Detailed Analysis of ISIS Routing Protocol on the Qwest Backbone:

A recipe for subsecond ISIS convergence

Cengiz Alaettinoglu
cengiz@packetdesign.com

Stephen Casner
casner@packetdesign.com

Why subsecond convergence?

- Increased network reliability
- Support for multi−service traffic
  - Voice over IP, ATM over IP, TDM over IP, ...
- Lower cost/complexity compared to layer 2 protection schemes like SONET

Where are we today?

- Current IP re−route times are typically tens of seconds
- We need to do better. There are two choices:
  - Figure out what’s wrong with IP routing and fix it
  - Replace IP routing with something else

Qwest Backbone ISIS Analysis

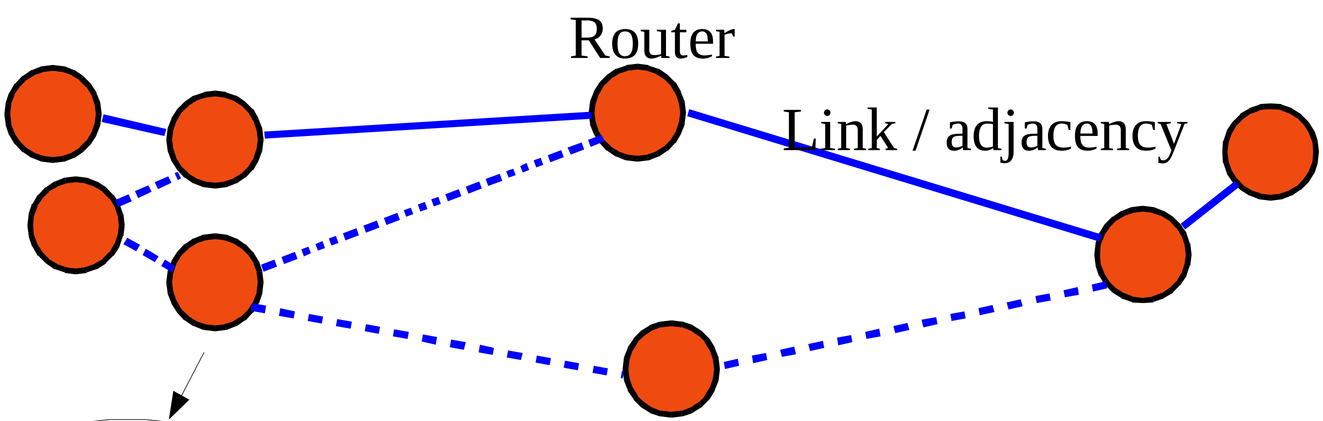
- Collected multiple week−long ISIS packet traces
- Identified problem areas:
  - causes of ISIS churn and stability
  - sequence of events and delays during routing convergence
- Conclude with a recipe for achieving subsecond ISIS convergence

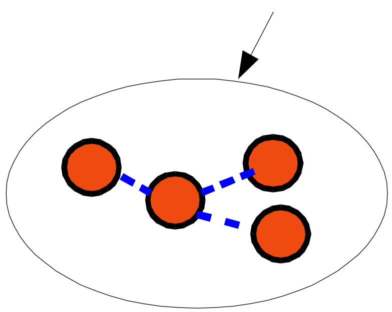
Qwest ISIS Findings Summary

- The backbone is extremely stable
  - 3 out of the 4 week−long data collection periods have no route change on our path
  - the churn is caused by few problem links

- Convergence times
  - Hard to find a link failure to diagnose
  - Convergence as fast as today’s technology allows
  - Can be improved to subsecond

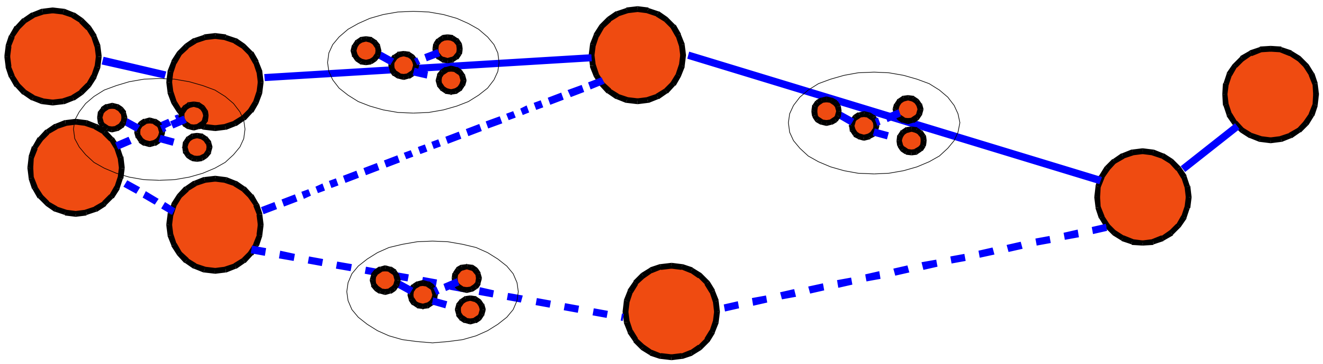
ISIS Routing Protocol 101: Link State Packets





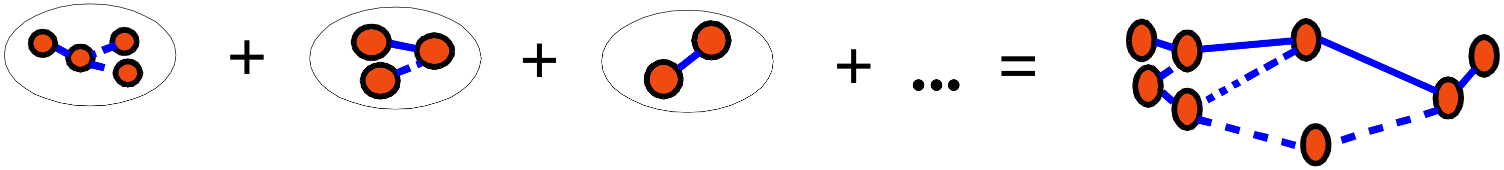
- An LSP describes the local topology of a router
- List of adjacencies
  - Hello protocol

ISIS Routing Protocol 101: Flooding



- LSPs are flooded
  - link propagation delays + per−hop processing delays

ISIS Routing Protocol 101: Shortest Path Calculation

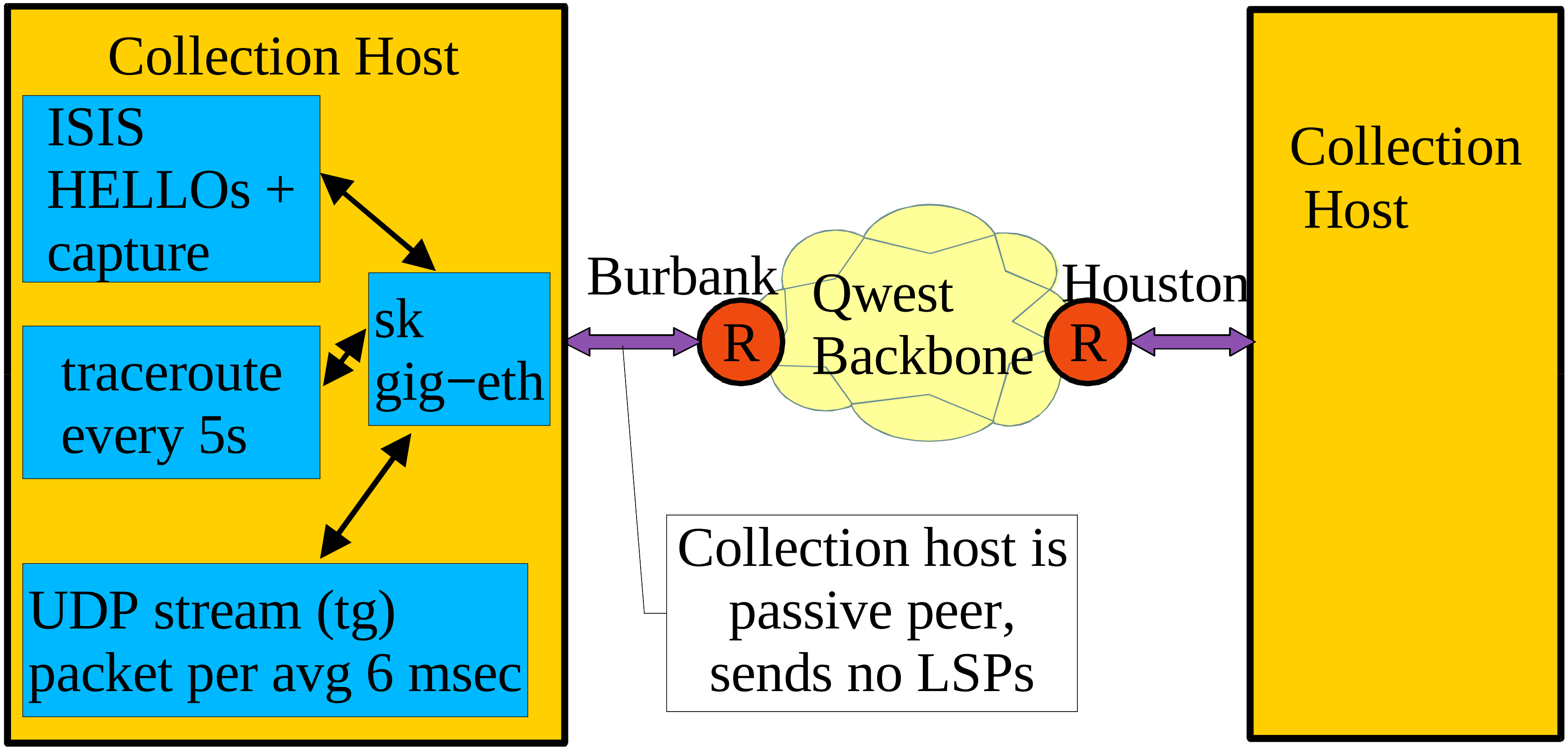


- A router constructs the current topology
- Dijkstra's SPF algorithm determines the routes

ISIS Routing Protocol 102: Convergence after a Topology Change

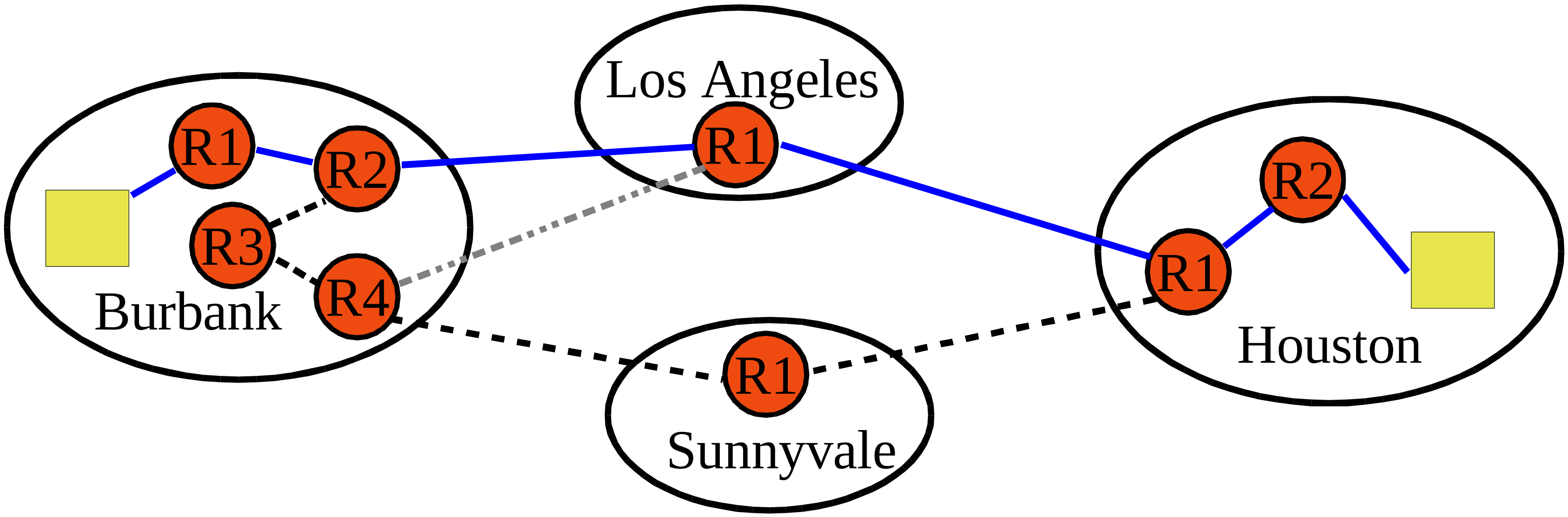
- Detection
  - Link up/down or peer reachability
  - Hardware detection is fast & preferred
  - Software detection using an HELLO protocol is slower but is a backup
- Propagation
  - Flood a Link State Packet (LSP)
  - Link propagation delays + per hop processing delay
  - Rate limiting may slow propagation
- New Route Computation
  - Run Dijkstra’s Shortest Path First (SPF) algorithm
  - CPU resource intensive
  - Rate limiting may delay SPF computation & consistency

Our Measurement Setup



- Multi−vendor backbone: Ciscos, Junipers, ...
- All point−to−point backbone: OC48, OC192, ...

Our Typical Path

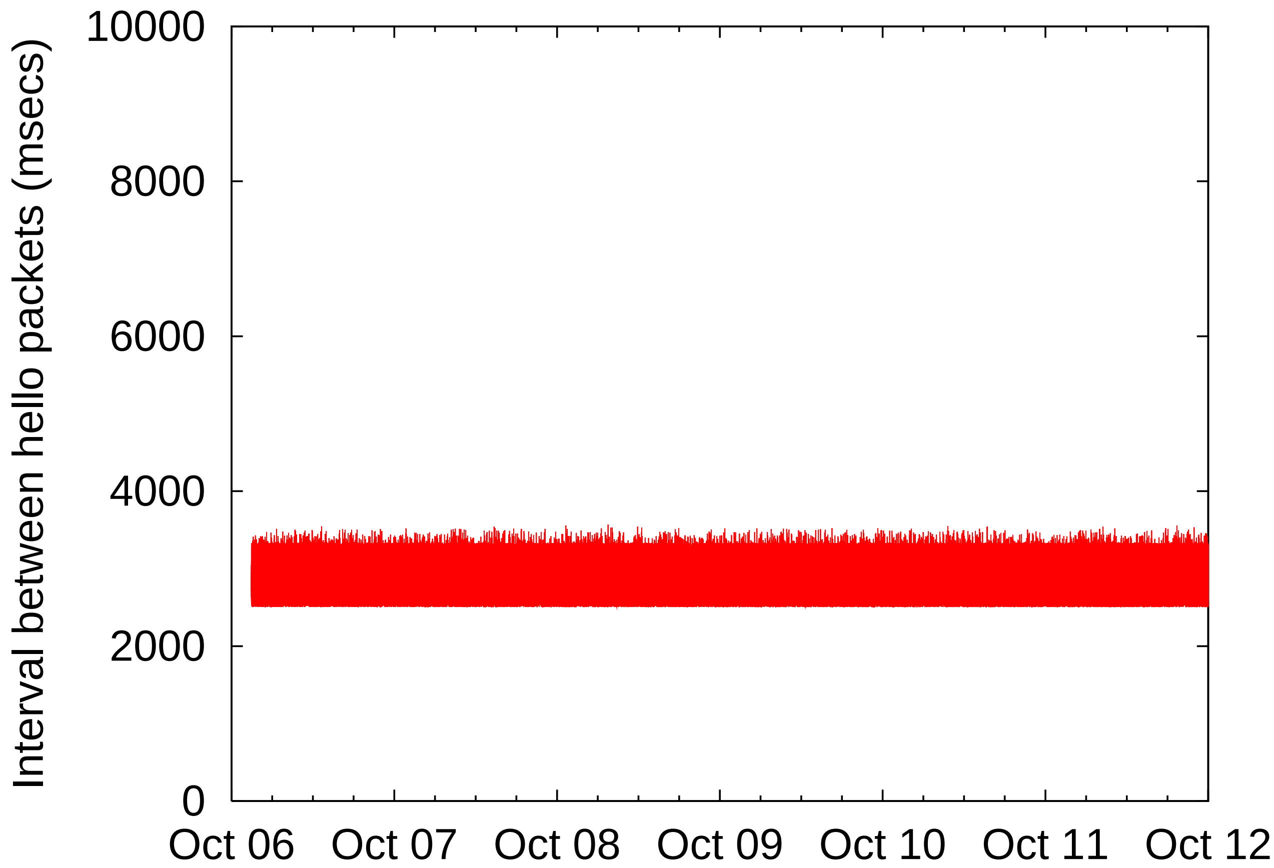


- 6 hops

ISIS Churn and Instability

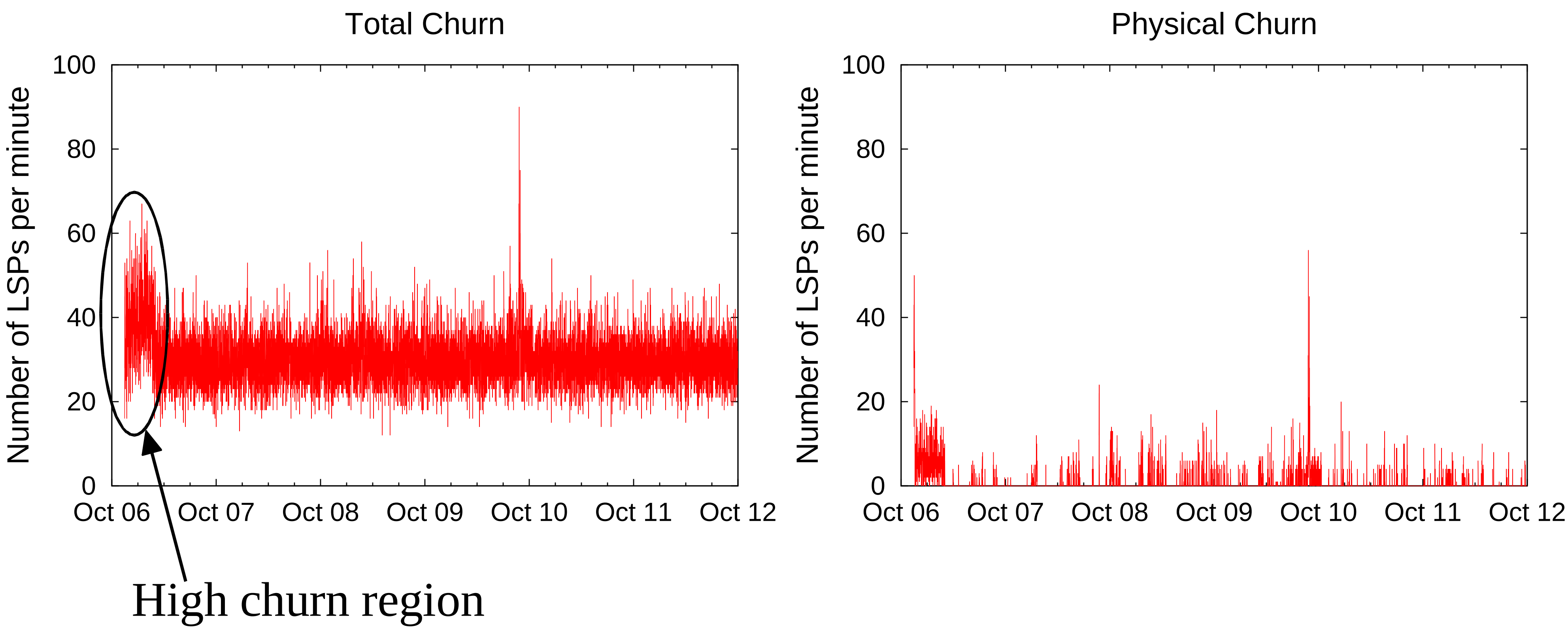
- Churn: number of LSPs received over a time period
  - requires SPF calculations
  - consumes CPU resources
- Busy CPU may cause HELLO packet misses
  - can falsely bring adjacencies down
  - increases churn
- Instability
  - churn => busy CPU => HELLO misses => more churn => ...
  - rate limits are for avoiding this instability

HELLO Packets



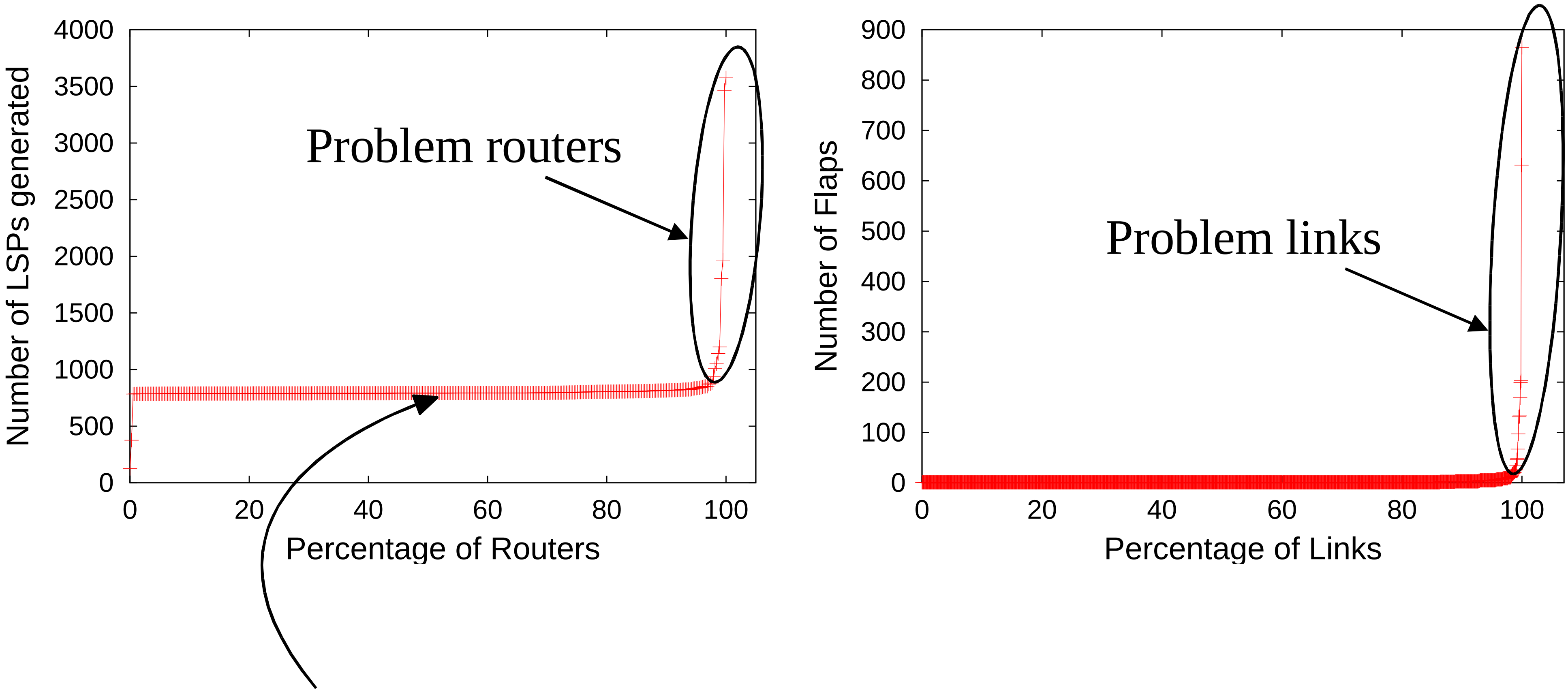
- Excellent HELLO behavior

Churn



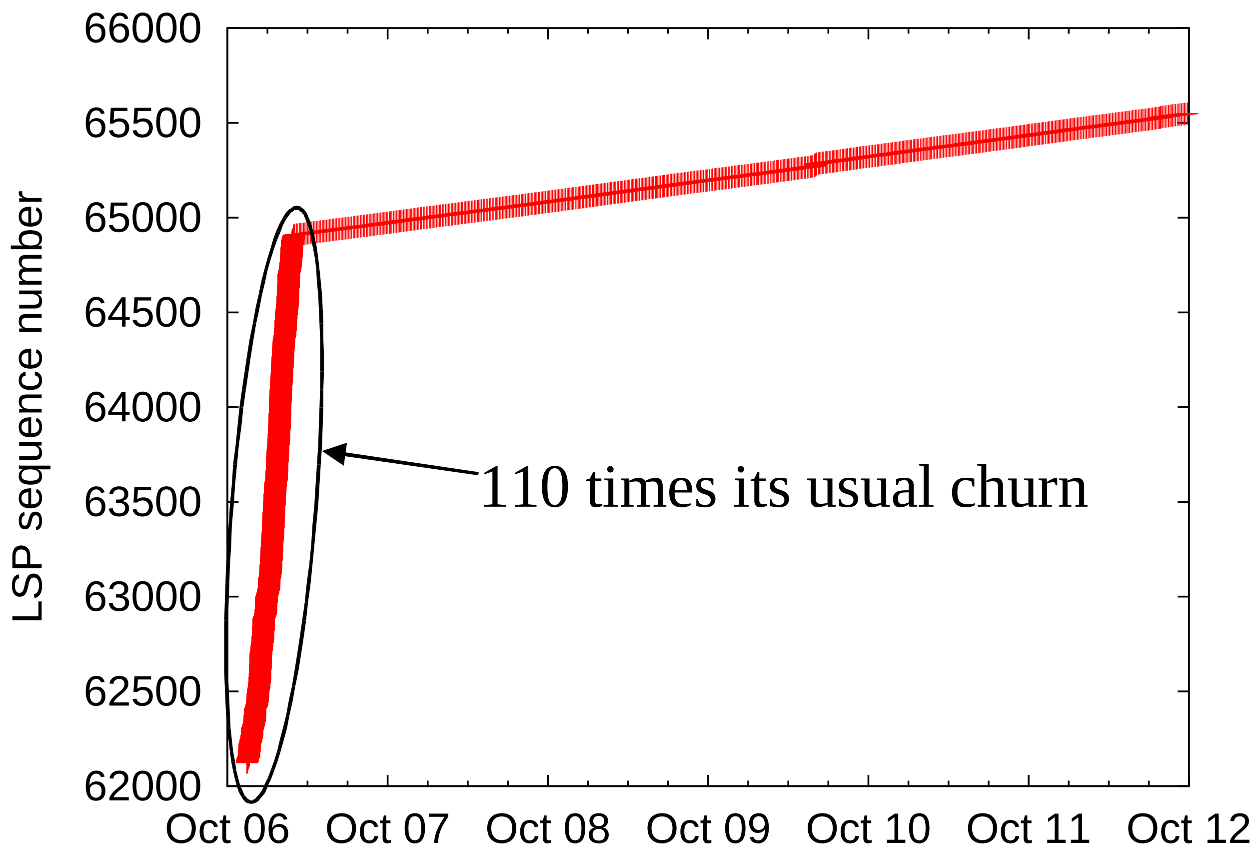
- Very stable network:
  - Average rate for total churn: 1 LSP every 2 seconds
  - 97.6% of the LSPs are state refreshes

Churn by Routers & Links



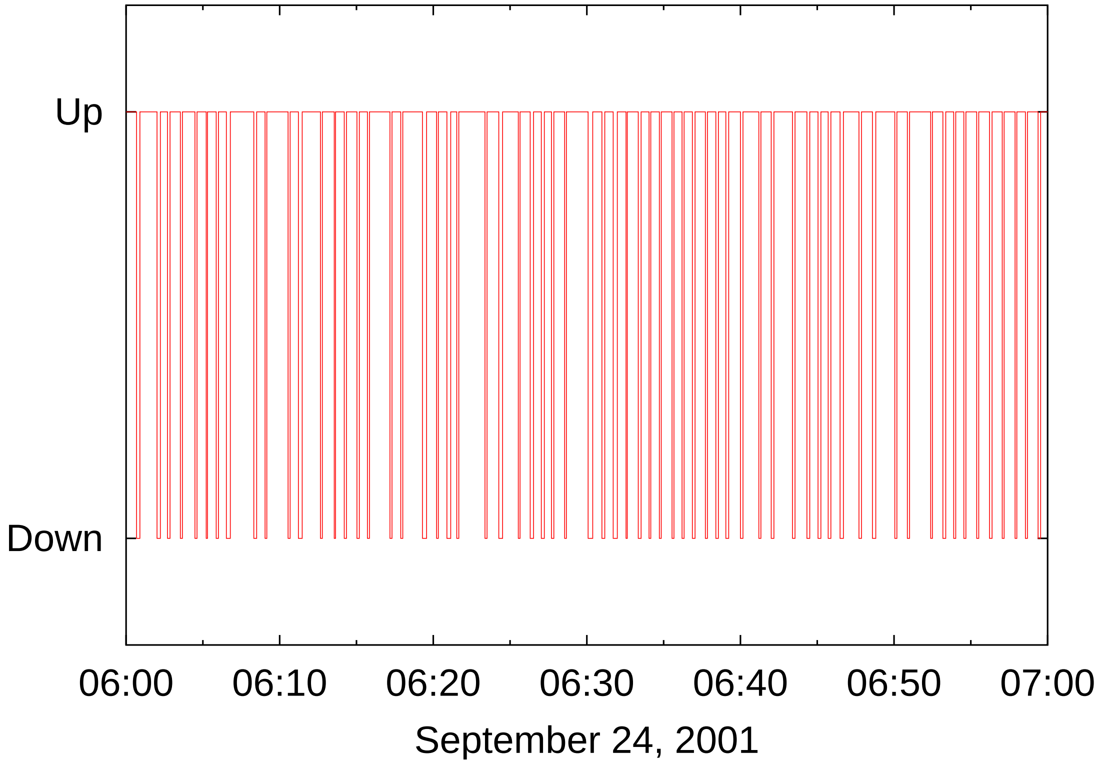
- About 800 LSPs per week per router is for refreshes
  - This can be configured to be less

One Atypical Router



- This router is responsible for the initial high churn

An Unstable Link

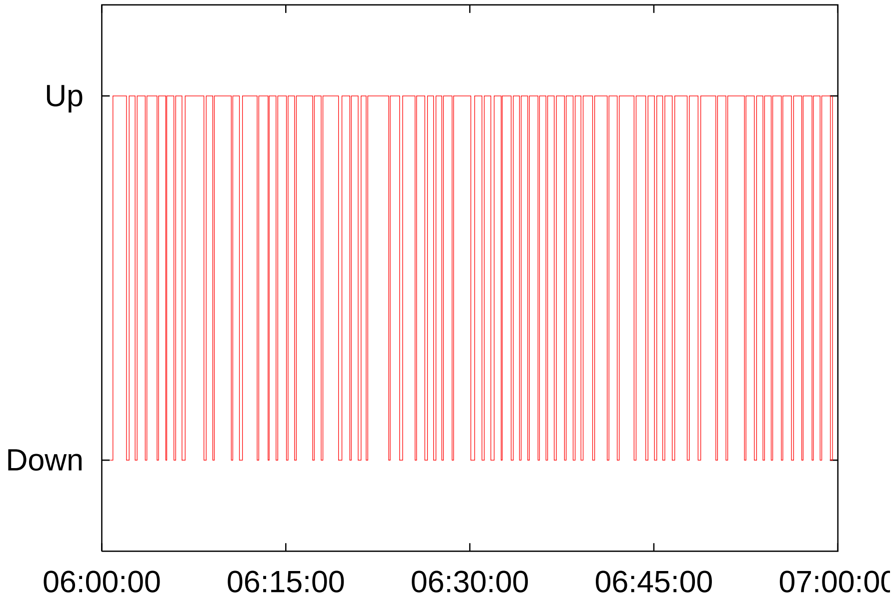


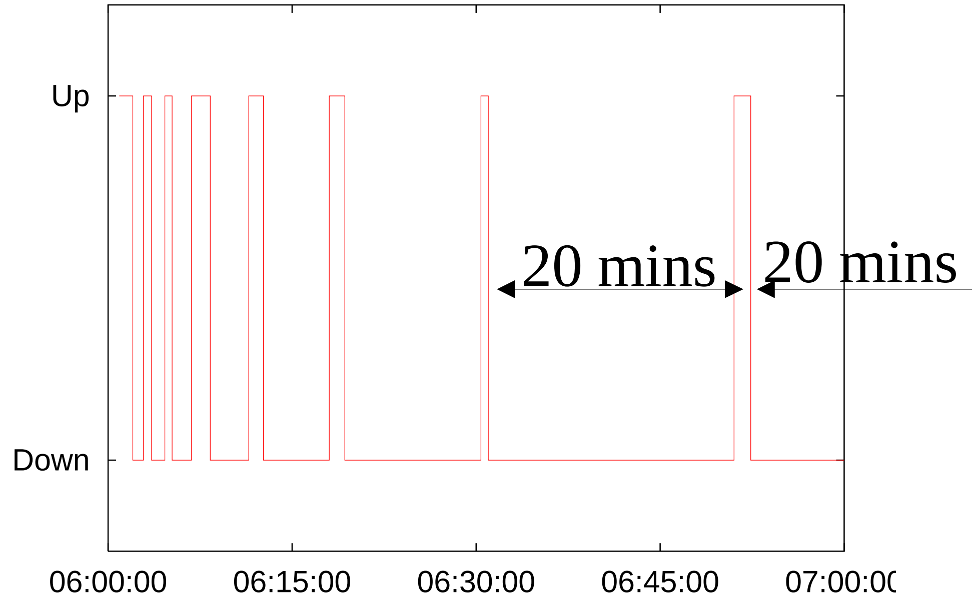
- No protection against instability
  - Goes on for a day
- Opposite of fast convergence requirement
  - 30 seconds to go down, 8 seconds to go up

Dealing with Instability

- SPF & LSP propagation rate limits don't reduce churn
    - does keep CPU from melting by ignoring change
- To reduce churn without impacting convergence:
  - Asymetric up/down filters for fast convergence
    - detect bad news fast
    - slow down on good news
  - Adaptive filters
    - linear or exponential adaptation to level of instability
  - Less CPU intensive incremental SPF algorithms

An Example Adaptive Filter





- An example exponential filter with 20 minute max penalty
- *It reduces the churn without hurting convergence*

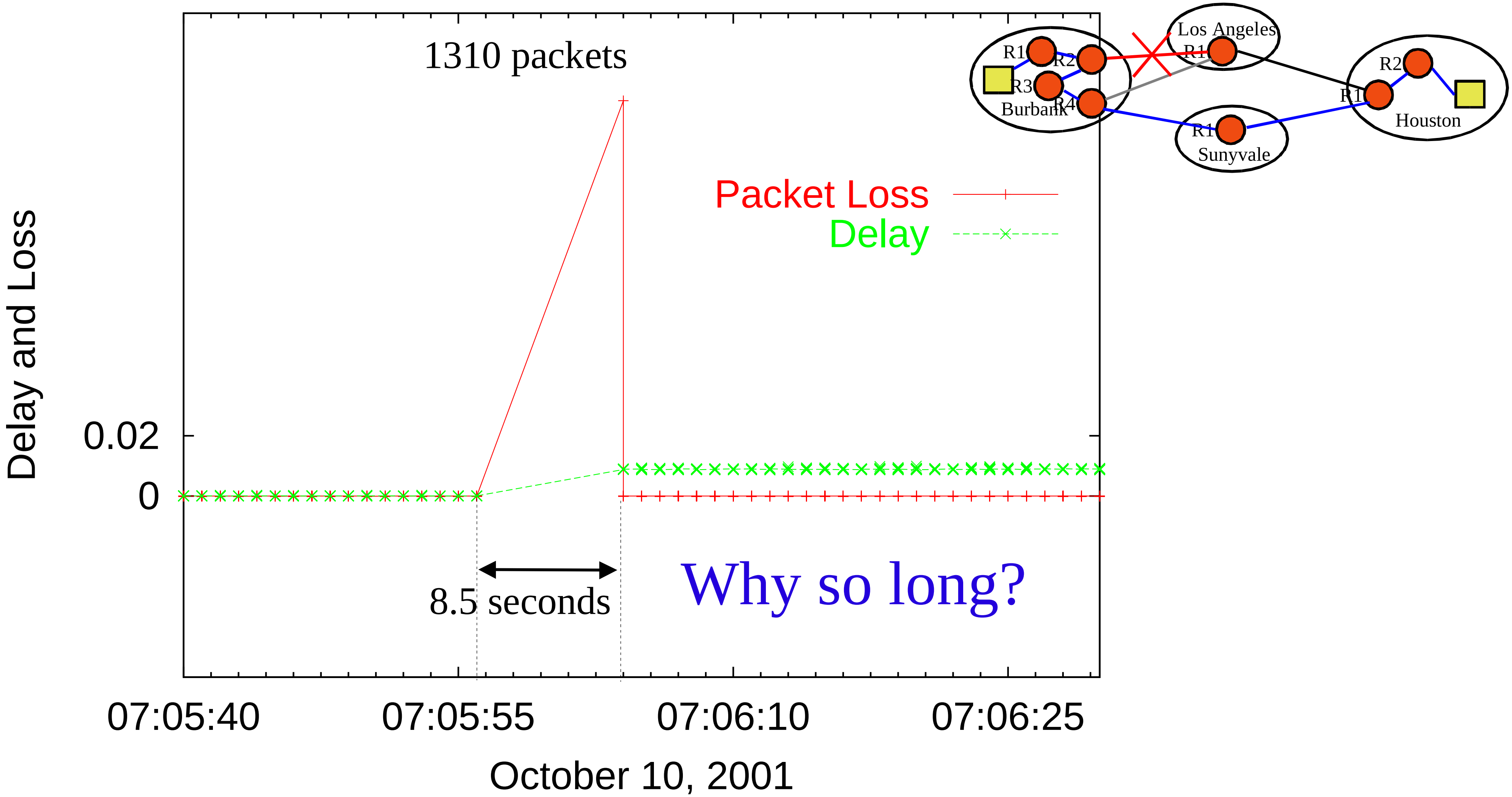
ISIS Convergence Delay

- Time from the physical change to new routing tables
  - Failure/repair detection
  - LSP propagation
  - Delay due to SPF−interval
  - SPF computation

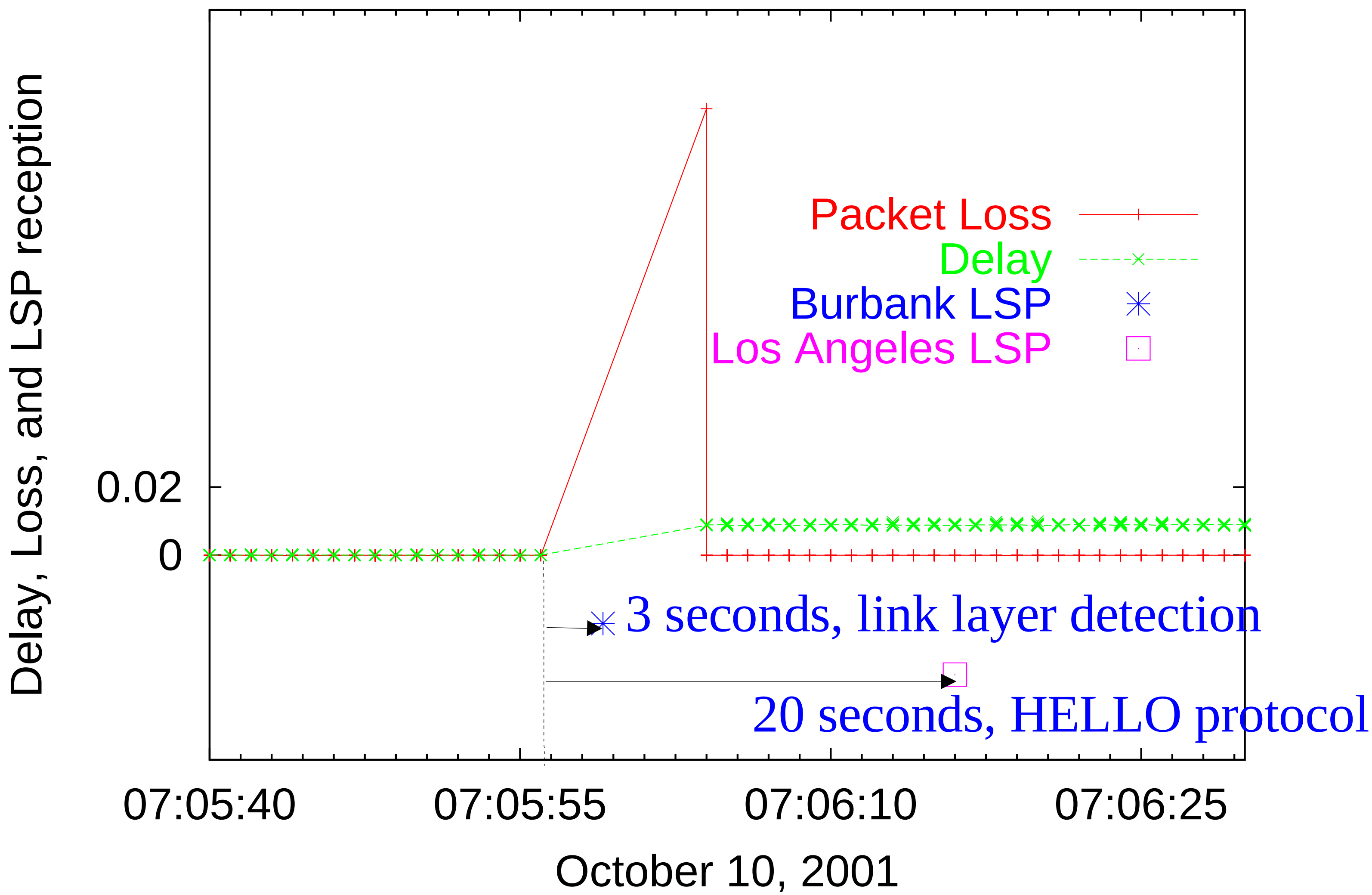
What Happens During Convergence?

- Routers perform SPF while their views of the network are not consistent, causing:
  - routing loops
  - black hole routes
  - suboptimal routes
- If fast convergence, this is not an issue. But,
  - Convergence times are not fast
  - On high speed links, lots of packets are affected
  - New services are less tolerant

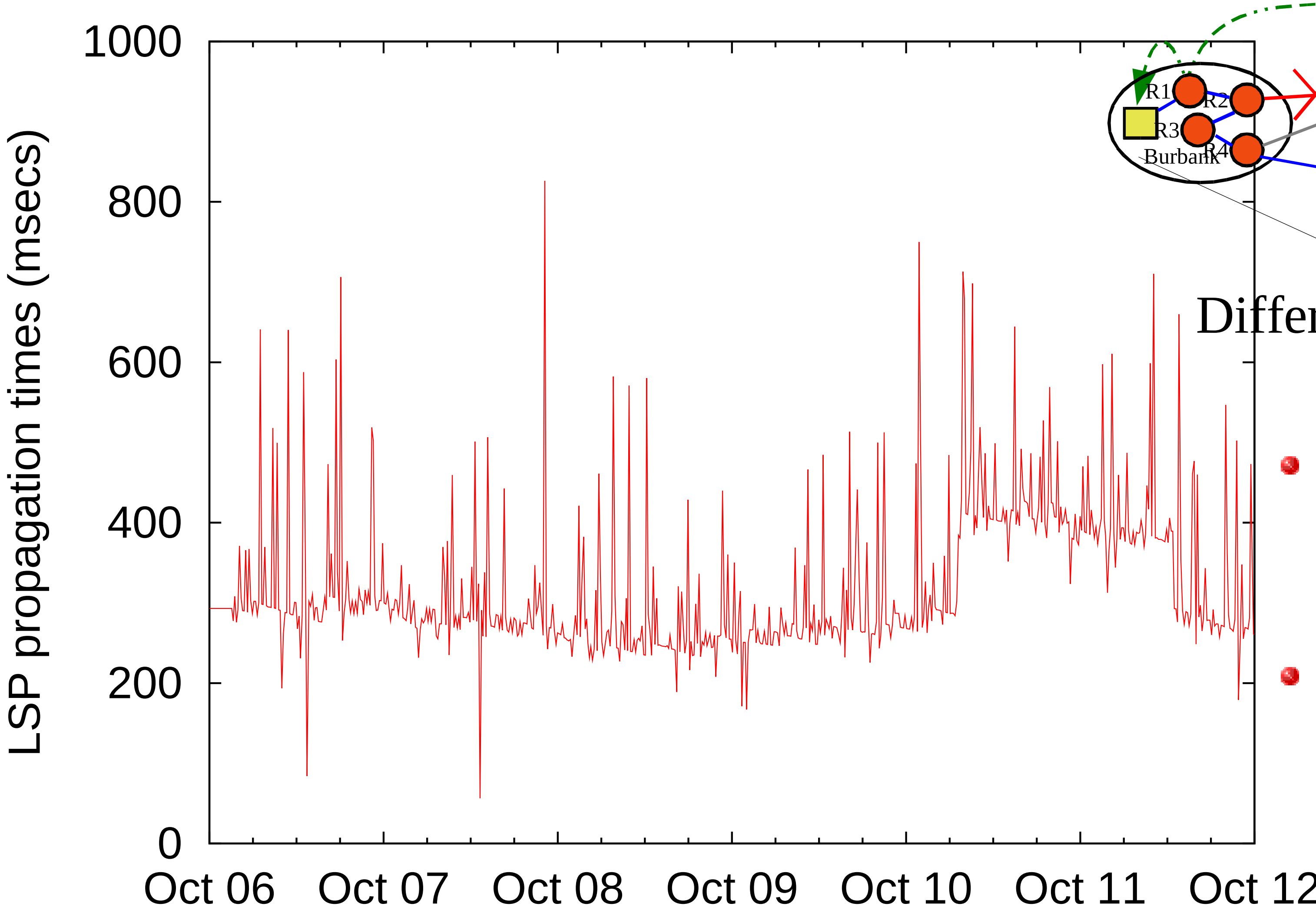
A Link Failure

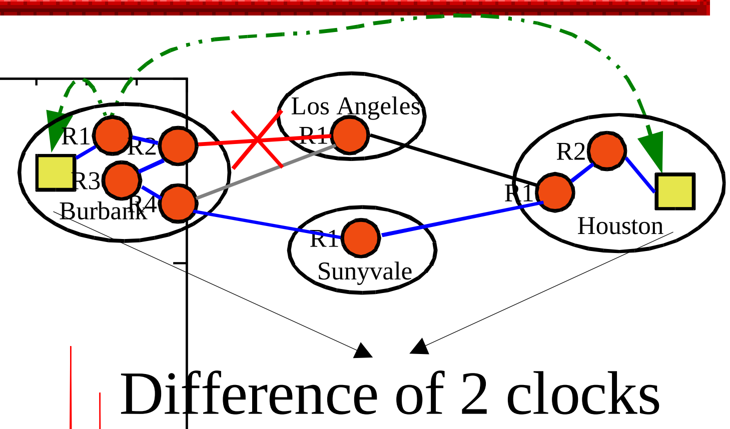


Slow Link Failure Detection



LSP Propagation Delay





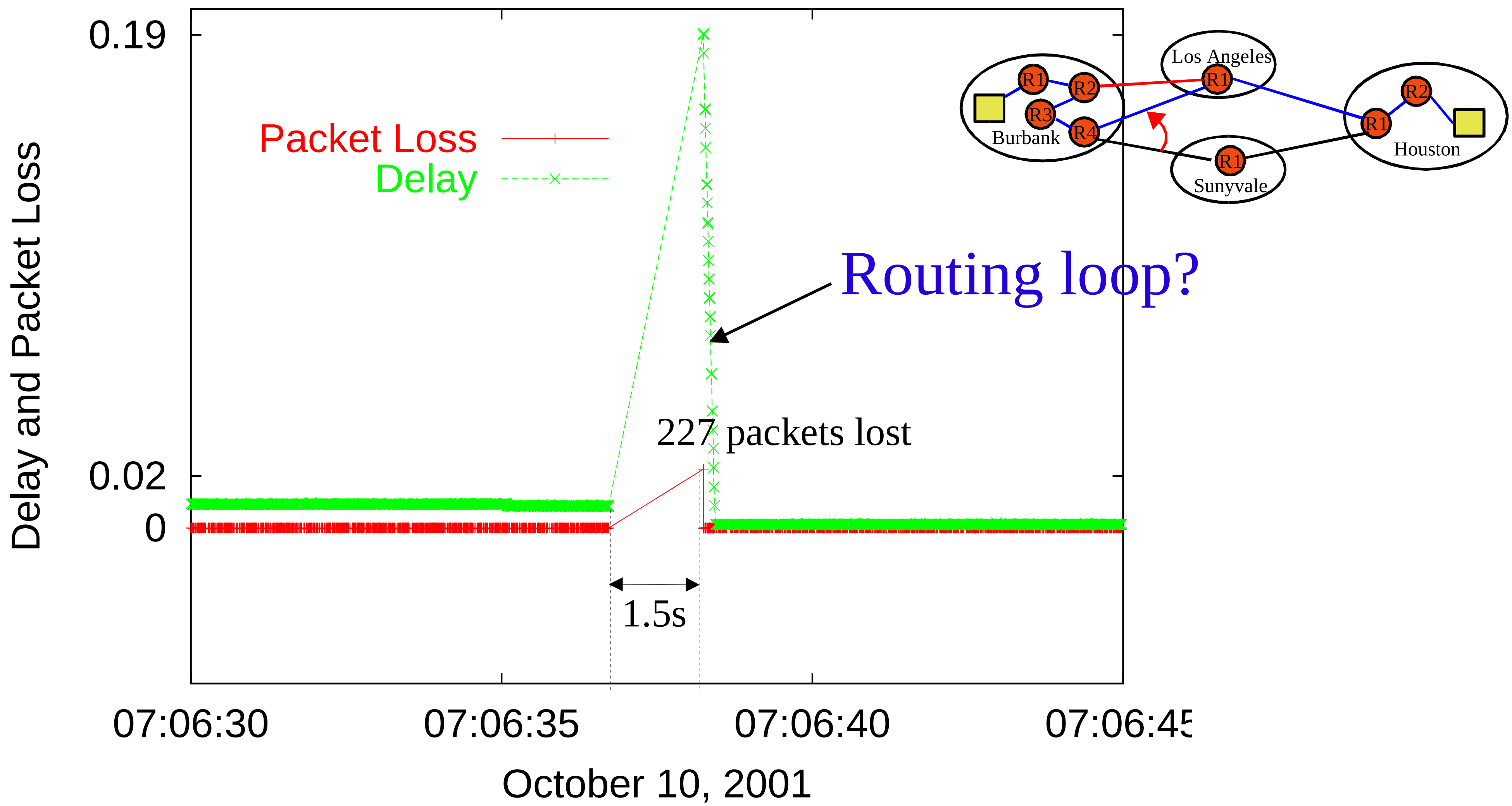
- LSP propagations are rate limited
- Their scheduling may be improved

SPF Rate Limiting

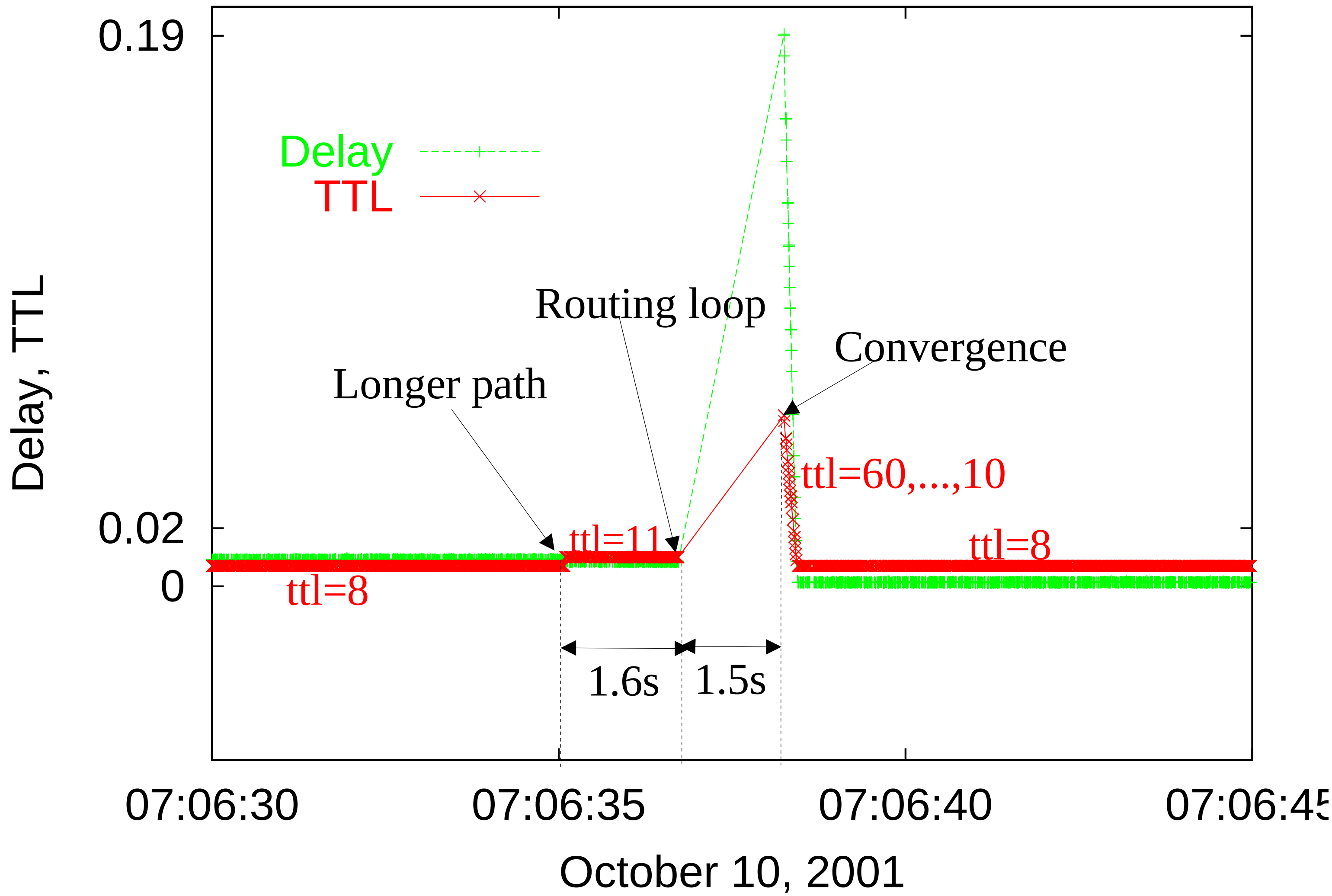
- spf−interval parameter delays SPF computation
  - default: SPF computation after 5 seconds from the change
  - visible in our convergence delay example

- Two goals:
  - to contain the CPU load
    - no more than one SPF computation per 5 seconds
  - to capture 2−4 LSPs reporting the failure in one SPF run
    - fails to do this in our case

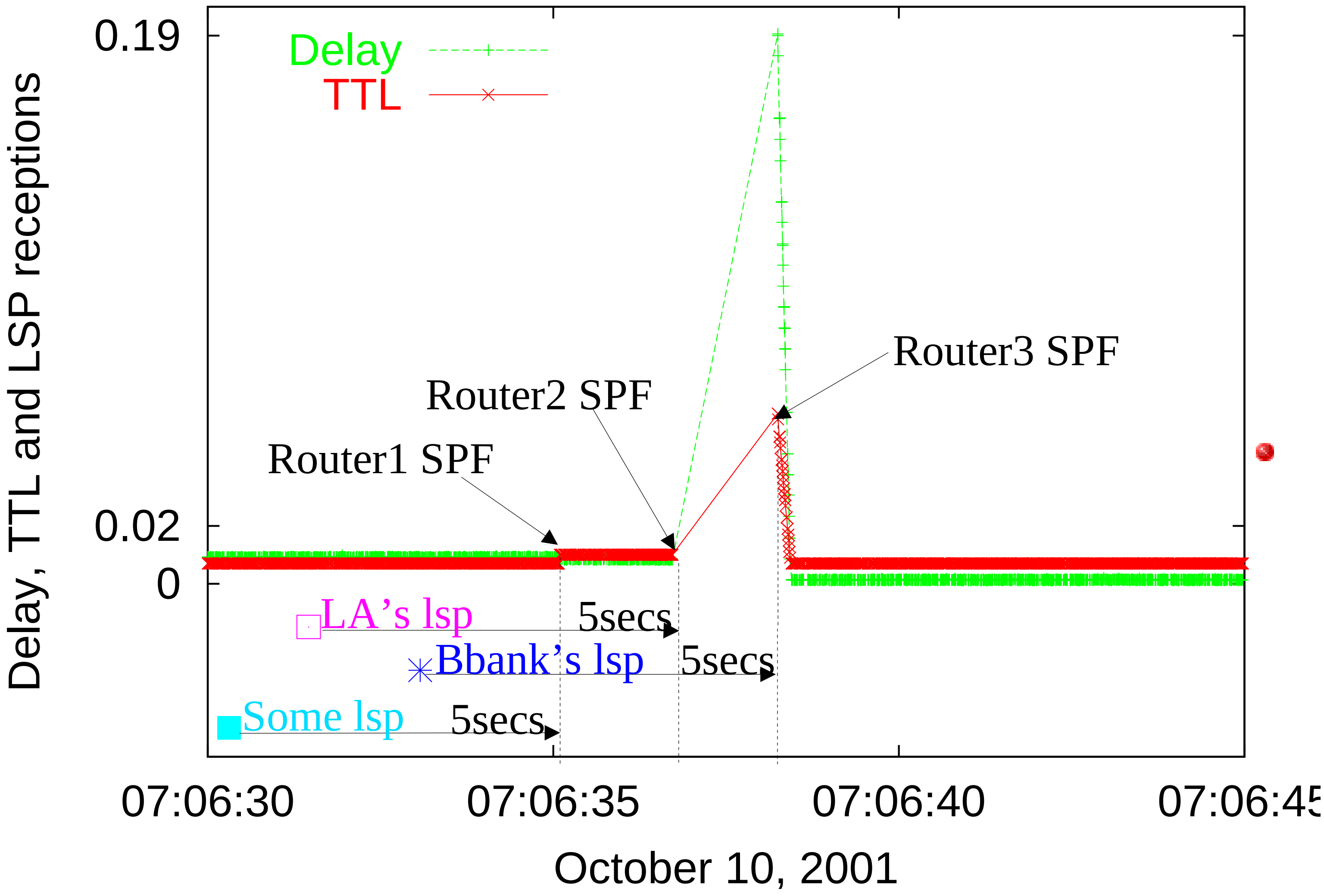
Why Loss and Delay at Link Repair?



TTLs Confirm the Routing Loop

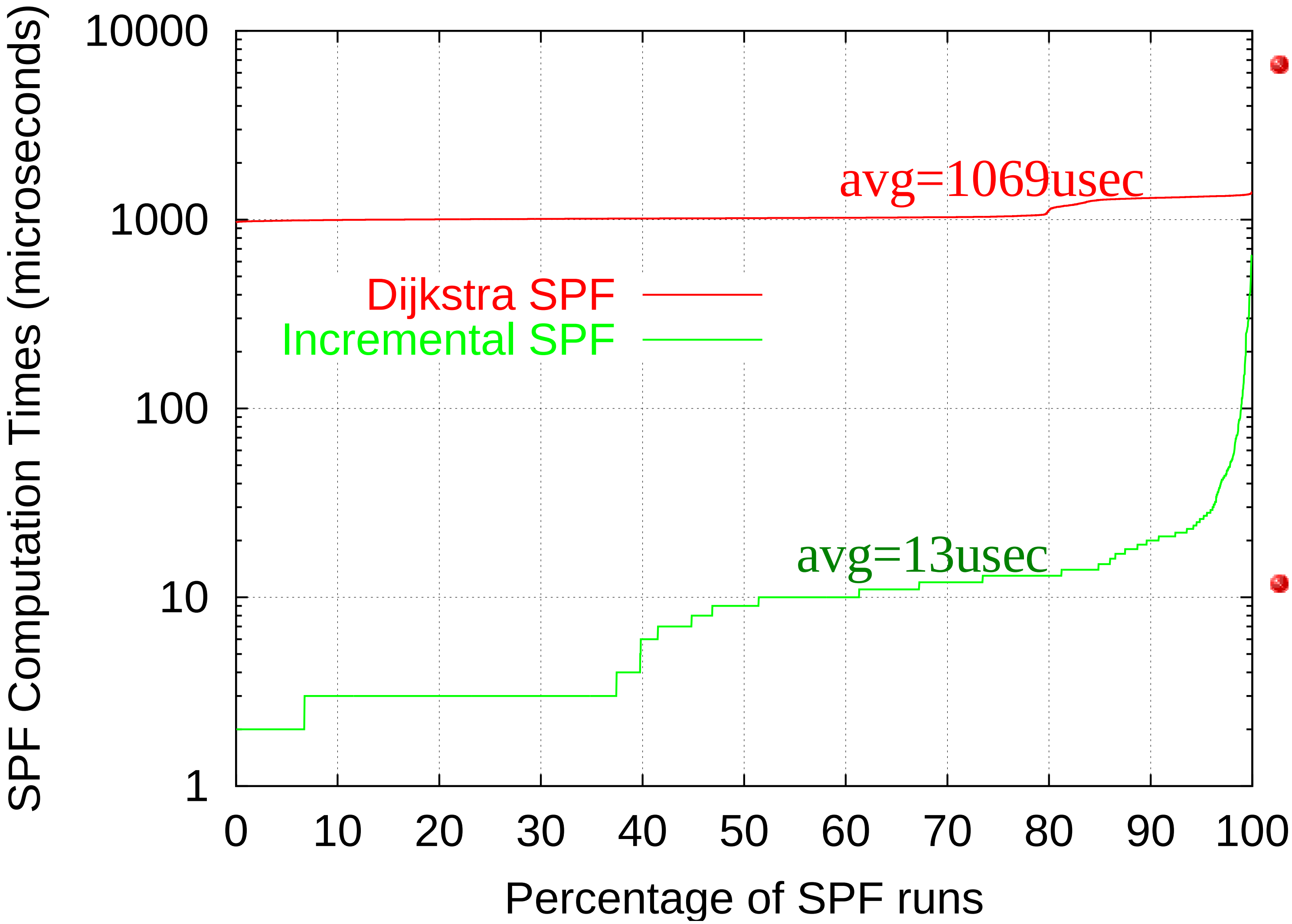


spf−interval Spreads SPFs



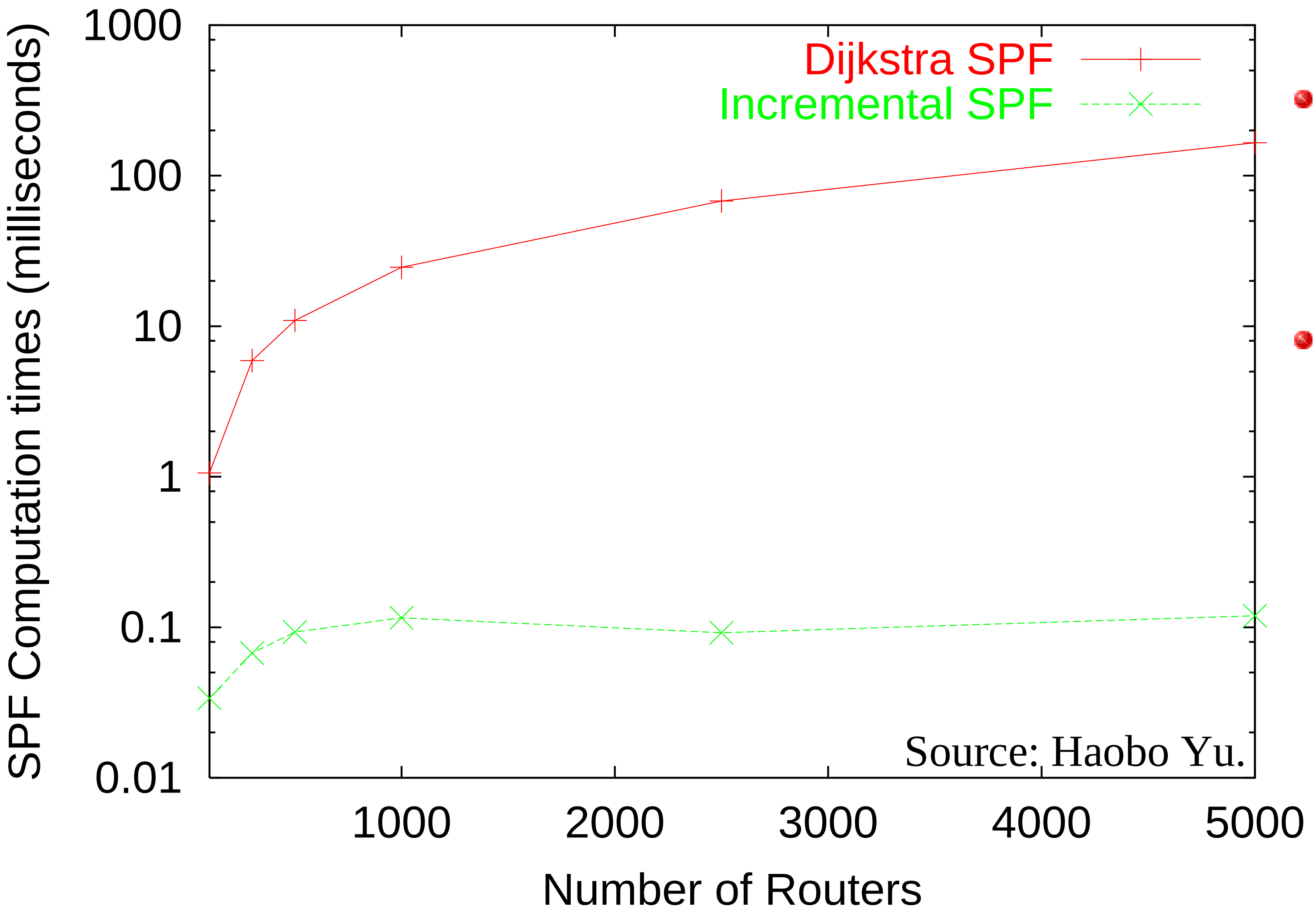
- 3 routers start timers after 3 different LSPs

SPF Computation Times



- Qwest topology and events using equal−cost multiple paths
- On average 84 times faster

SPF Scaling



- Using random topology and events
- Incremental SPF lets the network grow very large

Where does the time go?

- Detection times were several seconds, must be improved
- LSP Propagation times were subsecond, but still much larger than link propagation delays
- SPF rate limiting
  - spf−interval causes most of the convergence delay
  - spf−interval spreads SPFs, groups wrong set of LSPs into the same SPF

A Recipe for
Subsecond ISIS Convergence

Vendors: fast link failure detection

- Hardware detection is preferred
- Vendors have fast failure detection solution for MPLS fast reroute
- It will benefit convergence immediately

Vendors: adaptive and asymmetric Up/Down filters

- It will reduce the ISIS churn w/o hurting convergence

Operators: eliminate current LSP & SPF rate limits

- Adaptive asymmetric filters make it safe

Vendors: incremental SPF algorithm

- A must for avoiding CPU meltdowns even as the network gets bigger

Acknowledgements

- We would like to thank Chris Sieber, Paul Mabey and Shankar Rao of Qwest for providing access to their network for our study and for their review of our results.
- We would also like to thank Haobo Yu, Van Jacobson, Kathleen Nichols, and Kedar Poduri of Packet Design for their contributions to this work.