

Internet measurements: Fault detection, identification, and topology discovery

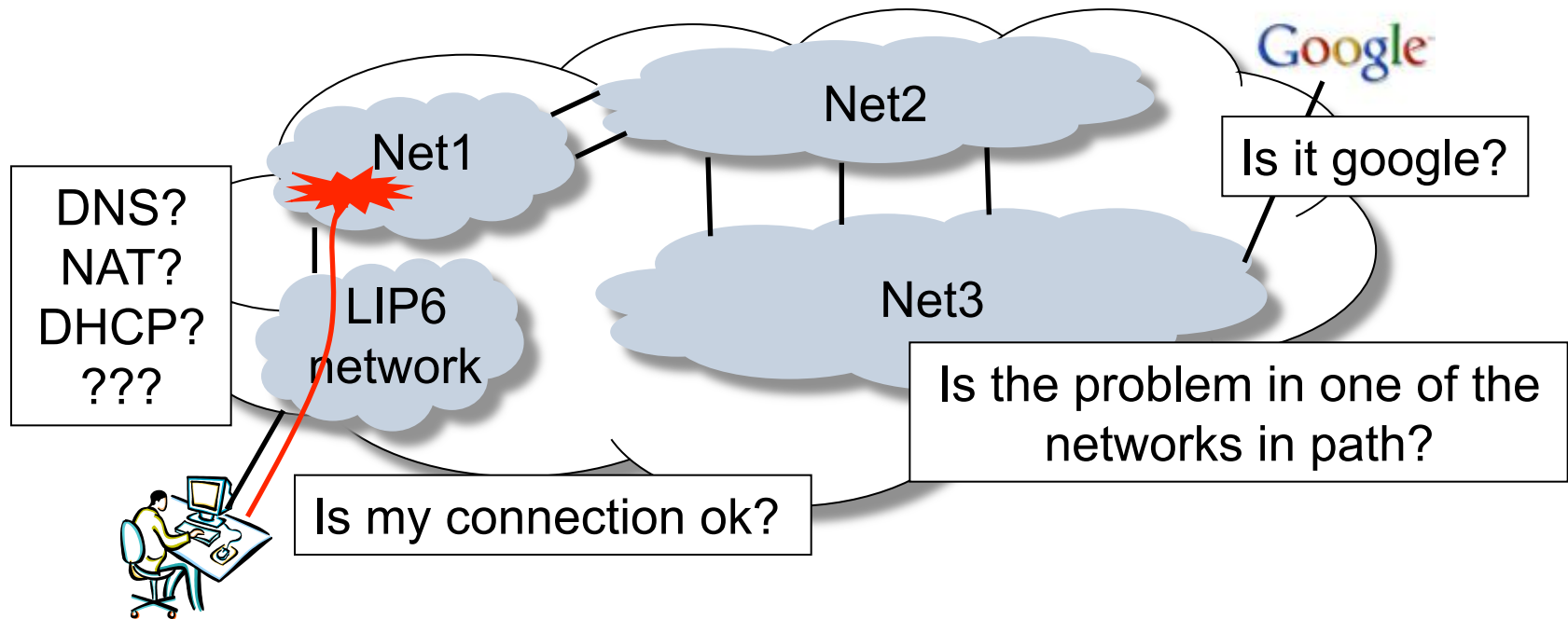
Renata Teixeira

Laboratoire LIP6

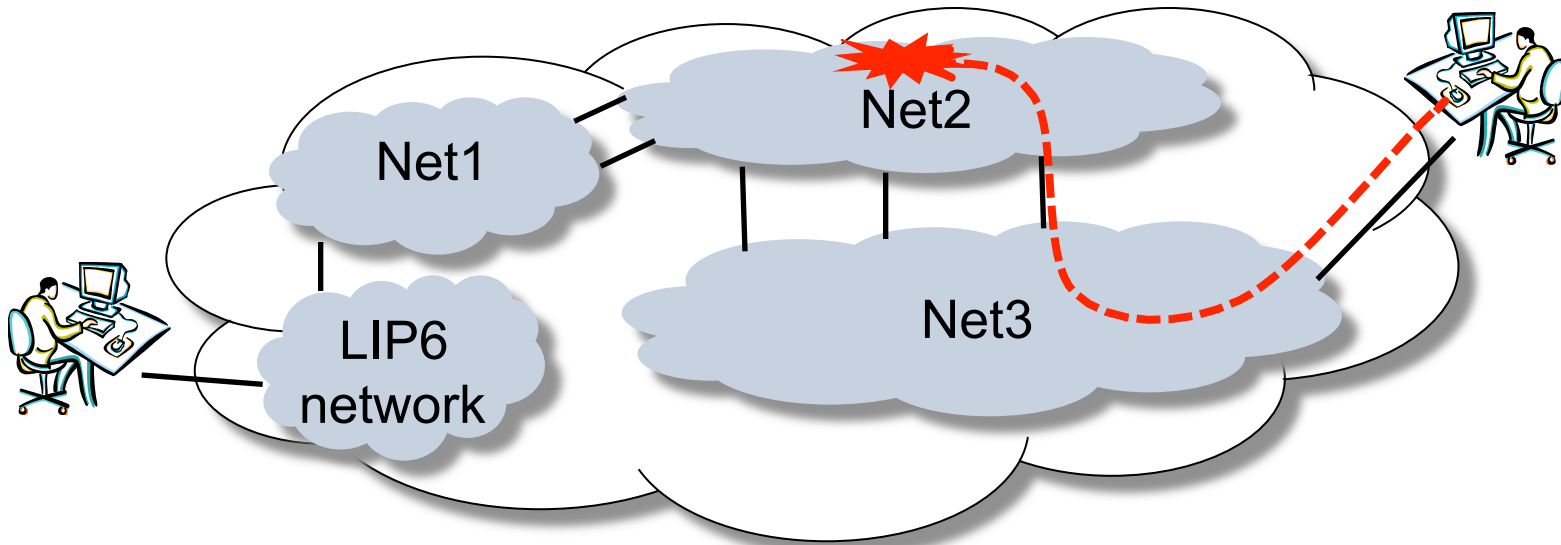
CNRS and UPMC Sorbonne Universités



Internet problems are hard to troubleshoot



Nobody has the full picture to diagnose problems



- End-users can't know what happens in network
- Network operators can't know user experience

End-to-end measurements to the rescue

- Available data depends on who you are
 - Network operators: Data from their own network
 - End-users: Data from their machine
- End-to-end measurements compensate for missing data
 - Network operators: Deploy monitoring hosts
 - End-users: Monitor some paths and cooperate with other users
- Troubleshooting is mostly manual and ad-hoc

Goal

Automatically troubleshoot network faults and performance disruptions

- Automatic detection: is there a problem?
 - Detect problems before users
 - How to detect problems that users care about?
- Fault identification: where is the problem?
 - Most common: traceroute
 - Assist in pinpointing the location of the problem
 - How to measure to obtain accurate location?

Overview

- Topology discovery
 - Topology mapping with traceroutes
 - Tracking topology changes
- Detection
 - Active versus passive fault detection techniques
 - Performance problems as perceived by users
- Identification
 - Network tomography for fault diagnosis

Overview

- Topology discovery
 - Topology mapping with traceroutes
 - Tracking topology changes
- Detection
 - Active versus passive fault detection techniques
 - Performance problems as perceived by users
- Identification
 - Network tomography for fault diagnosis

Measuring router topology

- With access to routers (or “from inside”)
 - Topology of one network
 - Routing monitors (OSPF or IS-IS)
- No access to routers (or “from outside”)
 - Multi-network topology or from end-hosts
 - Monitors issue active probes: traceroute

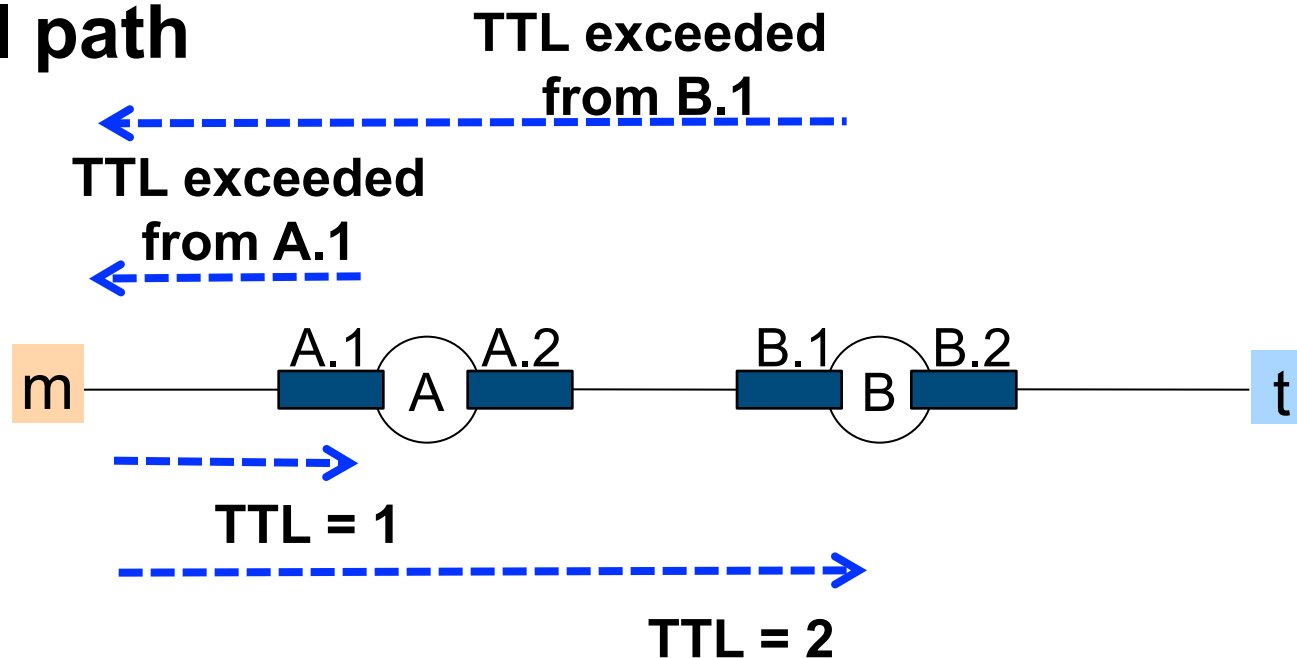
Topology from inside

- Routing protocols flood state of each link
 - Periodically refresh link state
 - Report any changes: link down, up, cost change
- Monitor listens to link-state messages
 - Acts as a regular router
 - AT&T's OSPFmon or Sprint's PyRT for IS-IS
- Combining link states gives the topology
 - Easy to maintain, messages report any changes

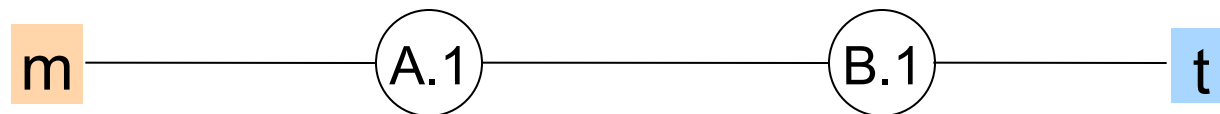
[Mortier] [Shaikh, 2004]

Inferring a path from outside: traceroute

Actual path



Inferred path



A traceroute path can be incomplete

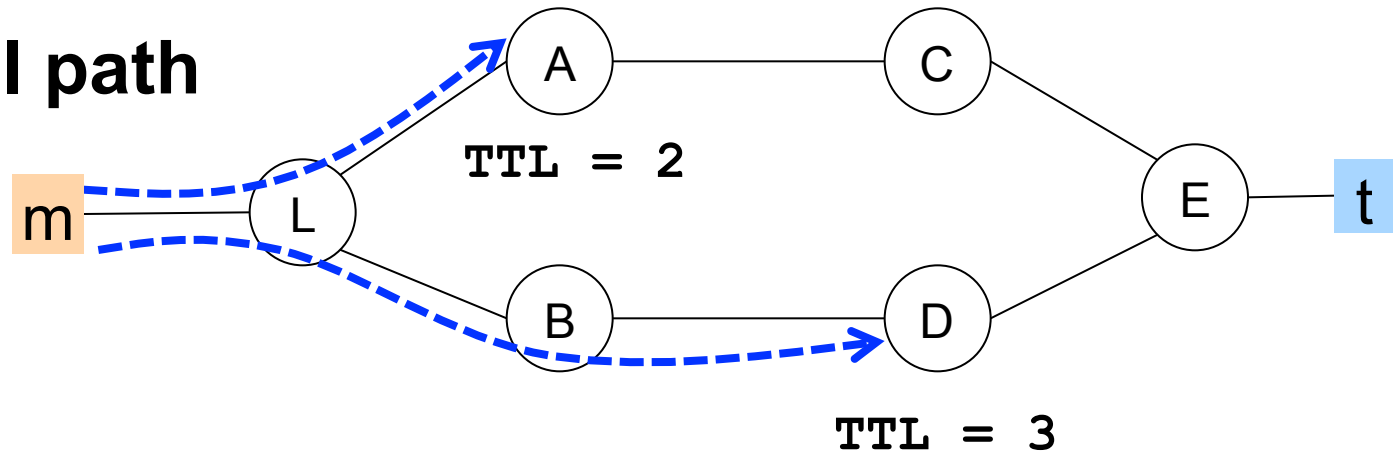
- Load balancing is widely used
 - Traceroute only probes one path
- Sometimes traceroute has no answer (stars)
 - ICMP rate limiting
 - Anonymous routers
- Tunnelling (e.g., MPLS) may hide routers
 - Routers inside the tunnel may not decrement TTL

Paris traceroute

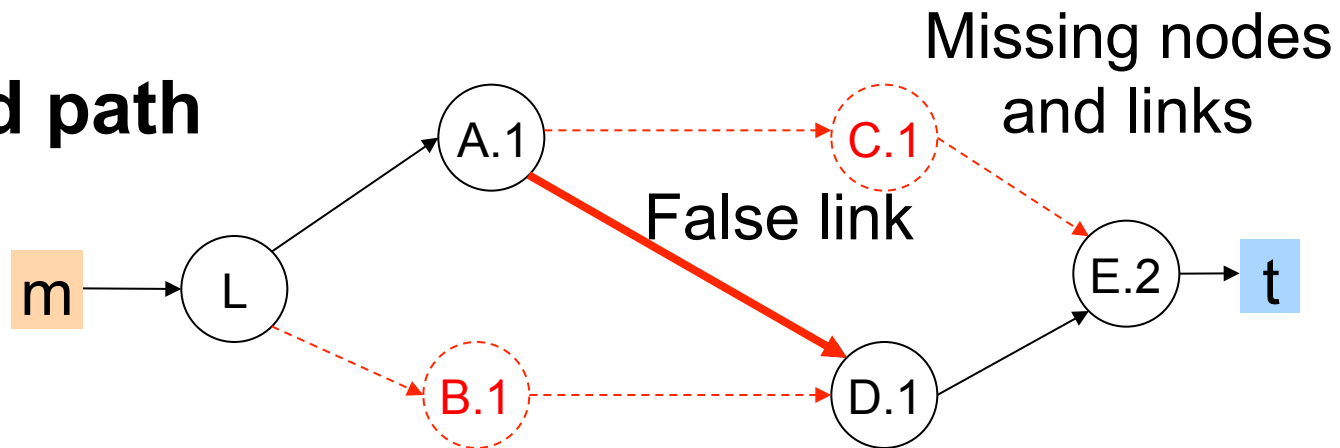
B. Augustin, X. Cuvellier, B. Orgogozo,
F. Viger, T. Friedman, M. Latapy,
C. Magnien (LIP6)
D. Veitch (U. Melbourne)

Traceroute under load balancing

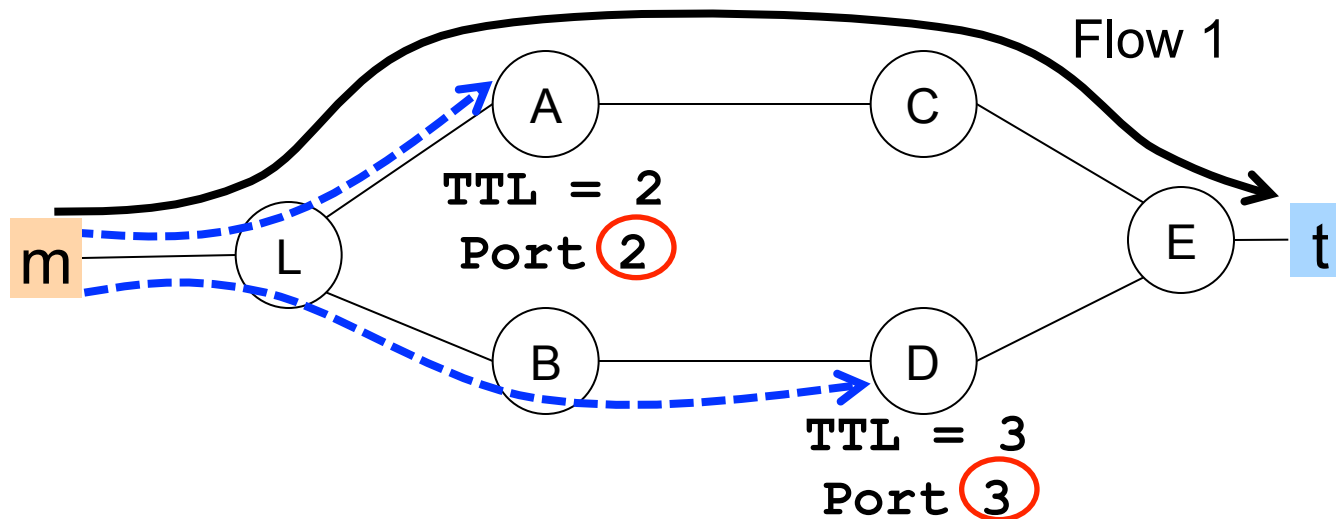
Actual path



Inferred path



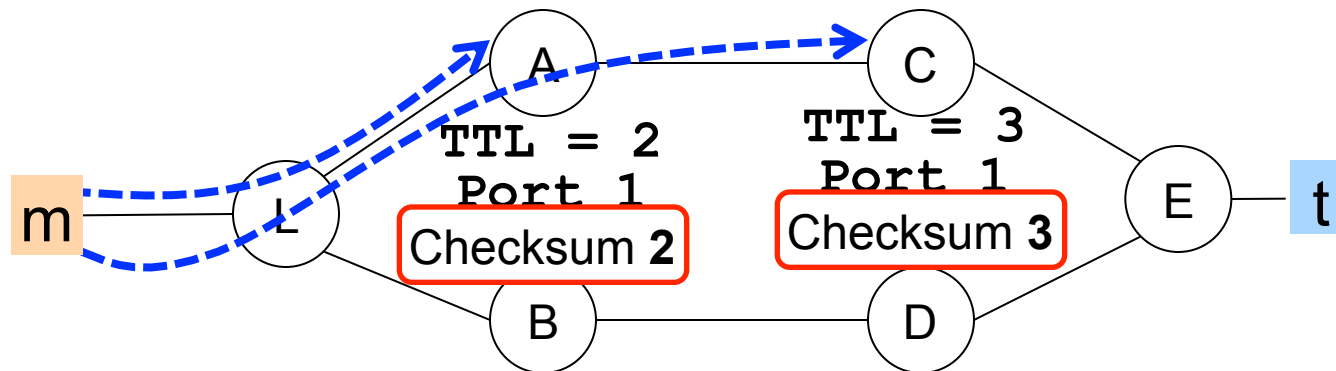
Errors happen even under per-flow load balancing



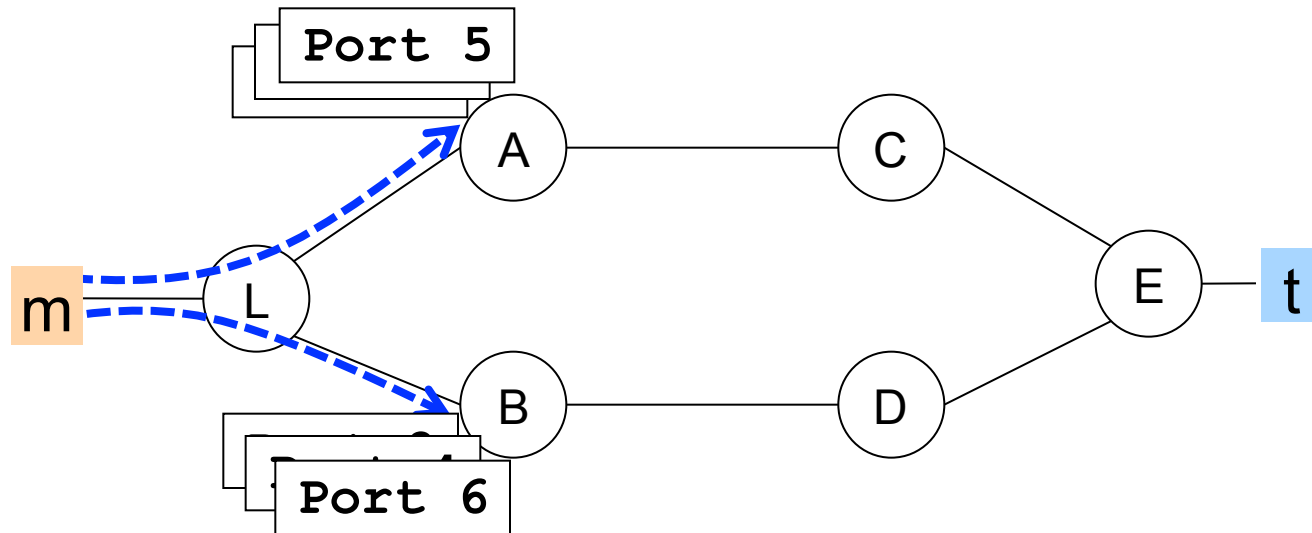
- Traceroute uses the destination port as identifier
 - Needs to match probe to response
 - Response only has the header of the issued probe

Paris traceroute: Removing false links

- Solves the problem with per-flow load balancing
 - Probes to a destination belong to same flow
- Changes the location of the probe identifier
 - Use the UDP checksum



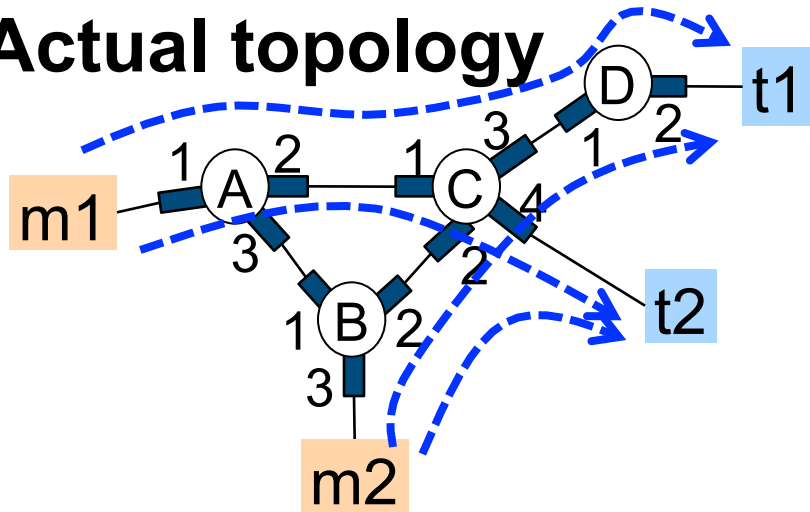
Paris traceroute: Tracing all the paths



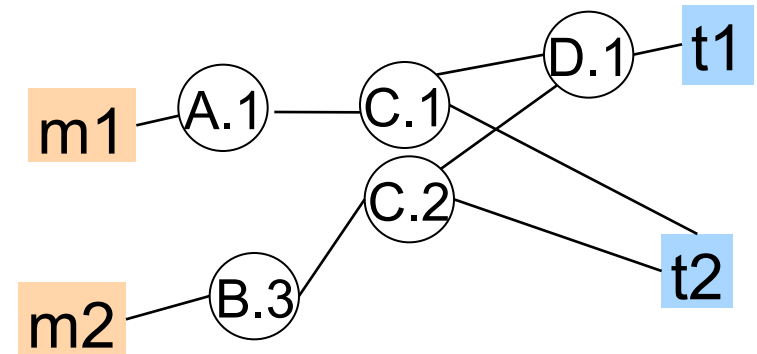
- Adaptive probing strategy
 - Vary flow id of probes
 - Send enough probes to enumerate all interfaces with high confidence

Topology from traceroutes

Actual topology



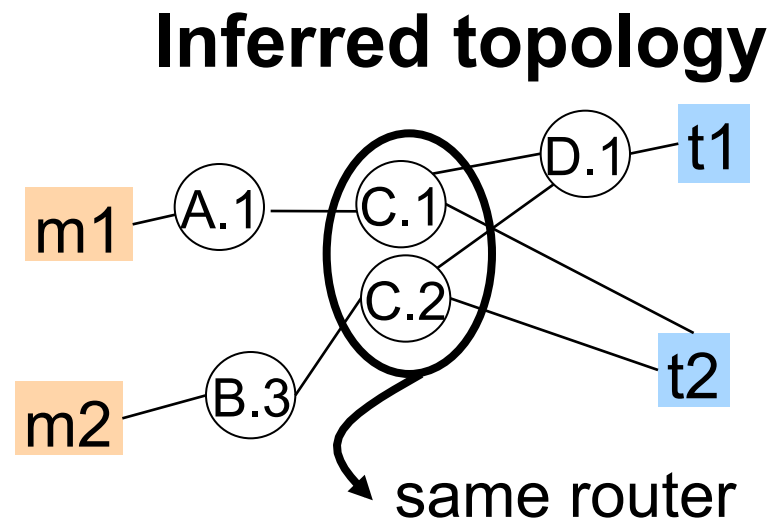
Inferred topology



- Inferred nodes = interfaces, not routers
- Coverage depends on monitors and targets
 - Misses links and routers
 - Some links and routers appear multiple times

Alias resolution: Map interfaces to routers

- Direct probing
 - Probe an interface, may receive response from another
 - Responses from the same router will have close IP identifiers and same TTL
- Record-route IP option
 - Records up to nine IP addresses of routers in the path



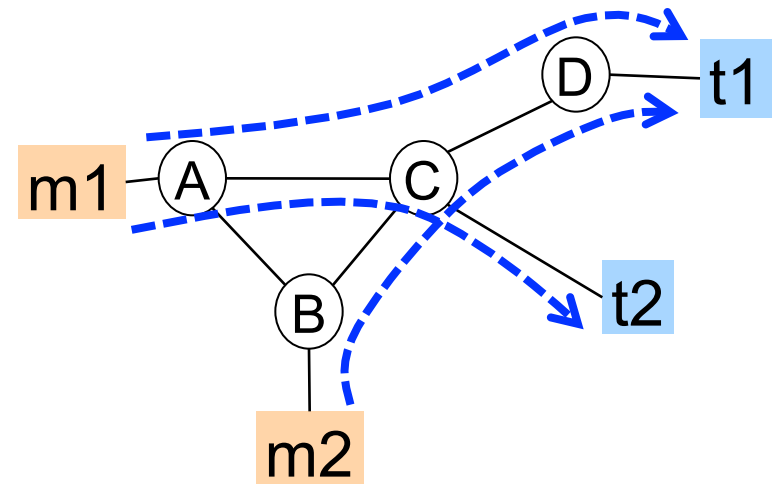
[Spring, 2002] [Sherwood, 2008]

Large-scale topology measurements

- Probing a large topology takes time
 - E.g., probing 1200 targets from PlanetLab nodes takes 5 minutes on average (using 30 threads)
 - Probing more targets covers more links
 - But, getting a topology snapshot takes longer
- Snapshot may be inaccurate
 - Paths may change during snapshot
- Hard to get up-to-date topology
 - To know that a path changed, need to re-probe

Faster topology snapshots

- Probing redundancy
 - Intra-monitor
 - Inter-monitor
- Doubletree
 - Combines backward and forward probing to eliminate redundancy



[Donnet, 2005]

Summary: Topology discovery

- Network operators
 - Own network: routing messages
 - Neighbor networks: traceroutes
- End users: combining traceroutes
 - Be aware of inaccuracies
 - False or missing links and nodes
 - Hidden hops: stars, tunneling

Overview

- **Topology discovery**
 - Topology mapping with traceroutes
 - Tracking topology changes
- **Detection**
 - Active versus passive fault detection techniques
 - Performance problems as perceived by users
- **Identification**
 - Network tomography for fault diagnosis

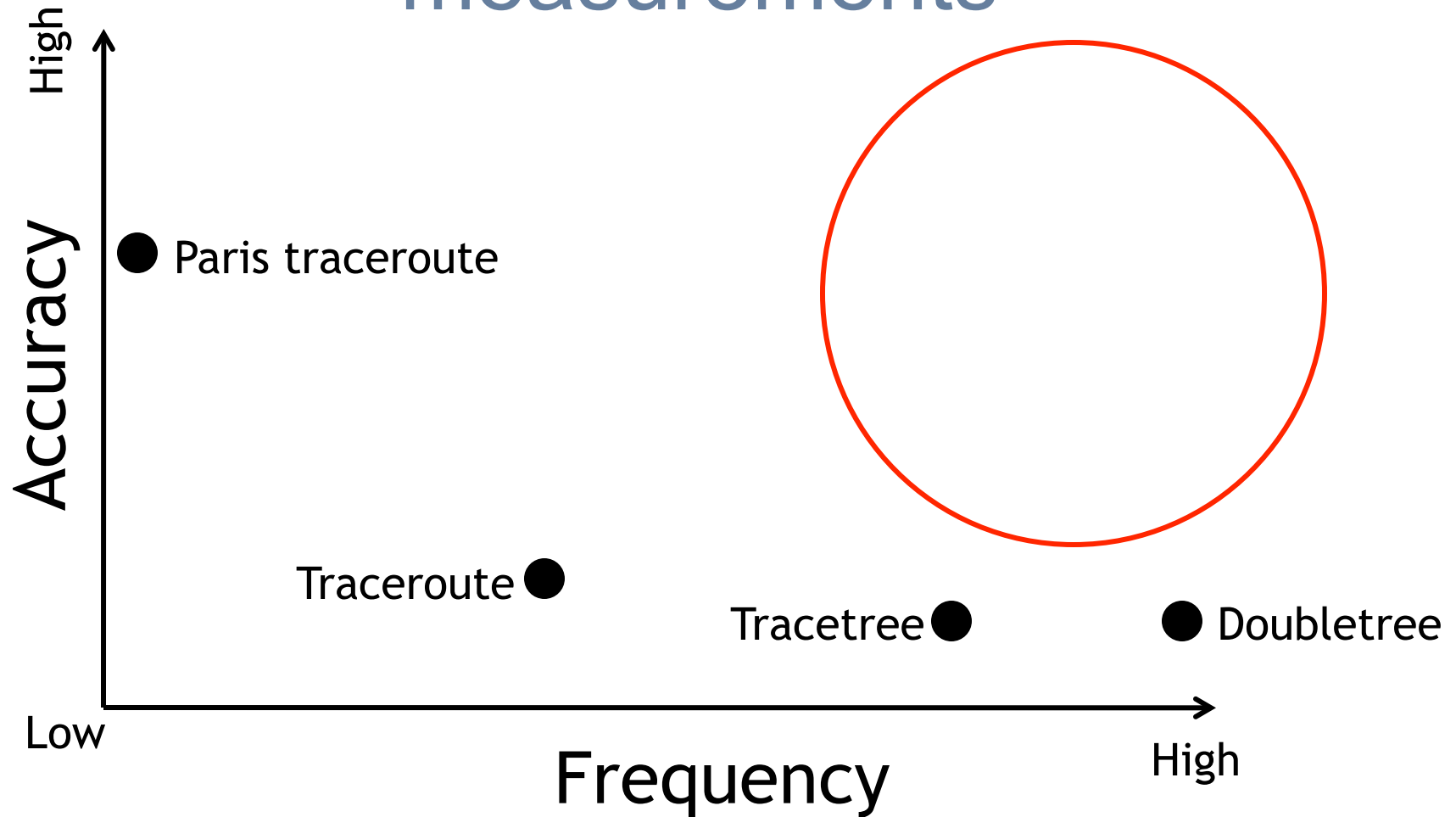
Predicting and tracking Internet path changes

I. Cunha (UPMC/Technicolor)
D. Veitch (U. Melbourne)
C. Diot (Technicolor)

Challenges of tracking topology with traceroute

- Cannot measure fast enough to detect all changes
 - Network and system limitations
- Accurate measurements require extra probes
 - Identify all paths under load balancing

Frequent vs. accurate measurements



Our approach

- Observation: Internet paths are mostly stable
 - Current techniques waste probes
- Probe according to path stability
- Separate tasks
 - Change detection: lightweight probing for speed
 - Path remapping: accuracy with Paris traceroute

Our contributions

- NN4: Predicting Internet path changes
 - Distinguish between stable and unstable paths
- DTrack: Tracking Internet path changes
 - Lightweight probing process to detect changes
 - Allocates more probes to unstable paths

Predicting path changes

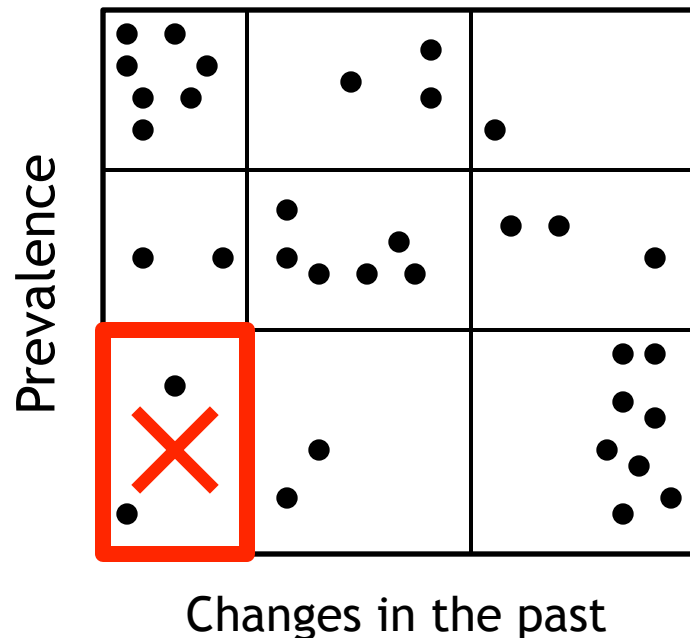
- Prediction goals
 - Time until the next change
 - Number of changes in a time interval
 - Whether a path will change in a time interval
- Identify features that can help with prediction
 - Features must be computable from traceroutes
 - Characteristics of the current path
 - Characteristics of the last path change
 - Behavior of the path in the recent past

Feature selection

- RuleFit to identify feature importance
 - Fraction of time path active in the past (prevalence)
 - Number of changes in the past
 - Number of previous occurrences of the current path
 - Path age
- Four most important features carry all the predictive information

NN4 predictor

- RuleFit is hard to integrate in other systems
- NN4 is based on the nearest-neighbor scheme
 - Compute neighbors by partitioning the path feature “state-space”
 - Prediction computed as the average behavior of all neighbors



NN4: summary

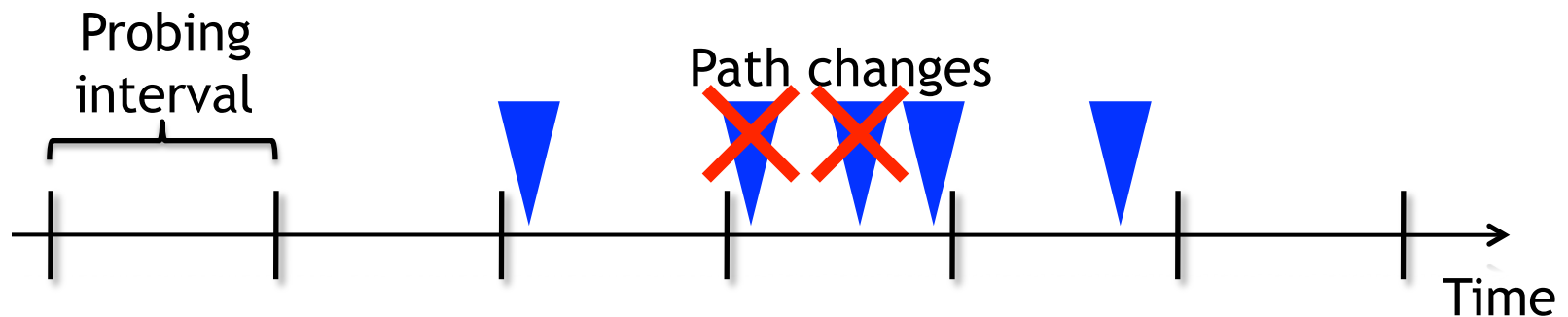
- NN4 is lightweight, easy to integrate, and as accurate as RuleFit
- Prediction is not highly accurate
 - It is possible to distinguish unstable/stable paths

DTrack

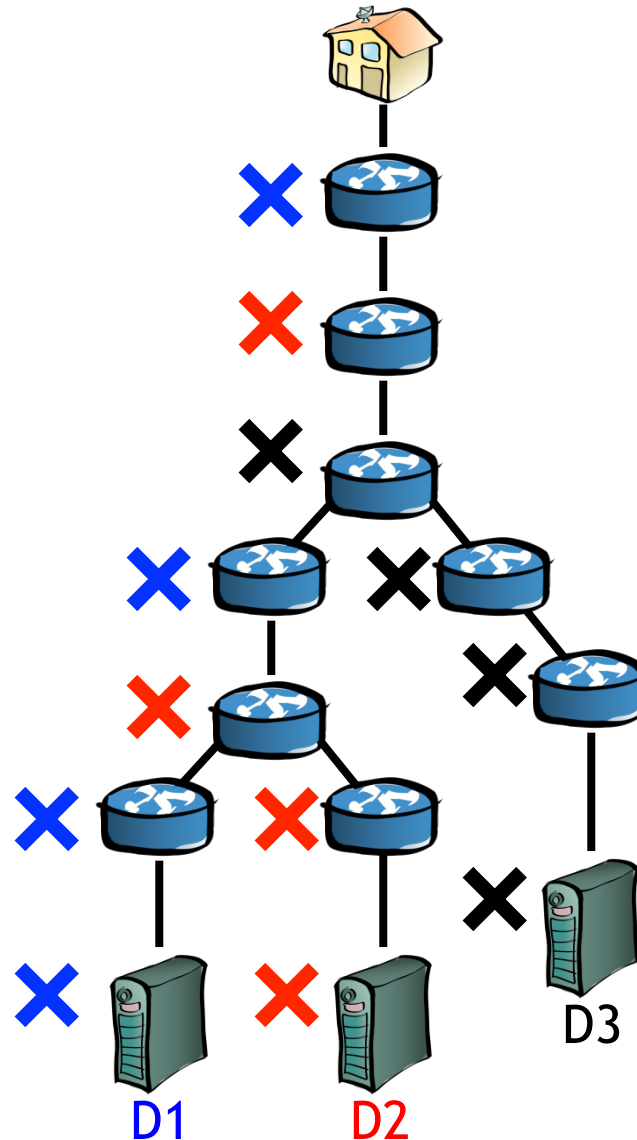
- Goal: Given a probing budget, detect as many changes as possible
- Allocates probing rates *per path* using NN4's predictions
- Targets probes along each path
 - Reduce redundant probes at shared links
 - Spread probes over time

Probe rate allocation

- Allocate rates that minimize number of missed changes
- Model changes in each path as a Poisson process
 - Estimate the rate of changes using NN4
- Compute missed changes as function of probing rate

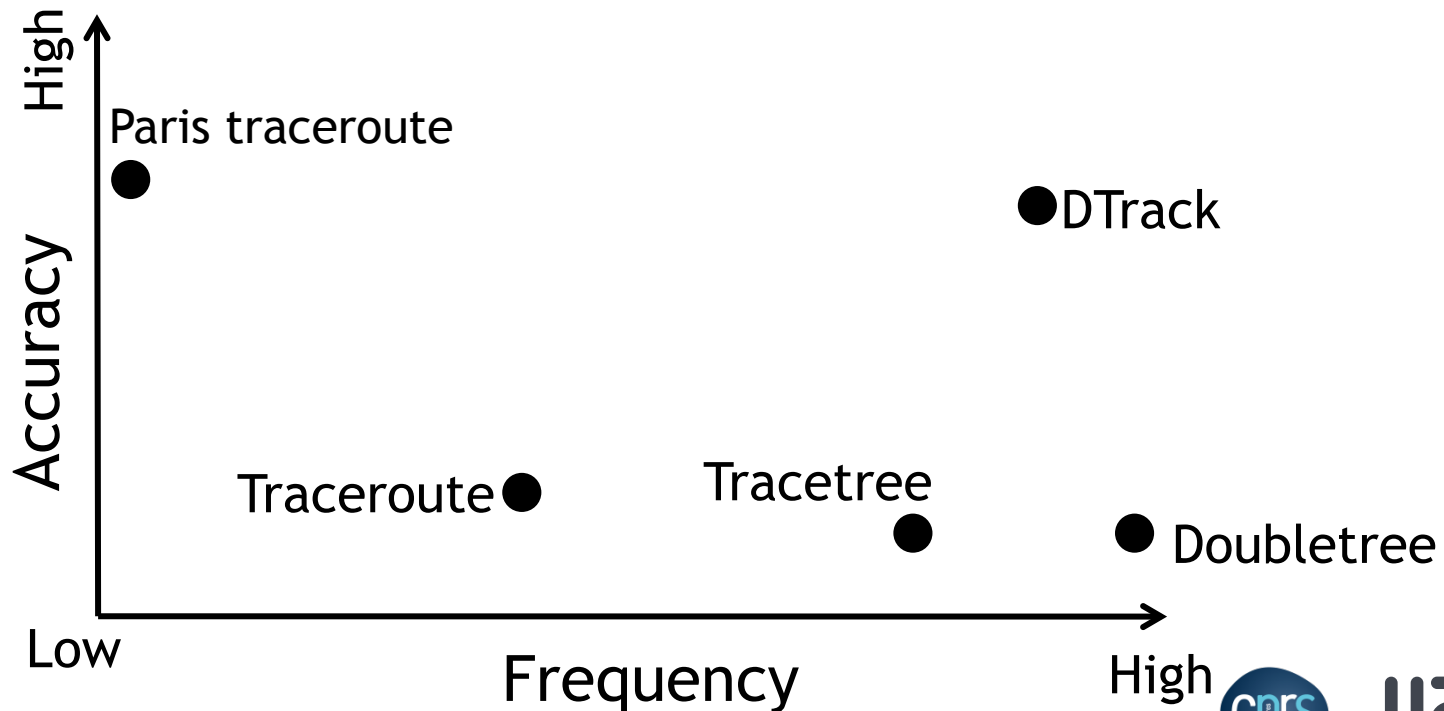


Probe targeting overview



DTrack summary

- Detects more changes than optimized traceroute
 - 2.2 times more changes in trace-driven simulations
 - 5 times more changes in PlanetLab



Overview

- Topology discovery
 - Topology mapping with traceroutes
 - Tracking topology changes
- Detection
 - Active versus passive fault detection techniques
 - Performance problems as perceived by users
- Identification
 - Network tomography for fault diagnosis

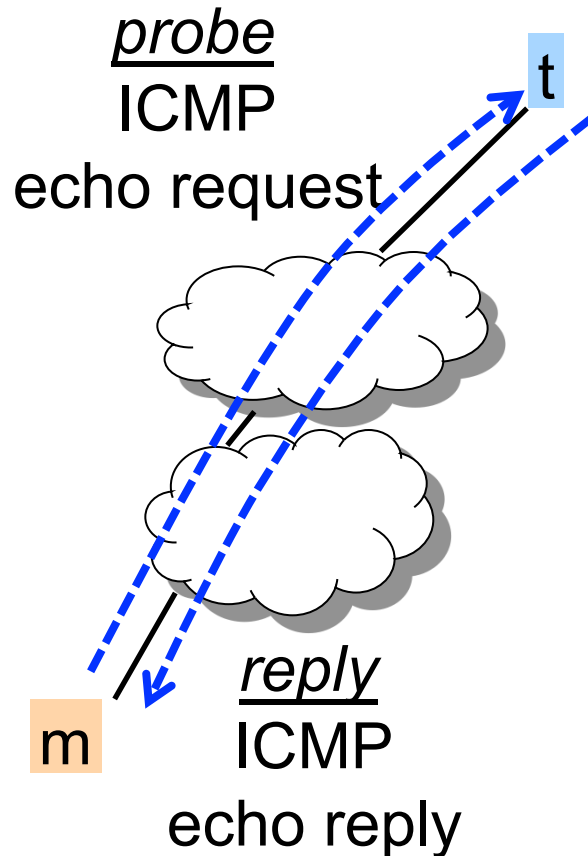
Faults versus performance disruptions

- Faults are persistent reachability problems
 - Cannot connect
- Performance disruptions
 - Web browsing is slow
 - Audio is choppy
 - Video is stalling

Fault detection techniques

- Active probing: ping
 - Send probe, collect response
 - From any end host
 - Works for network operators and end users
- Passive analysis of user's traffic
 - Tap incoming and outgoing traffic
 - At user's machines or servers: tcpdump, pcap
 - Inside the network: DAG card
 - Monitor status of TCP connections

Detection with ping



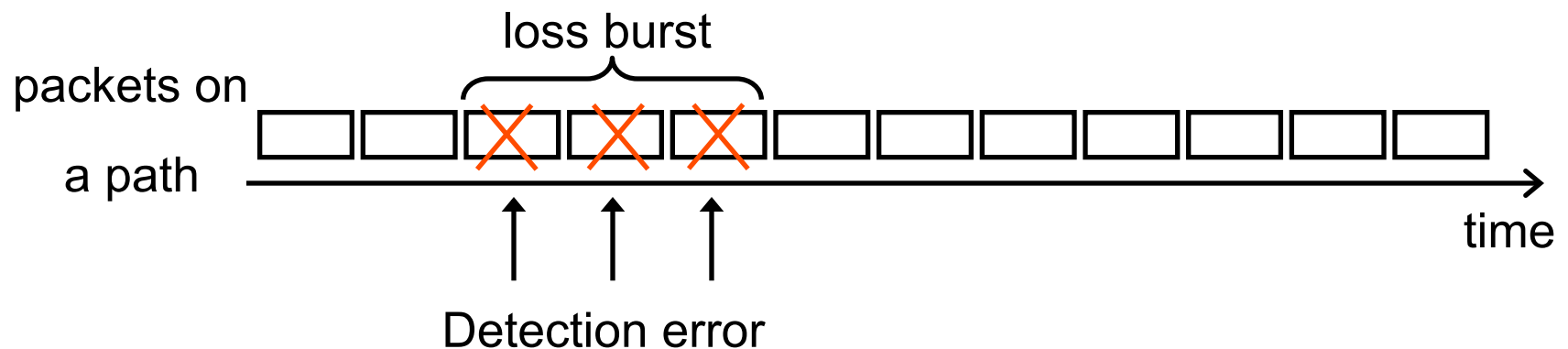
- If receives reply
 - Then, path is good ✓
- If no reply before timeout
 - Then, path is bad ✗

Persistent failure or measurement noise?

- Many reasons to lose probe or reply
 - Timeout may be too short
 - Rate limiting at routers
 - Some end-hosts don't respond to ICMP request
 - Transient congestion
 - Routing change
- Need to confirm that failure is persistent
 - Otherwise, may trigger false alarms



Failure confirmation

- Upon detection of a failure, trigger extra probes
- Goal: minimize detection errors
 - Sending more probes
 - Waiting longer between probes
- Tradeoff: detection error and detection time



[Cunha, 2009]

Passive detection at end hosts

- tcpdump/pcap captures packets
- Track status of each TCP connection
 - RTTs, timeouts, retransmissions
- Multiple timeouts indicate path is bad
 - If current seq. number $>$ last seq. number seen
 - Path is good 
 - If current seq. number = last seq. number seen
 - Timeout has occurred
 - After four timeouts, declare path as bad 

[Zhang, 2004]

Passive detection inside the network is hard

- Traffic volume is too high
 - Need special hardware
 - DAG cards can capture packets at high speeds
 - May lose packets
- Tracking TCP connections is hard
 - May not capture both sides of a connection
 - Large processing and memory overhead

Passive vs. active detection

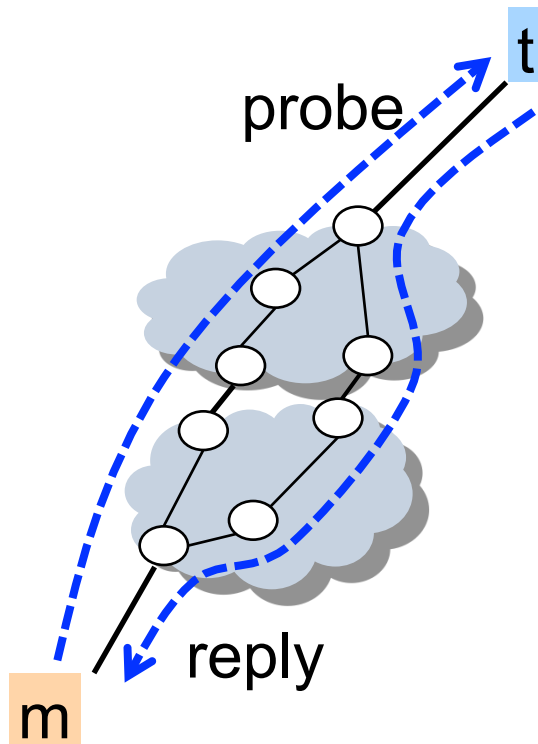
Passive

- + No need to inject traffic
- + Detects all failures that affect user's traffic
- + Responses from targets that don't respond to ping
- Not always possible to tap user's traffic
- Only detects failures in paths with traffic

Active

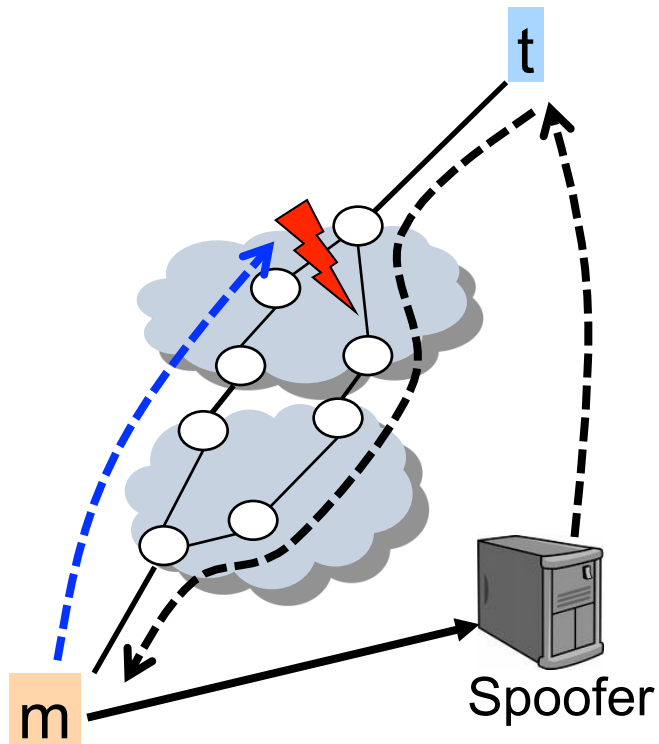
- + No need to tap user's traffic
- + Detects failures in any desired path
- Probing overhead
 - Cover a large number of paths
 - Detect failures fast

Is failure in forward or reverse path?



- Paths can be asymmetric
 - Load balancing
 - Hot-potato routing

Disambiguating one-way losses: Spoofing

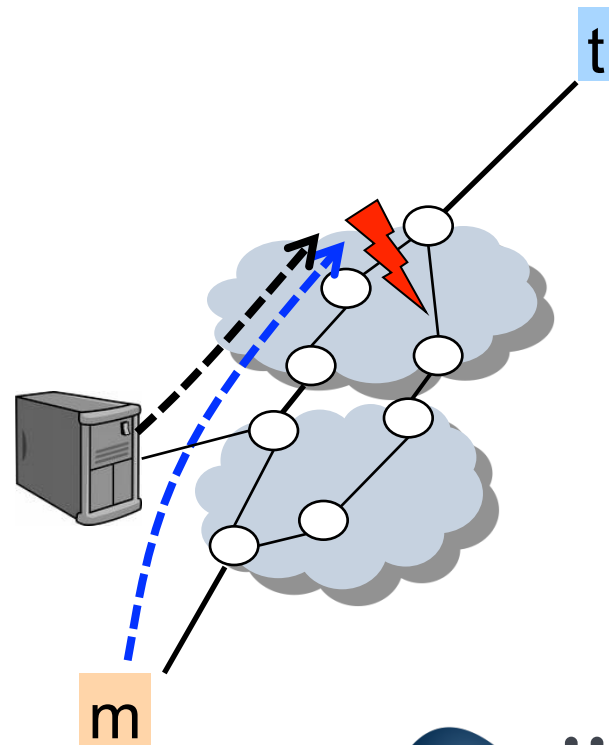


- Monitor requests to spoofer to send probe
- Spoofer sends spoofed probe with source address of the monitor
- If reply reaches the monitor, reverse path is good

[Katz-Bassett, 2008]

Limits of spoofing

- Network operators often drop spoofed packets
 - Spoofed packets are normally used for attacks
- Placement of spoofer
 - Paths from spoofer to targets need to be independent than paths from monitors



Summary: Fault detection

- End users: passive plus active probing
 - Passive measurements capture user's experience
 - Active probes
 - When path has no traffic
 - When TCP connections are too short
- Network operators: alarms plus active probing
 - Alarm systems directly report many faults
 - Active monitoring to capture customer's experience
 - Detect blackholes (i.e., faults that don't appear in alarms)
 - Detect faults in other networks

Overview

- Topology discovery
 - Topology mapping with traceroutes
 - Tracking topology changes
- **Detection**
 - Active versus passive fault detection techniques
 - Performance problems as perceived by users
- Identification
 - Network tomography for fault diagnosis

Understanding Users' Perception of Performance Disruptions at End-Hosts

Diana Joumblatt

Laboratoire LIP6 -- CNRS and UPMC Sorbonne Universités

Jaideep Chandrashekar, Nina Taft

Technicolor

User experience vs. typical network performance metrics

- Typical network performance metrics
 - Round-trip times (RTTs)
 - Throughput
 - Losses
- User perspective only in niche applications
 - Voice
 - Video
 - Gaming
- What is missing?
 - User perspective of overall online experience

Why user perspective matters?

- Automated diagnosis
 - Avoid unnecessary diagnosis when users don't care
 - Improve diagnosis when network performance seems ok and the user is annoyed
- Performance management
 - Adapt application performance to user needs

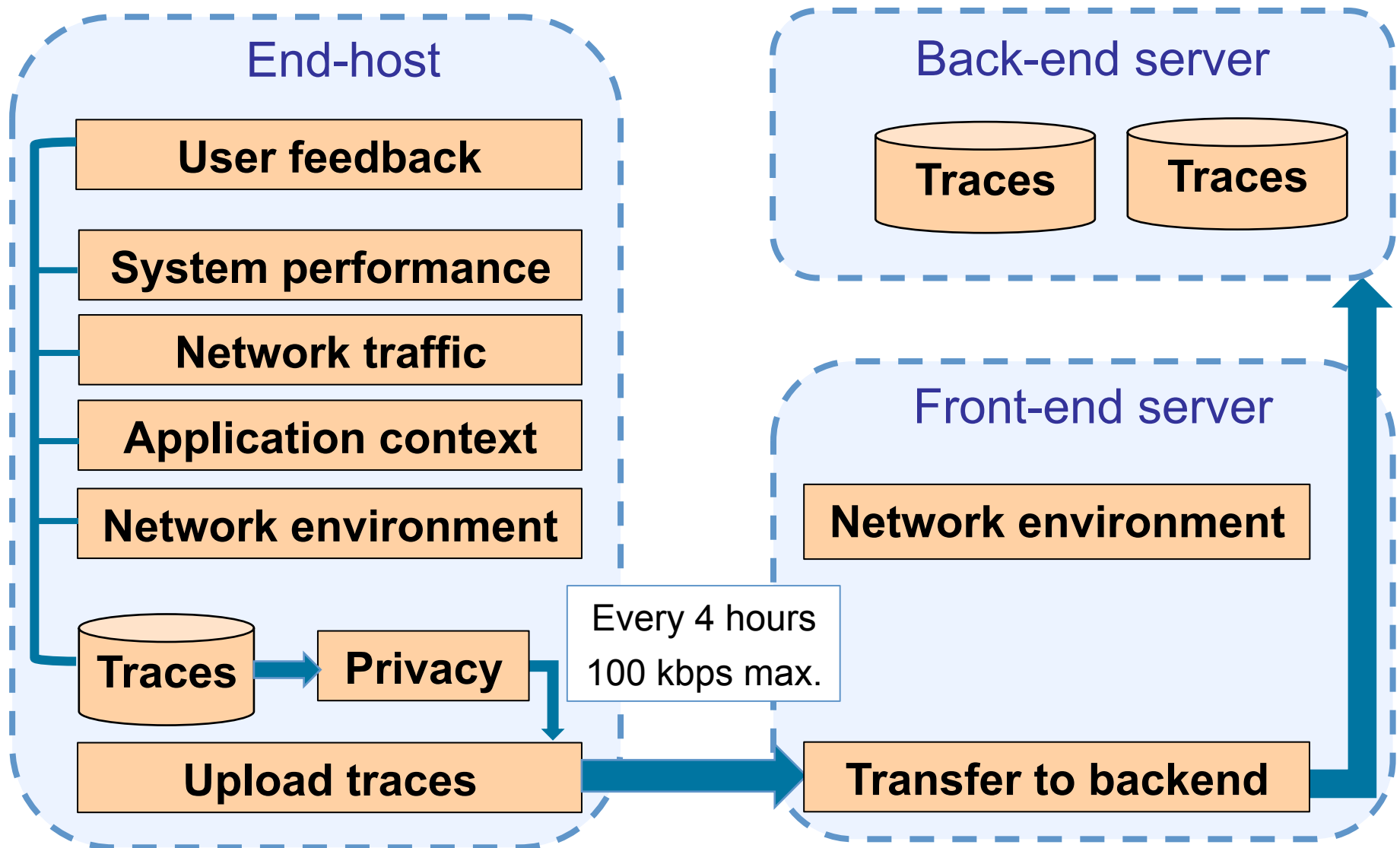
Challenges in measuring user perception

- User perception varies
 - Per user, per environment, per application
 - For a given user according to external factors
- Imbalance in number of samples
 - Can't collect frequent user feedback (~ 10 per day)
 - Orders of magnitude more network measurements ($\sim 10^3/10^4/10^5/\dots$)
- End-host data collection raises issues
 - Privacy
 - Machine overhead

Our approach

- Measure network performance with user feedback
 - HostView: end-host data collection
- Correlate network performance with user feedback

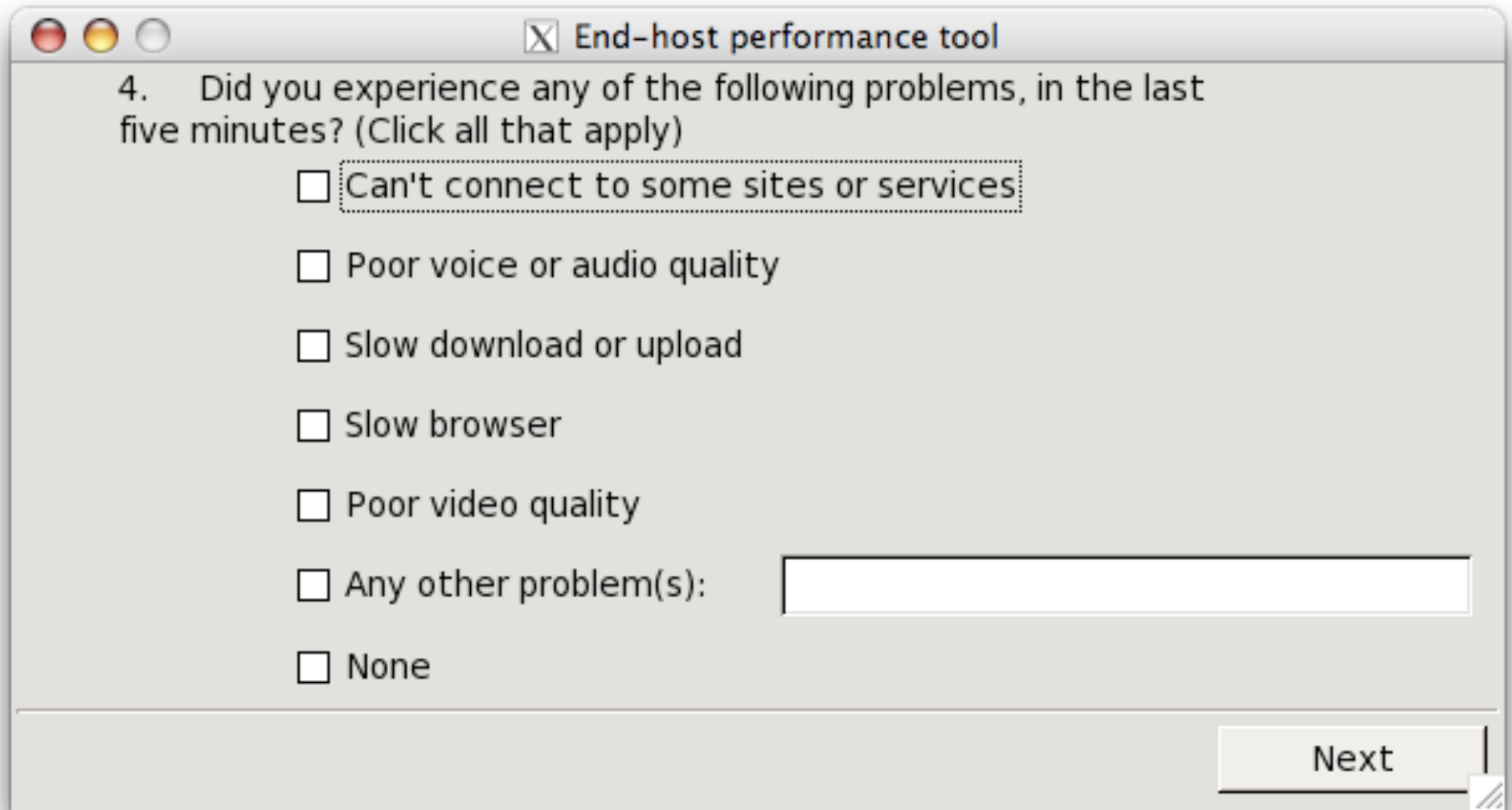
HostView overview



User feedback mechanisms

- System-triggered feedback
 - Experience sampling methodology (ESM)
 - Triggered based on state of machine
 - 5 short questions about network performance
 - At most 3 times a day
- User-triggered feedback
 - “I’m annoyed” button 😞
 - Same questions as in ESM
 - Can trigger as often as user wants

Example question



A screenshot of a survey window titled "End-host performance tool". The window has a standard macOS-style title bar with red, yellow, and green buttons. The main content area contains a question: "4. Did you experience any of the following problems, in the last five minutes? (Click all that apply)". Below the question are seven checkboxes with corresponding text: "Can't connect to some sites or services", "Poor voice or audio quality", "Slow download or upload", "Slow browser", "Poor video quality", "Any other problem(s):", and "None". The "Any other problem(s):" checkbox is followed by a text input field. At the bottom right of the window is a "Next" button.

☒ End-host performance tool

4. Did you experience any of the following problems, in the last five minutes? (Click all that apply)

- ☐ Can't connect to some sites or services
- ☐ Poor voice or audio quality
- ☐ Slow download or upload
- ☐ Slow browser
- ☐ Poor video quality
- ☐ Any other problem(s):
- ☐ None

Next

Deployment

- Recruiting volunteers
 - Leaflets at IMC 2010 and CS Mailing lists
 - 50 USD Amazon gift cards
 - Real-time feedback about network connection
- Data: 40 users
 - Nov 2010 – Feb 2011
 - 26 Mac OS and 14 Linux
 - 14 countries
 - Most users ran tool for one month

User vs. network reporting

- User perspective
 - Good and poor performance epochs as flagged by the user
- Network and system Perspective
 - Good and poor performance epochs as flagged by lower-level metrics that trigger atypical (anomalous) behavior automatically
- Question: Do these co-occur?

Anomalous performance

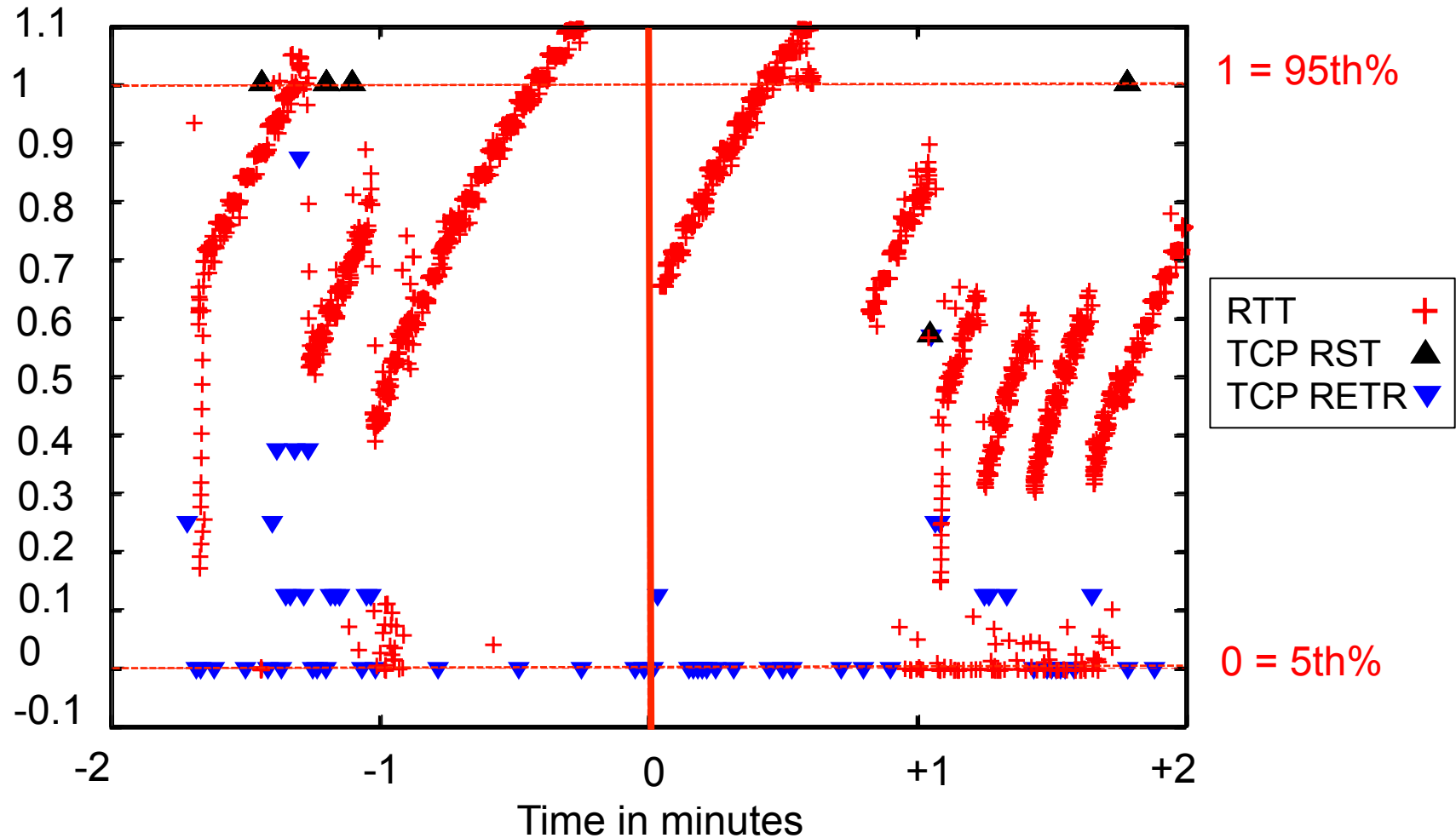
■ Metrics

- RTTs
- TCP retransmissions
- Wireless noise level
- Machine CPU load
- Data rates
- Wireless signal strength
- Any instance of TCP reset

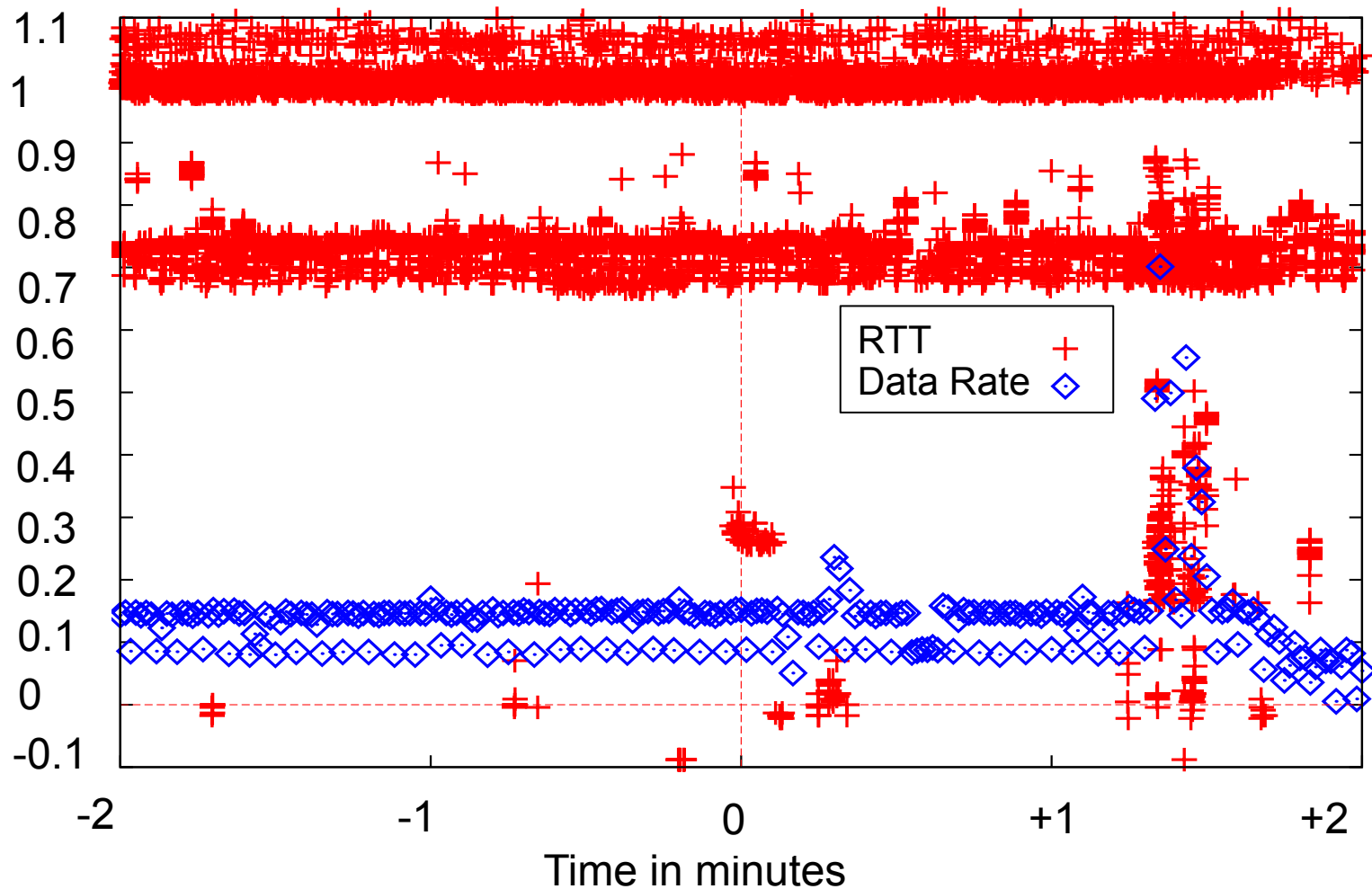
Above 95th percentile

Below 5th percentile



Can't connect to some sites or services



Everything is good!



Matching performance reports

<div> <div>User report</div> <div>Machine anomaly</div> </div>	Yes	No	No feedback
	<div>95 reports</div>	<div>352 reports</div>	
Yes	<div>  <div>82 reports</div> </div>	<div> <div>User doesn't care</div> <div>313 reports</div> </div>	Missing user feedback
No	<div> <div>Not reporting correct system metrics</div> <div>13 reports</div> </div>	<div>  <div>39 reports</div> </div>	Missing user feedback

Summary: Correlating user feedback with network performance

- Hard to get feedback from users
 - Many network performance samples without feedback
 - Users are diverse in how they report a problem
- Raw network metrics alone are not enough
 - Not all outliers affect the user perception

Next steps

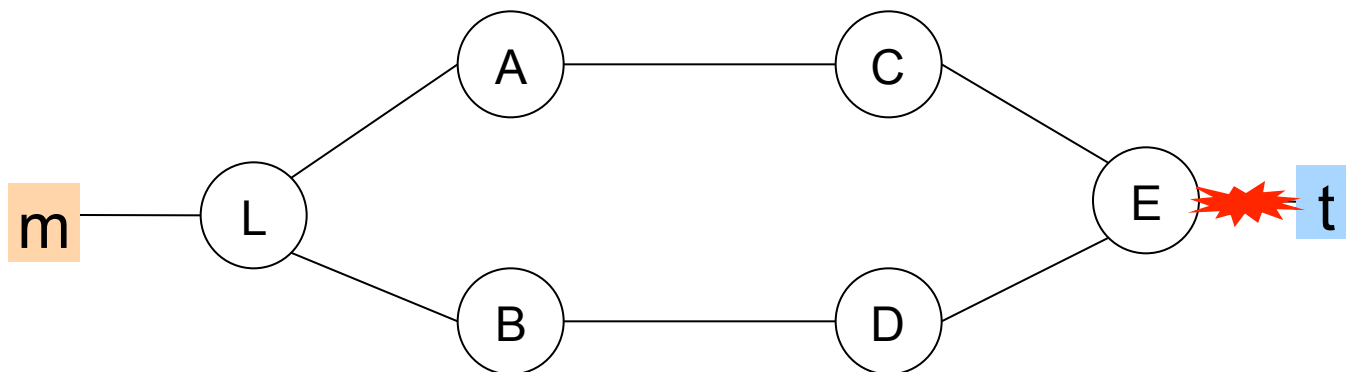
- Incorporate user context to define anomaly
 - Application
 - Environment
 - What annoys the user
- Apply machine learning techniques
 - Use HostView data to train models of user perception
- Build online detector of poor user experience

Overview

- Topology discovery
 - Topology mapping with traceroutes
 - Tracking topology changes
- Detection
 - Active versus passive fault detection techniques
 - Performance problems as perceived by users
- Identification
 - Network tomography for fault diagnosis

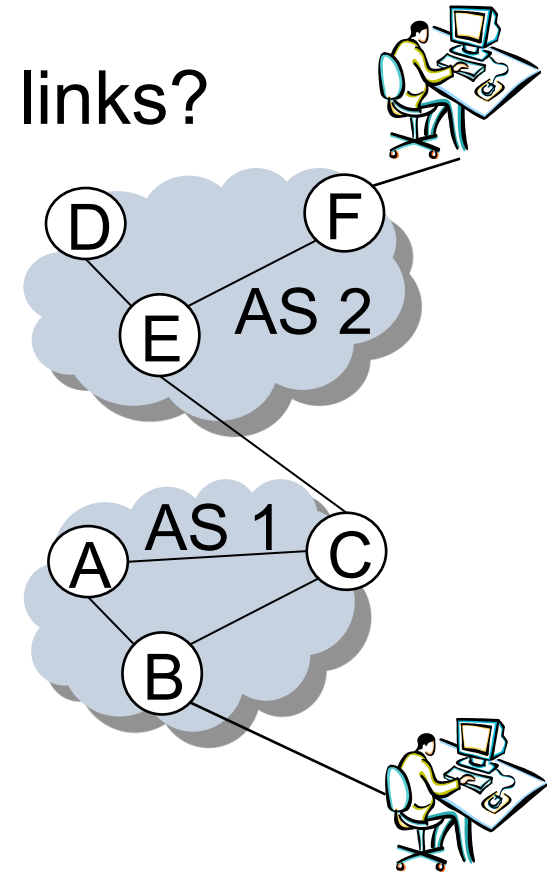
Fault identification with traceroute

- Forwarding loops and some reachability problems
- Traceroute identifies the effect, not the cause



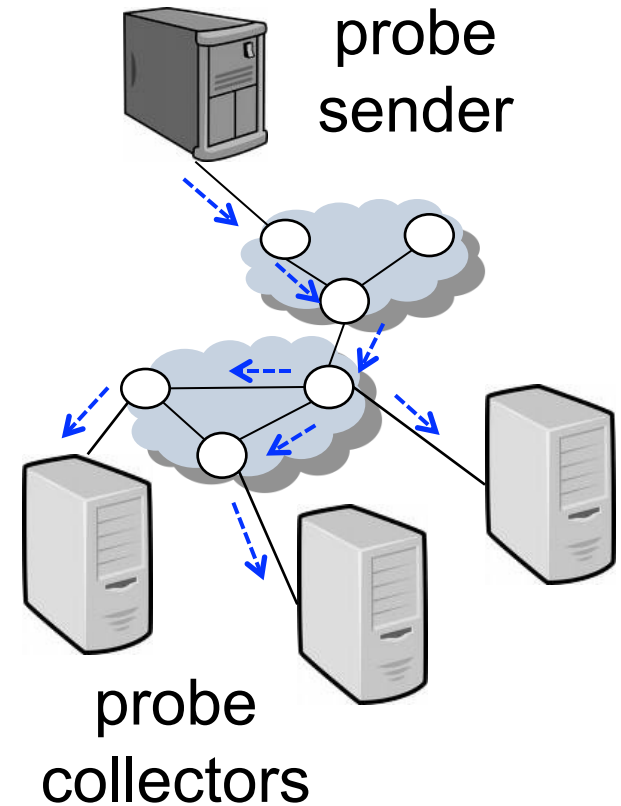
Network tomography to infer link performance

- What are the properties of network links?
 - Loss rate
 - Delay
 - Bandwidth
 - Connectivity
- Given end-to-end measurements
 - No access to routers



The origins

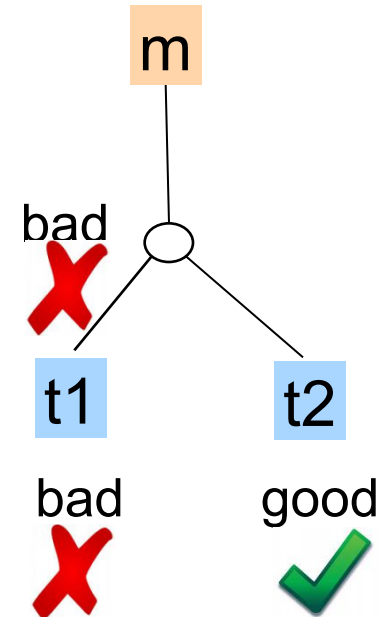
- MINC: Multicast-based Inference of Network-internal Characteristics
- Key idea: multicast probes
 - Exploit correlation in traces to estimate link properties



[MINC project, 1999]

Binary tomography

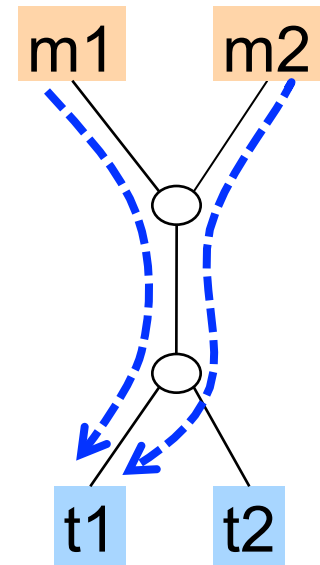
- Given
 - Complete network topology
 - Path reachability
- Find the smallest set of links that explains bad paths
 - If link is bad, all paths that cross the links are bad
 - Given bad links are uncommon



[Duffield, 2006]

Fault identification with binary tomography

- Fault monitoring needs multiple sources and targets
- Problem becomes NP-hard
 - Minimum hitting set problem
- Iterative greedy heuristic
 - Given the set of links in bad paths
 - Iteratively choose link that explains the max number of bad paths



Hitting set of link =
paths that traverse
the link

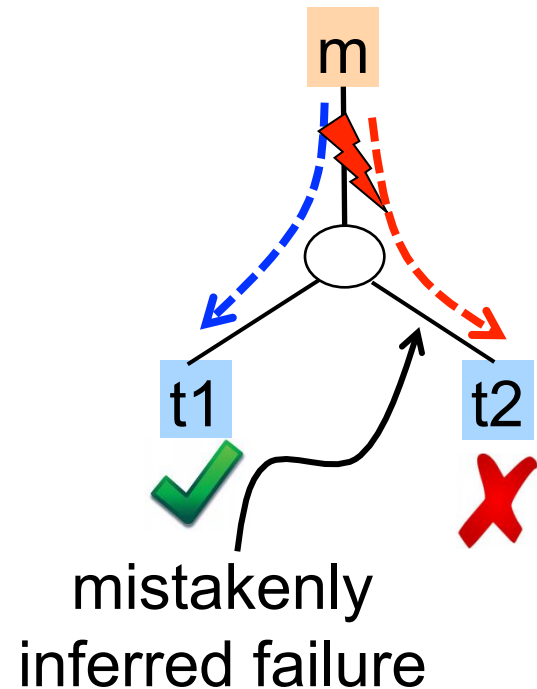
[Kompella, 2007] [Dhamdhere, 2007]

Practical issues

- Topology is often unknown
 - Need to measure accurate topology
- Multicast not available
 - Need to extract correlation from unicast probes
 - Even using probes from different monitors
- Control of targets is not always practical
 - Need one-way performance from round-trip probes
- Links can fail for some paths, but not all
 - Need to extend tomography algorithms

Uncorrelated measurements lead to errors

- Lack of synchronization leads to inconsistencies
 - Probes cross links at different times
 - Path may change between probes



Sources of inconsistencies

- In measurements from a single monitor
 - Probing all targets can take time
- In measurements from multiple monitors
 - Hard to synchronize monitors for all probes to reach a link at the same time
 - Impossible to generalize to all links

Inconsistent measurements with multiple monitors

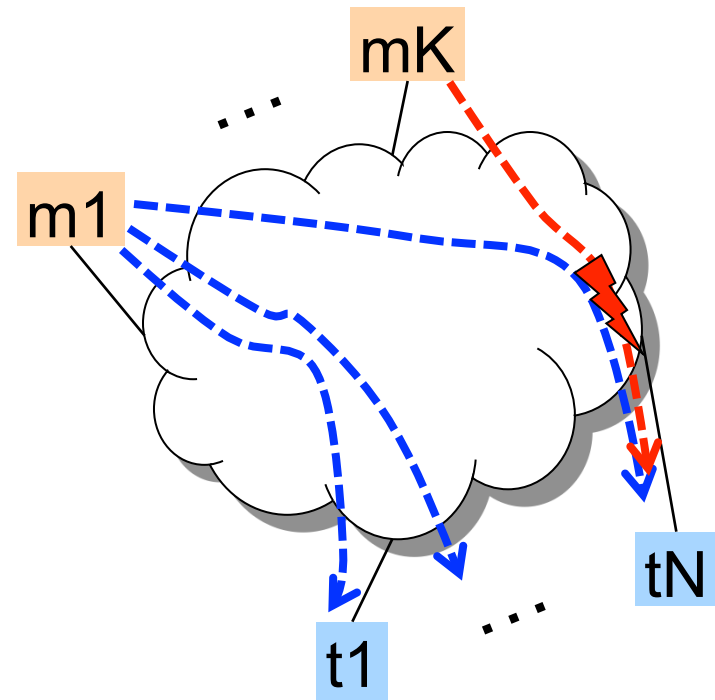
path reachability

m1,t1	good	mK,t1	good
-------	------	-------	------

⋮	⋮	⋮	⋮
---	---	---	---

m1, tN	good	mK, tN	bad
--------	------	--------	-----

inconsistent
measurements



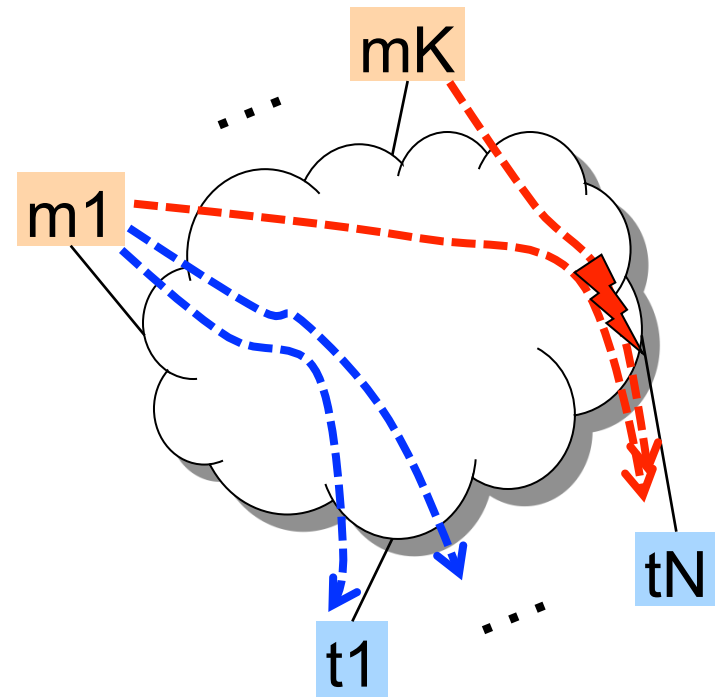
Solution:

Reprobe paths after failure

path reachability

m1,t1	good	mK,t1	good
⋮	⋮	⋮	⋮
m1, tN	bad	mK, tN	bad

- Consistency has a cost
 - Delays fault identification
 - Cannot identify short failures



Summary: Tomography

- Trade-off: consistency vs. identification speed
 - Faster identification leads to false alarms
 - Slower identification misses short failures
- Network operators
 - Too many false alarms are unmanageable
 - Longer failures are the ones that need intervention
- End users
 - Even short failures affect performance

Thank you!

- HostView: <http://cmon.lip6.fr/EMD>
 - Measure end-host performance with user feedback
 - Mac OS and Linux
- HomeNet Profiler: <http://cmon.lip6.fr/hnp>
 - One-shot test
 - Home network configuration
 - WiFi performance
 - Internet access performance

REFERENCES

Topology from inside

- IS-IS monitoring
 - R. Mortier, “Python Routeing Toolkit (‘PyRT’)”, <https://research.sprintlabs.com/pyrt/>
- OSPF monitoring
 - A. Shaikh and A. Greenberg, “OSPF Monitoring: Architecture, Design and Deployment Experience”, NSDI 2004
- Commercial products
 - Packet Design: <http://www.packetdesign.com/>

Topology with traceroute

- Tracing accurate paths under load-balancing
 - B. Augustin *et al.*, “Avoiding traceroute anomalies with Paris traceroute”, IMC, 2006.
 - D. Veitch, B. Augustin, R. Teixeira, and T. Friedman, " Failure Control in Multipath Route Tracing", in Proc. of IEEE Infocom, April 2009.
- Reducing overhead to trace topology of a network and alias resolution with direct probing
 - N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP Topologies with Rocketfuel”, SIGCOMM 2002.
- Use of record route to obtain more accurate topologies
 - R. Sherwood, A. Bender, N. Spring, “DisCarte: A Disjunctive Internet Cartographer”, SIGCOMM, 2008.

Optimizing topology discovery

- Reducing overhead to take a topology snapshot
 - B. Donnet, P. Raoult, T. Friedman, and M. Crovella, “Efficient Algorithms for Large-Scale Topology Discovery”, SIGMETRICS, 2005.
- Tracking topology changes
 - I. Cunha, R. Teixeira, D. Veitch, and C. Diot, "Predicting and Tracking Internet Path Changes, in Proc. of ACM SIGCOMM, August 2011.

Reducing overhead of active fault detection

- Selection of paths to probe
 - H. Nguyen and P. Thiran, “Active measurement for multiple link failures diagnosis in IP networks”, PAM, 2004.
 - Yigal Bejerano and Rajeev Rastogi, “Robust monitoring of link delays and faults in IP networks”, INFOCOM, 2003.
- Selection of the frequency to probe paths
 - H. X. Nguyen , R. Teixeira, P. Thiran, and C. Diot, " Minimizing Probing Cost for Detecting Interface Failures: Algorithms and Scalability Analysis", INFOCOM, 2009.

Performance disruptions as perceived by users

- HostView design
 - D. Joumlatt, R. Teixeira, J. Chandrashekar, N. Taft, "HostView: Annotating end-host performance measurements with user feedback", in Proc. of ACM HotMetrics Workshop, June 2010.
- Correlation of network performance and user feedback
 - D. Joumlatt, R. Teixeira, J. Chandrashekar, N. Taft, "Performance of Networked Applications: The Challenges in Capturing the User's Perception", in Proc. of ACM SIGCOMM Workshop on Measurements Up the Stack, August 2011.

Network tomography theory

- Survey on network tomography
 - R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, “Network Tomography: Recent Developments”, Statistical Science, Vol. 19, No. 3 (2004), 499-517.
- Traffic matrix estimation
 - Y. Vardi, “Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data”, Journal of the American Statistical Association, Vol. 91, 1996.
- Inference of link performance/connectivity
 - MINC project: <http://gaia.cs.umass.edu/minc/>
 - A. Adams et al., “The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior”, IEEE Communications Magazine, May 2000.

Binary tomography

- Single-source tree algorithm
 - N. Duffield, “Network Tomography of Binary Network Performance Characteristics”, IEEE Transactions on Information Theory, 2006.
- Applying tomography in one network
 - R. R. Kompella, J. Yates, A. Greenberg, A. C. Snoeren, “Detection and Localization of Network Blackholes”, IEEE INFOCOM, 2007.
- Applying tomography in multiple network topology
 - A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, “NetDiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data”, CoNEXT, 2007.
- Obtaining accurate path status for binary tomography
 - I. Cunha, R. Teixeira, N. Feamster, C. Diot, “Measurement Methods for Fast and Accurate Blackhole Identification with Binary Tomography”, in Proc. of ACM Internet Measurement Conference, November 2009.

Internet-wide fault detection systems

- Detection with BGP monitoring plus continuous pings, spoofing to disambiguate one-way failures, traceroute to locate faults
 - E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, “Studying Black Holes in the Internet with Hubble”, NSDI, 2008.
- Detection with passive monitoring of traffic of peer-to-peer systems or content distribution networks, traceroutes to locate faults
 - M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang, “PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services”, OSDI, 2004.