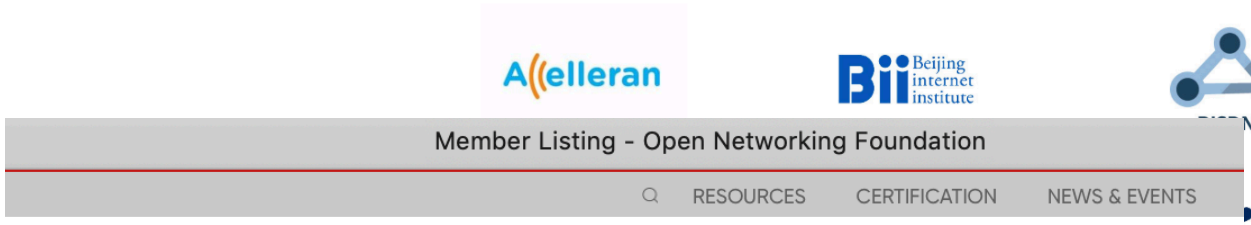
This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows. While radical, this design is backwards-compatible with existing hosts and switches.

We have implemented Ethane in both hardware and software, supporting both wired and wireless hosts. Our operational Ethane network has supported over 300 hosts for the past four months in in Stanford University's network, and this deployment experience has significantly affected Ethane's design.

Viable path forward

in multi-vendor networks comes from human-error and f IT budgets is spent on maintenance and operations [16].

There have been many attempts to make networks more manageable and more secure. One approach introduces proprietary middleboxes that can exert their control effectively only if placed at network choke-points. If traffic accidentally flows (or is maliciously diverted) around the middlebox, the network is no longer managed nor secure [25]. Another approach is to add functionality to existing networks—to provide tools for diagnosis, to offer controls for VLANs, access-control lists, and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to allow auditing. This can be done by adding a new layer of protocols, scripts, and applications [1, 10] that help automate configuration



Collaborating Innovators

Partners

Innovators

Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

ABSTRACT

This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows. While radical, this design is backwards-compatible with existing hosts and switches.

We have implemented Ethane in both hardware and software, supporting both wired and wireless hosts. Our operational Ethane network has supported over 300 hosts for the past four months in in Stanford[illegible] has signific[illegible]

No clear threat model: a “resonance” paper

downtime in multi-vendor networks comes from human-error and that 80% of IT budgets is spent on maintenance and operations [16].

There have been many attempts to make networks more manageable and more secure. One approach introduces proprietary middleboxes that can exert their control effectively only if placed at network choke-points. If traffic accidentally flows (or is maliciously diverted) around the middlebox, the network is no longer managed nor secure [25]. Another approach is to add functionality to existing networks—to provide tools for diagnosis, to offer controls for VLANs, access-control lists, and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to al- adding a new layer of protocols, that help automate configuration

Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

Architectural notions?

ABSTRACT

This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows. While radical, this design is backwards-compatible with existing hosts and switches.

We have implemented Ethane in both hardware and software, supporting both wired and wireless hosts. Our operational Ethane network has supported over 300 hosts for the past four months in in Stanford University's network, and this deployment experience has significantly affected Ethane's design.

downtime in multi-vendor networks comes from human-error and that 80% of IT budgets is spent on maintenance and operations [16].

There have been many attempts to make networks more manageable and more secure. One approach introduces proprietary middleboxes that can exert their control effectively only if placed at network choke-points. If traffic accidentally flows (or is maliciously diverted) around the middlebox, the network is no longer managed nor secure [25]. Another approach is to add functionality to existing networks—to provide tools for diagnosis, to offer controls for VLANs, access-control lists, and filters to isolate users, to instrument the routing and spanning tree algorithms to support better connectivity management, and then to collect packet traces to allow auditing. This can be done by adding a new layer of protocols, scripts, and applications [1, 10] that help automate configuration

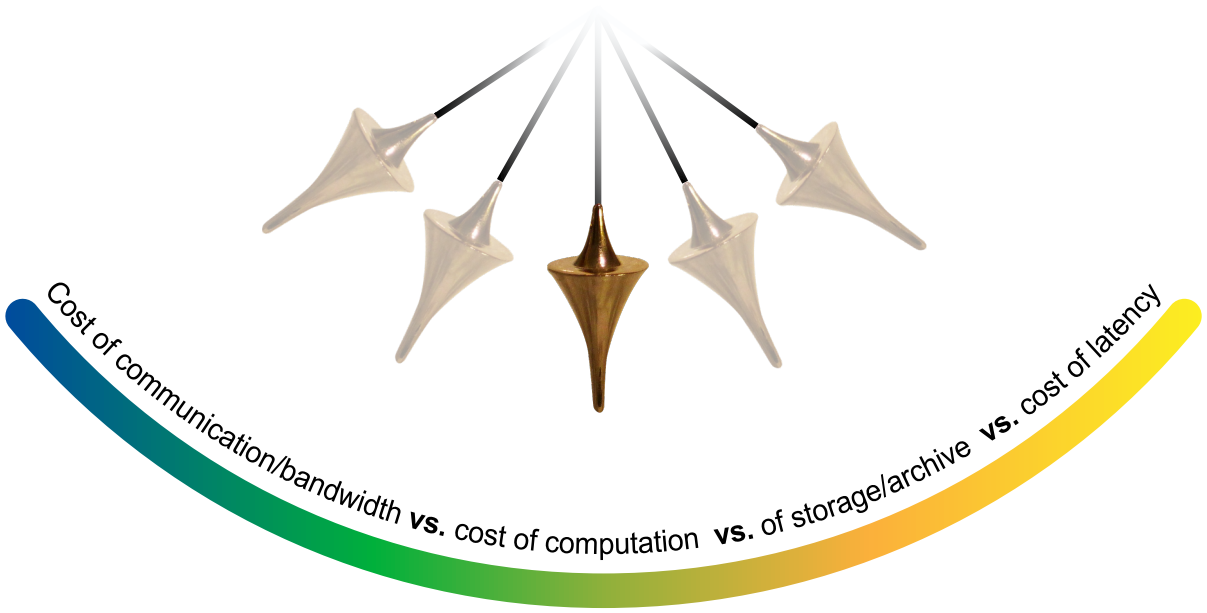
Ethane Architecture

- Changes basic notion of Ethernet forwarding
  - New notion is more complex (switches though are simpler)
- Switches maintain per-flow state
- Strongly enforces **default deny**
- Strongly enforces compliance with policy
- Strong awareness of higher-level identity
  - Can perceive/control user network activity
  - Can reason about policy in high-level terms

Ethane's Scalability Premise?

- *Flows are not exceedingly short*

THE PENDULUM OF SYSTEMS DESIGN



AS THESE CHANGE
IN RELATIVE TERMS,
SO DO ARCHITECTURES

e.g. **mobile handsets**
expensive local computation,
expensive communication ⇒
communication becomes cheaper
⇒ transformative for app design

e.g. **cloud**
IF have sufficient bandwidth
⇒ can leverage cheap remote
computation

What's not to like?

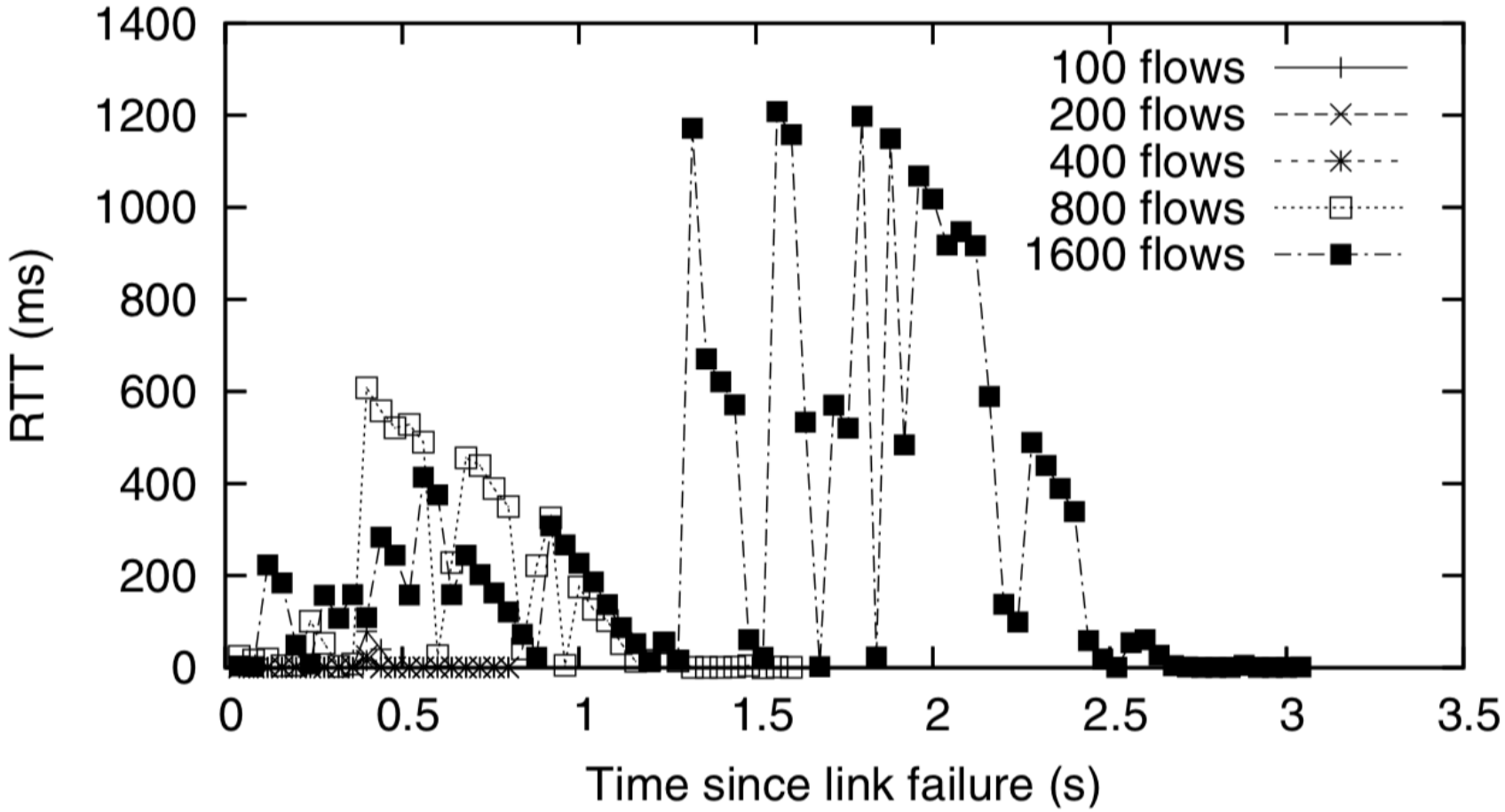


Figure 10: Round-trip latencies experienced by packets through a diamond topology during link failure.

It is also worth noting that the flow table can be several orders-of-magnitude smaller than the forwarding table in an equivalent Ethernet switch. In an Ethernet switch, the table is sized to minimize broadcast traffic: as switches flood during learning, this can swamp links and makes the network less secure.[5] As a result, an Ethernet switch needs to remember all the addresses it's likely to encounter; even small wiring closet switches typically contain a million entries. Ethane Switches, on the other hand, can have much smaller

two-way hashing scheme [9]. A typical commercial enterprise Ethernet switch today holds 1 million Ethernet addresses (6MB, but larger if hashing is used), 1 million IP addresses (4MB of TCAM),

High-end $27K (2009)

Table 1. Scalability Table

Name	WS-SUP720-3B	WS-SUP720-3BXL	VS-S720-10G-3C *	VS-S720-10G-3CXL*
MAC Entries	64,000	64,000	96,000	96,000
Routes	256,000 (IPv4); 128,000 (IPv6)	1,000,000 (IPv4); 500,000 (IPv6)	256,000 (IPv4); 128,000 (IPv6)	1,000,000 (IPv4); 500,000 (IPv6)

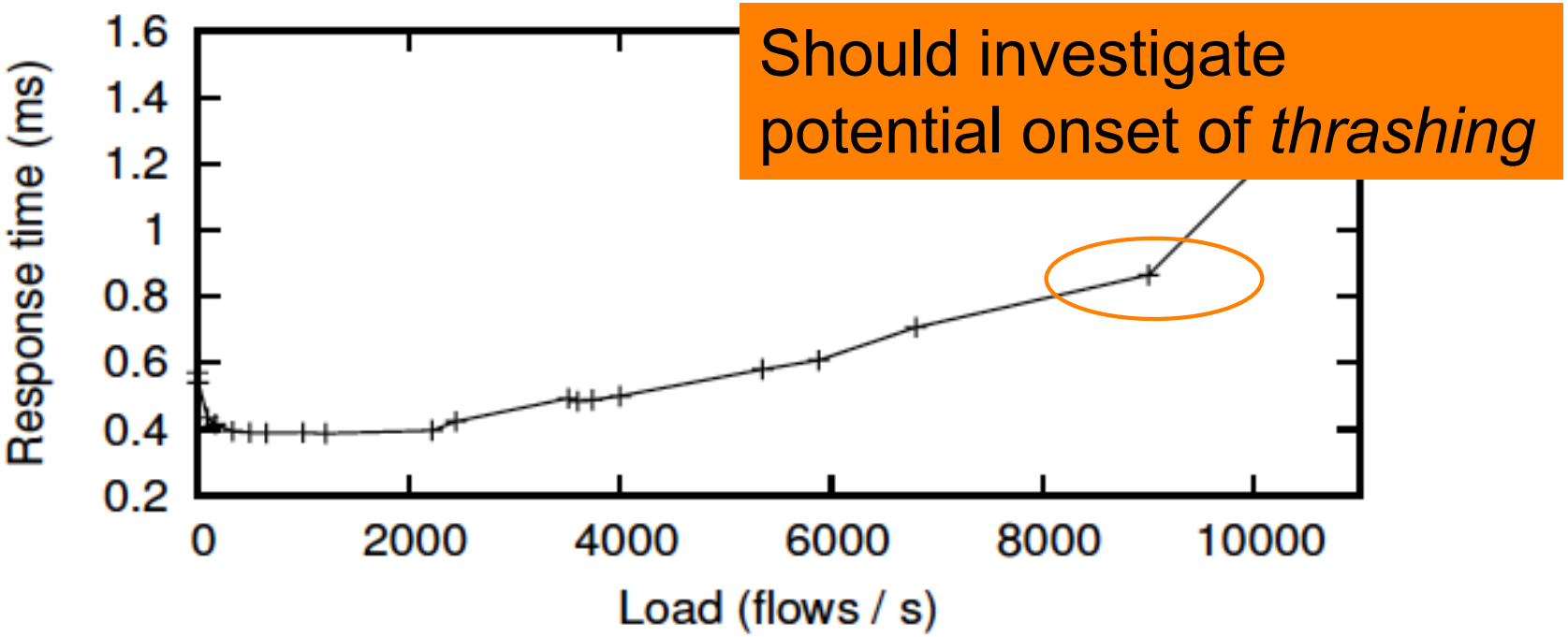


Figure 6: Flow-setup times as a function of Controller load. Packet sizes were 64B, 128B and 256B, evenly distributed.

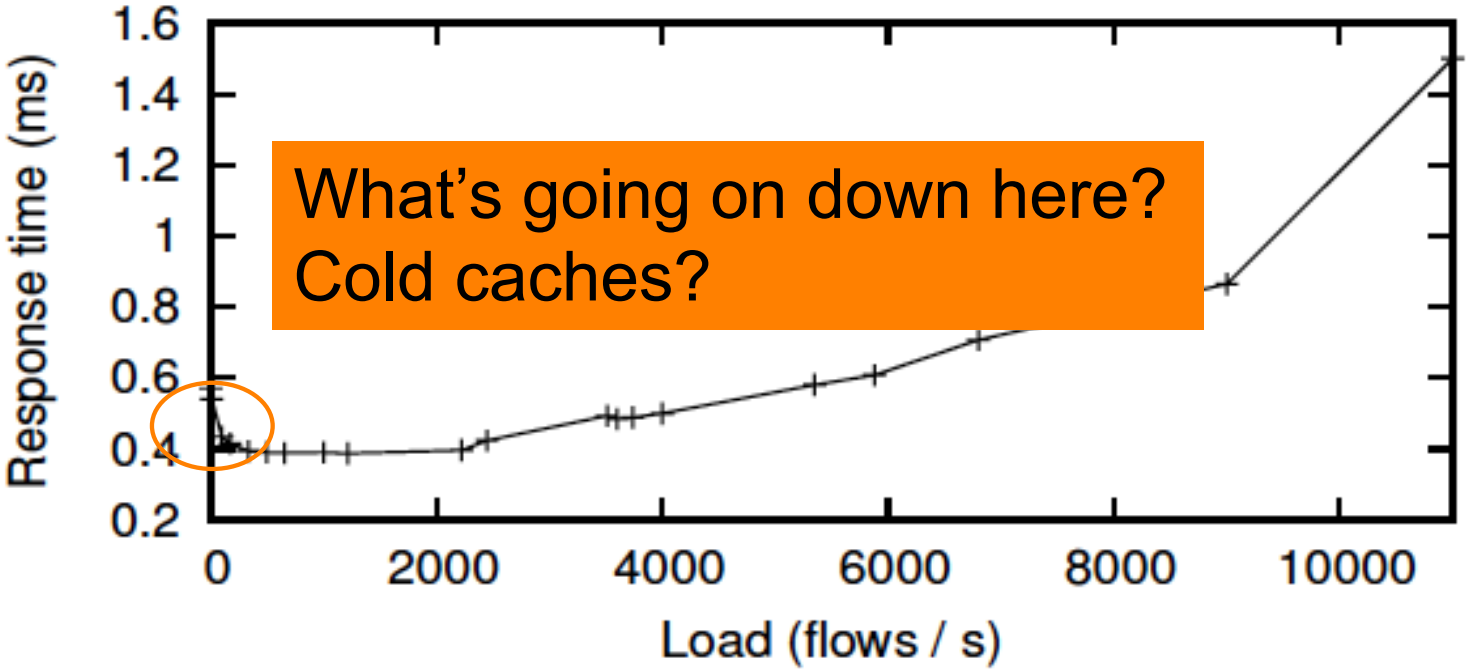


Figure 6: Flow-setup times as a function of Controller load. Packet sizes were 64B, 128B and 256B, evenly distributed.

heads. The Controller was configured with a policy file of 50 rules and 100 registered principles; routes were precalculated and cached. Under these conditions, the system could handle 650,845 bind events per second and 16,972,600 permission checks per second. The

A check takes < 60 nsec??

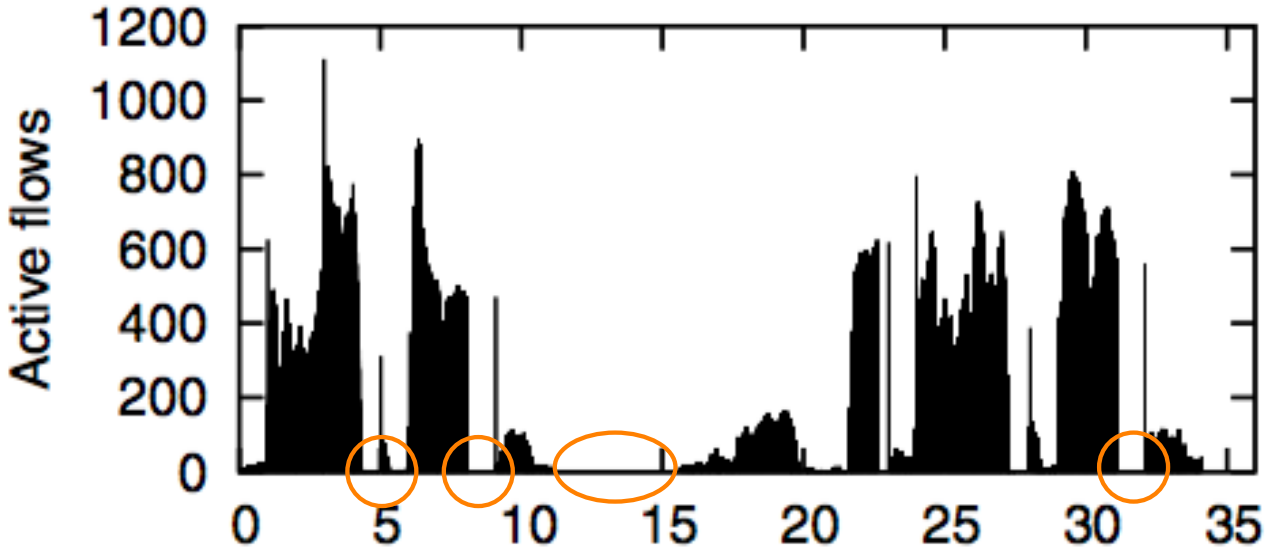


Figure 7: Active

There are *no* flows for *hours* at a site w/ 8,000 hosts?

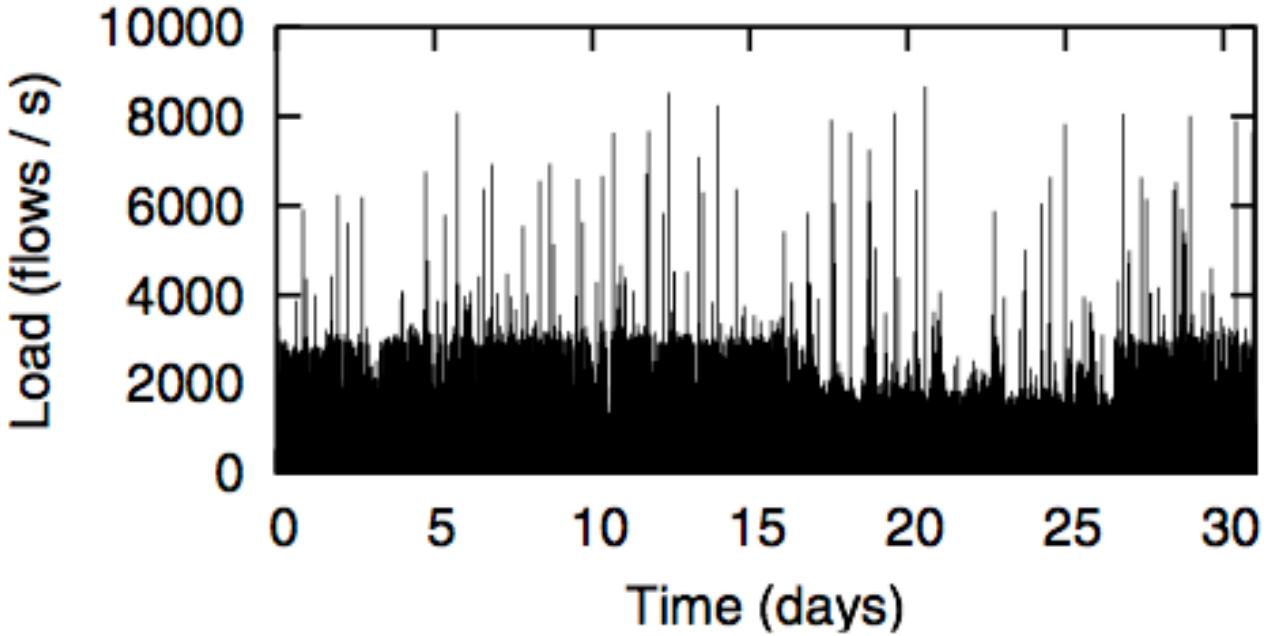


Figure 8: Flow-request rate for Stanford network.

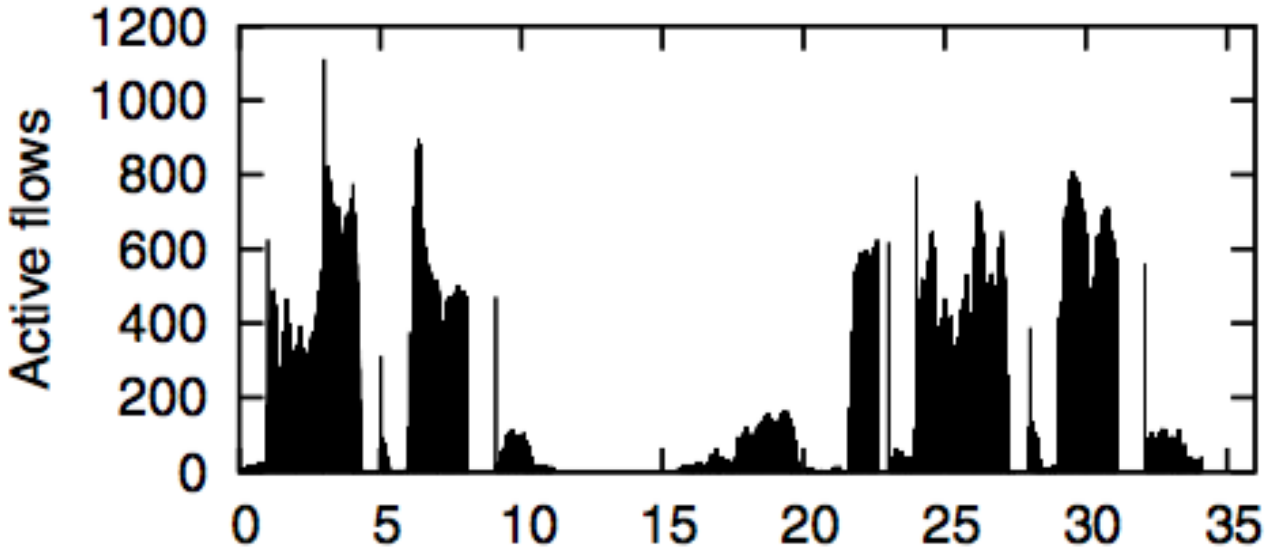


Figure 7: Active

Complete misinterpretation of LBL dataset: it is a series of traces *one-port-at-a-time* for *inter-subnet* traffic

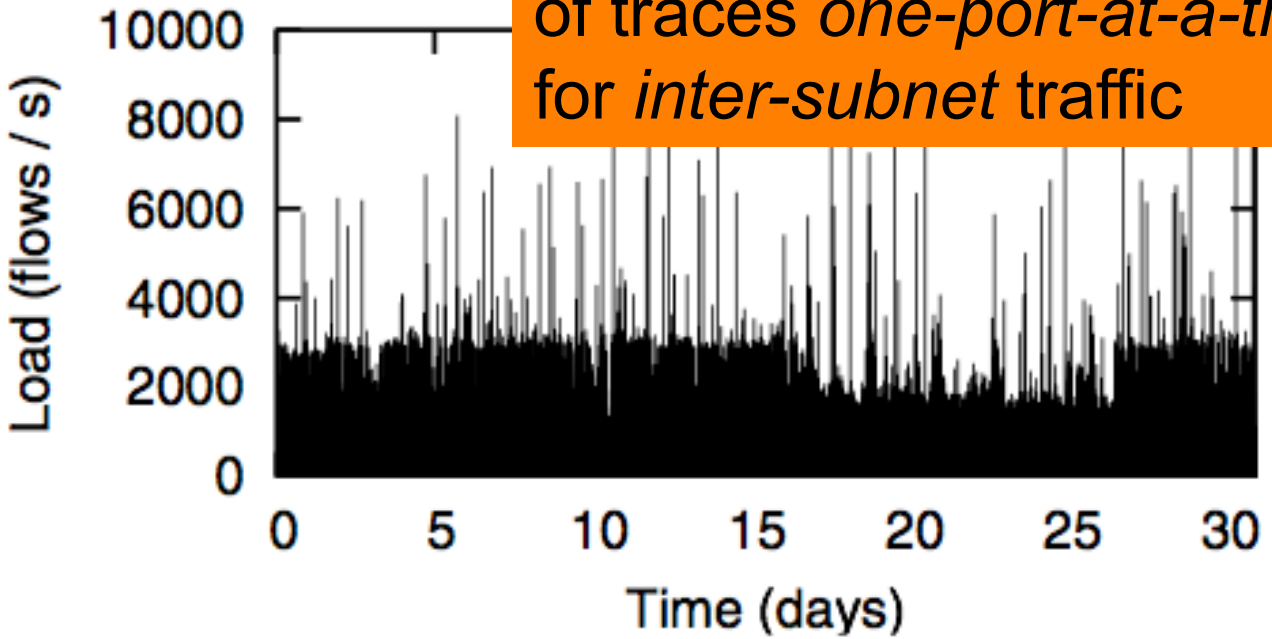


Figure 8: Flow-request rate for Stanford network.

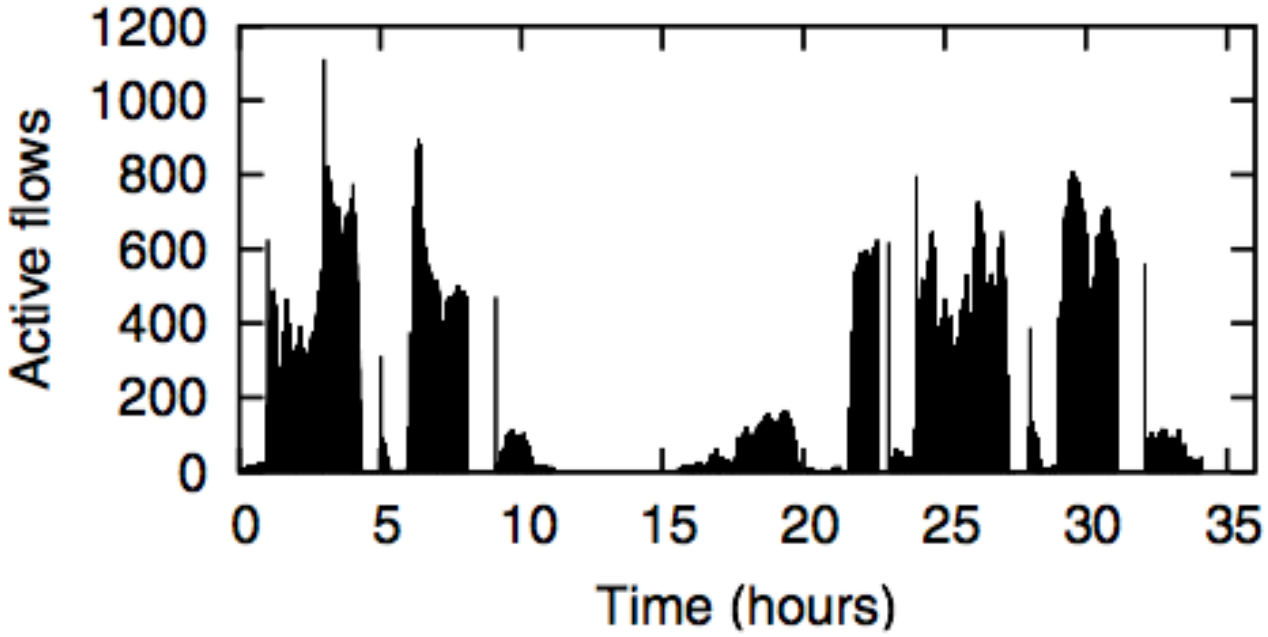


Figure 7: Active flows for LBL network [19].

OTOH, fact that authors ran system for real overcomes a lot of these sorts of evaluation concerns

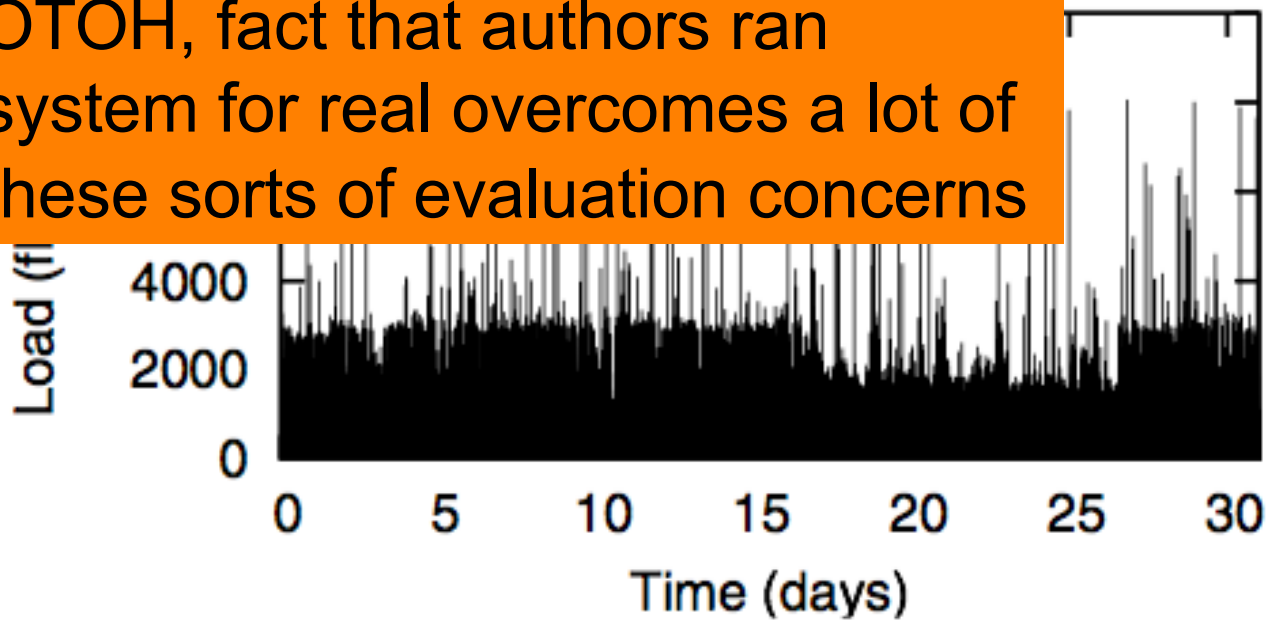


Figure 8: Flow-request rate for Stanford network.