Active BGP Measurement with BGP-Mux

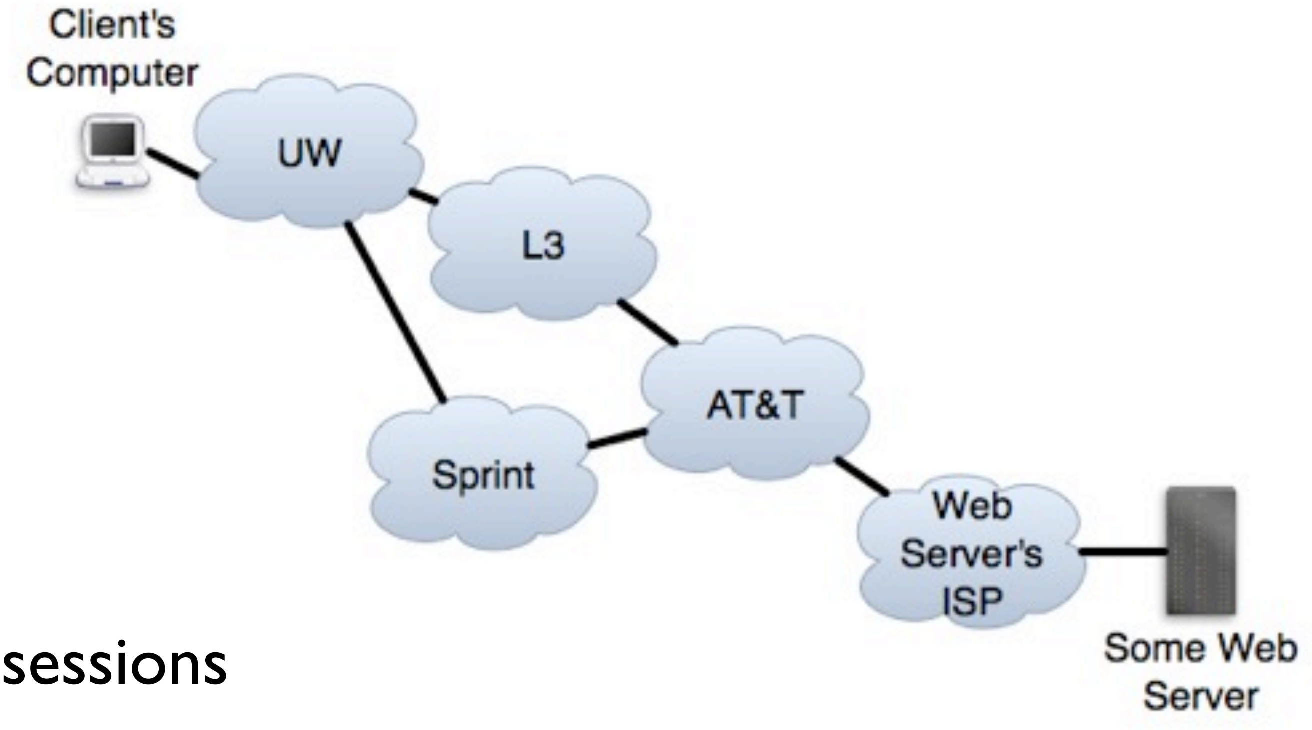
Ethan Katz-Bassett (USC)
with testbed and some slides hijacked from
Nick Feamster and Valas Valancius

Before I Start

- Georgia Tech system, I am just an enthusiastic user
    - Nick Feamster and his students:
    - Valas Valancius
    - Bharath Ravi

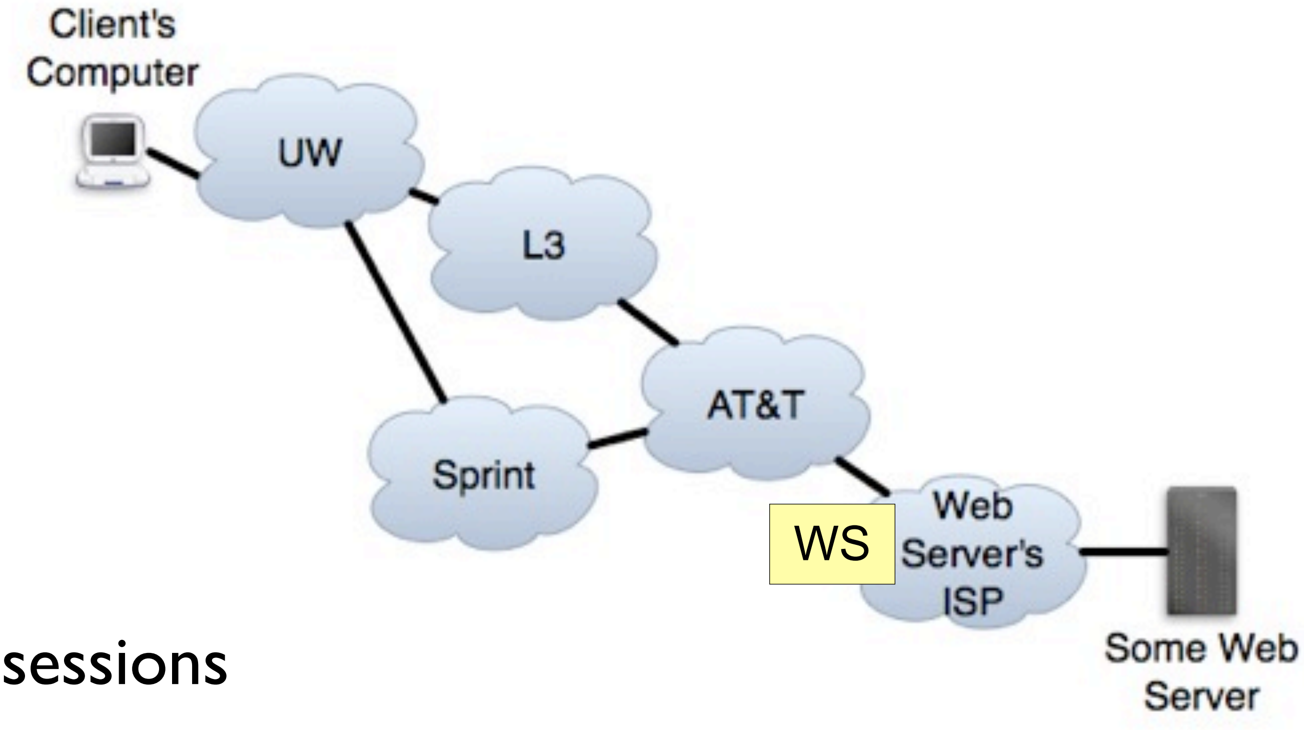
- Questions for the audience:
    - What would you use this system for? What should we use it for?
    - How do we get more ASes to connect to us?
        - Getting them to agree to peer
        - Then, getting the connection to work

Networks Use BGP to Interconnect



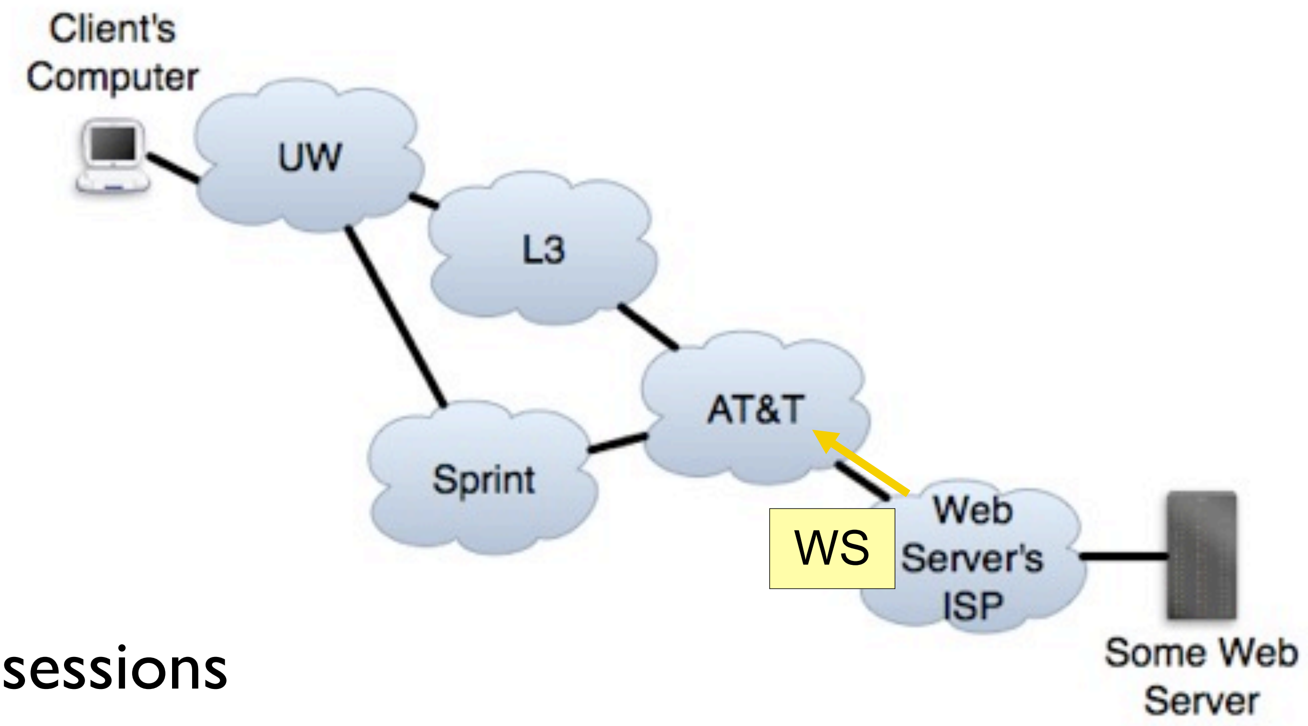
- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



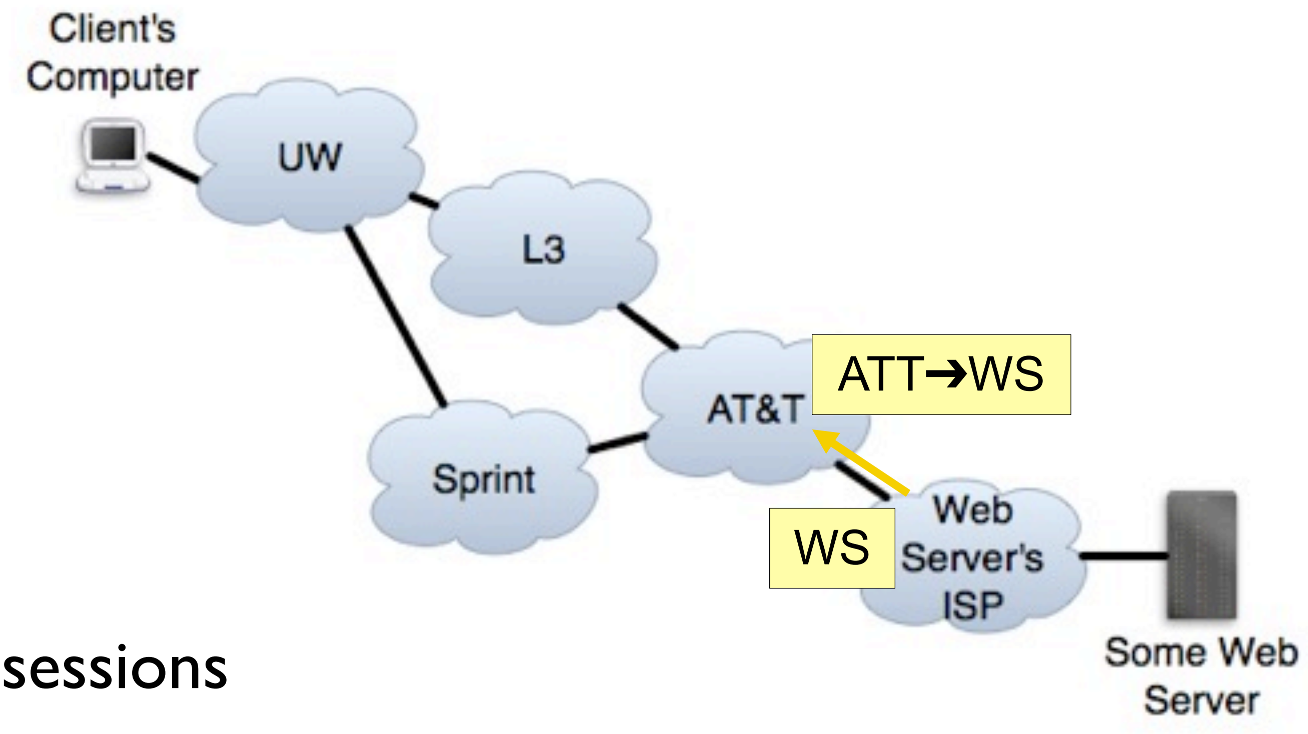
- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



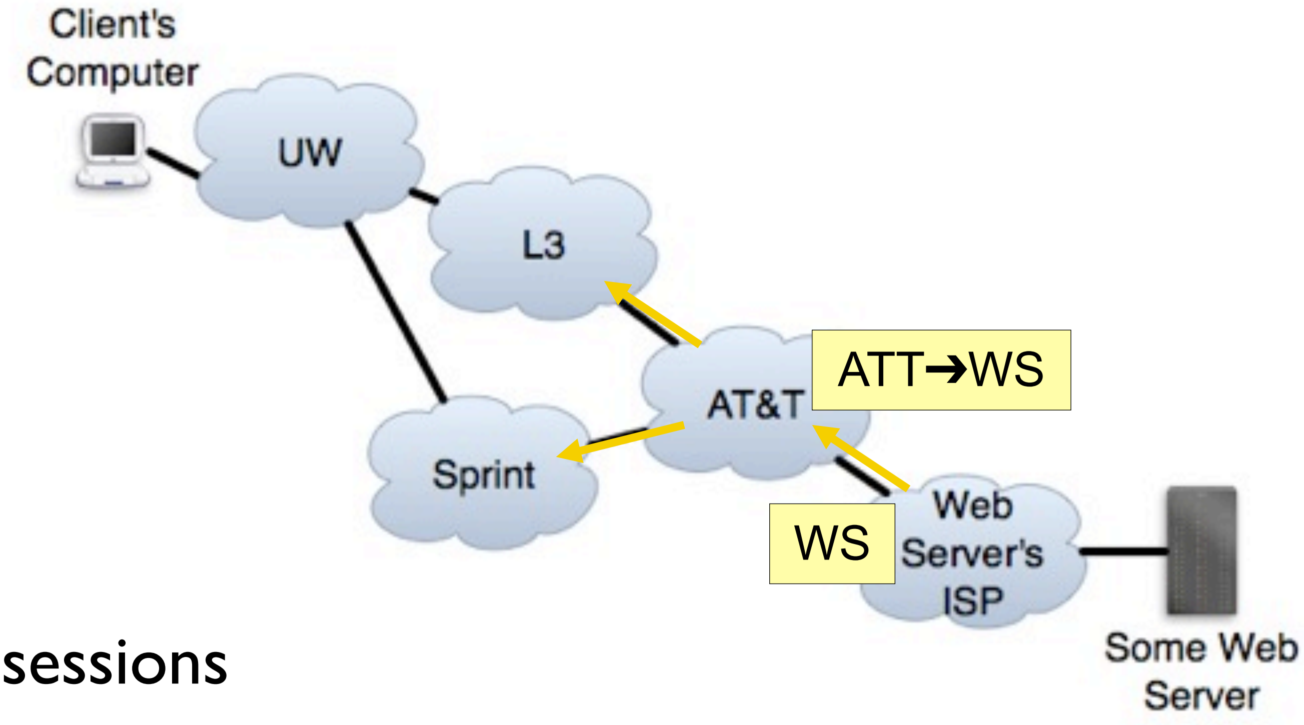
- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



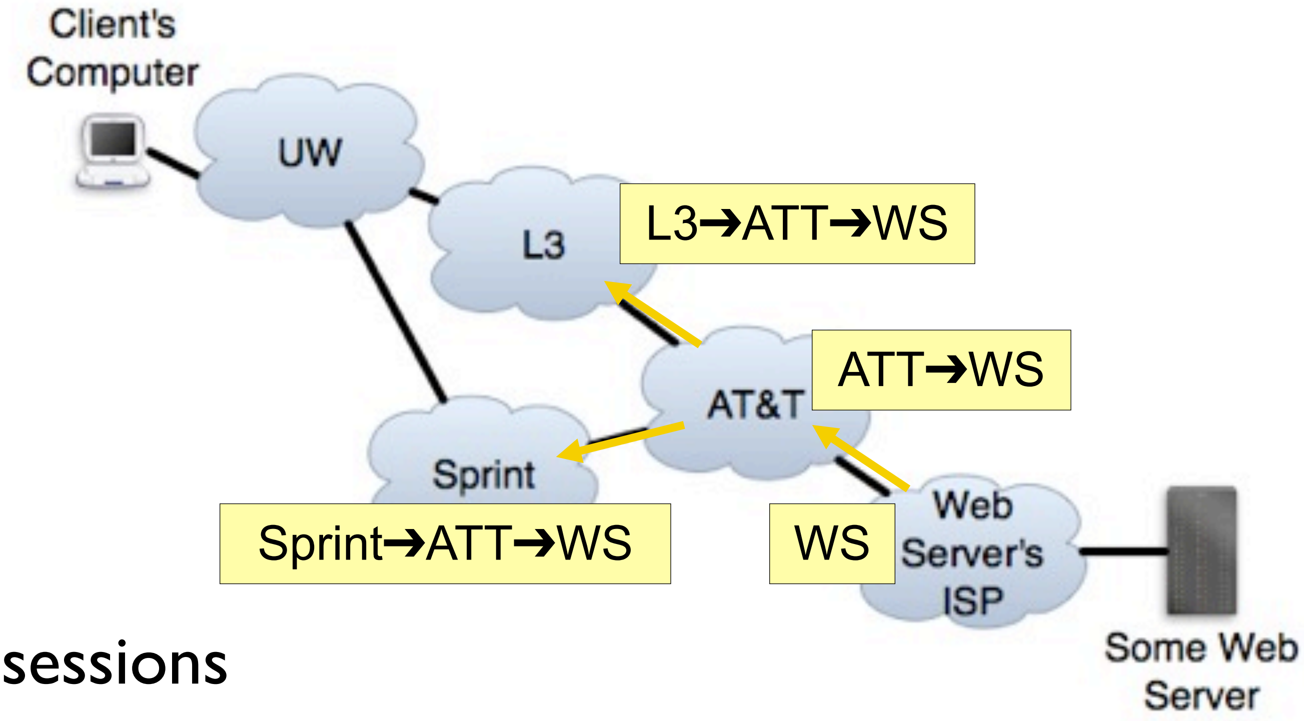
- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



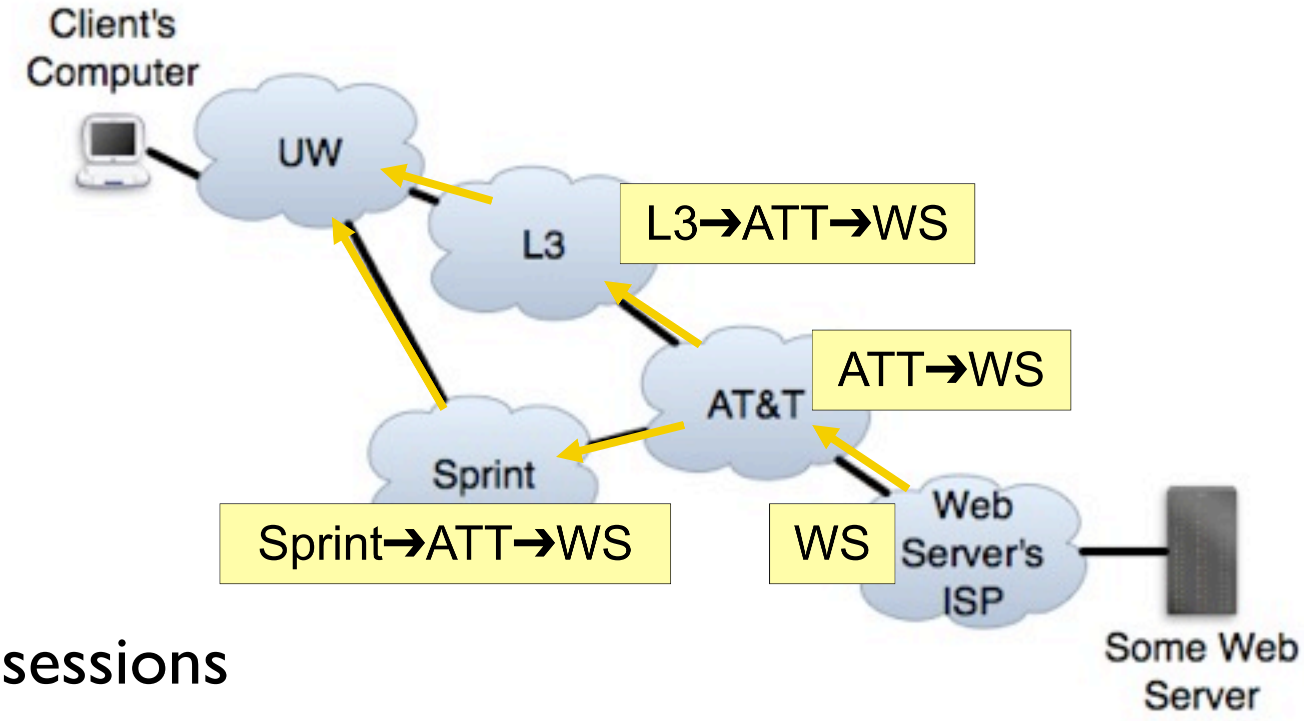
- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



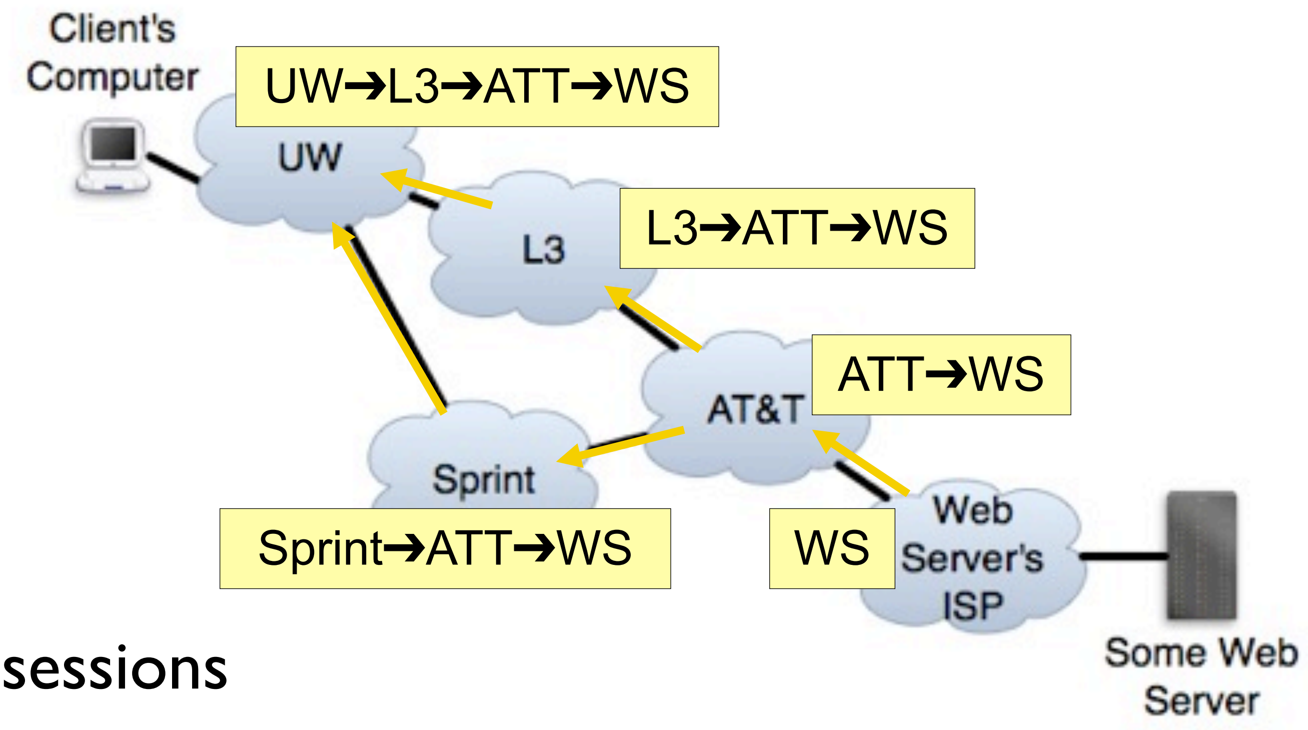
- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



UW➔L3➔ATT➔WS

L3➔ATT➔WS

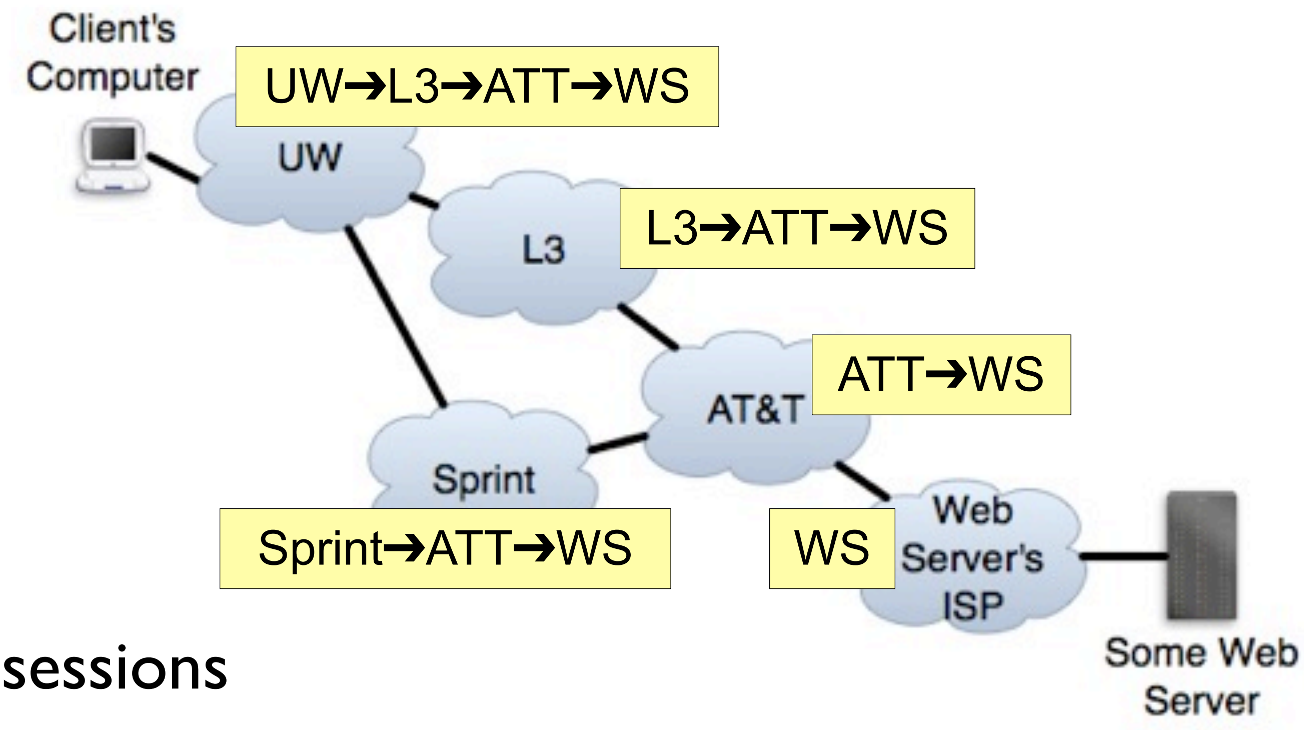
ATT➔WS

Sprint➔ATT➔WS

WS

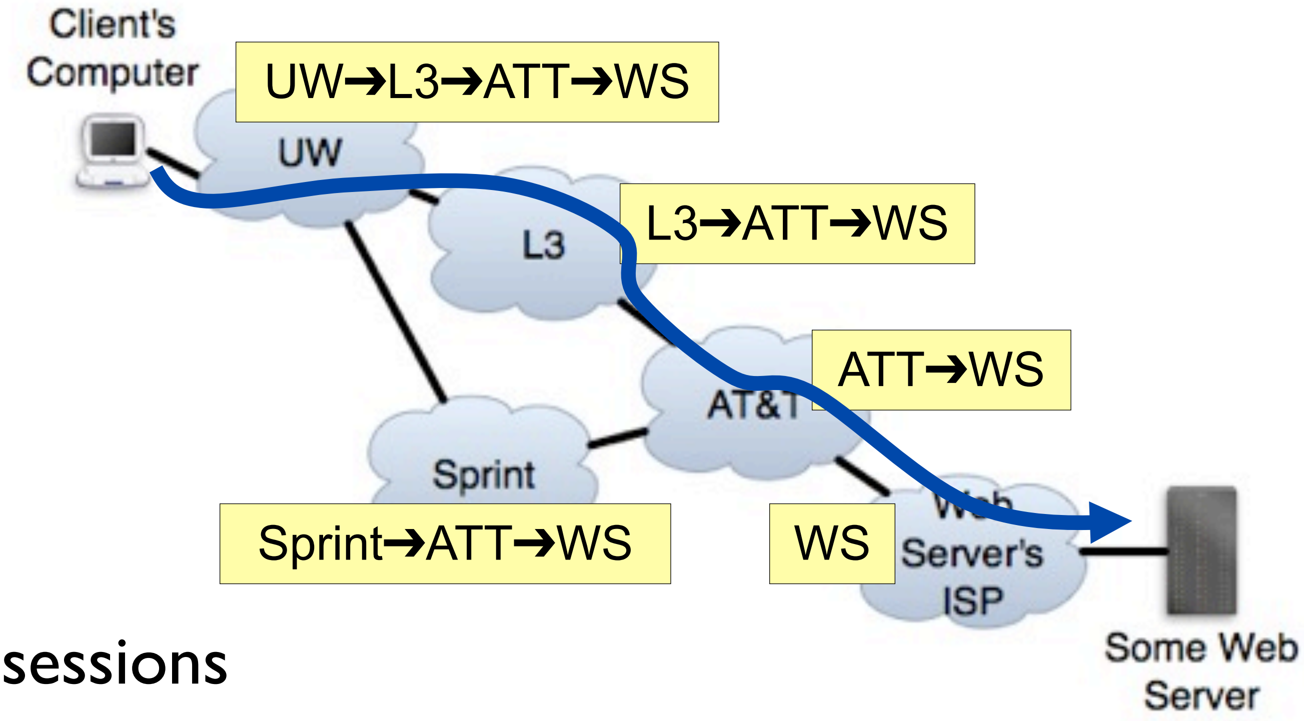
- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Networks Use BGP to Interconnect



- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic

Virtual Networks Need BGP, too

Say I have some neat new routing ideas. I want to test them:

- Emulate the type of AS (CDN, stub, etc) of my choice
  - Choose a set of providers, peers, and customers
- Inbound:
  - Choose routes from those providers
  - Send traffic along those routes
- Outbound:
  - Announce my prefix(es) to neighbors of choice, with communities, etc
  - Receive traffic to prefix(es)
- And everyone else should be able to do this, also

Traditionally, BGP Experiments are Hard

I have some neat new routing ideas. How do I test them?

- Passive observation
  - E.g., RouteViews, RIPE
  - Receive feeds only
- Limited "active" measurements
  - E.g., Beacons
  - Generally, regular announcements and withdrawals
- Know the right people
  - Negotiate the ability to make announcements
  - High overhead, limited deployment

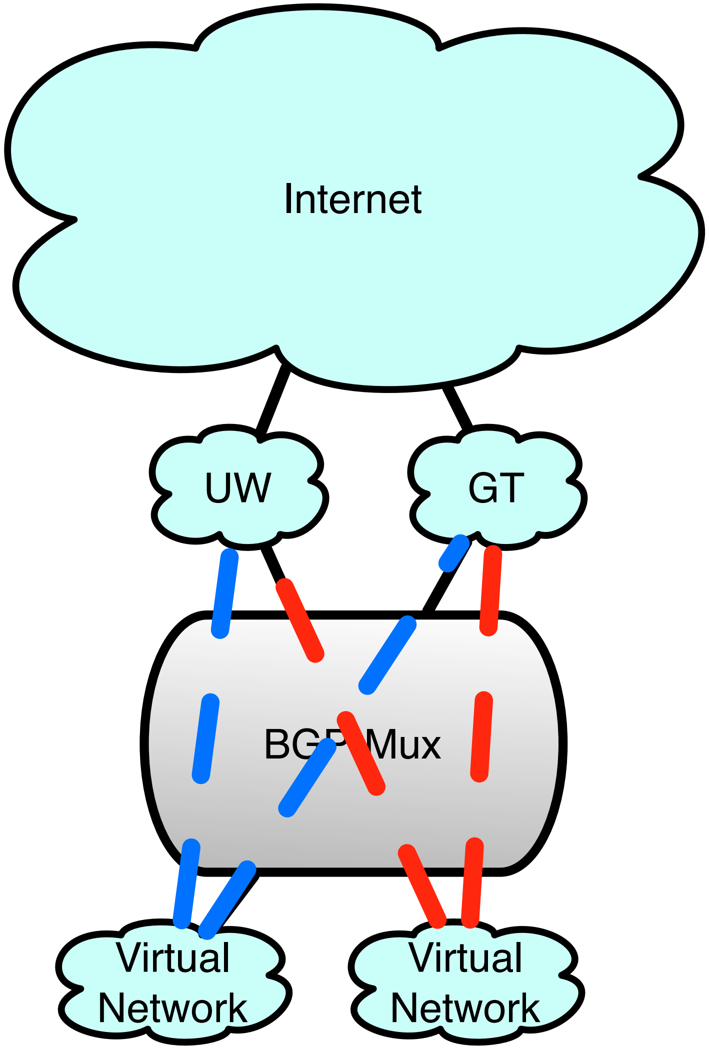
All limit what you can do

What I Need to Get What I Want

- Resources
  - IP address space
  - AS number
- Connectivity & contracts
  - BGP peering with real ASes
  - Data plane forwarding
- Time and money

BGP-Mux Provides All This For You

- Resources
  - IP address space
    184.164.224.0/19
  - AS number
    AS47065
- Connectivity & contracts
  - BGP peering with real ASes
    5 Universities as providers
  - Data plane forwarding
    Send & receive traffic
- Time and money
  One-time cost



Design Requirements

- **Session transparency**: BGP updates should appear as they would with direction connection
- **Session stability**: Upstreams should not see transient behavior
- **Isolation**: Individual networks should be able to set their own policies, forward independently, etc
- **Scalability**: BGP-Mux should support many networks

A Project Using BGP-Mux

LIFEGUARD: **L**ocating **I**nternet **F**ailures **E**ffectively and **G**enerating **U**sable **A**lternate **R**outes **D**ynamically

- Locate the ISP / link causing the problem

- Suggest that other ISPs reroute around the problem
  - What would we like to add to BGP to enable this?
  - What can we deploy today, using only available protocols and router support?

Our Goal for Failure Avoidance

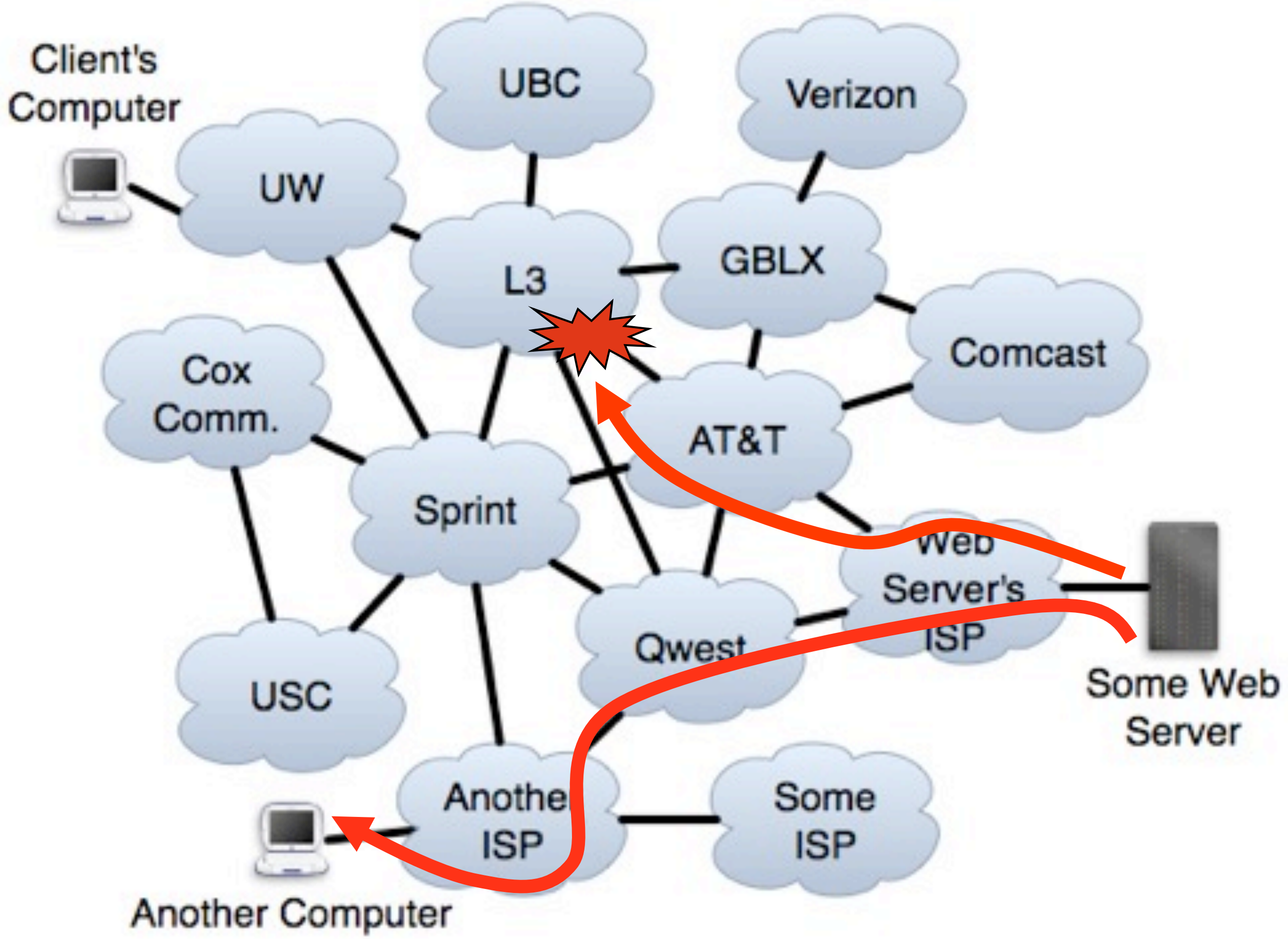
- Enable content / service providers to repair persistent routing problems affecting them, regardless of which ISP is causing them

Setting

- Assume we can locate problem
- Assume we are multi-homed / have multiple data centers
- Assume we speak BGP

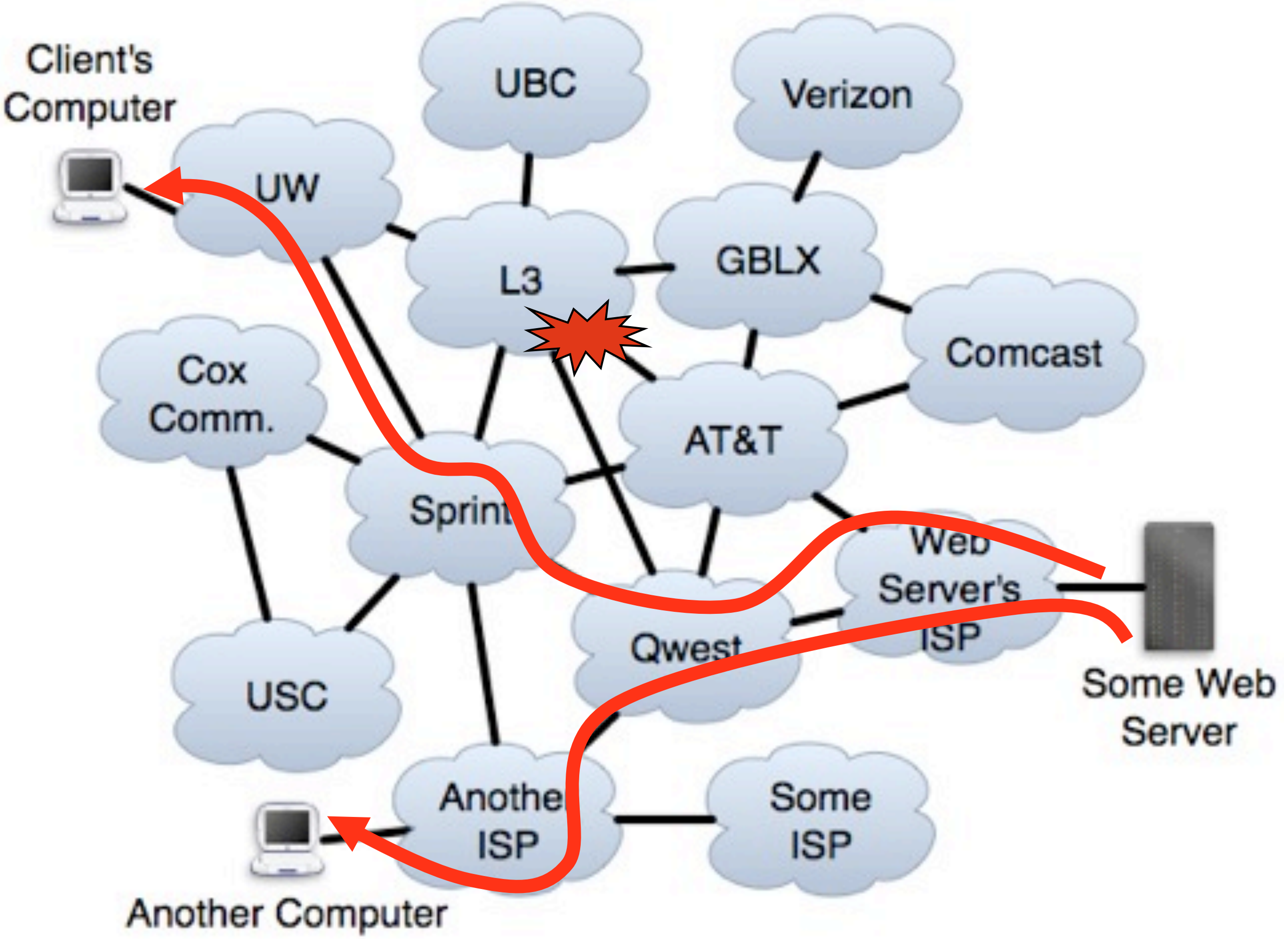
- We use BGP-Mux to speak BGP to the real Internet:
  5 US universities as providers

Self-Repair of Forward Paths



Straightforward: Choose a path that avoids the problem.

Self-Repair of Forward Paths



Straightforward: Choose a path that avoids the problem.

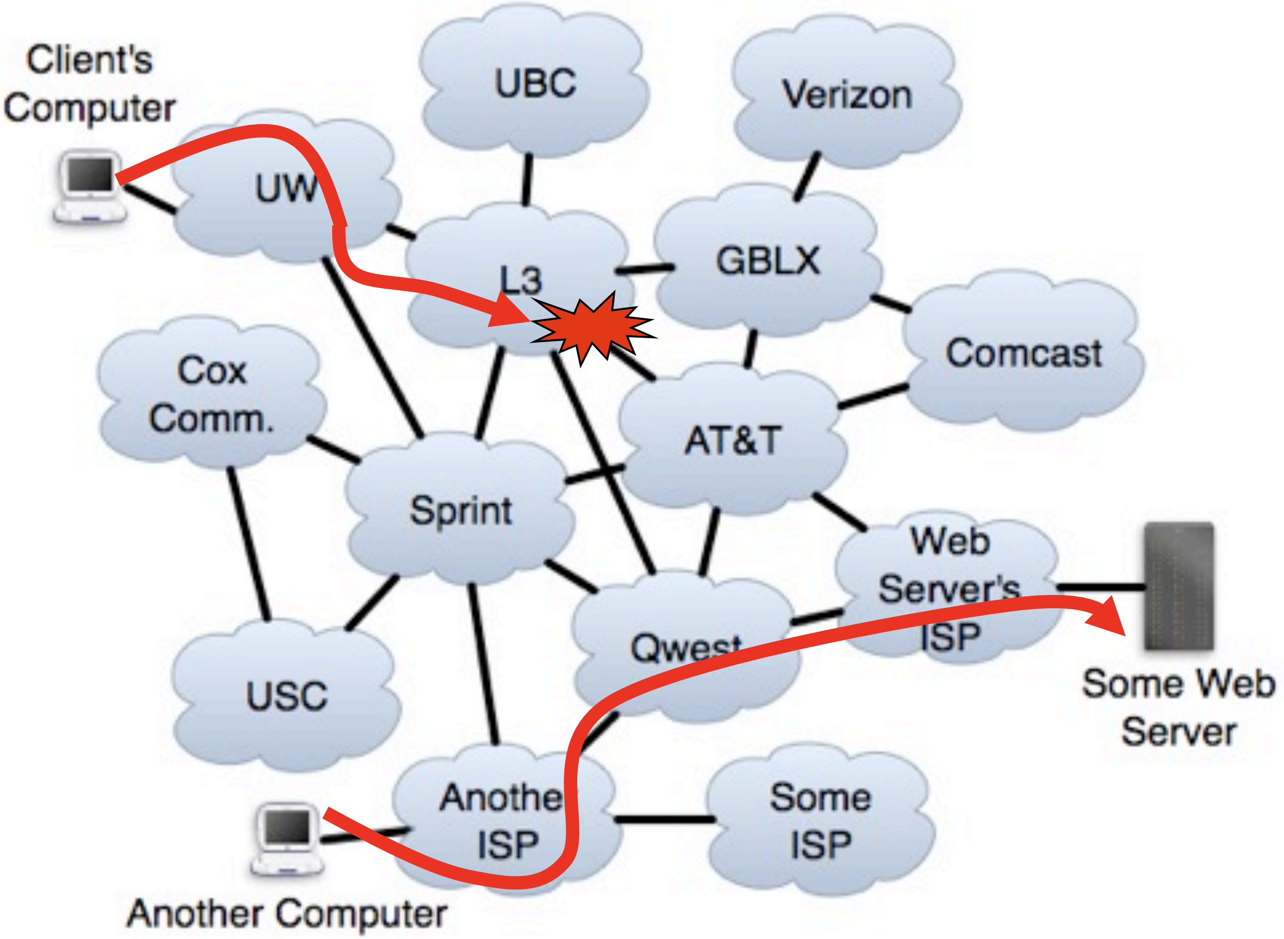
A Mechanism for Failure Avoidance

Forward path: Choose route that avoids ISP or ISP-ISP link

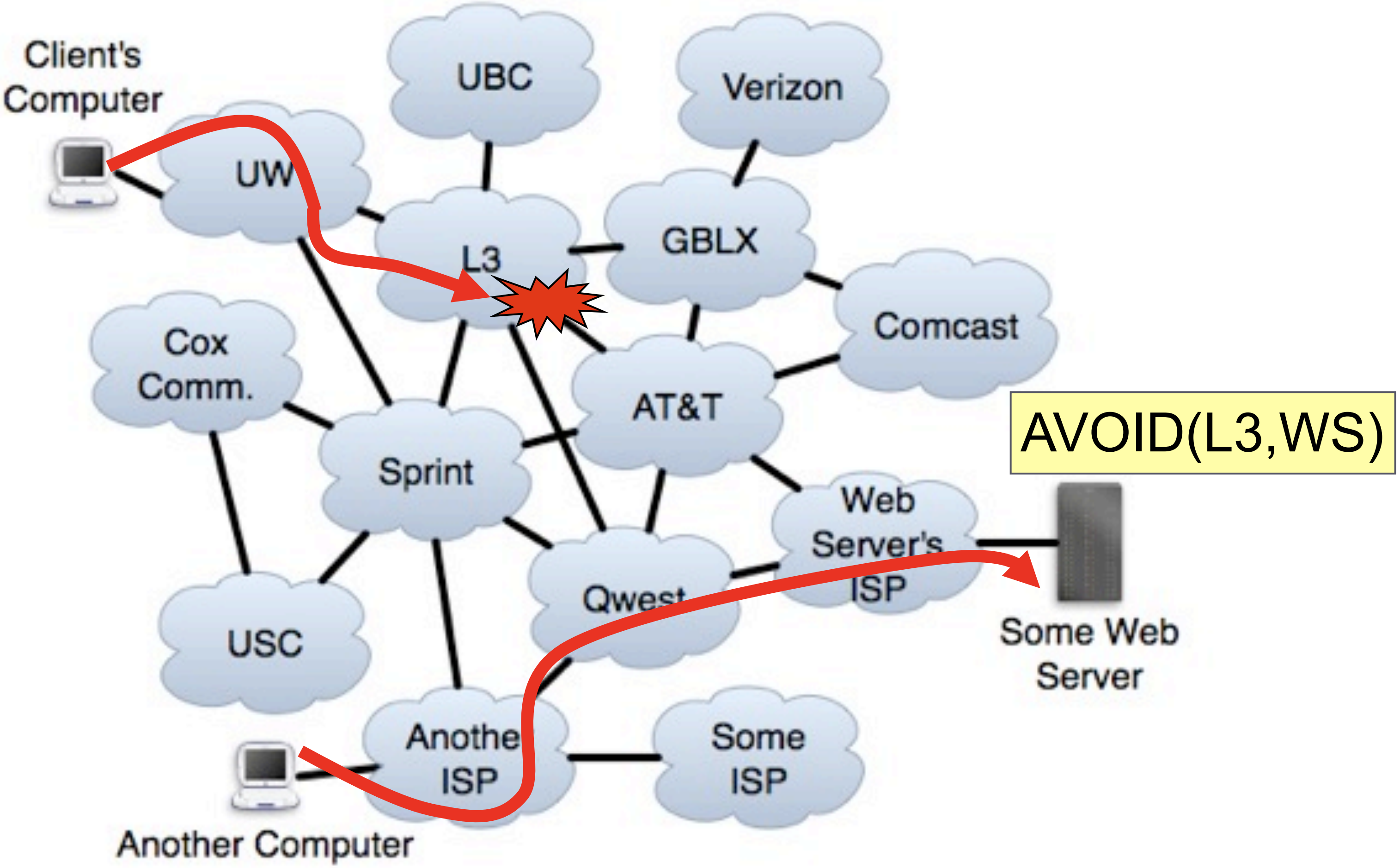
Reverse path: Want others to choose paths to my prefix P that avoid ISP or ISP-ISP link X

- Want a BGP announcement AVOID(X,P):
  - Any ISP with a route to P that avoids X uses such a route
  - Any ISP not using X need only pass on the announcement

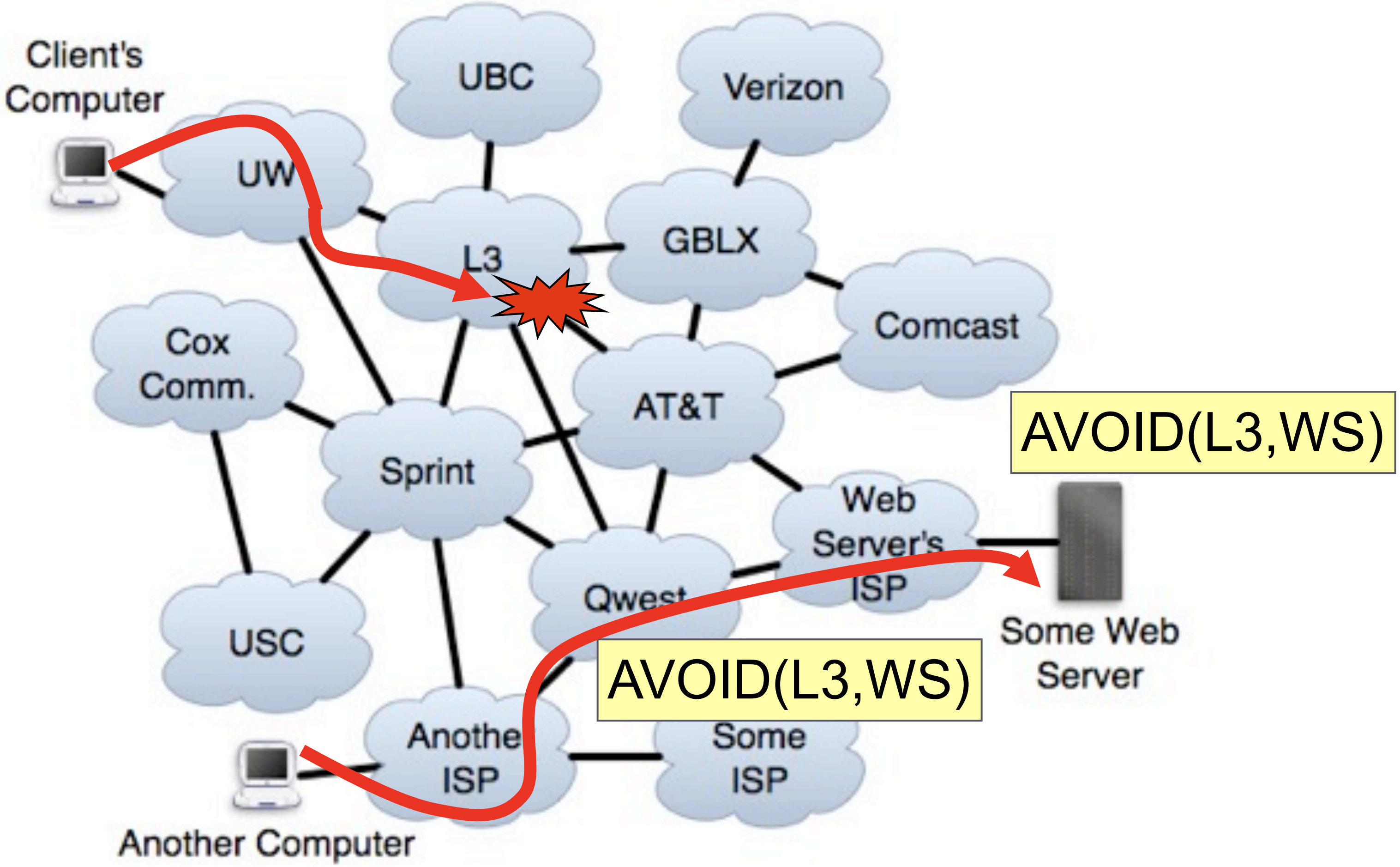
Ideal Self-Repair of Reverse Paths



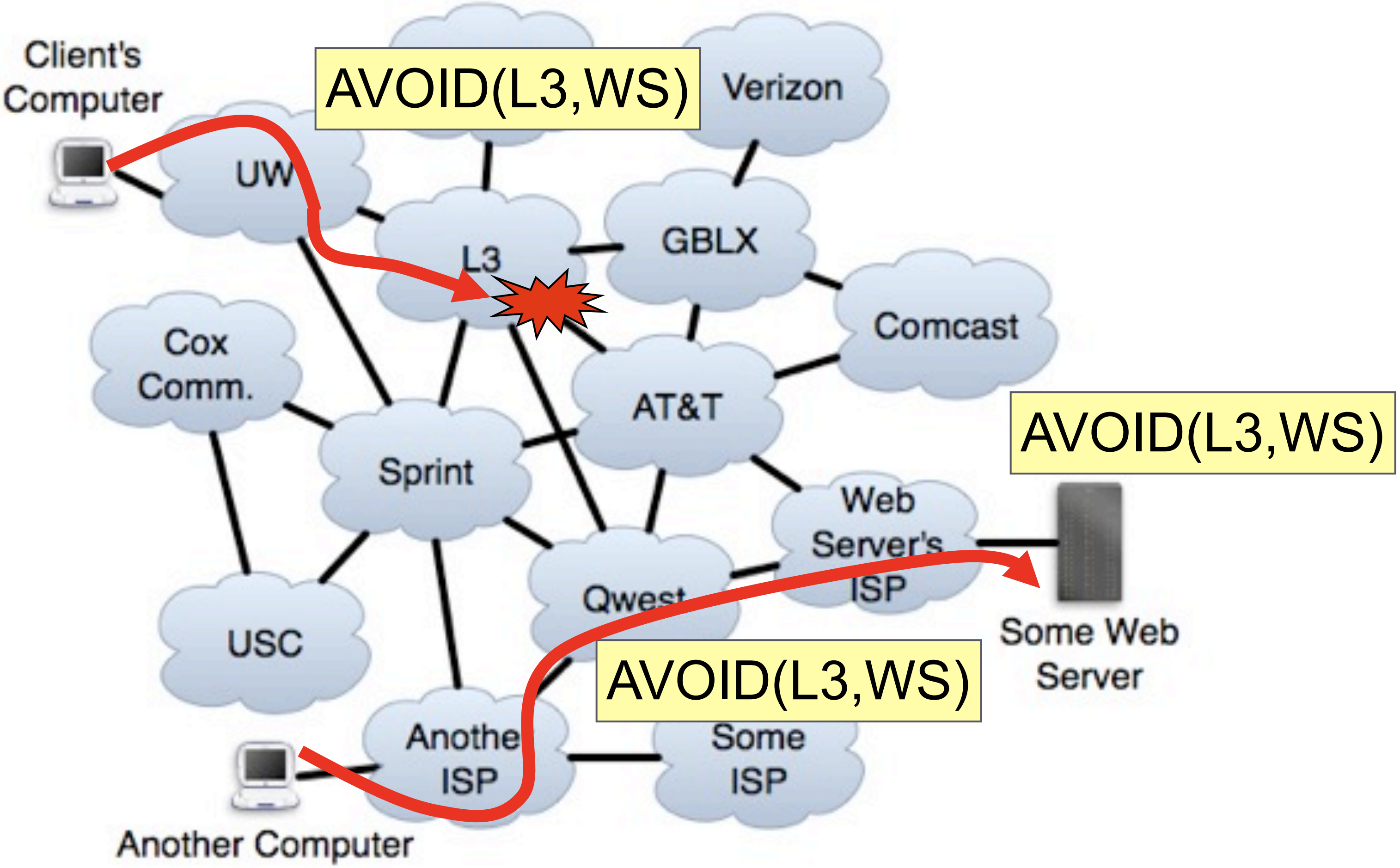
Ideal Self-Repair of Reverse Paths



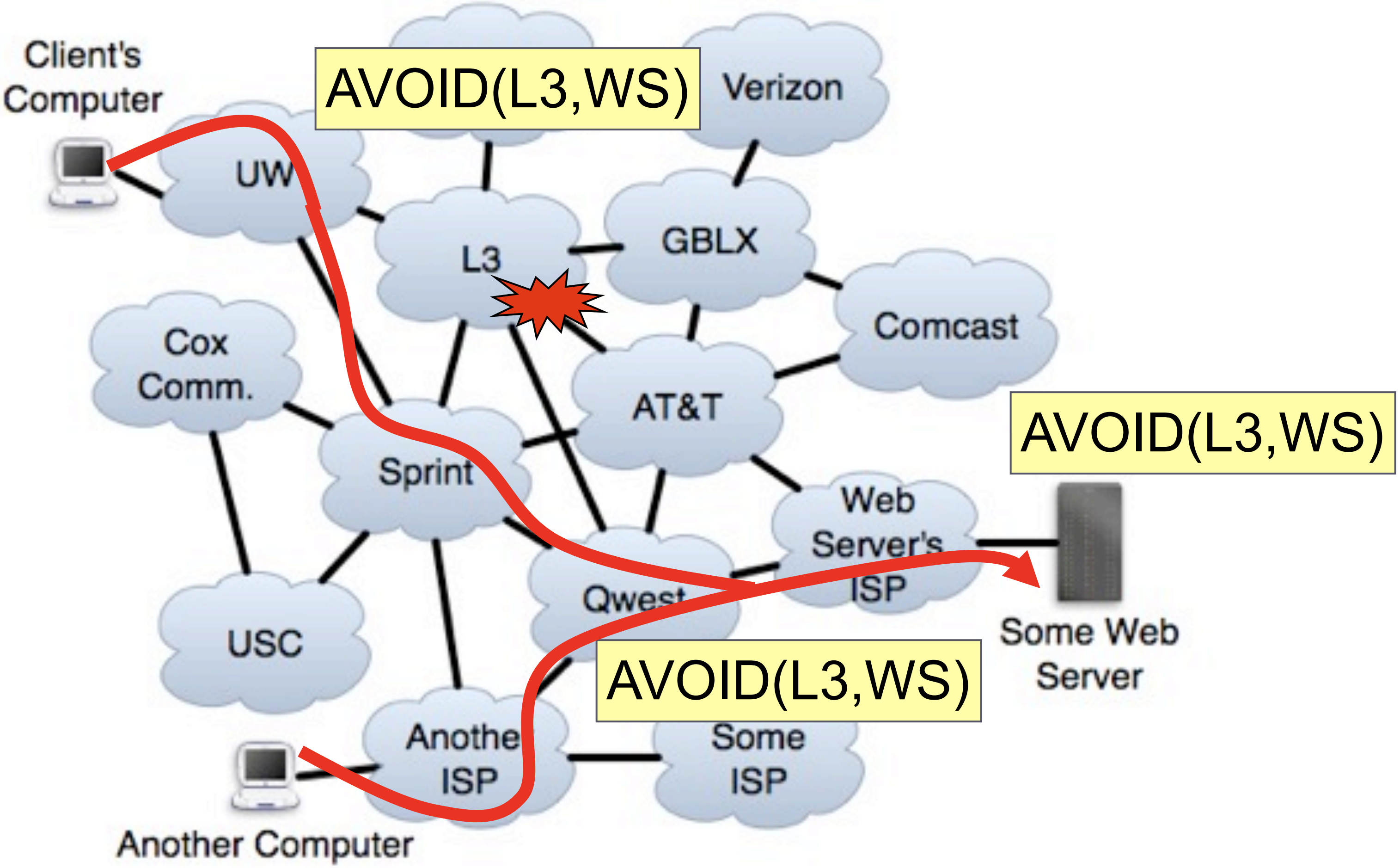
Ideal Self-Repair of Reverse Paths



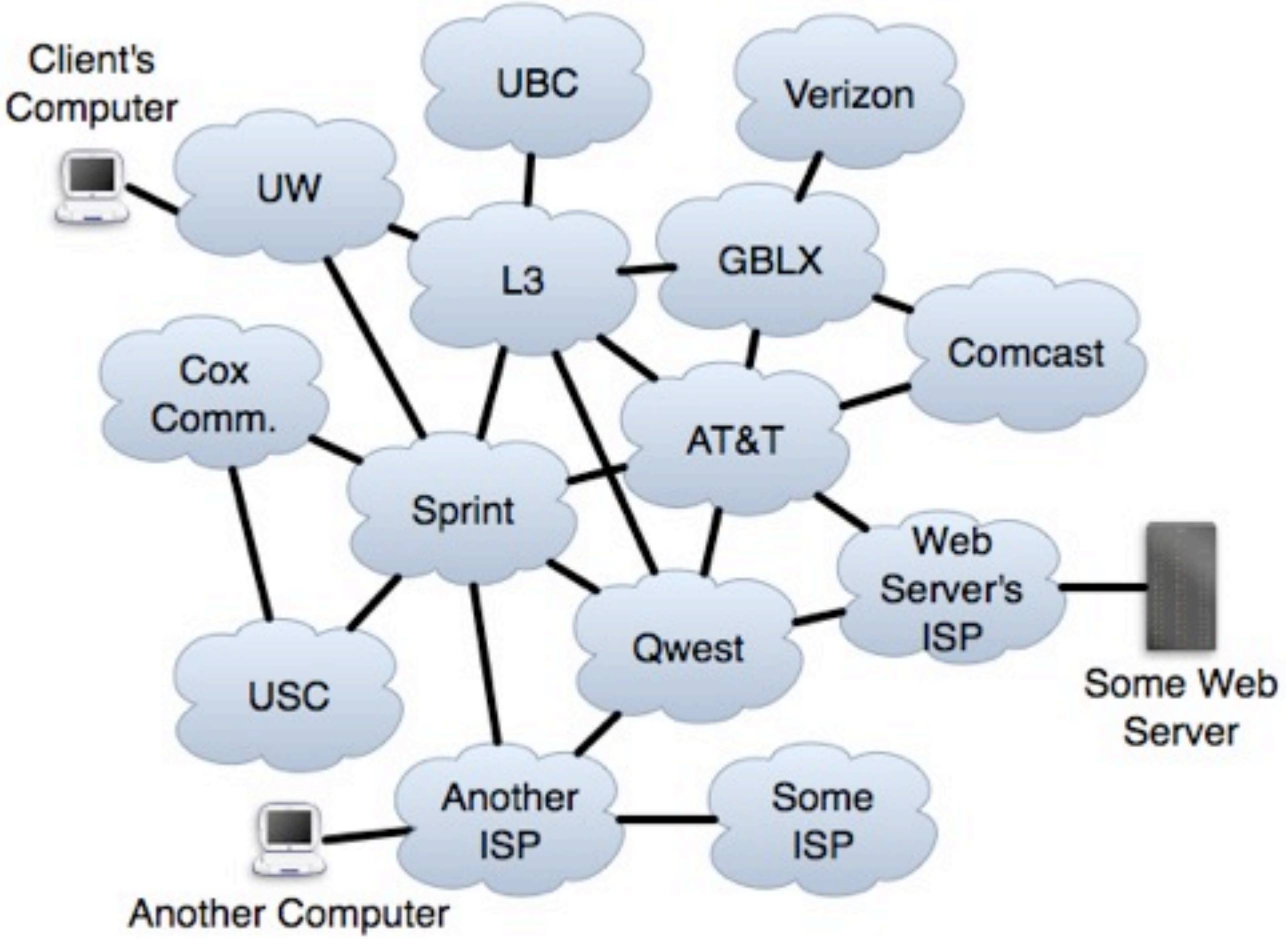
Ideal Self-Repair of Reverse Paths



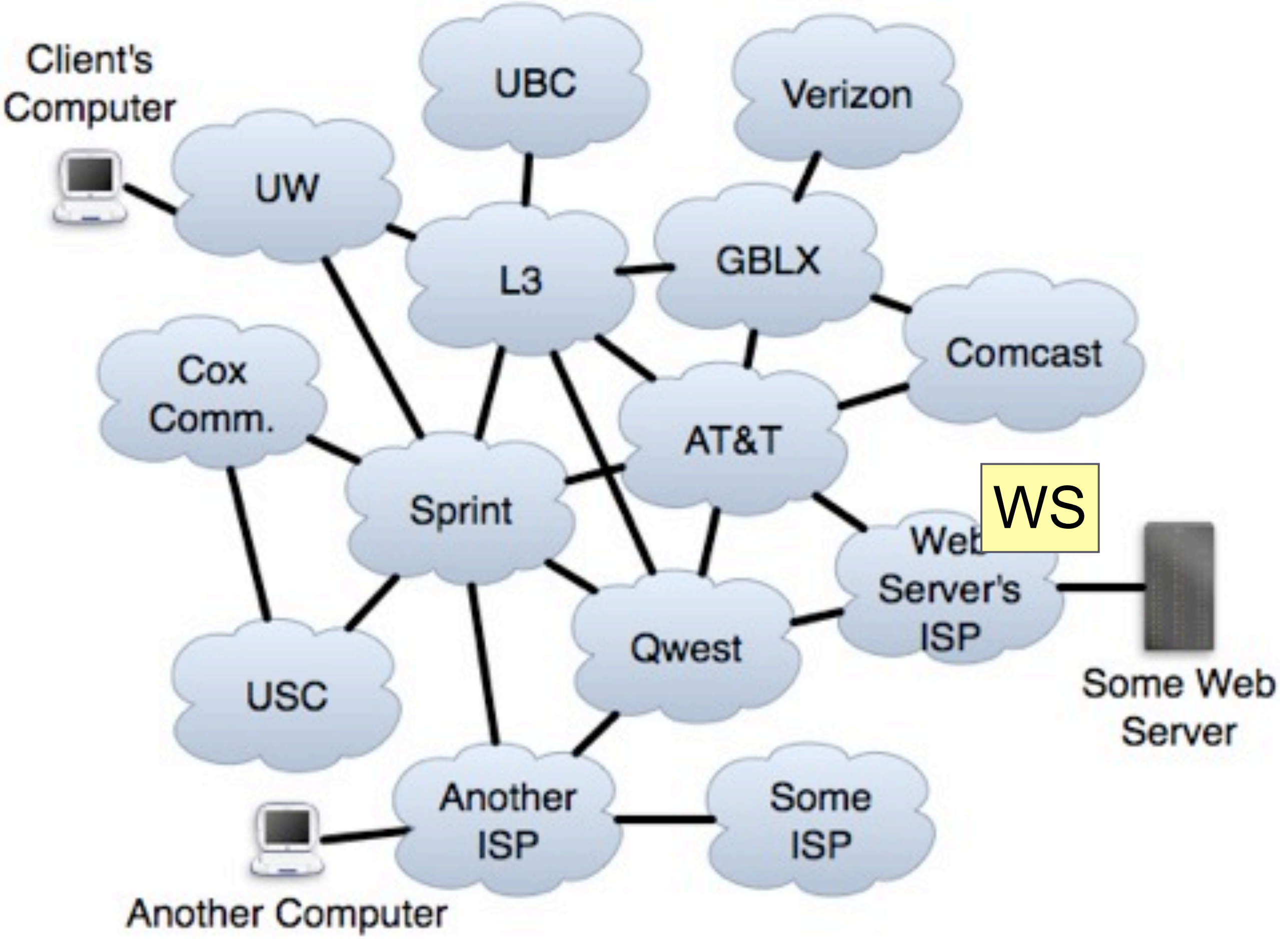
Ideal Self-Repair of Reverse Paths



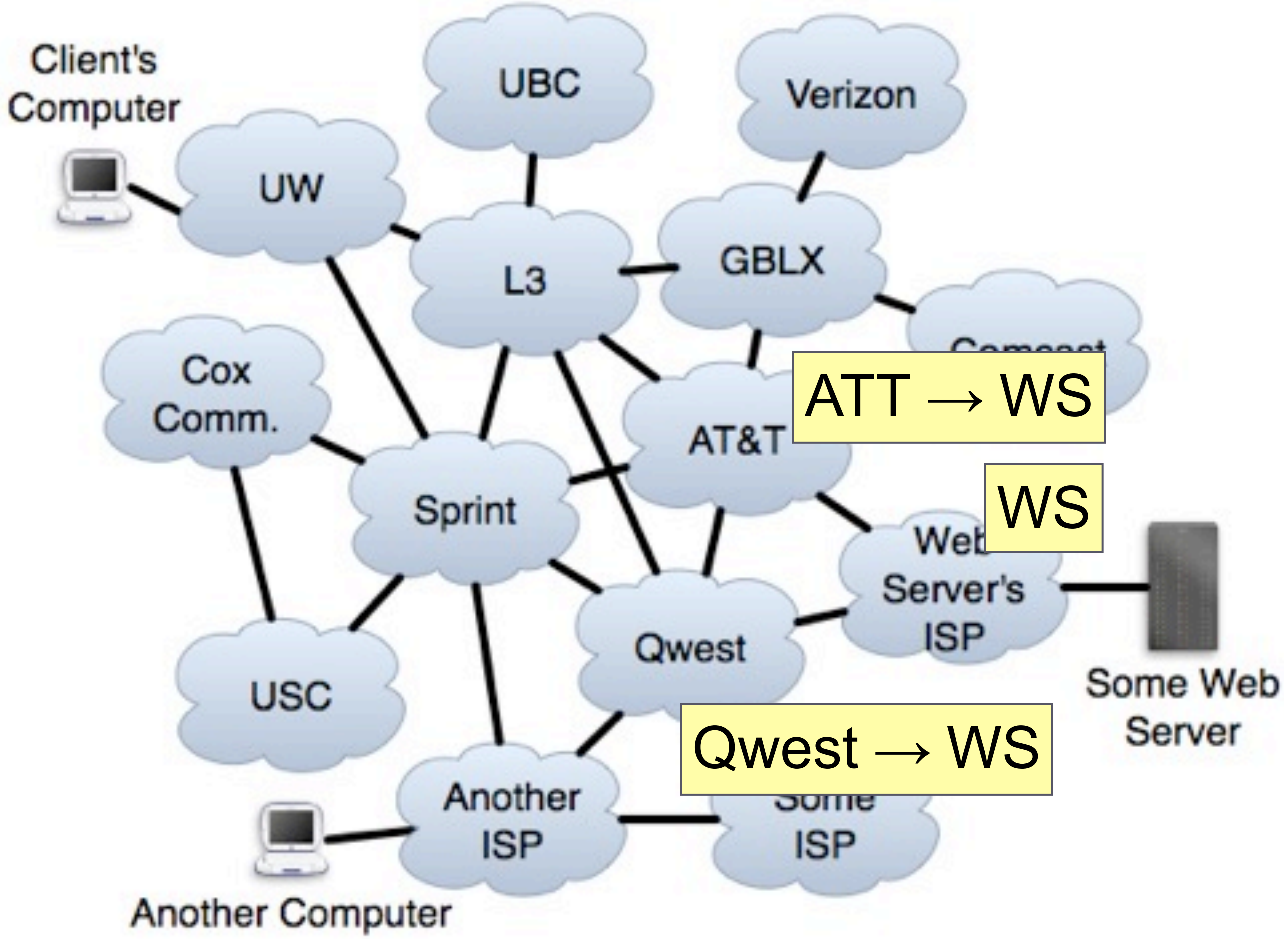
Practical Self-Repair of Reverse Paths



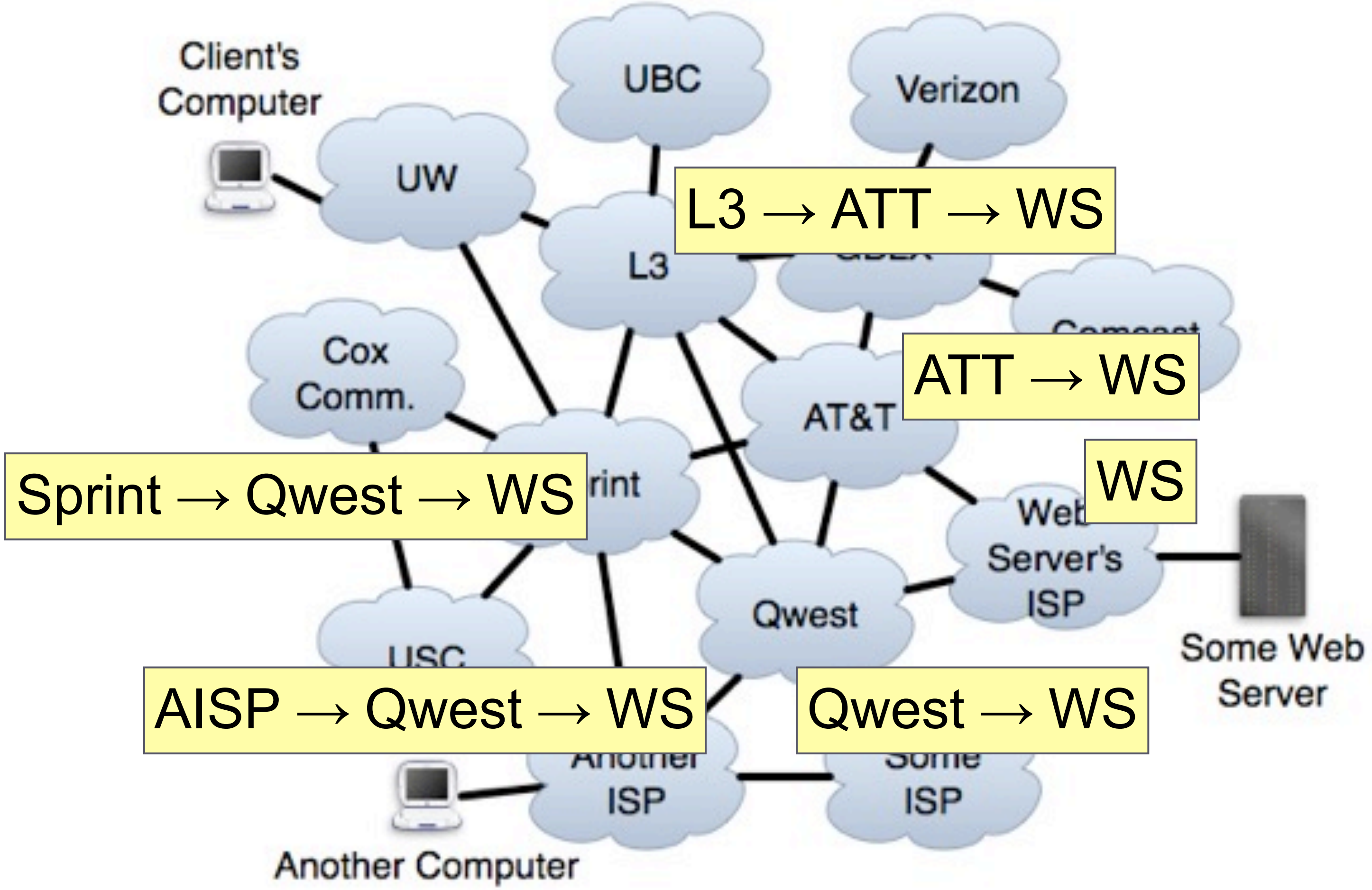
Practical Self-Repair of Reverse Paths



Practical Self-Repair of Reverse Paths



Practical Self-Repair of Reverse Paths



L3 → ATT → WS

ATT → WS

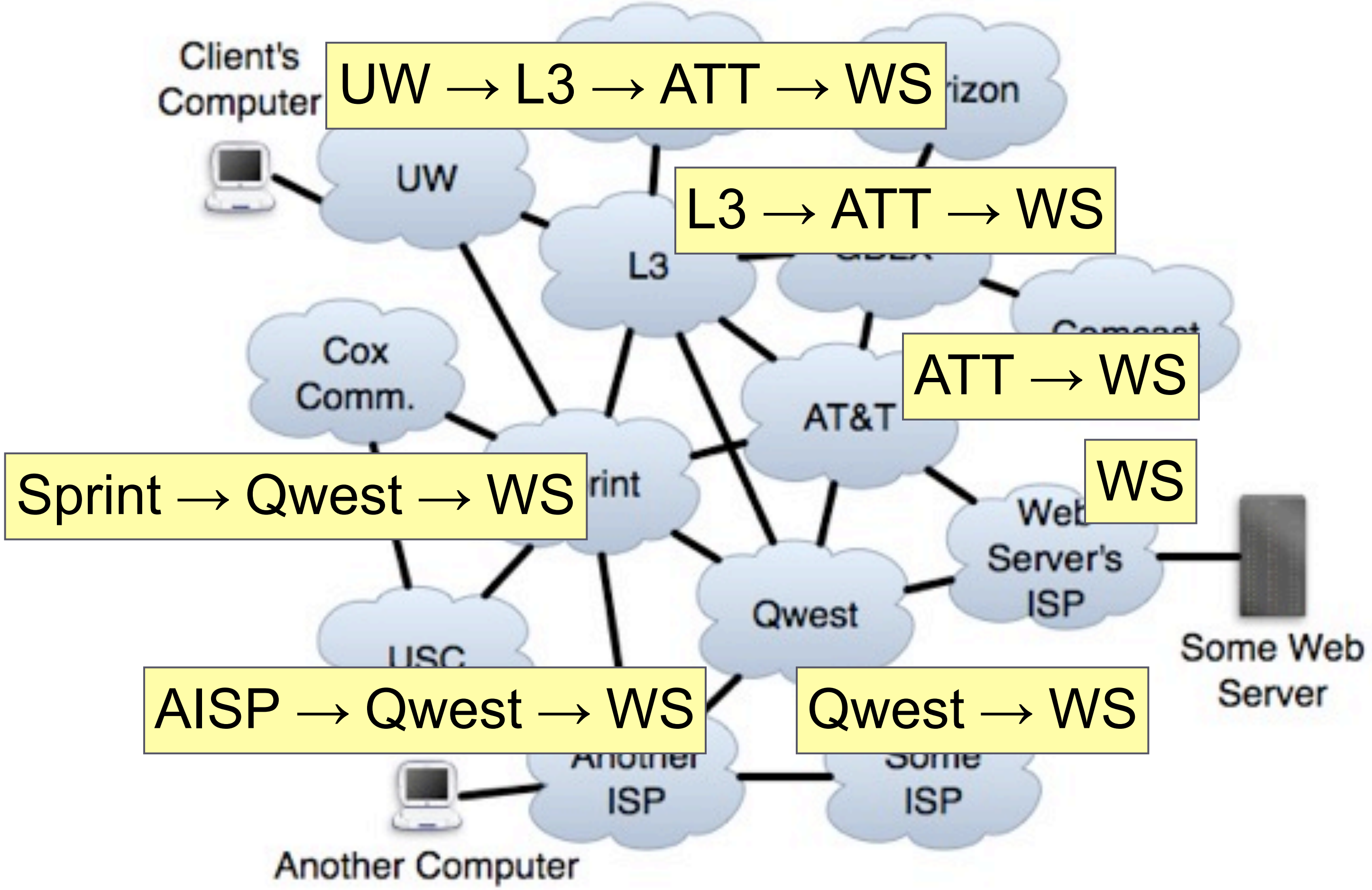
WS

Sprint → Qwest → WS

AISP → Qwest → WS

Qwest → WS

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

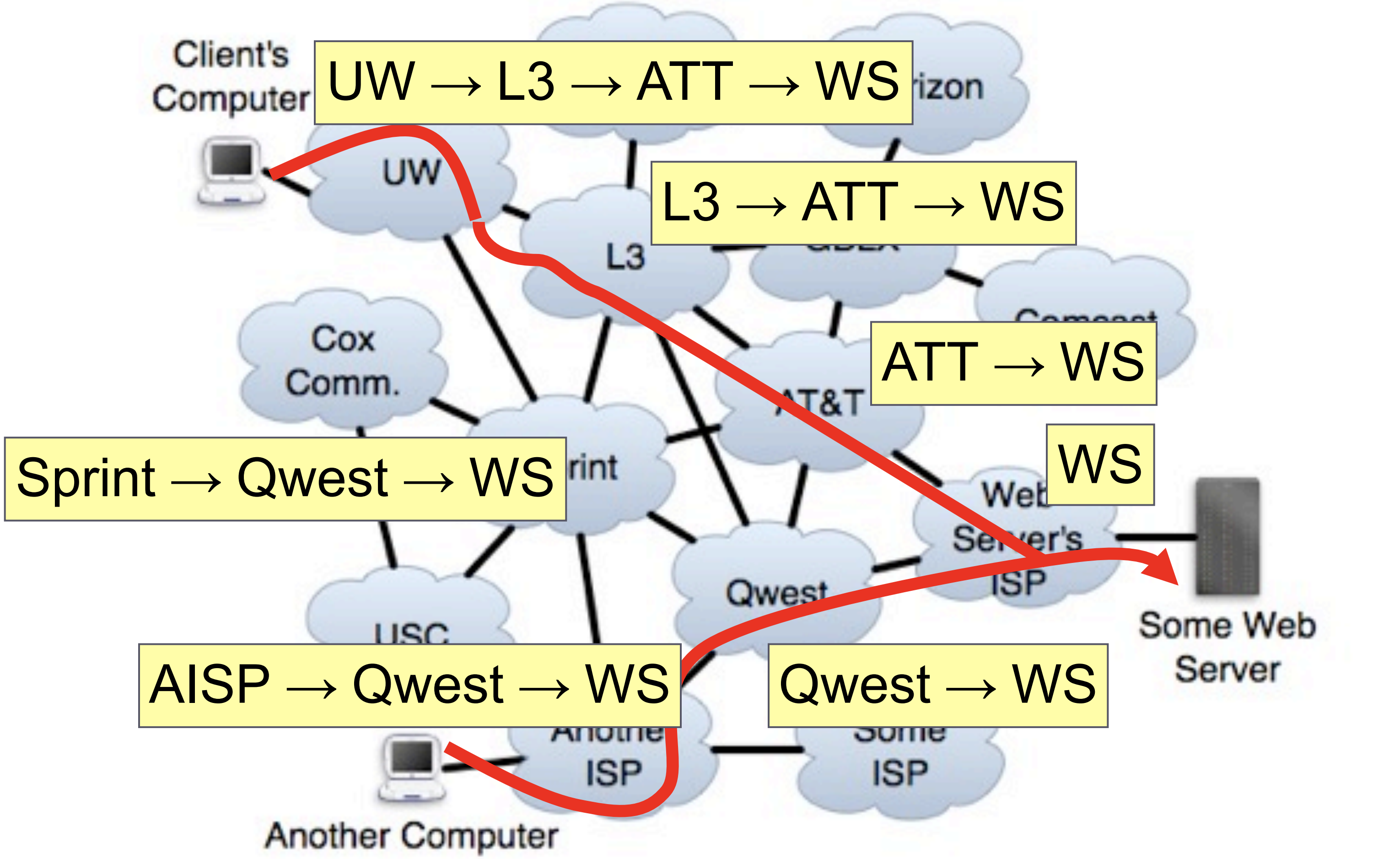
WS

Sprint → Qwest → WS

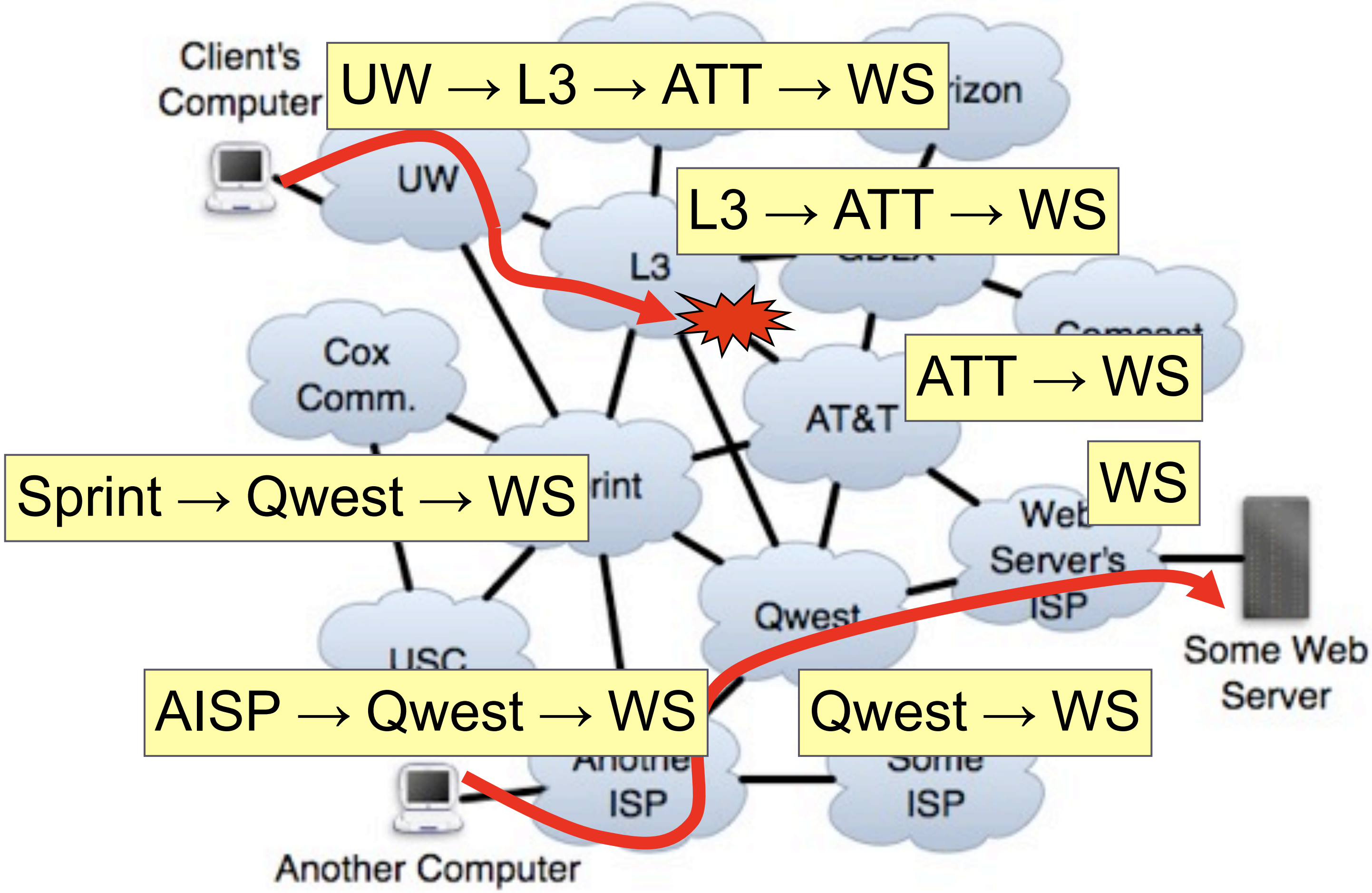
AISP → Qwest → WS

Qwest → WS

Practical Self-Repair of Reverse Paths



Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

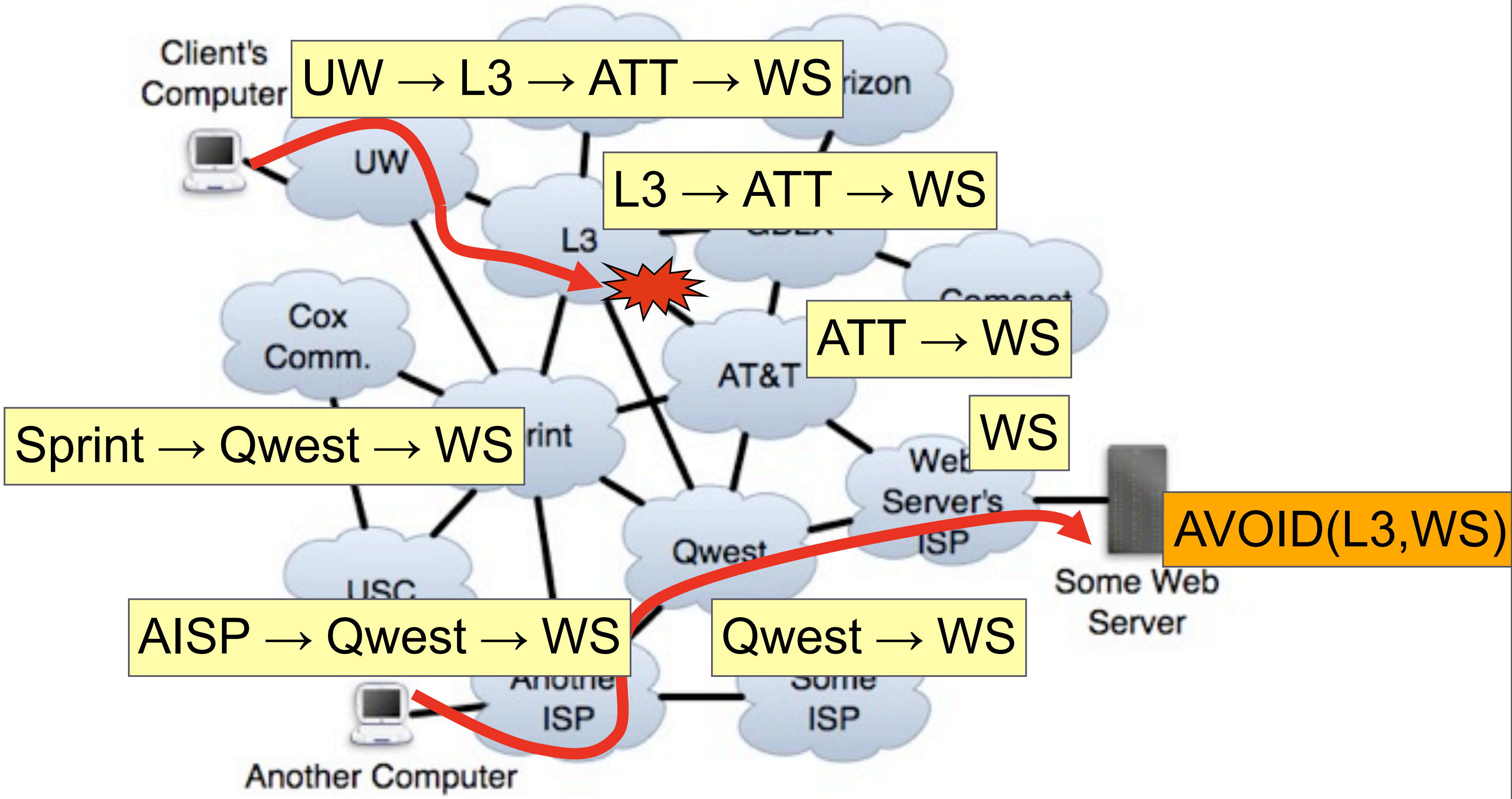
WS

Sprint → Qwest → WS

AISP → Qwest → WS

Qwest → WS

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

WS

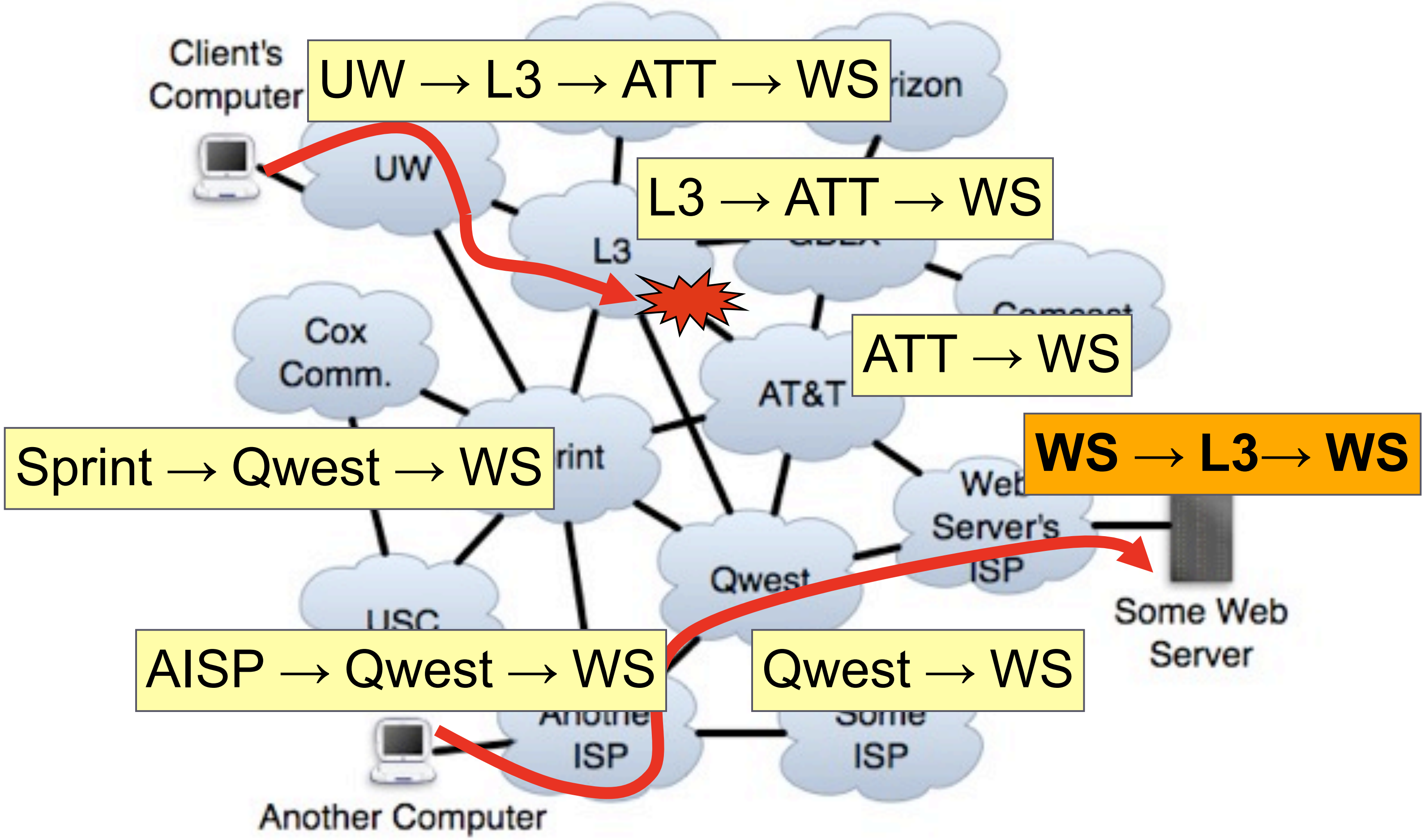
Sprint → Qwest → WS

AISP → Qwest → WS

Qwest → WS

AVOID(L3,WS)

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

WS → L3→ WS

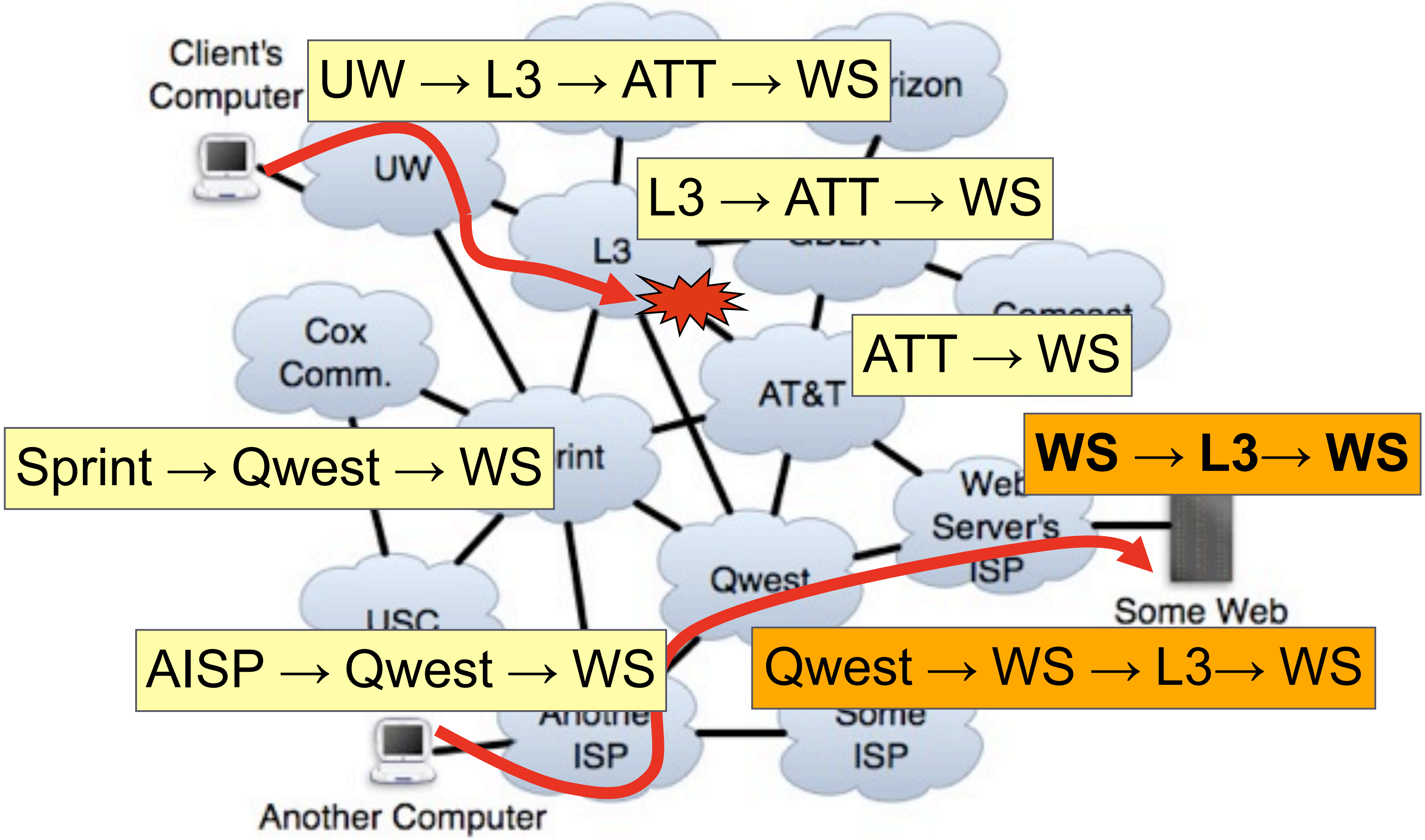
Sprint → Qwest → WS

AISP → Qwest → WS

Qwest → WS

BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

WS → L3→ WS

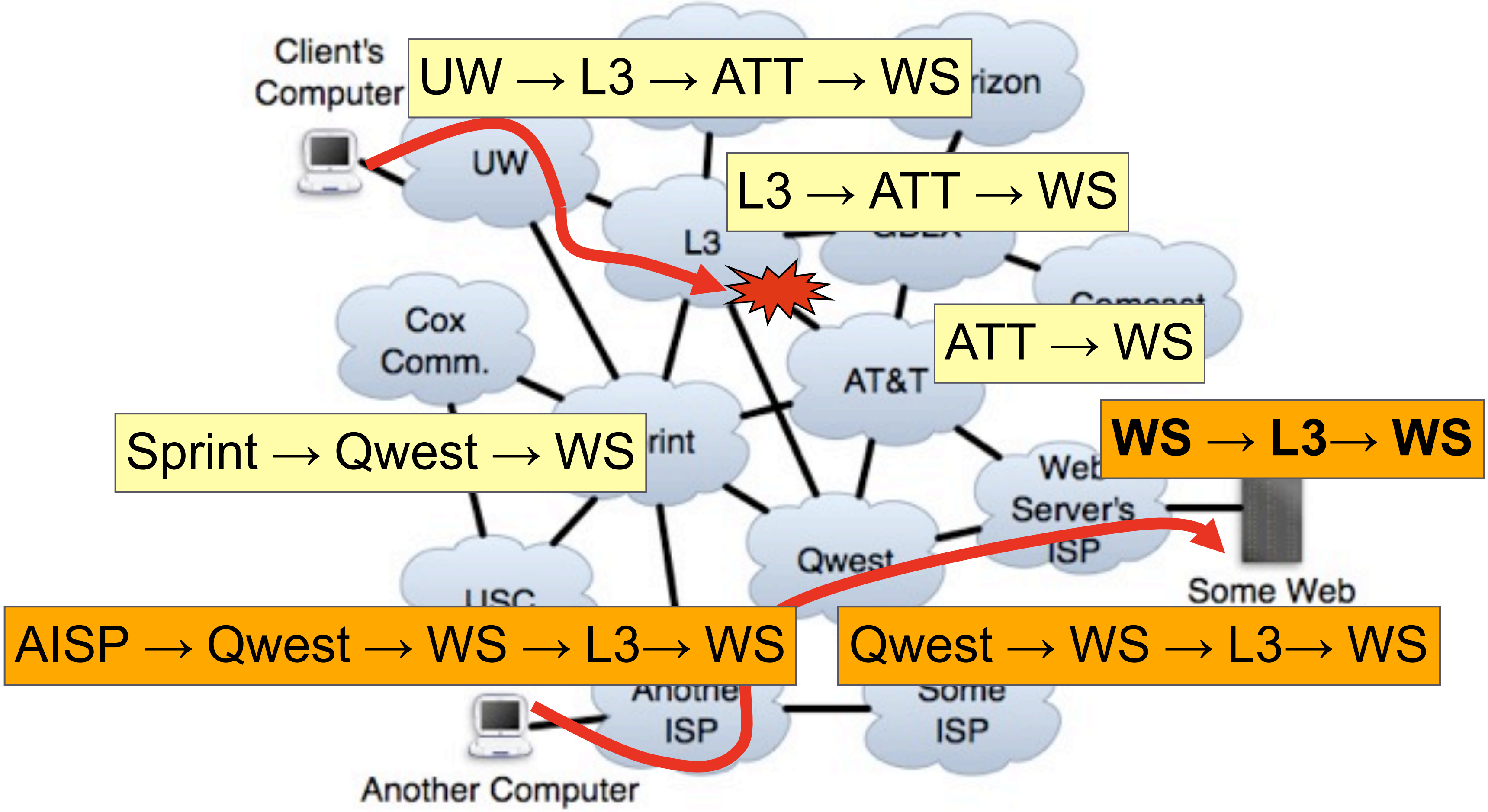
Sprint → Qwest → WS

AISP → Qwest → WS

Qwest → WS → L3→ WS

BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

Sprint → Qwest → WS

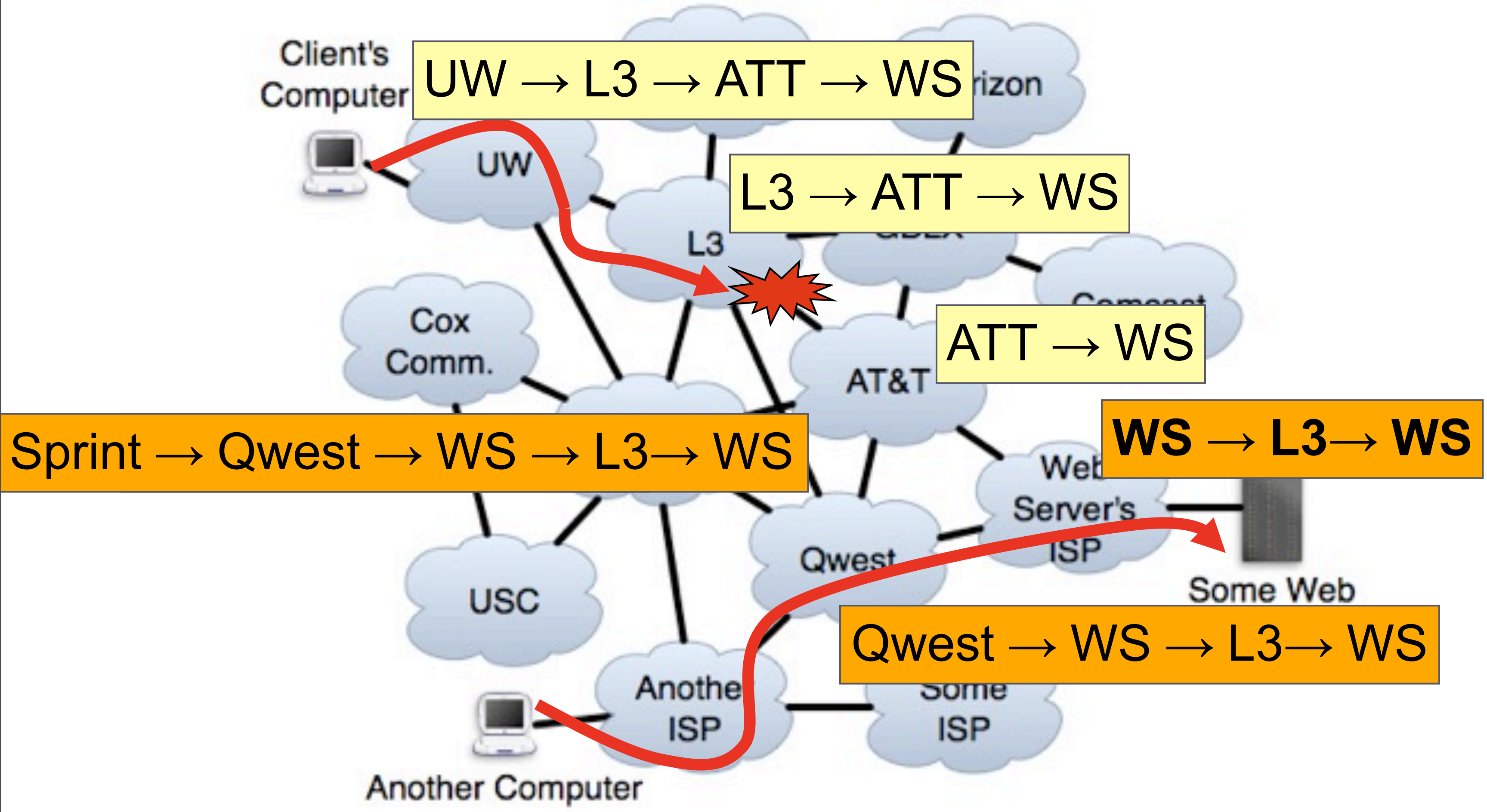
WS → L3→ WS

AISP → Qwest → WS → L3→ WS

Qwest → WS → L3→ WS

BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

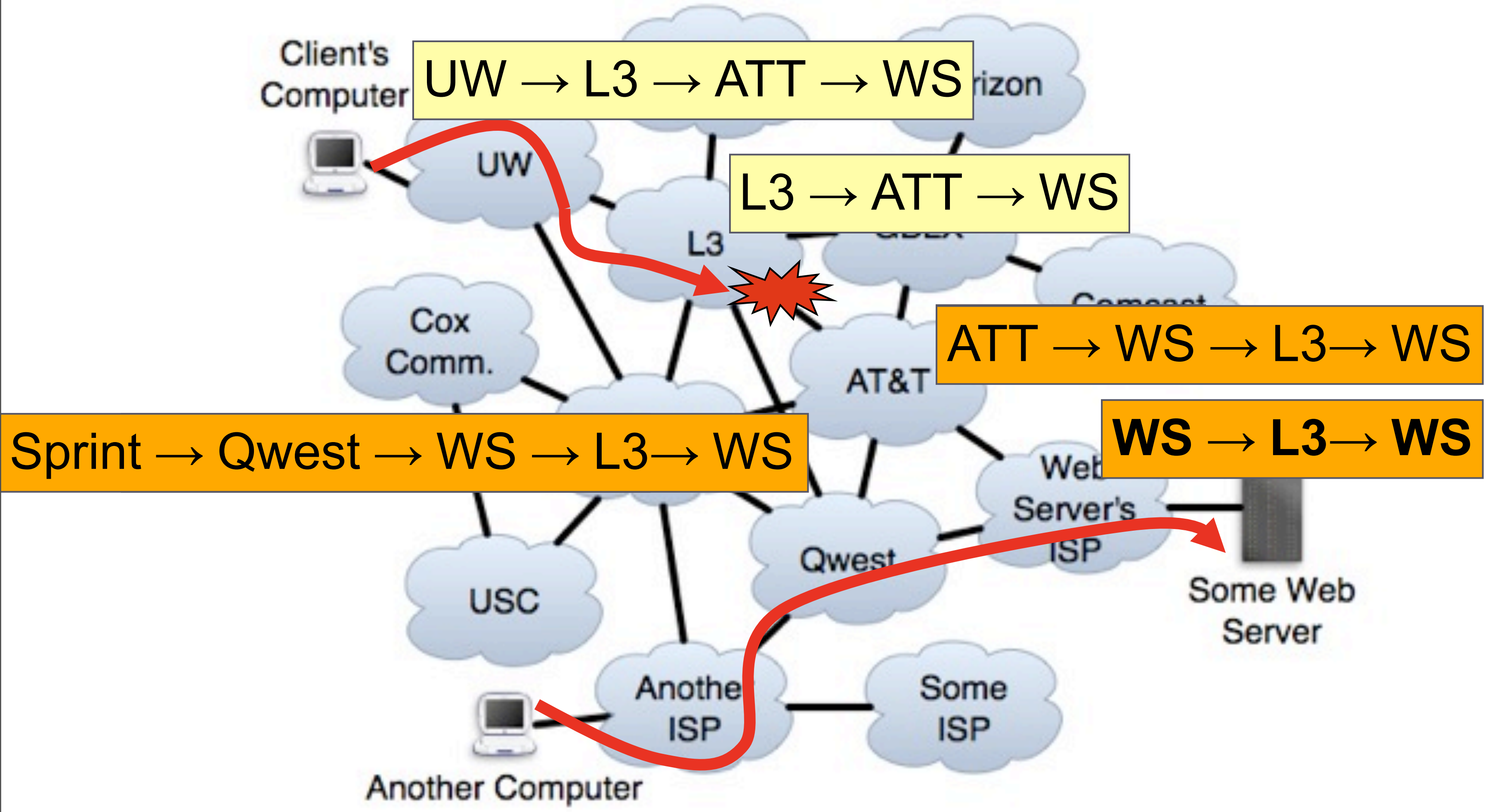
Sprint → Qwest → WS → L3→ WS

WS → L3→ WS

Qwest → WS → L3→ WS

BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

L3 → ATT → WS

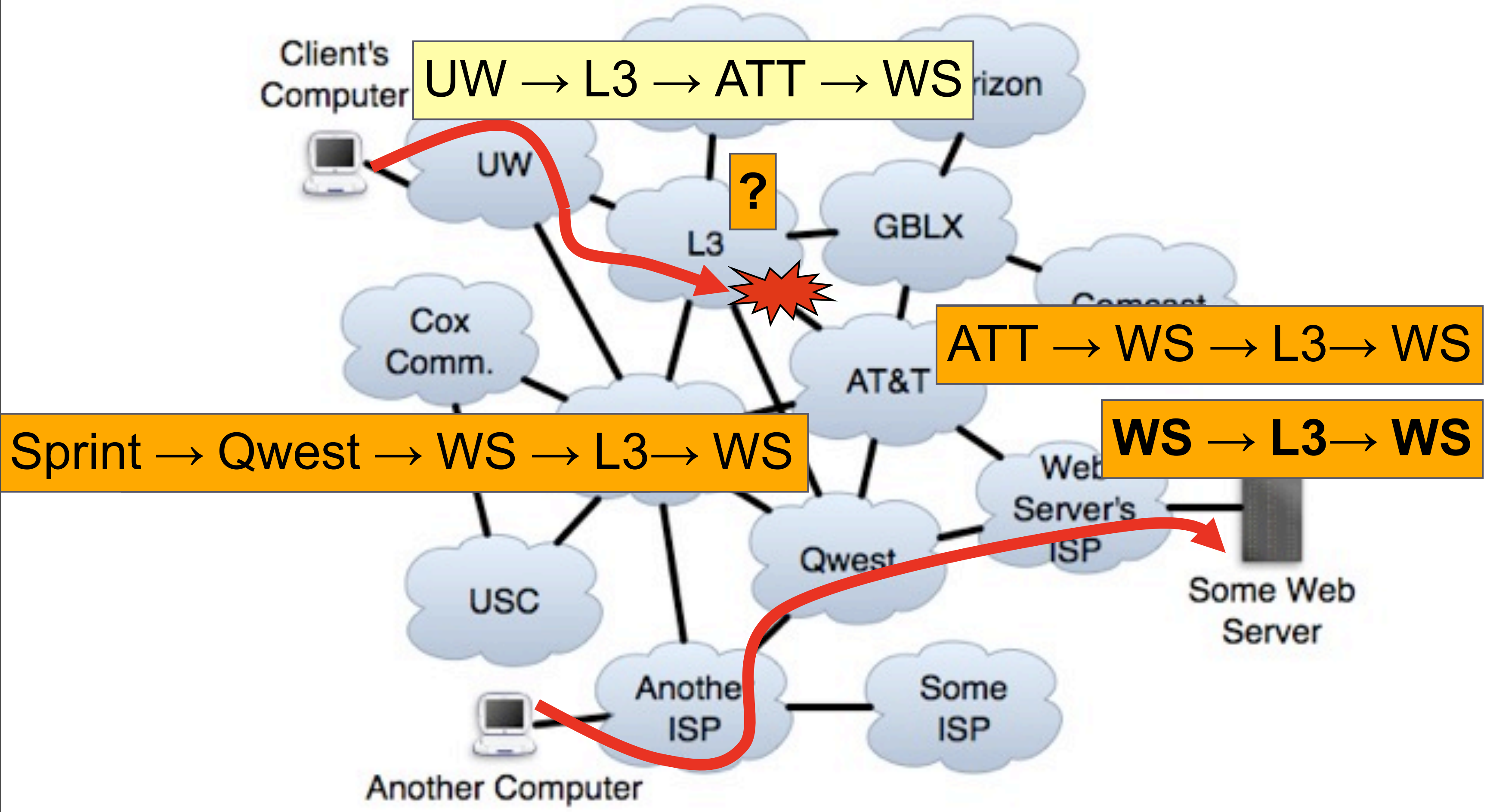
ATT → WS → L3→ WS

WS → L3→ WS

Sprint → Qwest → WS → L3→ WS

BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



UW → L3 → ATT → WS

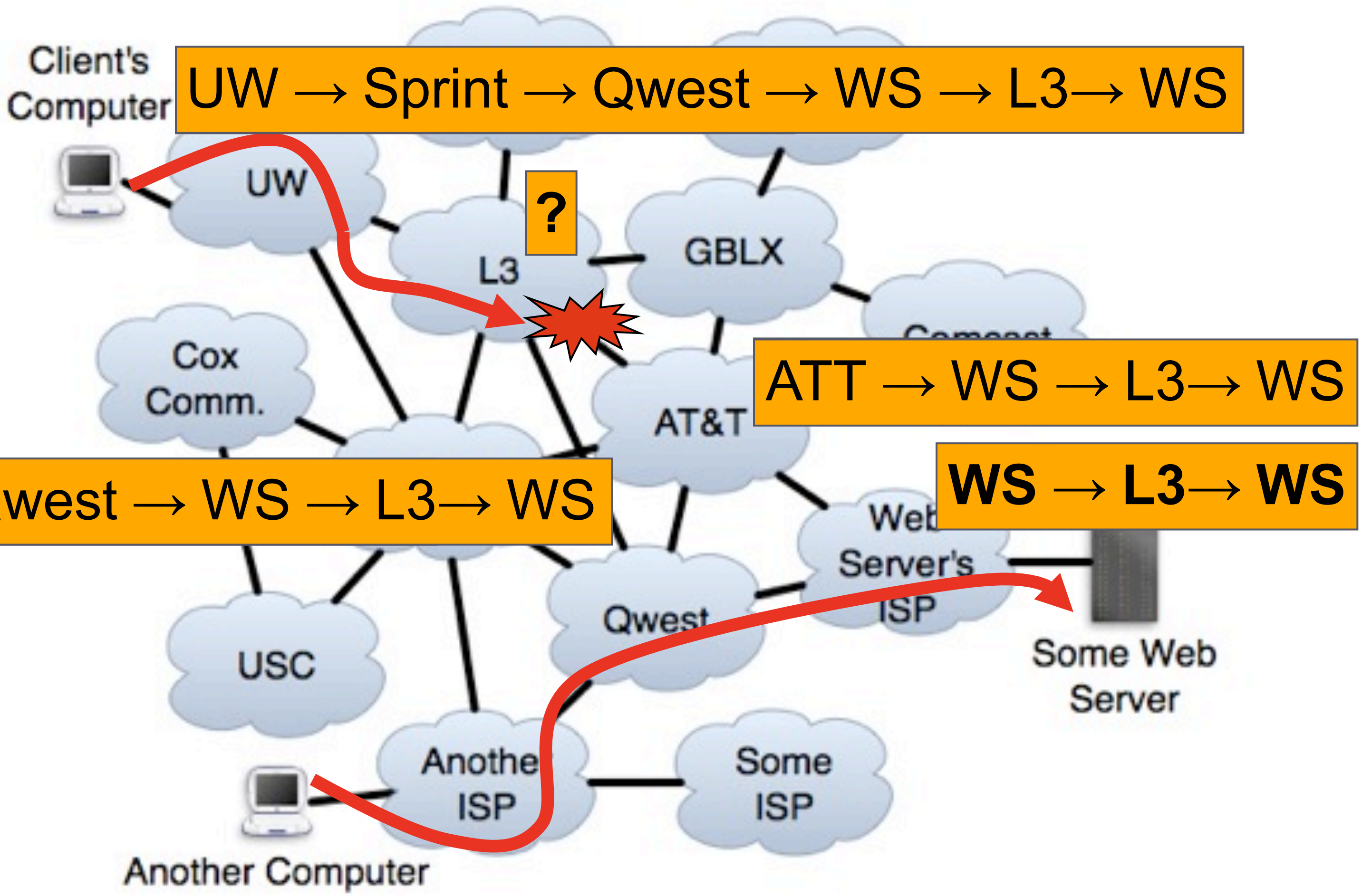
ATT → WS → L3→ WS

WS → L3→ WS

Sprint → Qwest → WS → L3→ WS

BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



UW → Sprint → Qwest → WS → L3→ WS

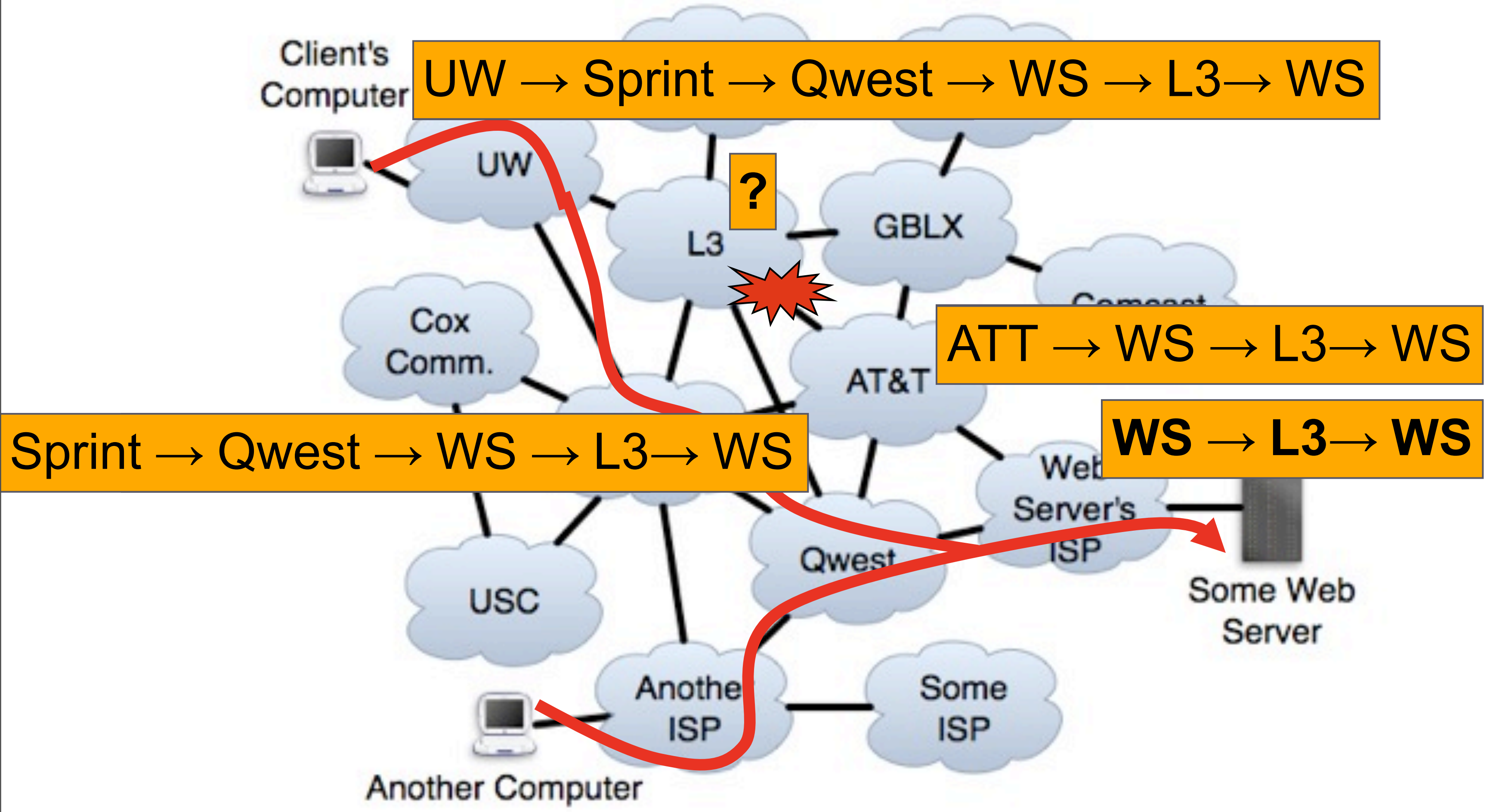
ATT → WS → L3→ WS

WS → L3→ WS

Sprint → Qwest → WS → L3→ WS

BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



UW → Sprint → Qwest → WS → L3→ WS

ATT → WS → L3→ WS

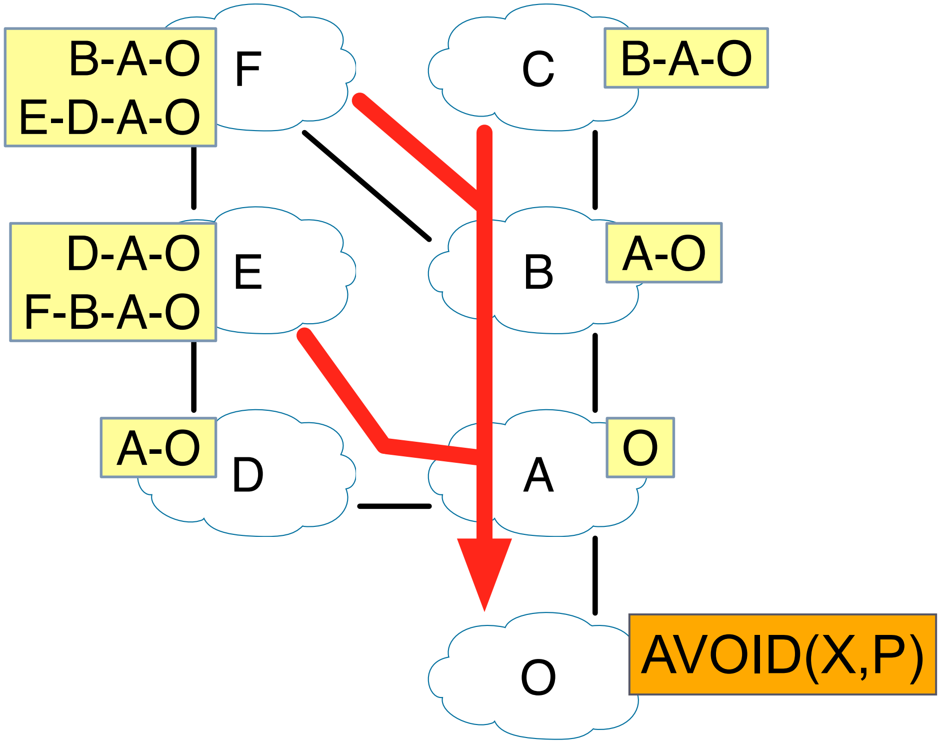
WS → L3→ WS

Sprint → Qwest → WS → L3→ WS

BGP loop prevention encourages switch to working path.

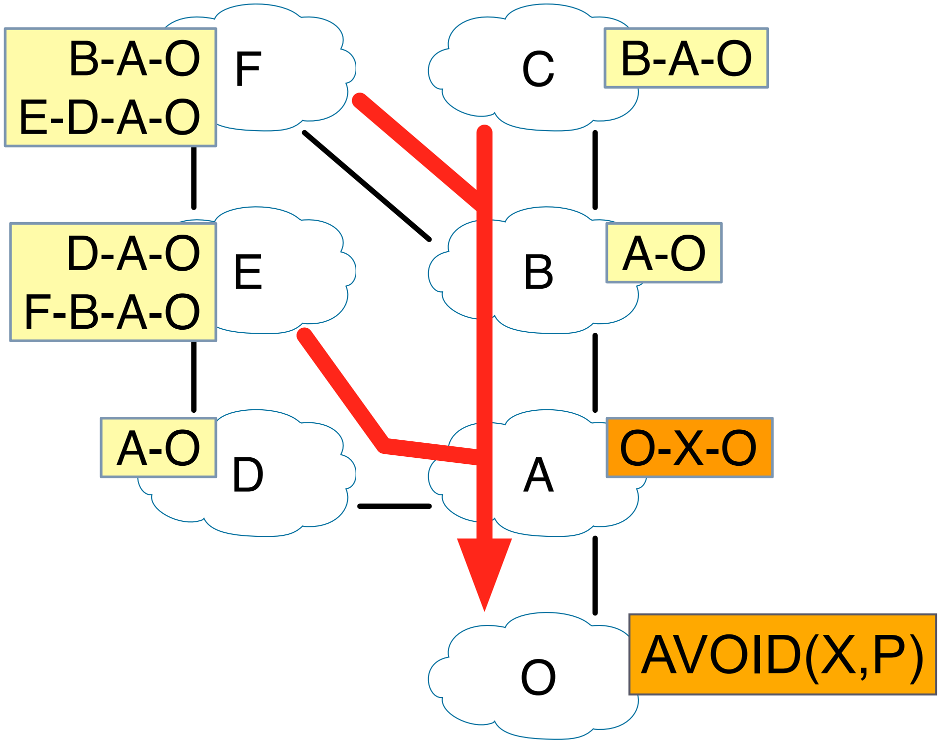
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



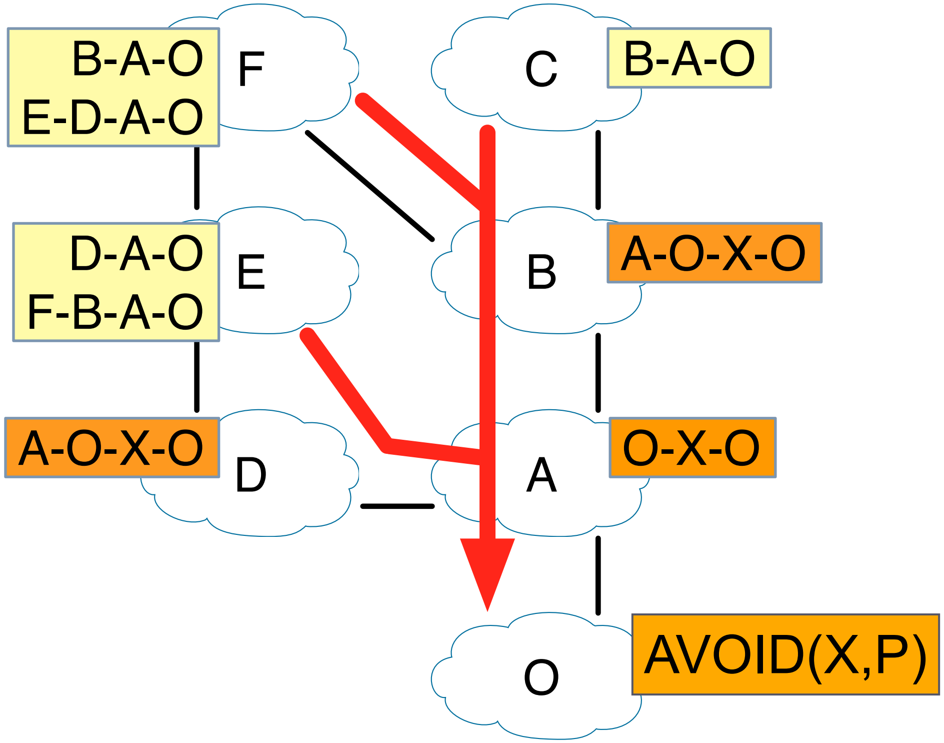
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



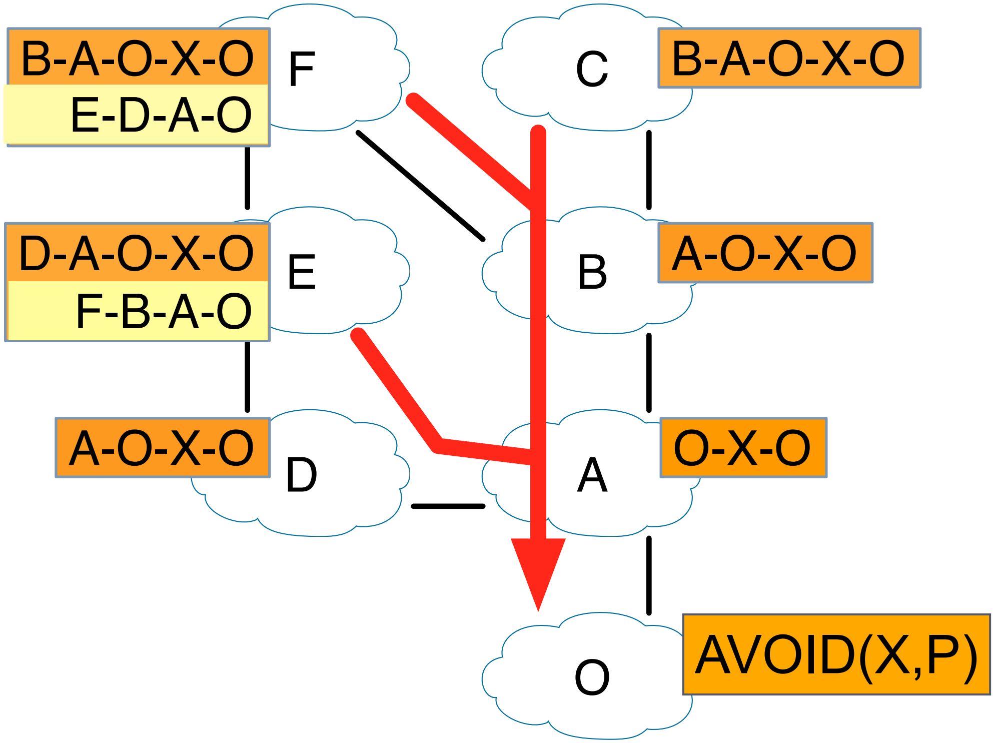
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



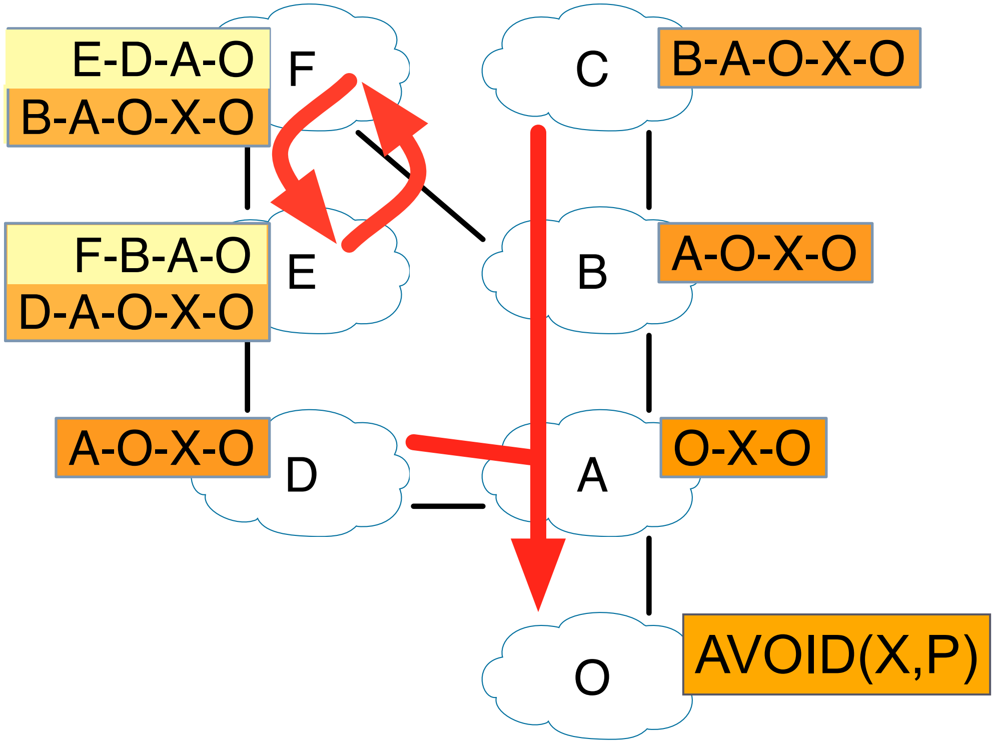
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



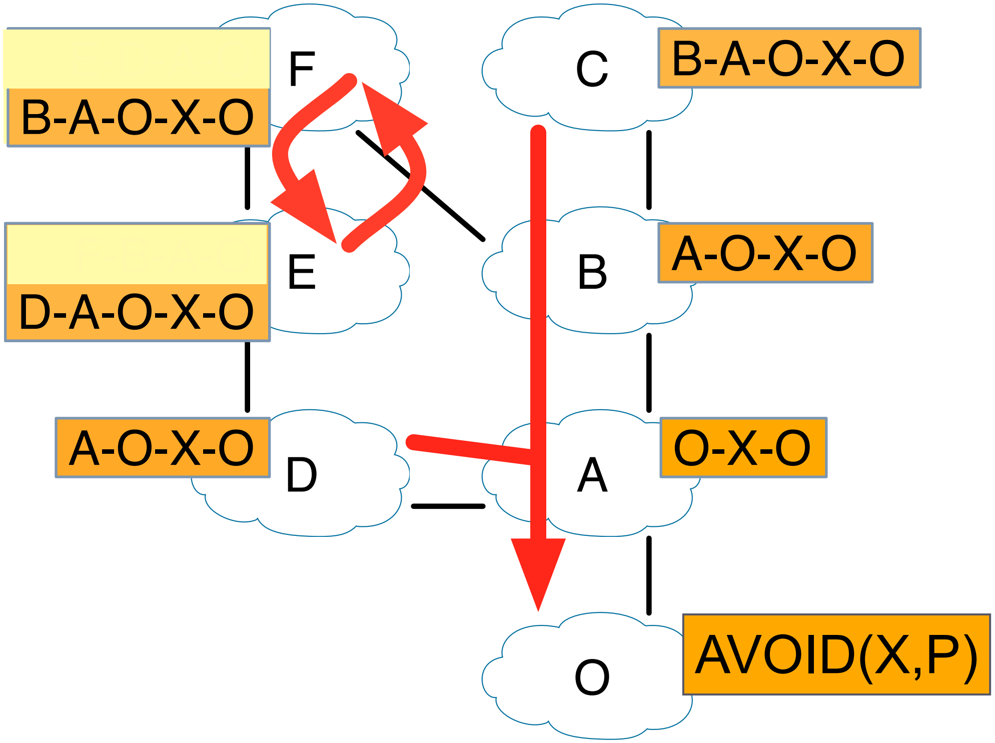
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



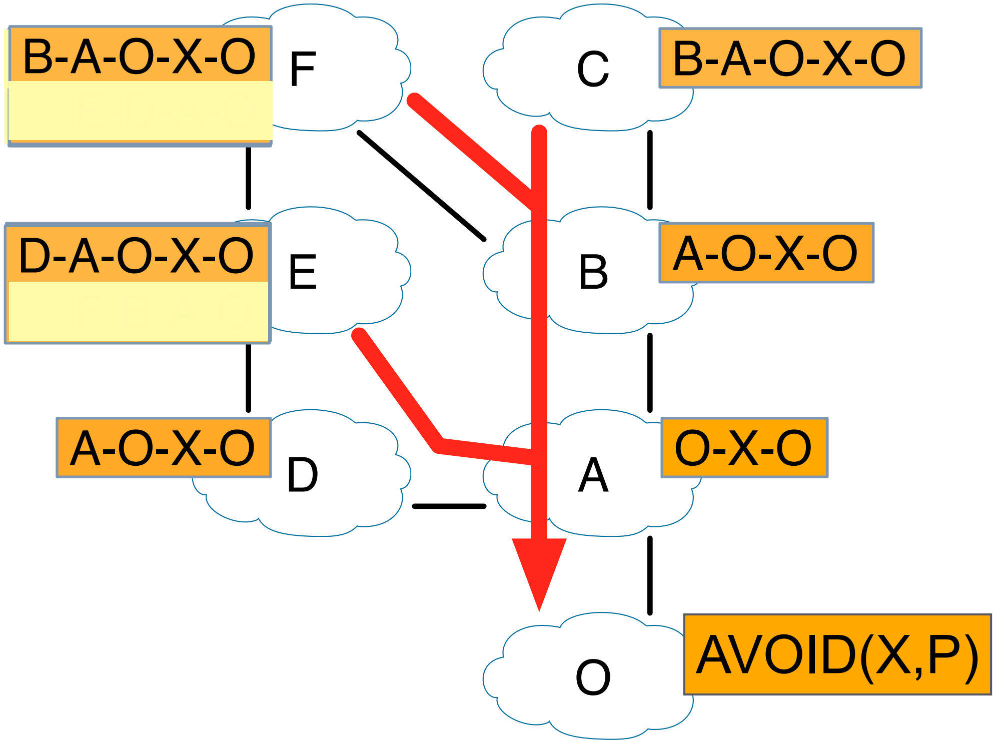
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



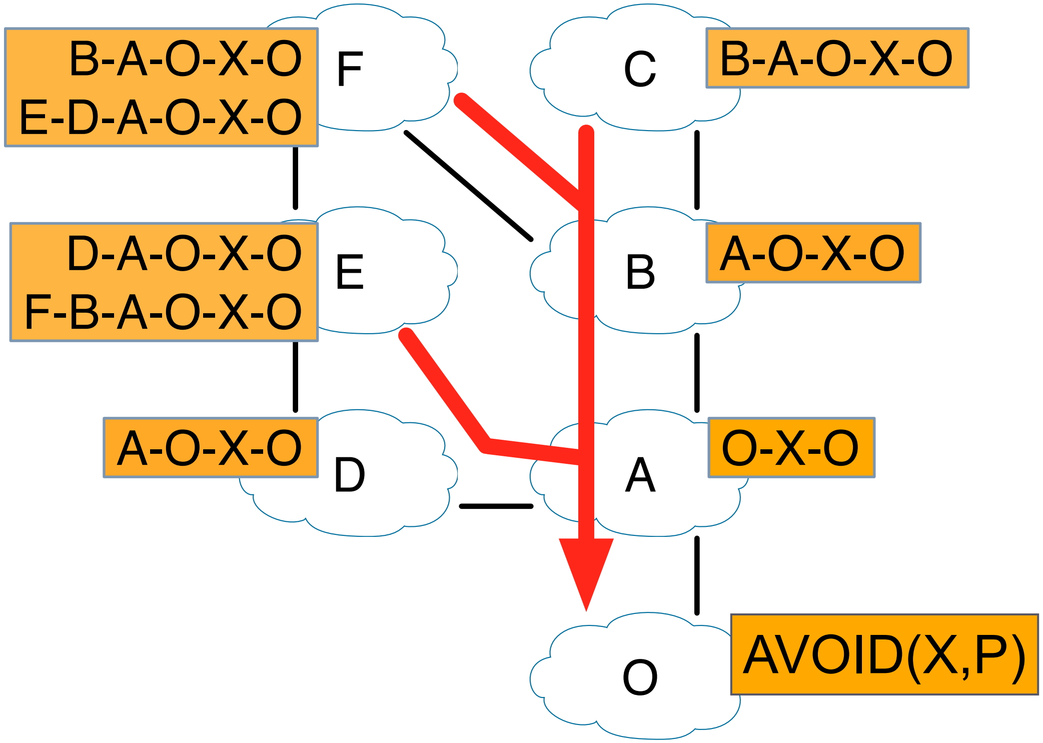
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



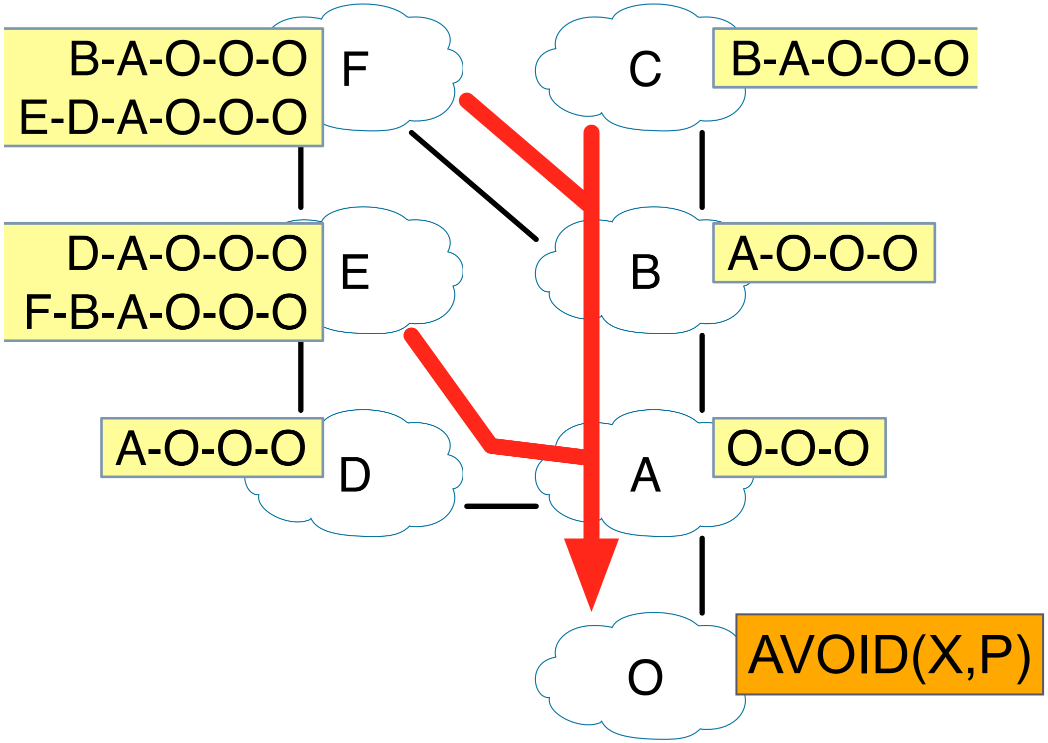
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss



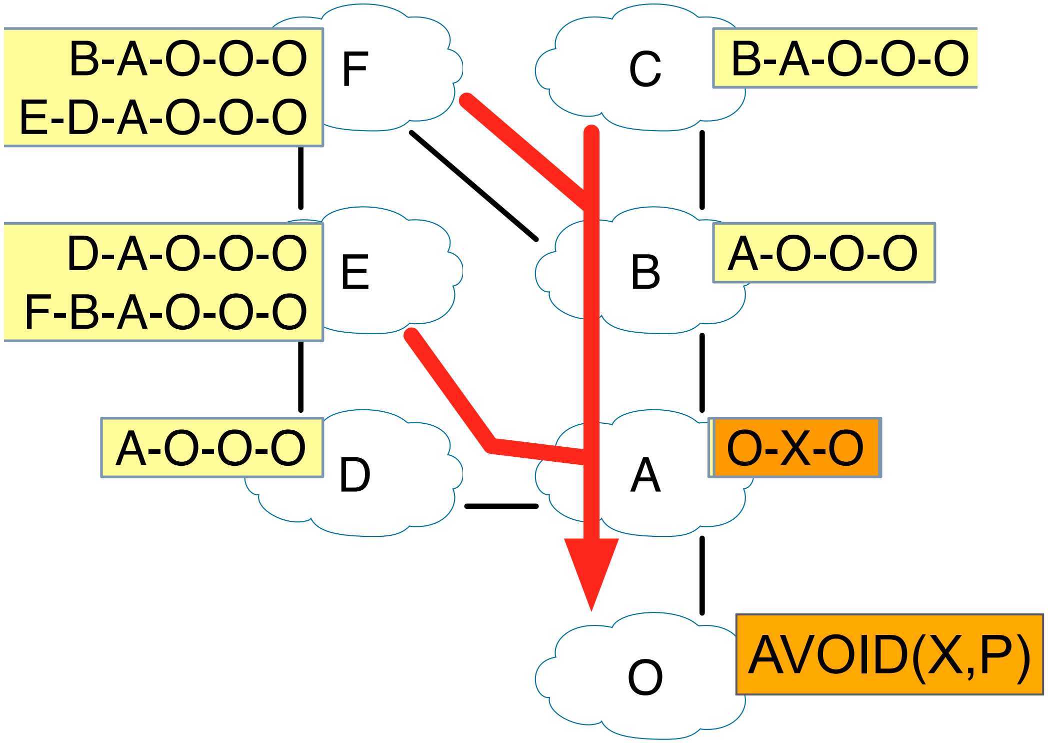
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  (1) next hop ISP
  (2) path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison



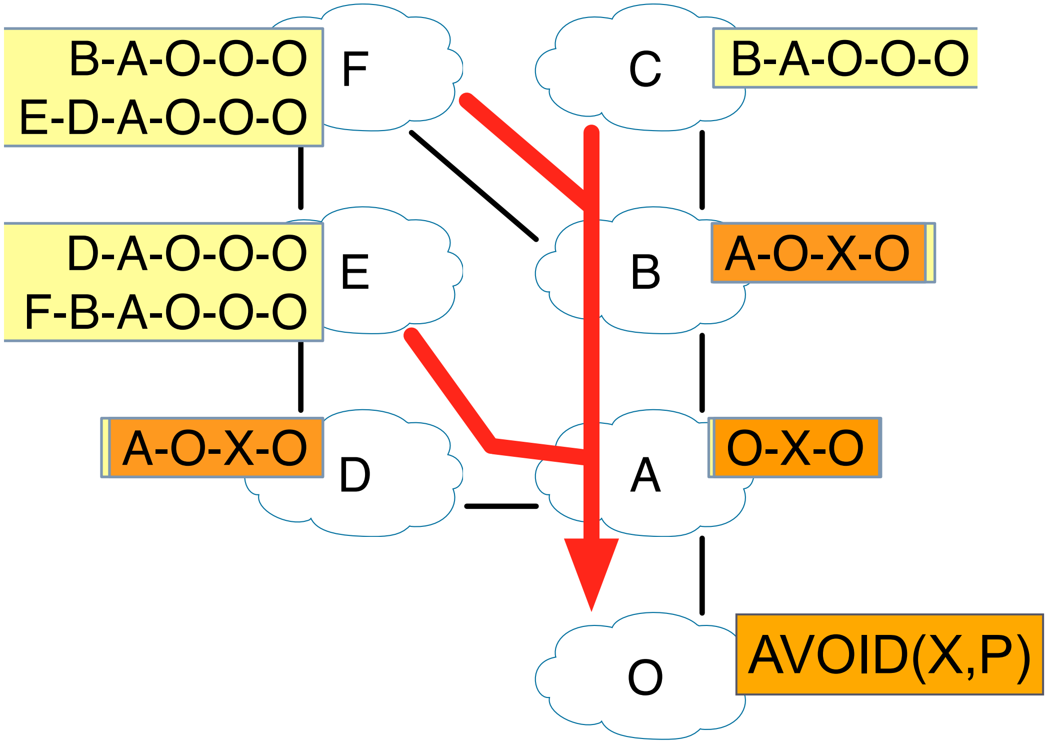
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  (1) next hop ISP
  (2) path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison



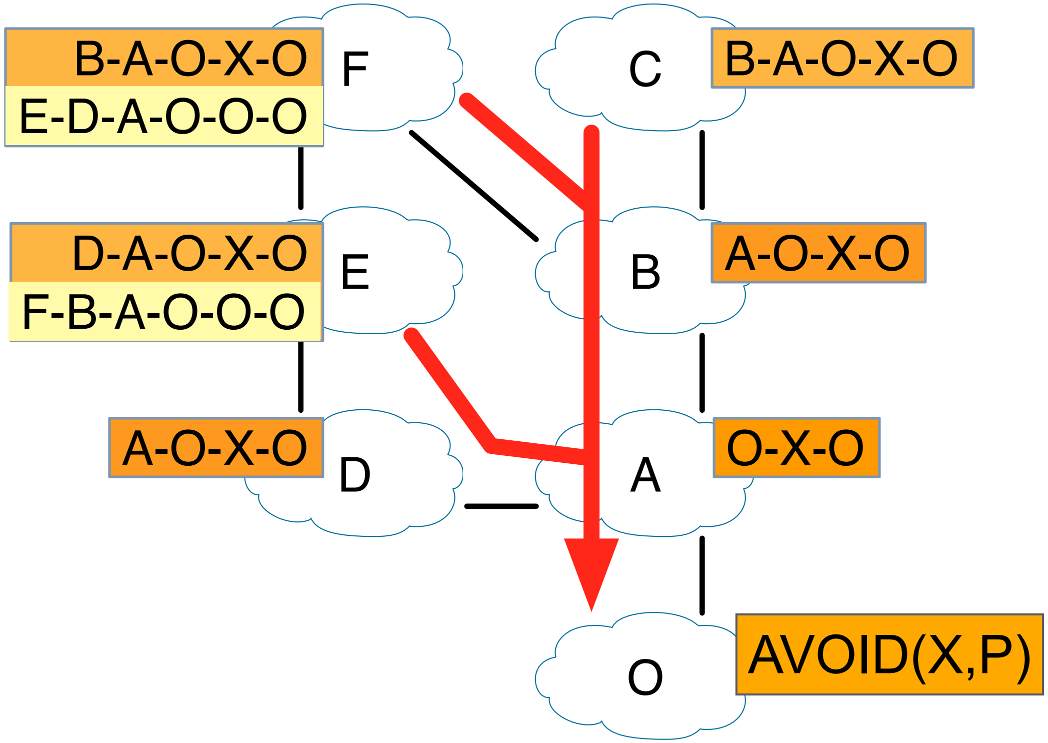
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  (1) next hop ISP
  (2) path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison



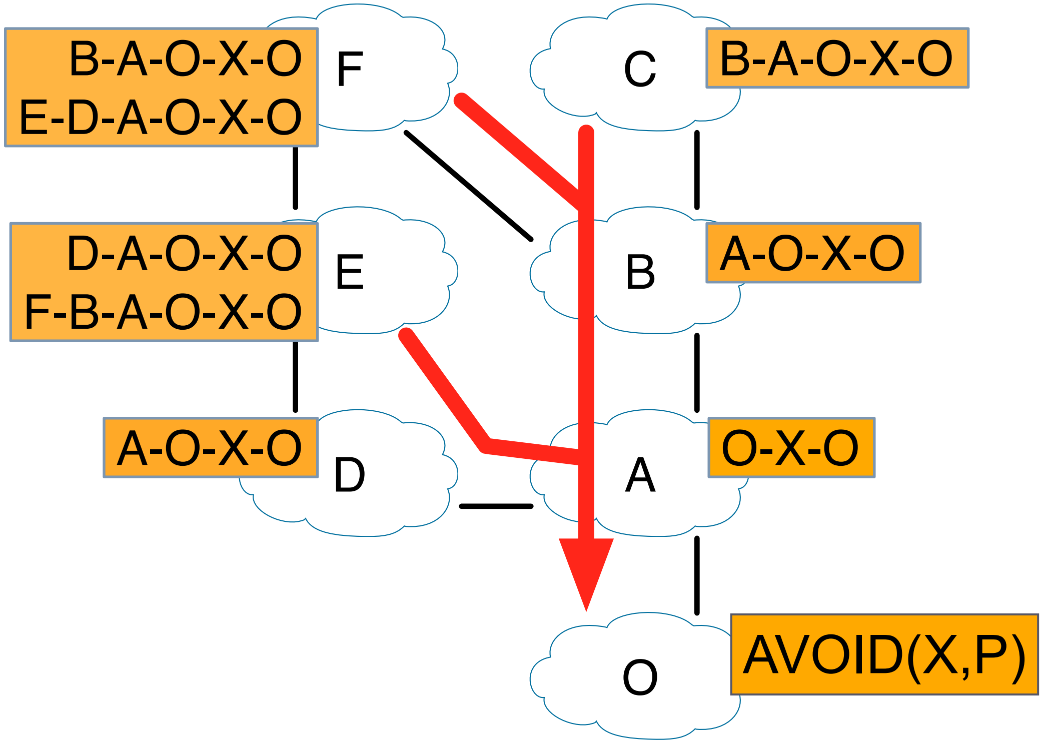
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  (1) next hop ISP
  (2) path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison



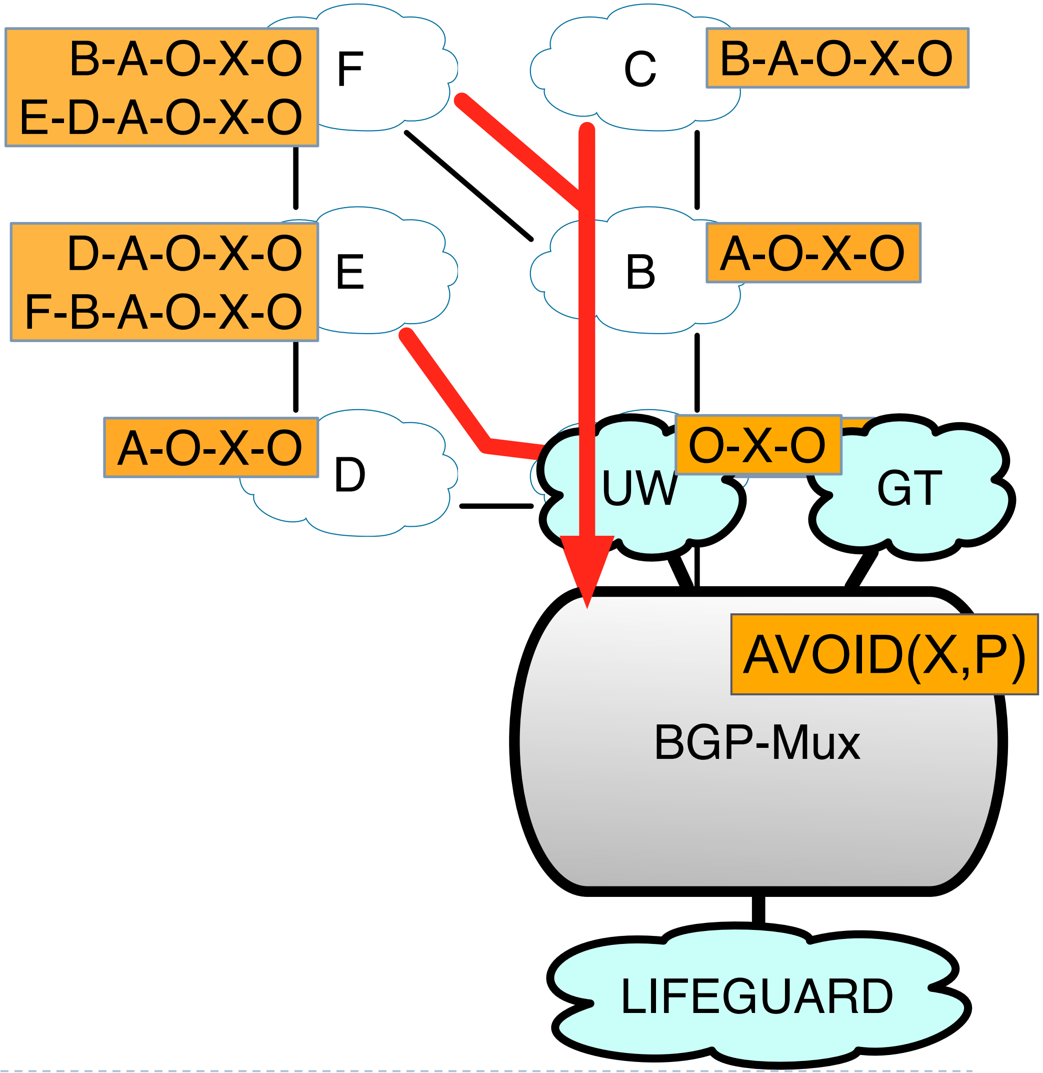
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  (1) next hop ISP
  (2) path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison

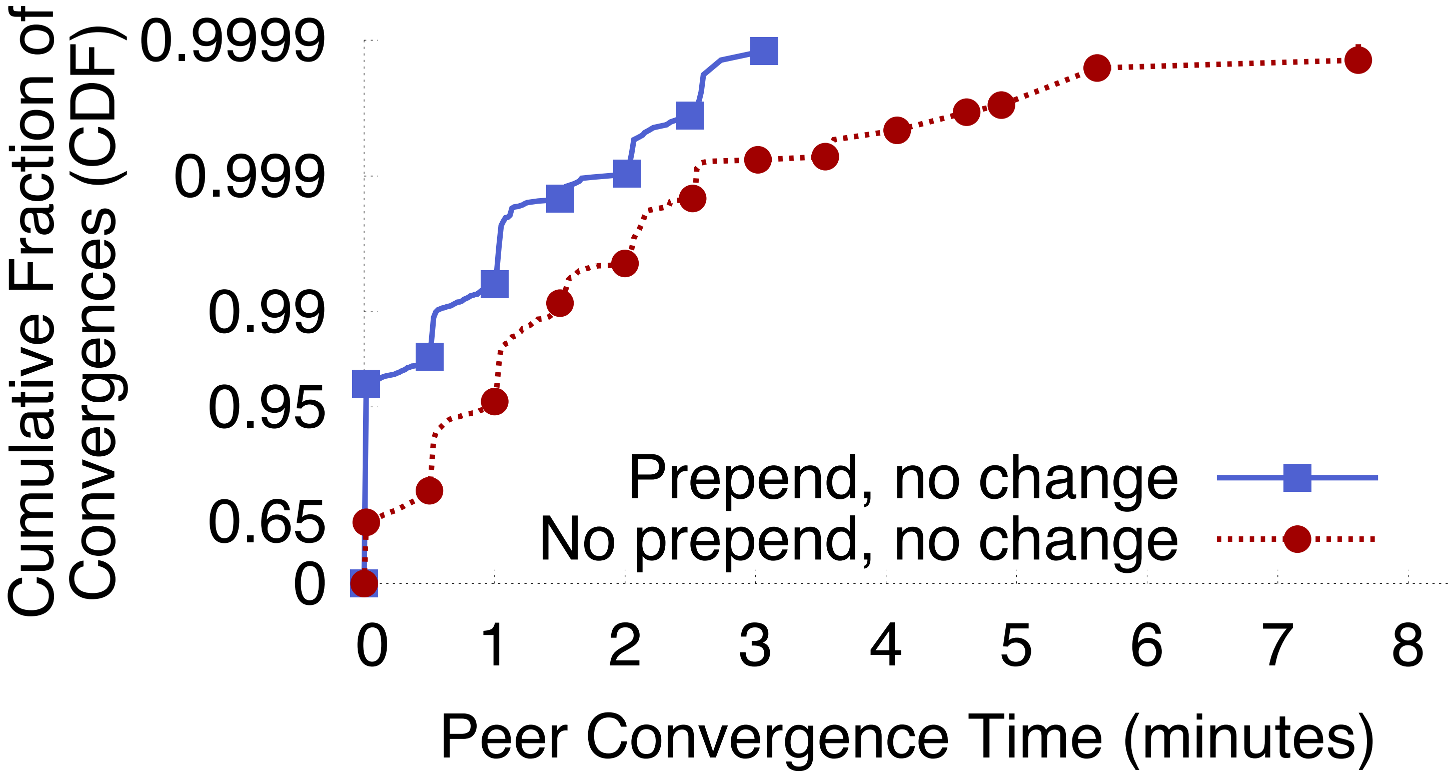


Prepend to Reduce Path Exploration

- Most routing decisions based on:
  (1) next hop ISP
  (2) path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison



Tested Idea Using BGP-Mux



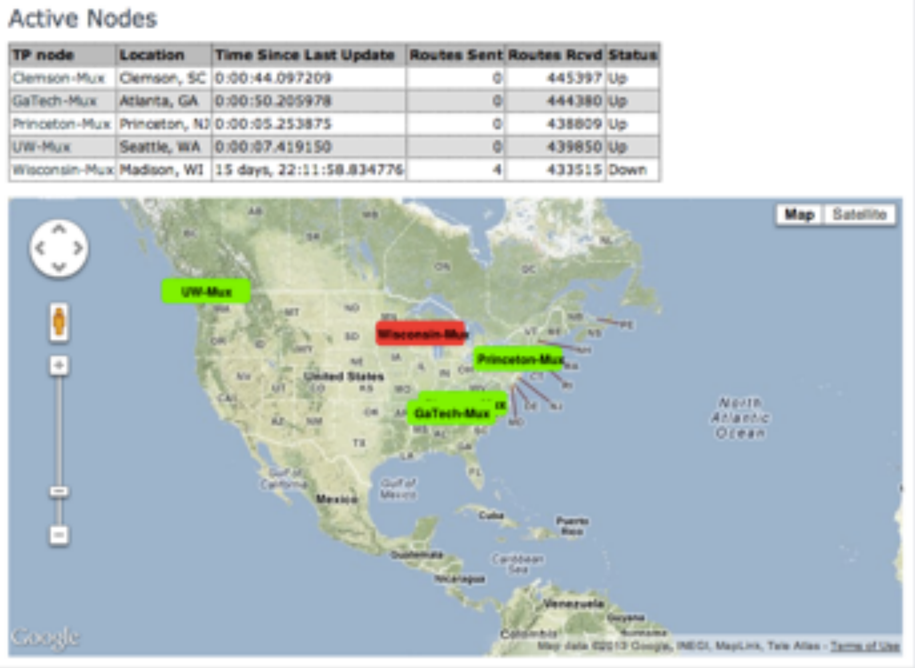
- With no prepend, only 65% of unaffected ISPs converge instantly
- With prepending, 95% of unaffected ISPs re-converge instantly, 98%<1/2 min.
- Also speeds convergence to new paths for affected peers

Summary

- BGP-Mux lets researchers experiment with BGP in the wild
  - Transparent to experiments and stable to upstream
- Initial experiments using it:
  - LIFEGUARD: reroute around ASes or links
  - PoiRoot: root cause analysis of BGP path changes
    - Expose routing preferences
    - Induce changes to use as ground truth
  - PECAN: joint content and network routing
    - Measure performance of alternate paths

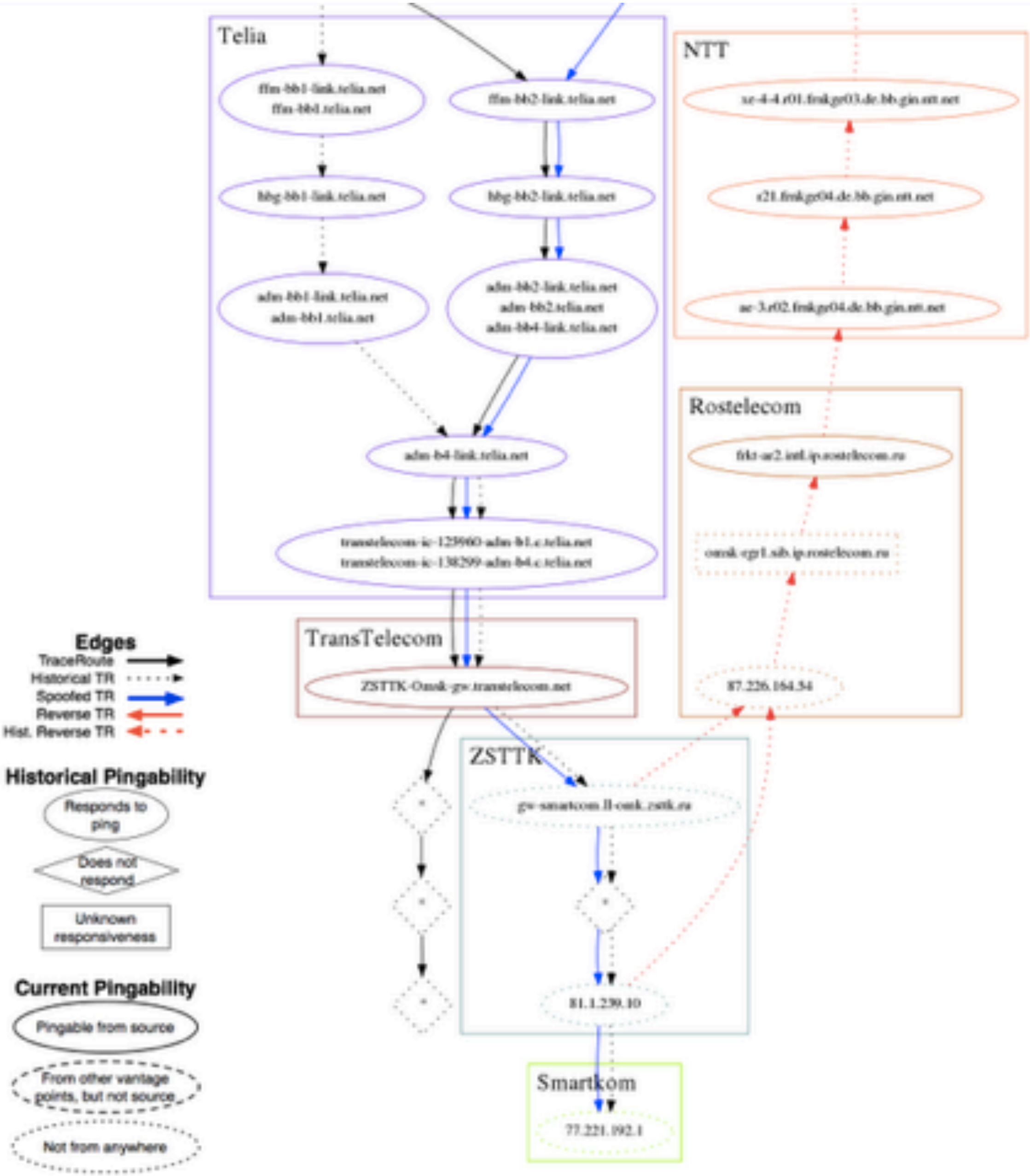
Those Three Questions

- Data sharing
  - Reverse traceroute data now online
  - Other researchers passively observed our active BGP updates
  - Use the testbed yourself
- Visualization: http://tp.gtnoise.net/



Active Nodes

TP node	Location	Time Since Last Update	Routes Sent	Routes Rcvd	Status
Clemson-Mux	Clemson, SC	0:00:44.097209	0	445397	Up
GaTech-Mux	Atlanta, GA	0:00:50.205978	0	444380	Up
Princeton-Mux	Princeton, NJ	0:00:05.253875	0	438809	Up
UW-Mux	Seattle, WA	0:00:07.419150	0	439850	Up
Wisconsin-Mux	Madison, WI	15 days, 22:11:58.834776	4	433515	Down


Telia
ffm-bb1-link.telia.net
ffm-bb1.telia.net
ffm-bb2-link.telia.net
hbg-bb1-link.telia.net
hbg-bb2-link.telia.net
adm-bb1-link.telia.net
adm-bb1.telia.net
adm-bb2-link.telia.net
adm-bb2.telia.net
adm-bb4-link.telia.net
adm-b4-link.telia.net
transtelecom-ic-125960-adm-b1.c.telia.net
transtelecom-ic-138299-adm-b4.c.telia.net
NTT
xe-4-4.r01.frnkge03.de.bb.gin.ntt.net
r21.frnkge04.de.bb.gin.ntt.net
ae-3.r02.frnkge04.de.bb.gin.ntt.net
Rostelecom
frkt-ar2.intl.ip.rostelecom.ru
omsk-cgr1.sib.ip.rostelecom.ru
87.226.164.34
TransTelecom
ZSTTK-Omsk-gw.transtelecom.net
ZSTTK
gw-smartcom.ll-omk.zsttk.ru
*
*
*
*
81.1.239.10
Smartkom
77.221.192.1
Edges
TraceRoute
Historical TR
Spoofed TR
Reverse TR
Hist. Reverse TR
Historical Pingability
Responds to ping
Does not respond
Unknown responsiveness
Current Pingability
Pingable from source
From other vantage points, but not source
Not from anywhere

Conclusion

- BGP-Mux lets researchers experiment with BGP in the wild
  - Transparent to experiments and stable to upstream
  - Georgia Tech system, I am just an enthusiastic user

- **LIFEGUARD**: Let edge networks reroute around failures

- Questions for the audience:
  - What would you use this system for? What should we use it for?
  - How do we get more ASes to connect to us?
    - Getting them to agree to
    - Then, getting the connection to work
      - □ VLAN between BGP-Mux and border router
      - □ Ability to advertise BGP routes