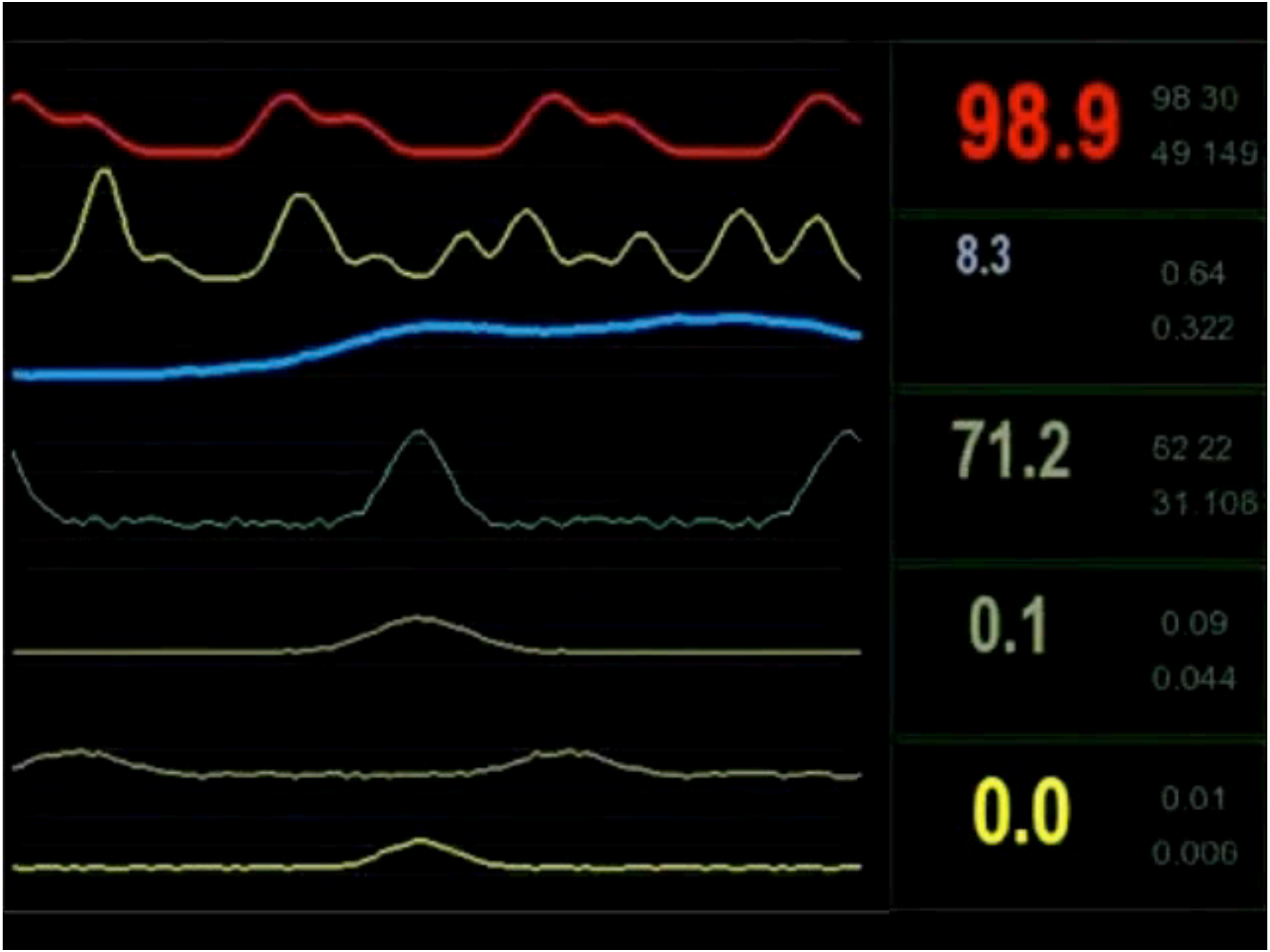
LIFEGUARD: Practical Repair of Persistent Route Failures

Ethan Katz-Bassett (USC)

Colin Scott, David Choffnes, Italo Cunha,
Valas Valancius, Nick Feamster, Harsha Madhyastha,
Tom Anderson, Arvind Krishnamurthy

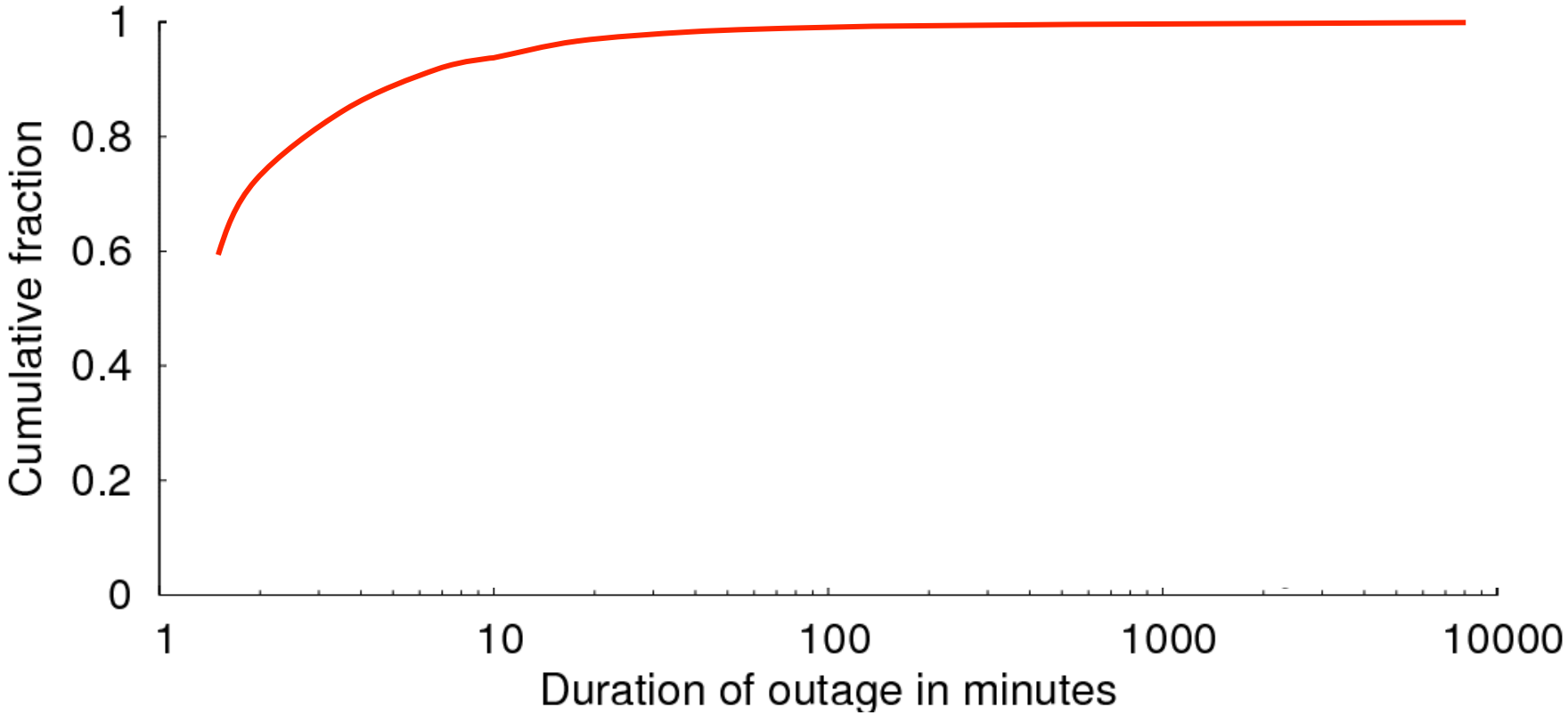
This work is generously funded in part by Google, Cisco and the NSF.


YouTube
0:00 / 6:66


98.9
98 30
49 149
8.3
0 64
0.322
71.2
62 22
31 108
0.1
0 09
0.044
0.0
0 01
0.006

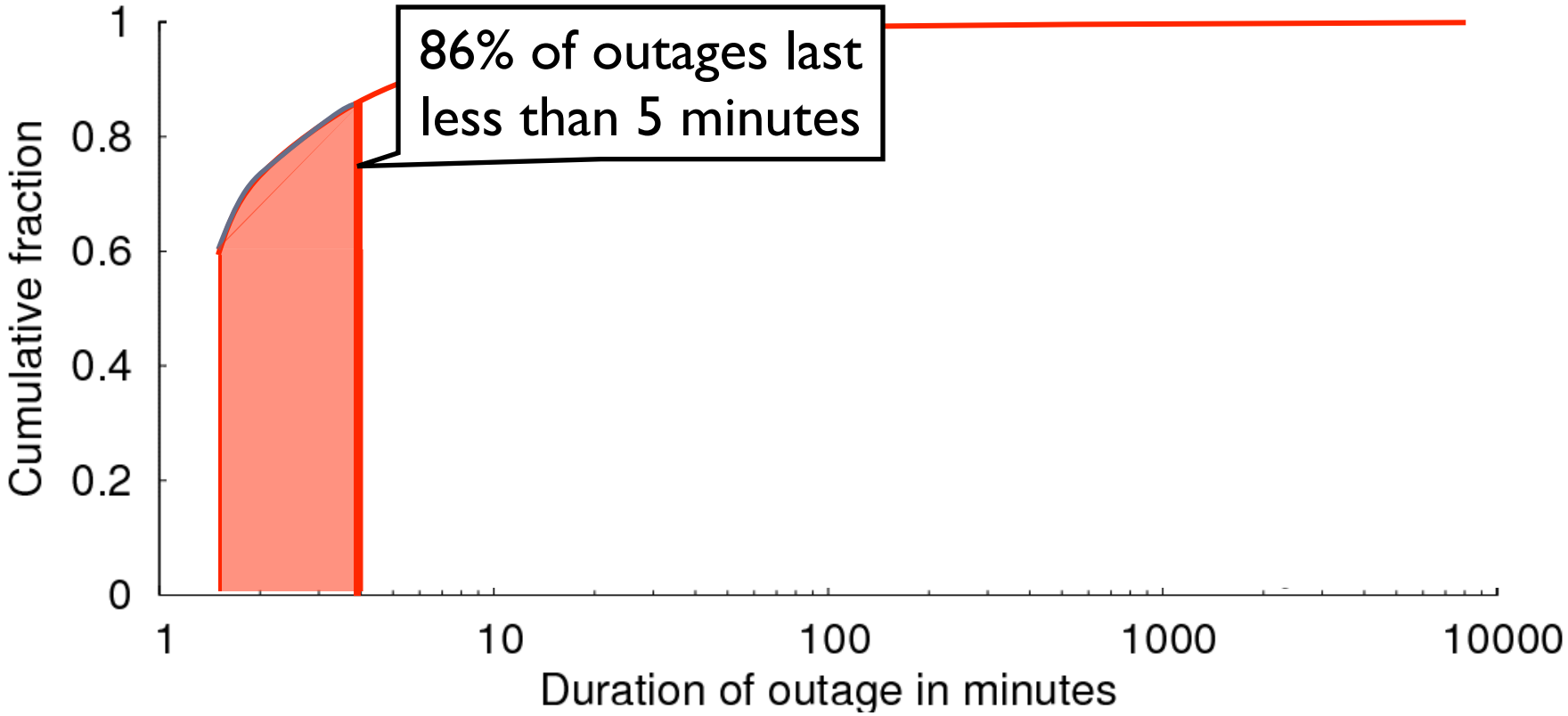
Long Outages Cause Most Unavailability

- Monitor outages from Amazon's EC2
- Fraction of outages of duration ≤ ***X***?
- Fraction of unavailability due to outages of duration ≤ ***X***?



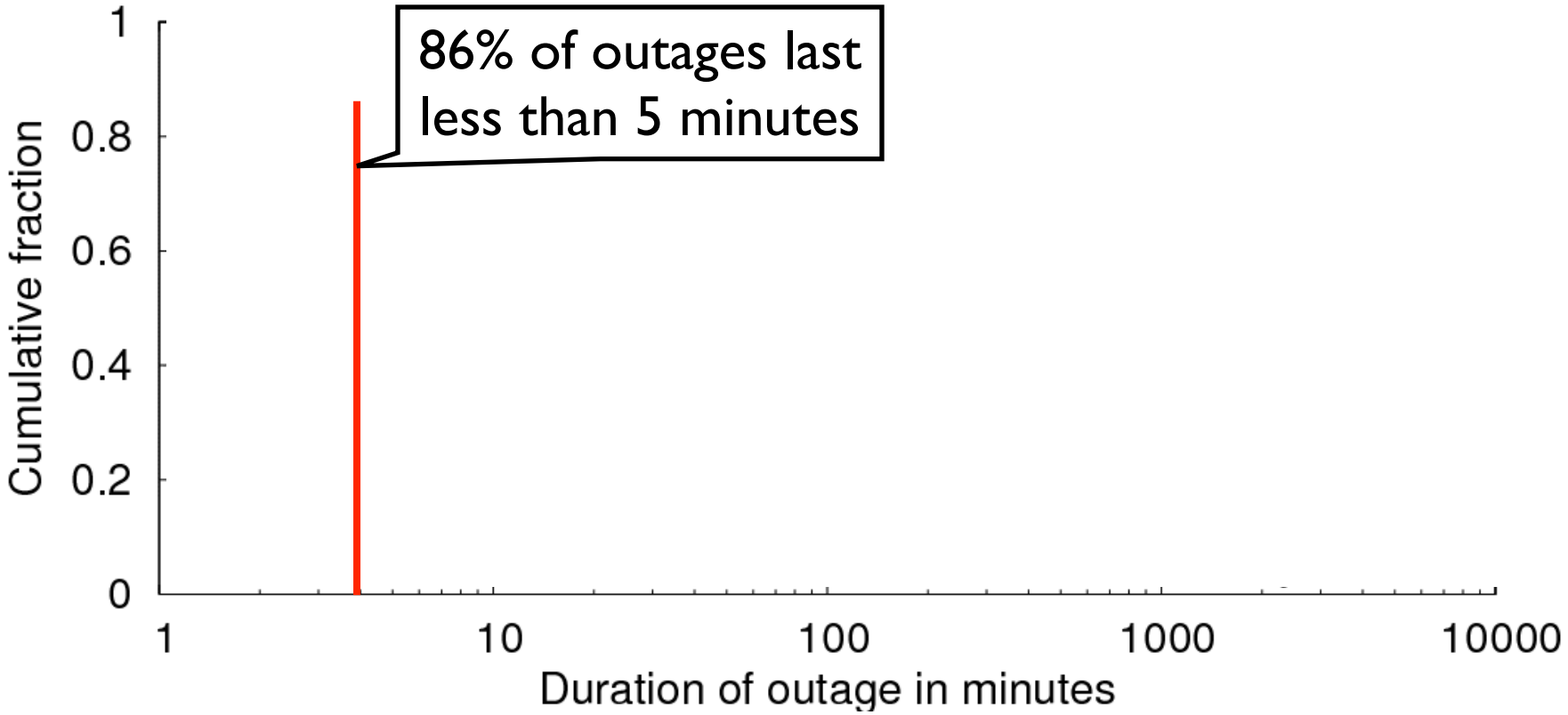
Long Outages Cause Most Unavailability

- Monitor outages from Amazon's EC2
- Fraction of outages of duration ≤ ***X***?
- Fraction of unavailability due to outages of duration ≤ ***X***?



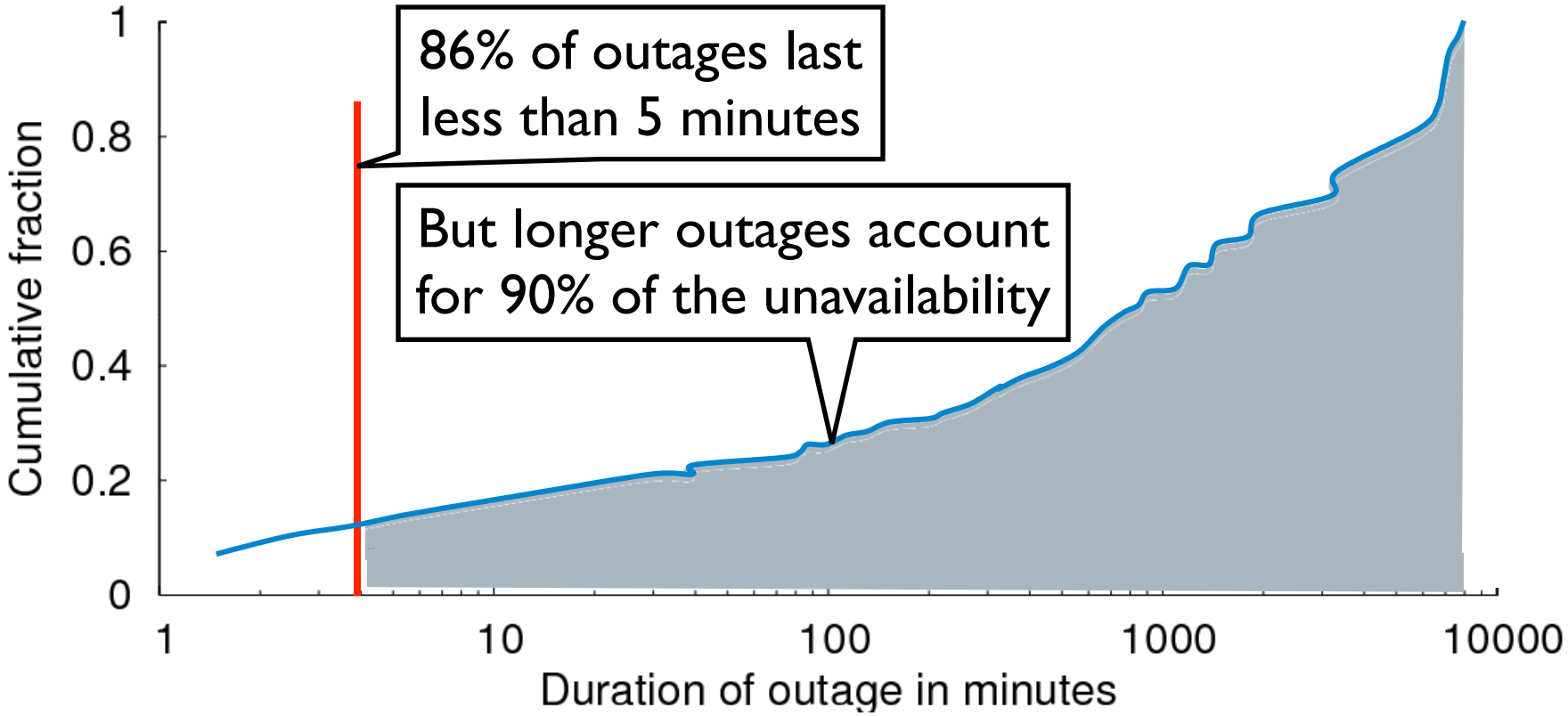
Long Outages Cause Most Unavailability

- Monitor outages from Amazon's EC2
- ~~Fraction of outages of duration ≤ *X*?~~
- Fraction of unavailability due to outages of duration ≤ ***X***?



Long Outages Cause Most Unavailability

- Monitor outages from Amazon's EC2
- ~~Fraction of outages of duration ≤ *X*?~~
- Fraction of unavailability due to outages of duration ≤ ***X***?



Operators Struggle to Locate Failures

"Traffic attempting to pass through Level3's network in the Washington, DC area is getting lost in the abyss. Here's a trace from Verizon residential to Level3." *Outages mailing list, Dec. 2010*

Operators Struggle to Locate Failures

"Traffic attempting to pass through Level3's network in the Washington, DC area is getting lost in the abyss. Here's a trace from Verizon residential to Level3." *Outages mailing list, Dec. 2010*

Mailing List User 1

1 Home router

2 Verizon in Baltimore

3 Verizon in Philly

4 Alter.net in DC

5 Level3 in DC

6 * * *

7 * * *

Operators Struggle to Locate Failures

"Traffic attempting to pass through Level3's network in the Washington, DC area is getting lost in the abyss. Here's a trace from Verizon residential to Level3." *Outages mailing list, Dec. 2010*

Mailing List User 1

1 Home router
2 Verizon in Baltimore
3 Verizon in Philly
4 Alter.net in DC
5 Level3 in DC
6 * * *
7 * * *

Mailing List User 2

1 Home router
2 Verizon in DC
3 Alter.net in DC
4 Level3 in DC
5 Level3 in Chicago
6 Level3 in Denver
7 * * *
8 * * *

Reasons for Long-Lasting Outages

Long-term outages are:

- Repaired over slow, human timescales
- Not well understood
- Caused by routers advertising paths that do not work
  - E.g., corrupted memory on line card causes black hole
  - E.g., bad cross-layer interactions cause failed MPLS tunnel
- Complicated by lack of visibility into or control over routes in other ISPs

Our Approach and Outline

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

Our Approach and Outline

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
  - Building blocks
  - Example
  - Description of technique
- Suggest that other ISPs reroute around the problem

Building blocks for failure isolation

LIFEGUARD can use:

- Ping to test reachability
- Traceroute to measure forward path
- Distributed vantage points (VPs)
  - PlanetLab for our experiments
  - Some can source spoof
- Reverse traceroute to measure reverse path (NSDI '10)
- Atlas of historical forward/reverse paths between VPs and targets

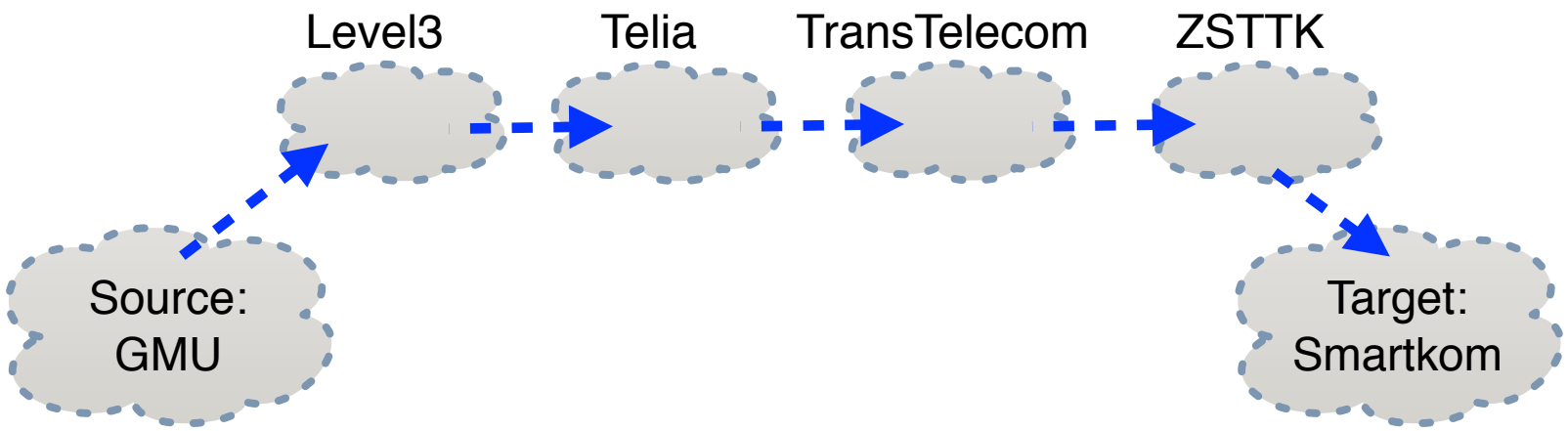
How does **LIFEGUARD** locate a failure?

Before outage:

- Historical atlas enables reasoning about changes
- Traceroute yields only path from GMU to target
- Reverse traceroute reveals path asymmetry

How does **LIFEGUARD** locate a failure?

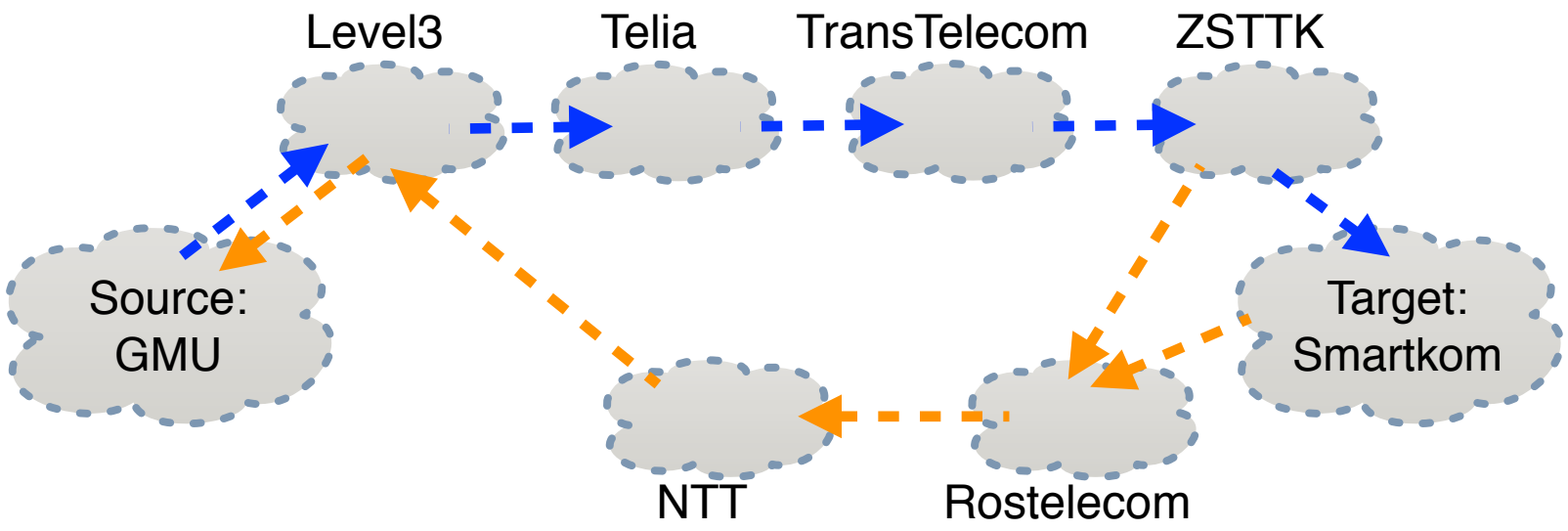
Before outage:



- Historical atlas enables reasoning about changes
- Traceroute yields only path from GMU to target
- Reverse traceroute reveals path asymmetry

How does **LIFEGUARD** locate a failure?

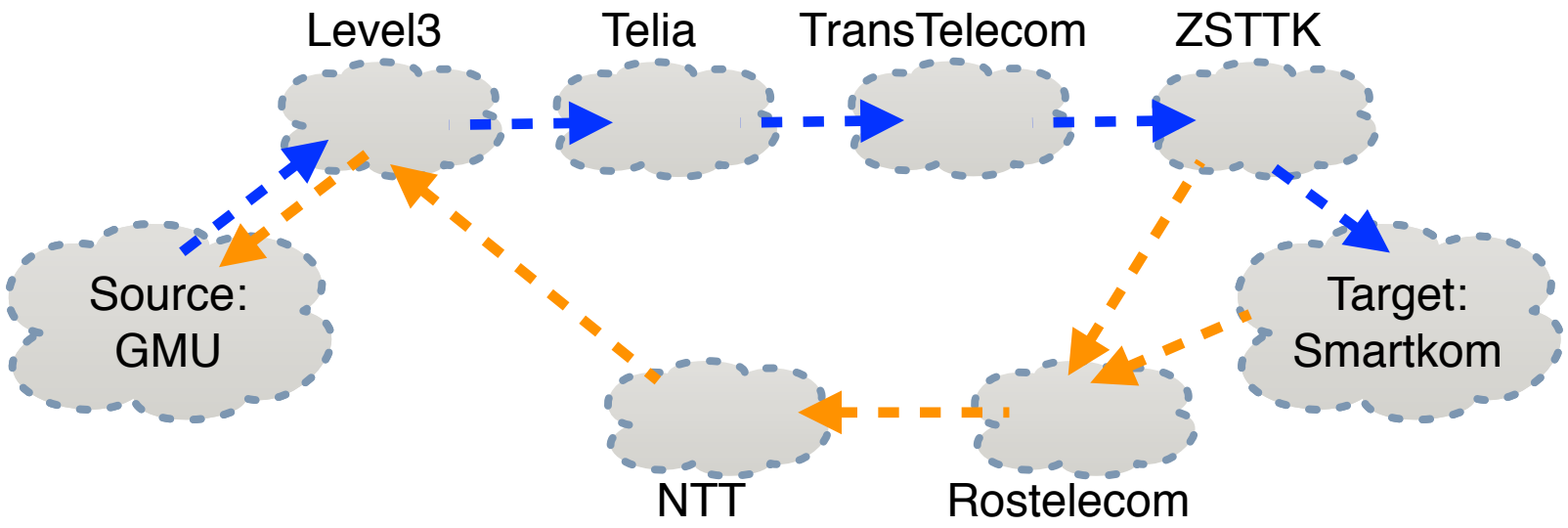
Before outage:



- Historical atlas enables reasoning about changes
- Traceroute yields only path from GMU to target
- Reverse traceroute reveals path asymmetry

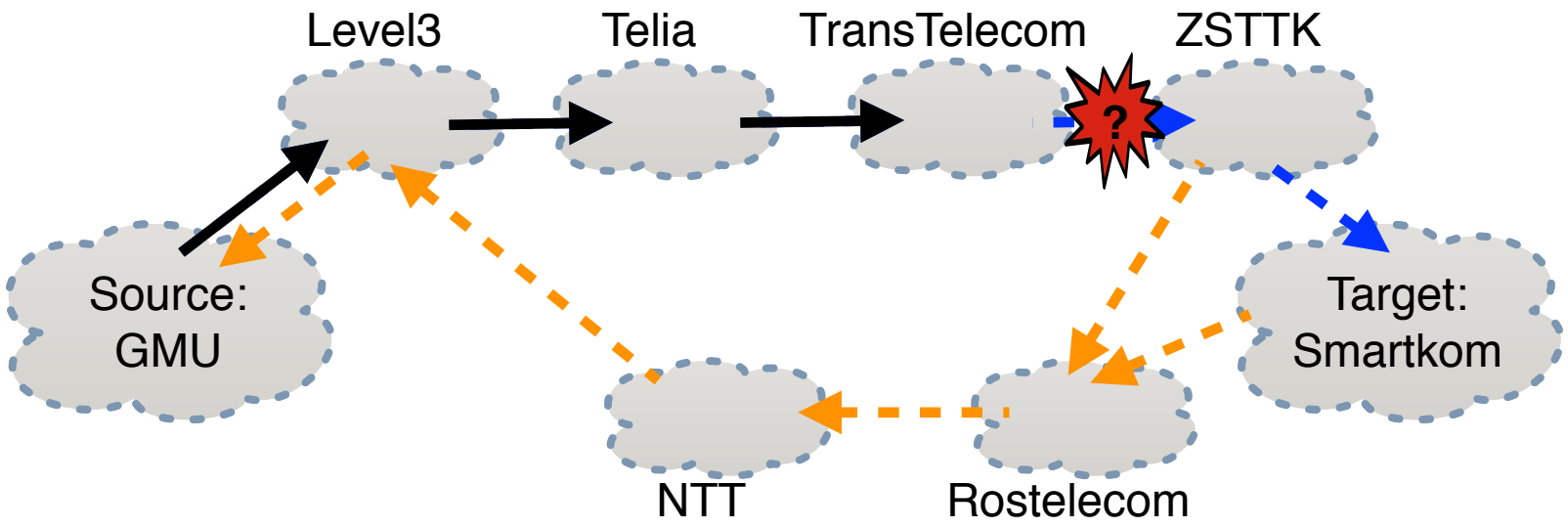
How does **LIFEGUARD** locate a failure?

During outage:



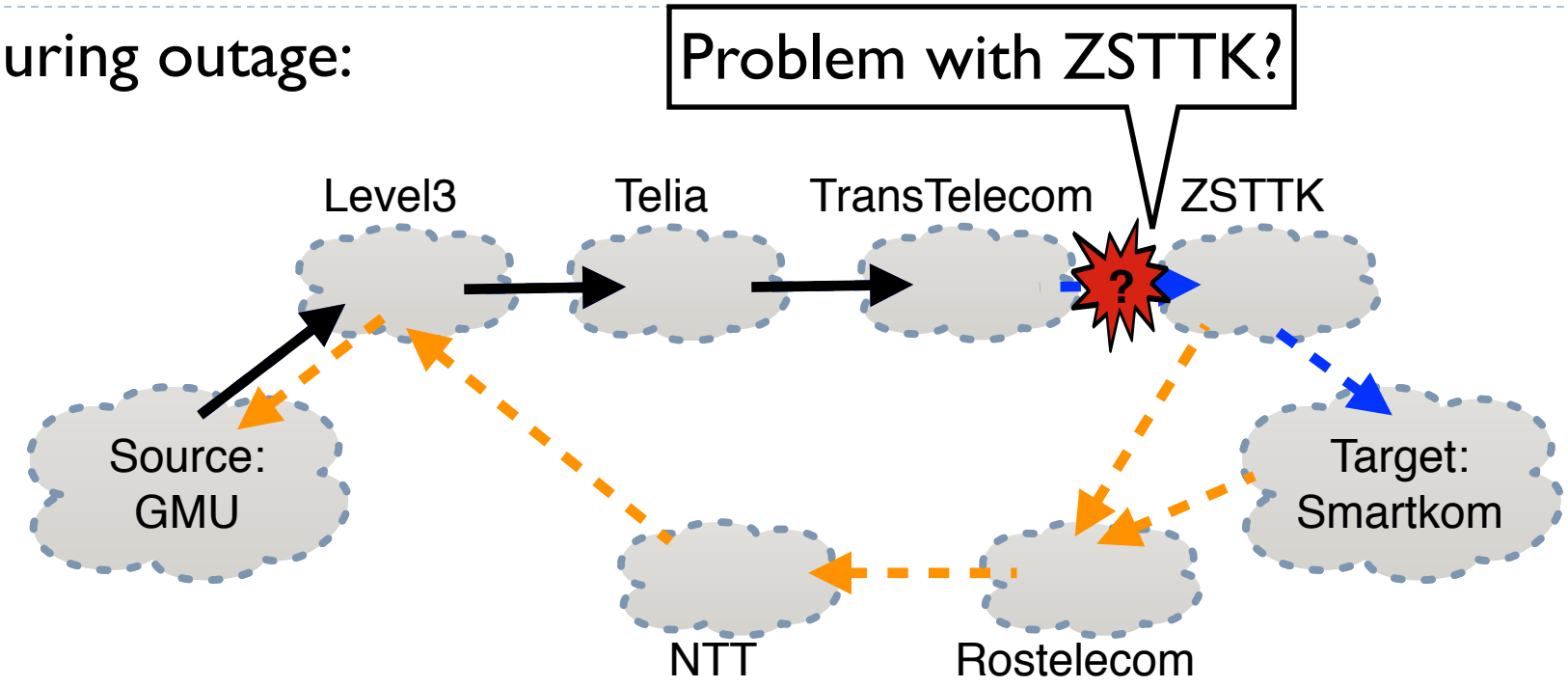
How does **LIFEGUARD** locate a failure?

During outage:



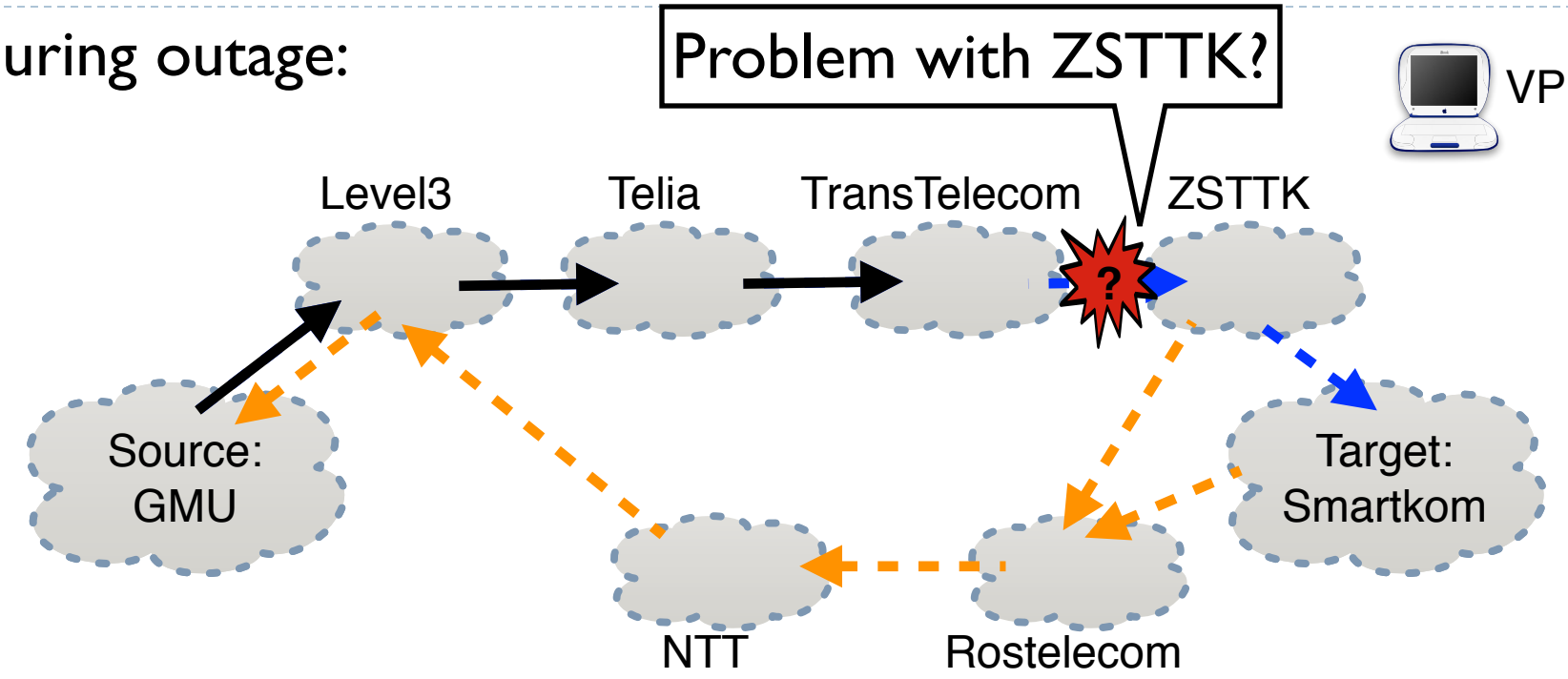
How does **LIFEGUARD** locate a failure?

During outage:



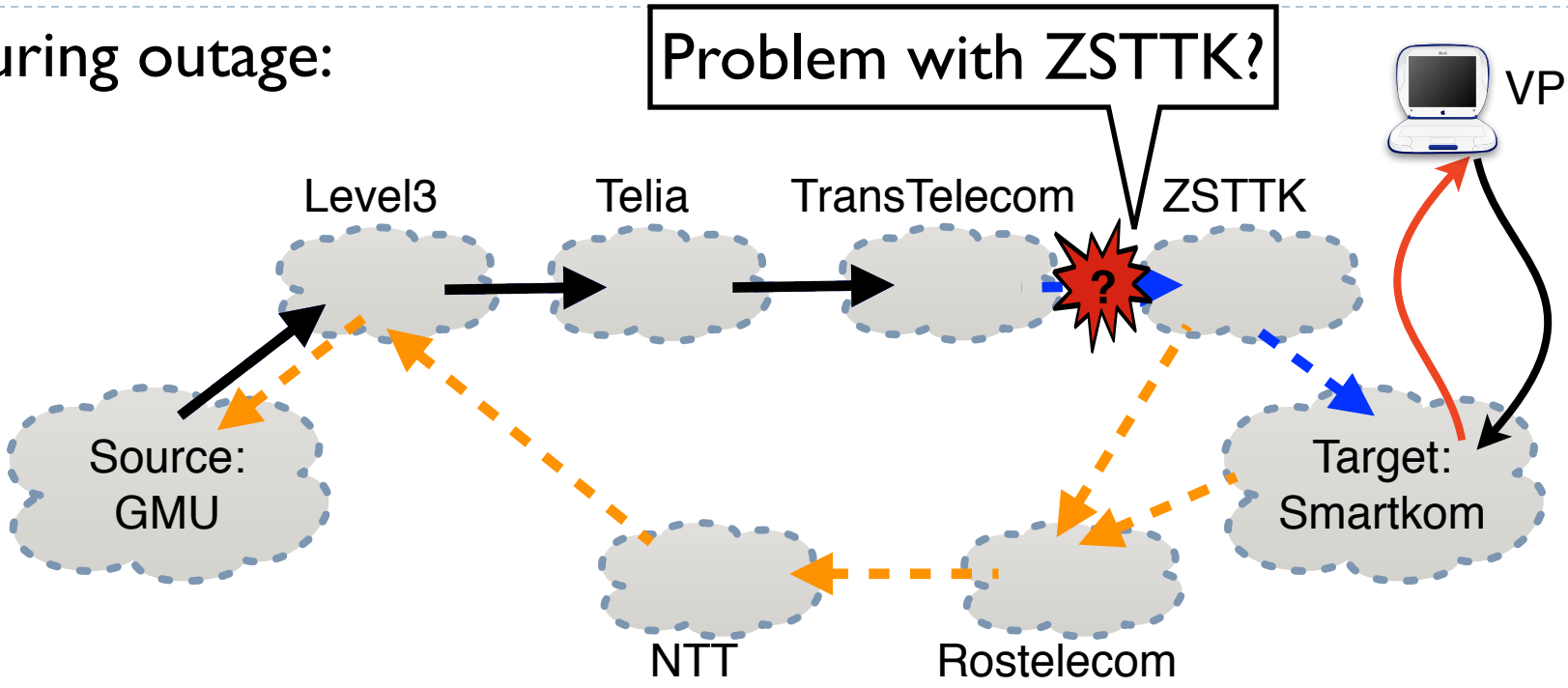
How does **LIFEGUARD** locate a failure?

During outage:



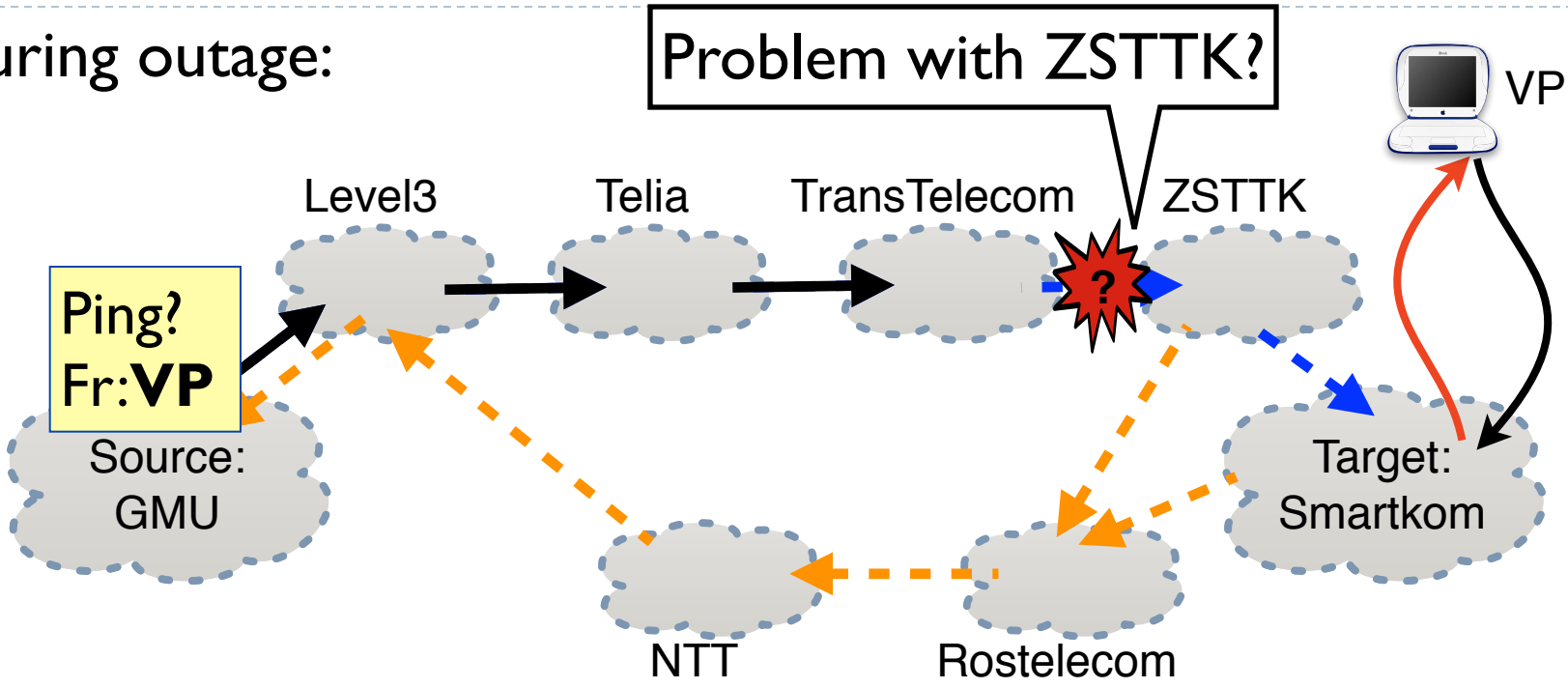
How does **LIFEGUARD** locate a failure?

During outage:



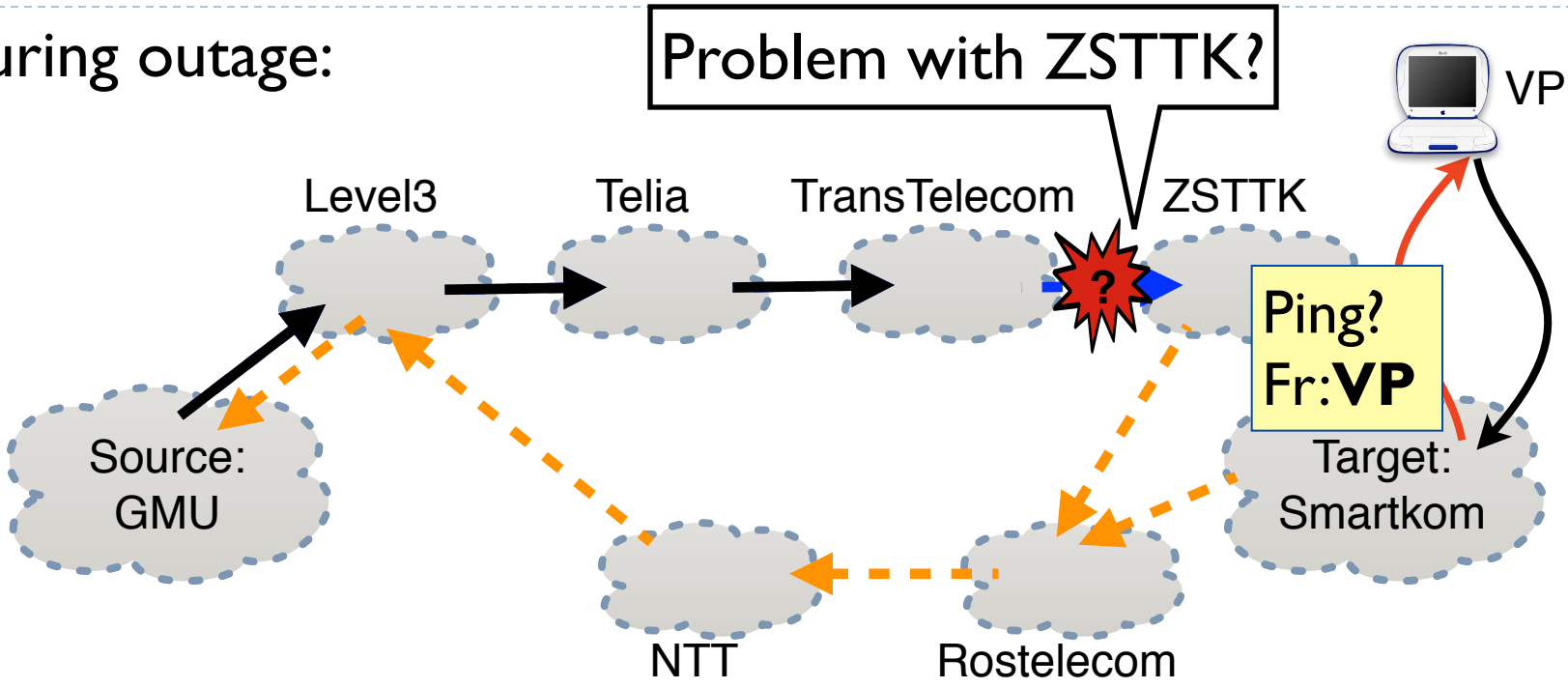
How does **LIFEGUARD** locate a failure?

During outage:



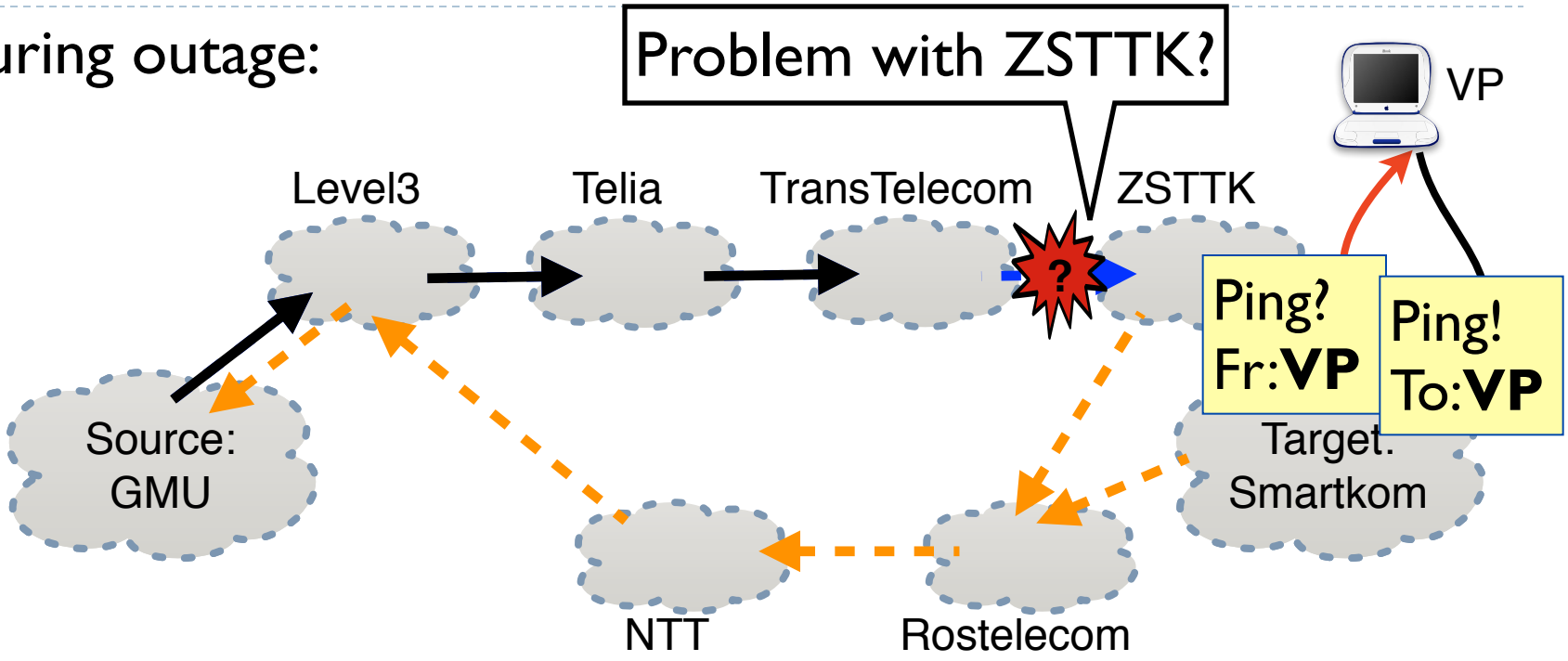
How does **LIFEGUARD** locate a failure?

During outage:



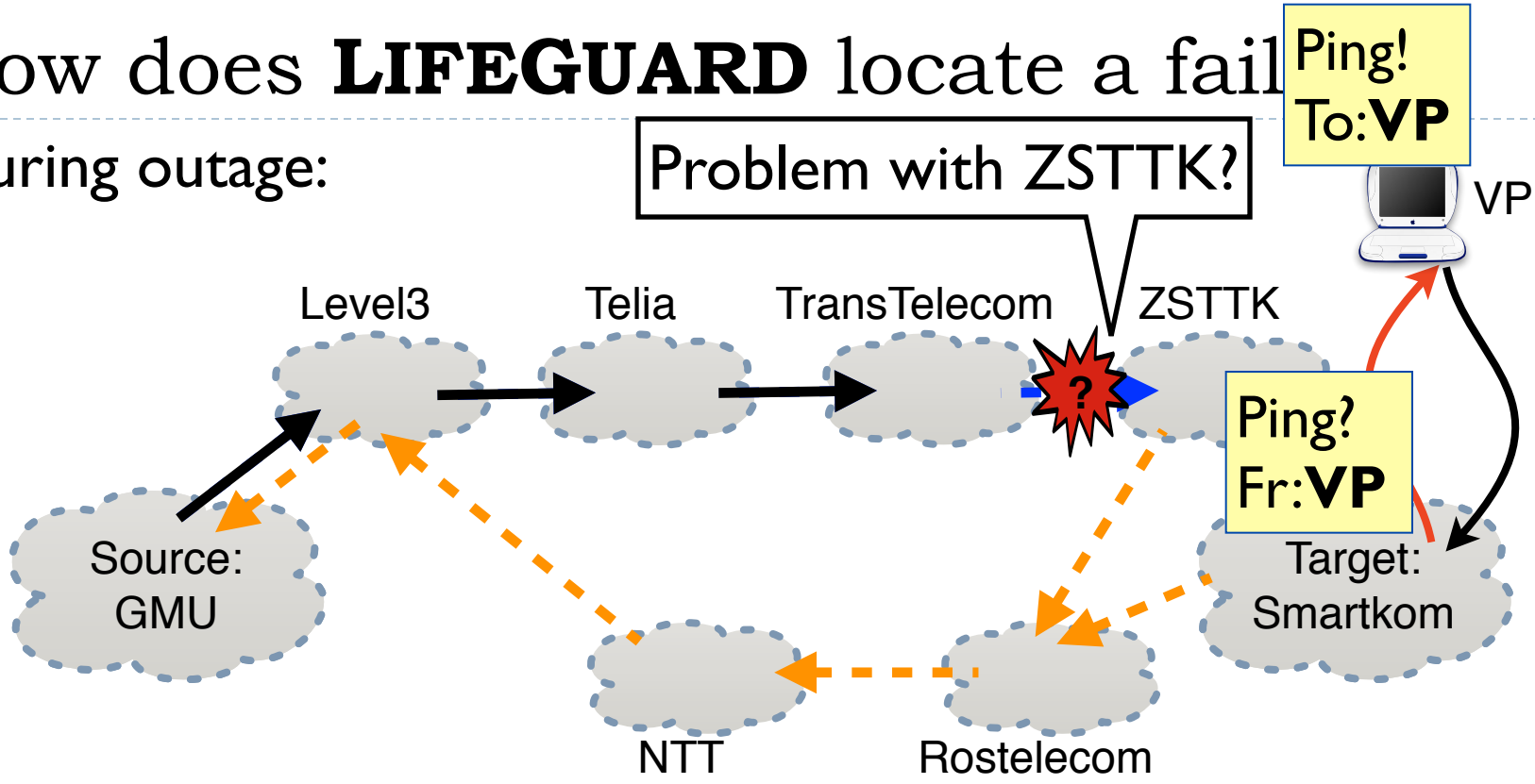
How does **LIFEGUARD** locate a failure?

During outage:



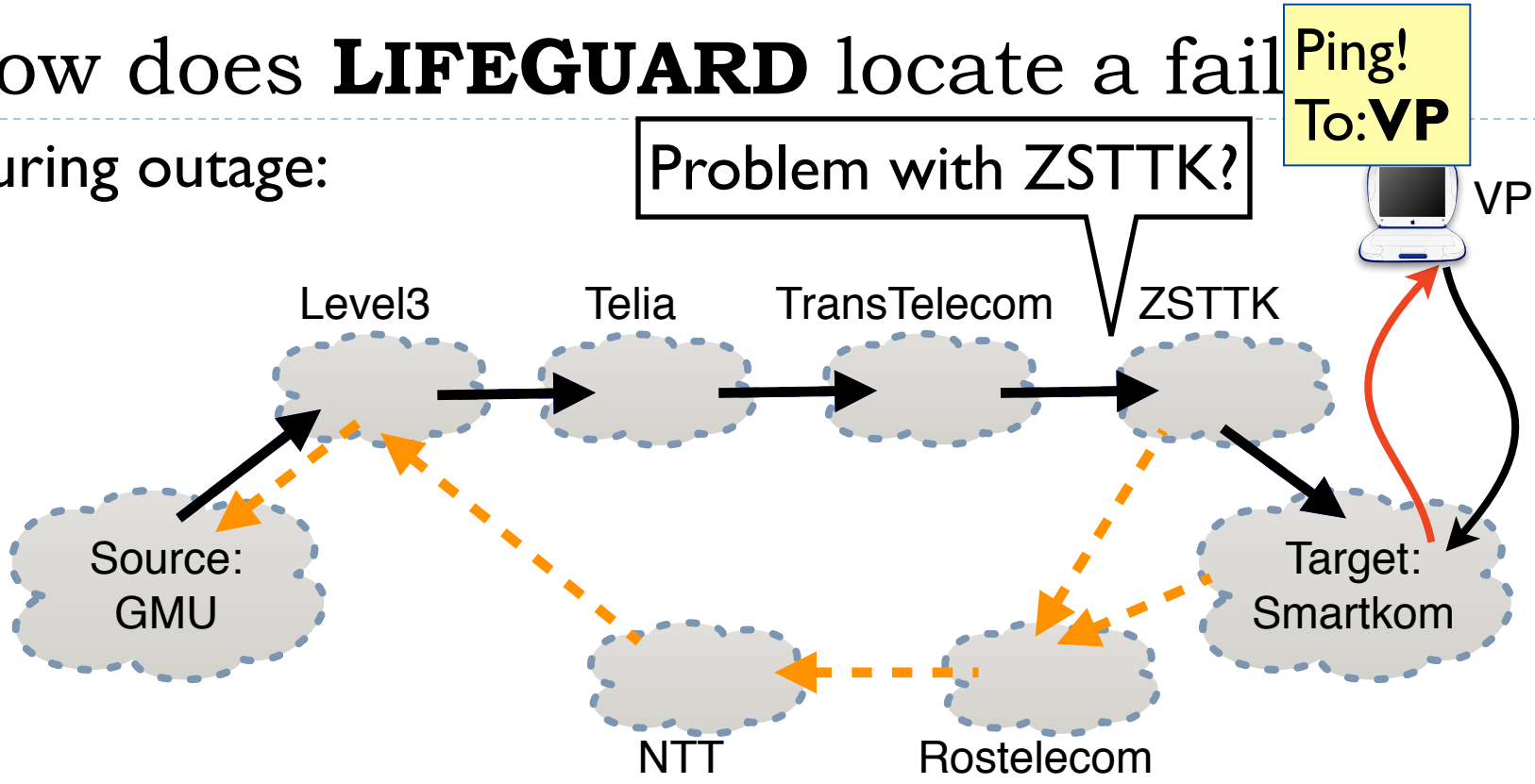
How does **LIFEGUARD** locate a fail

During outage:



How does **LIFEGUARD** locate a fail

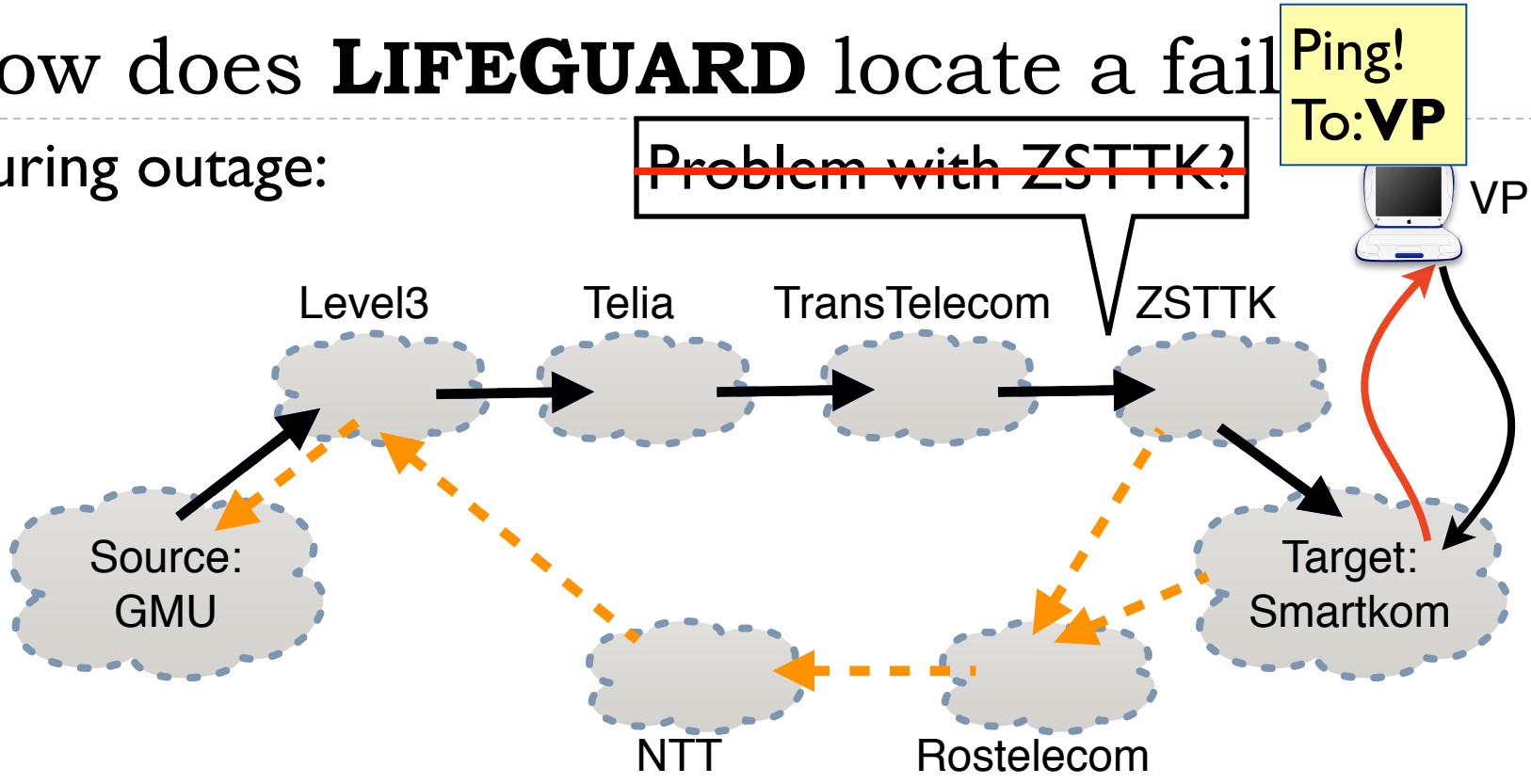
During outage:



- Forward path works

How does **LIFEGUARD** locate a fail

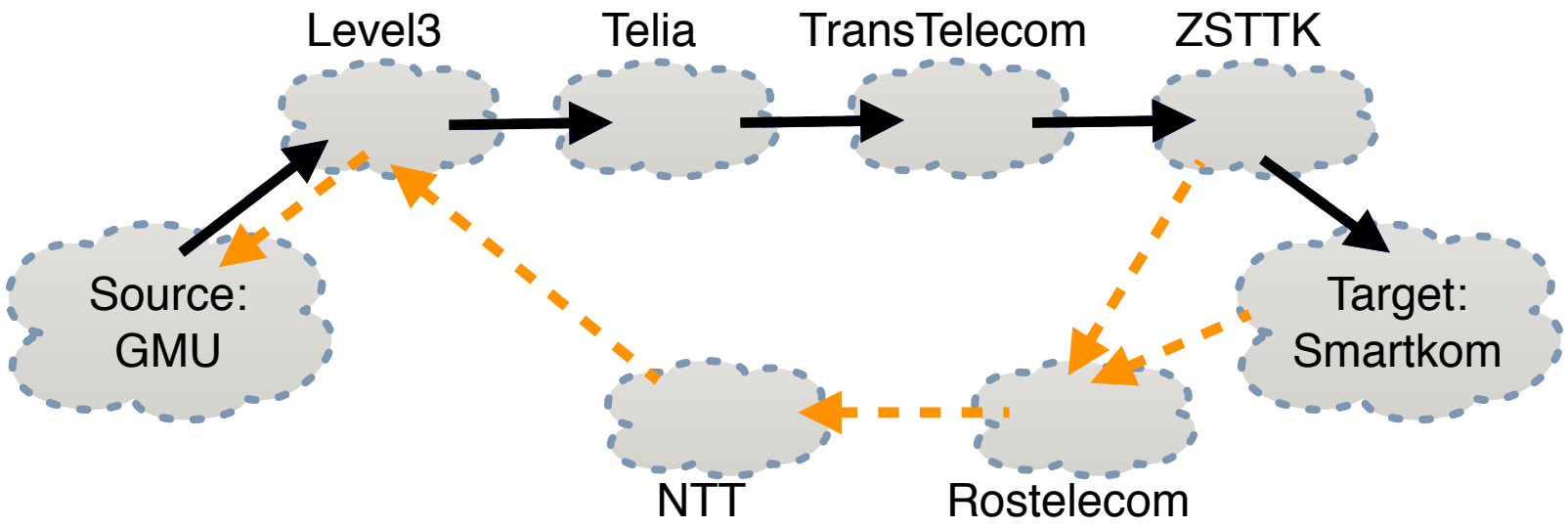
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

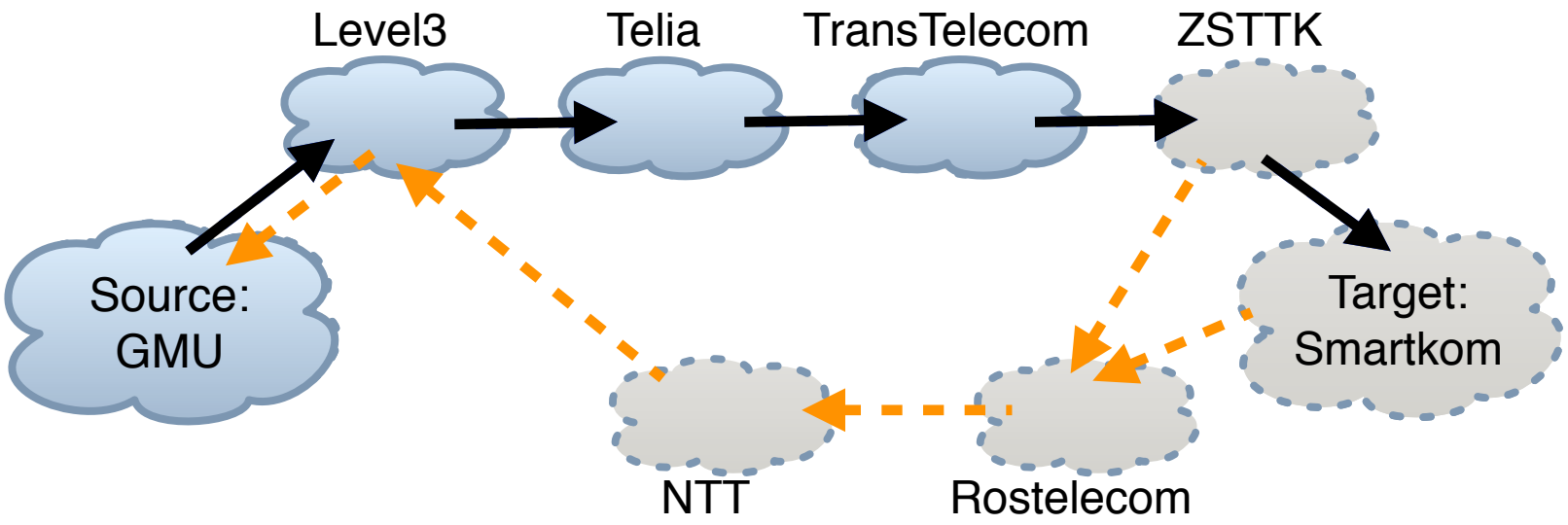
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

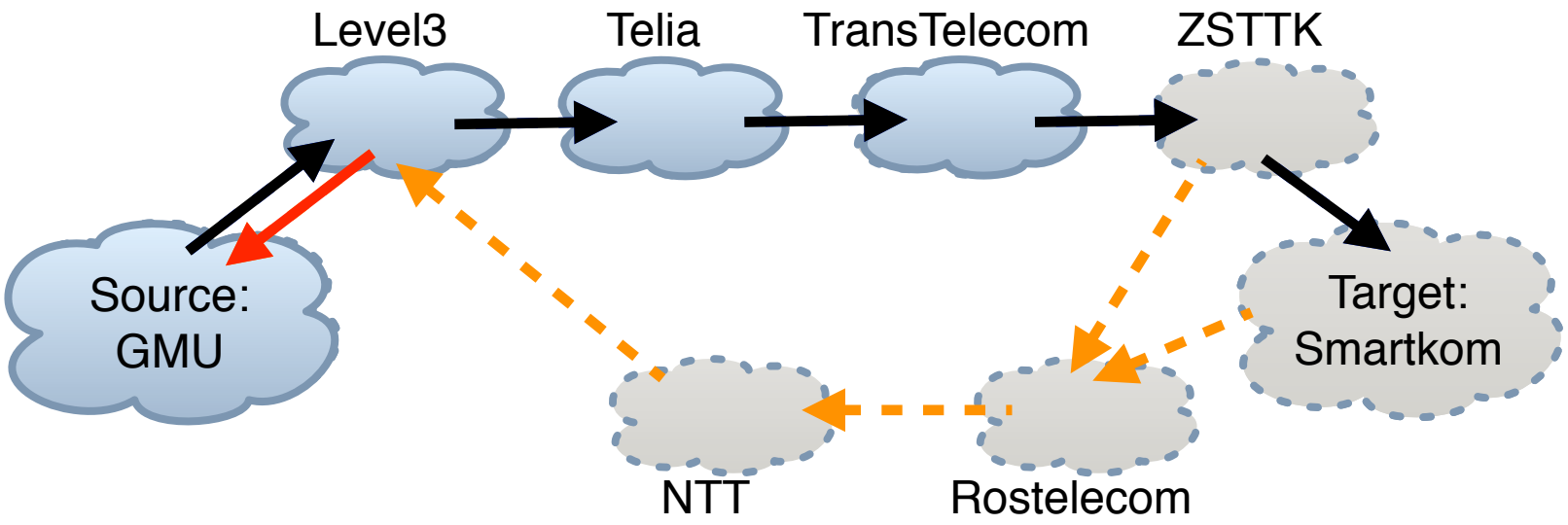
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

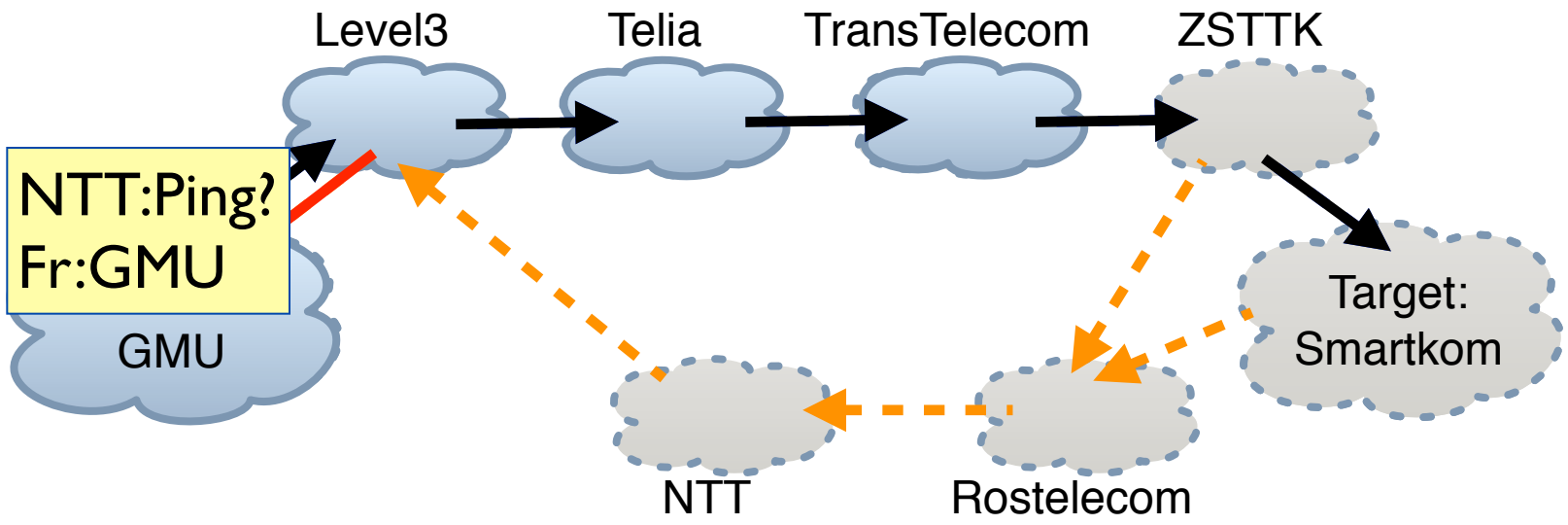
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

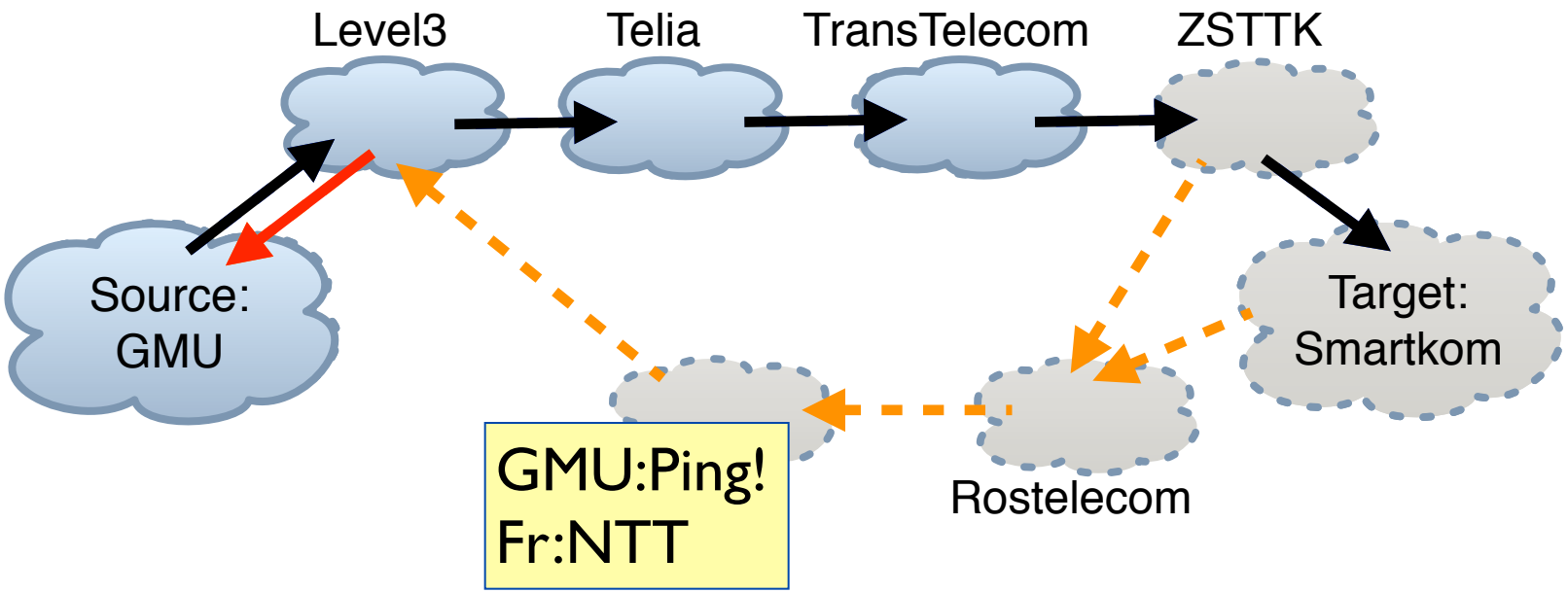
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

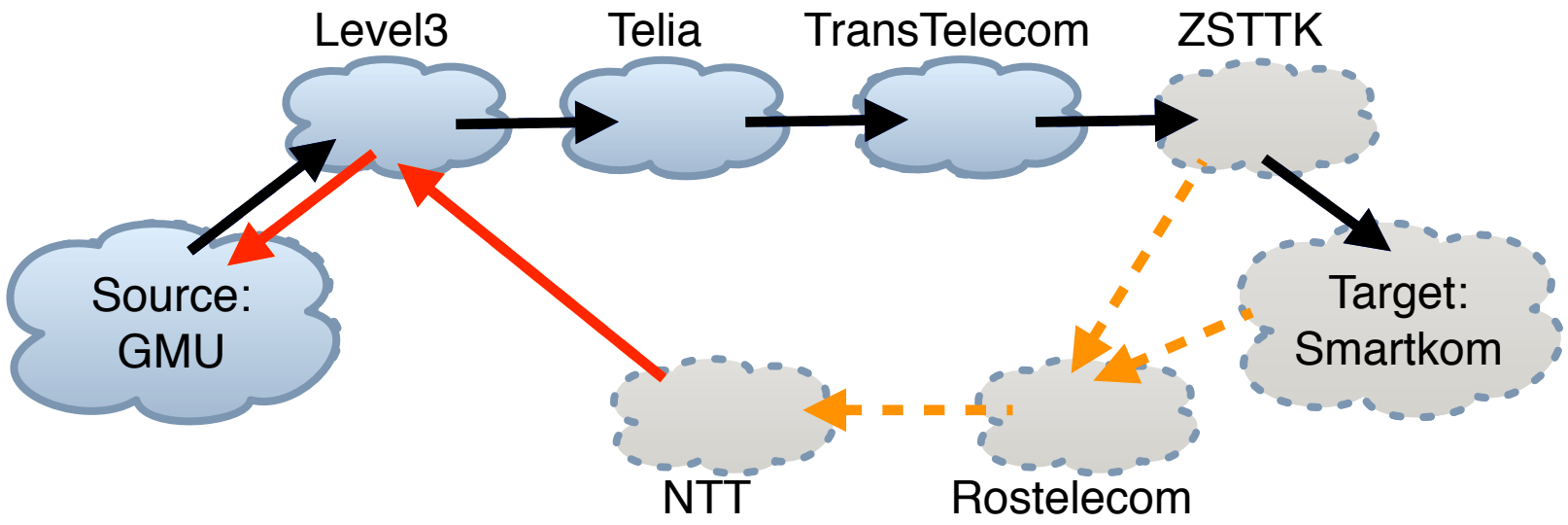
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

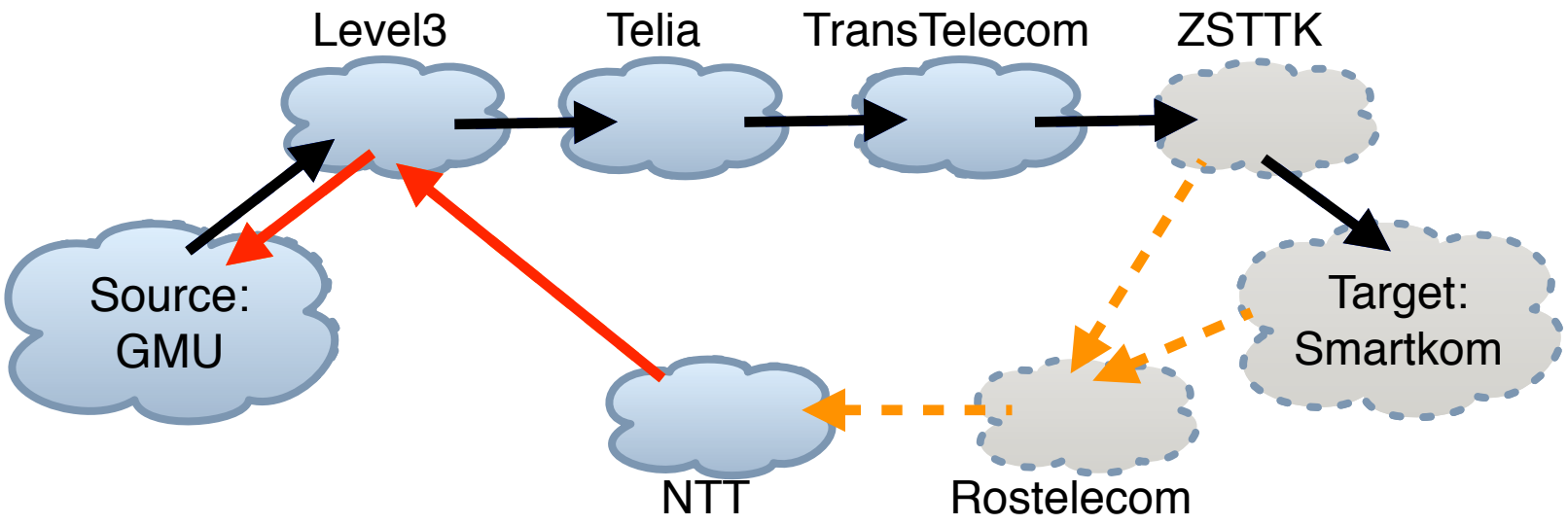
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

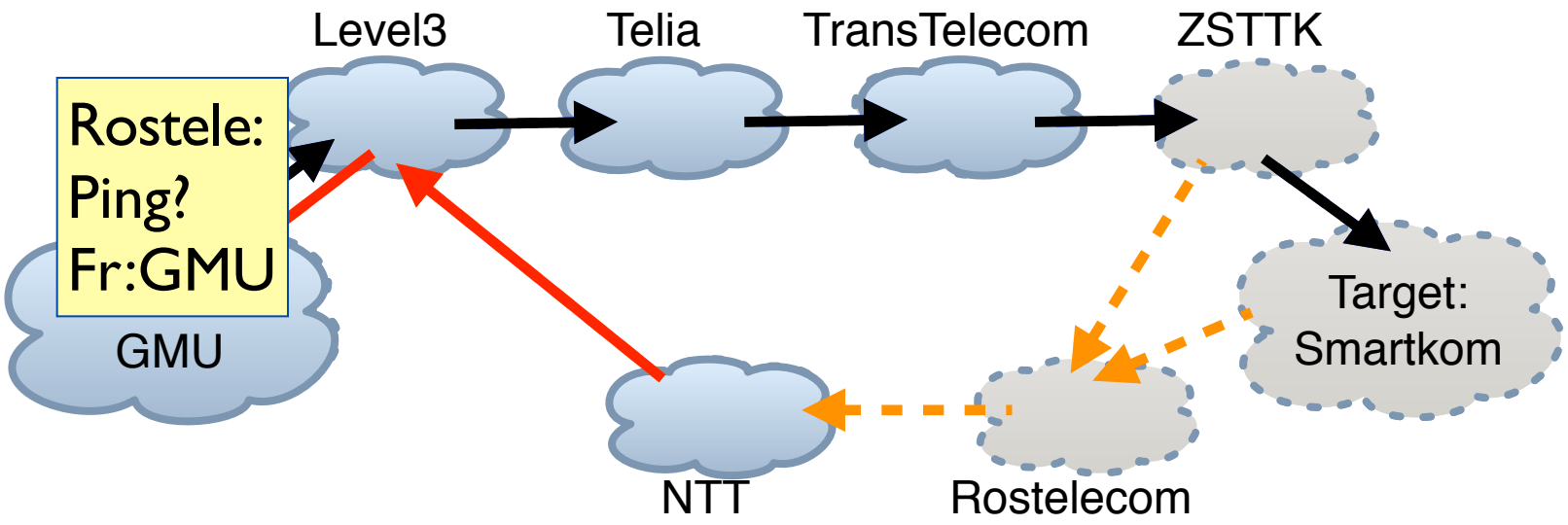
During outage:



- Forward path works

How does **LIFEGUARD** locate a failure?

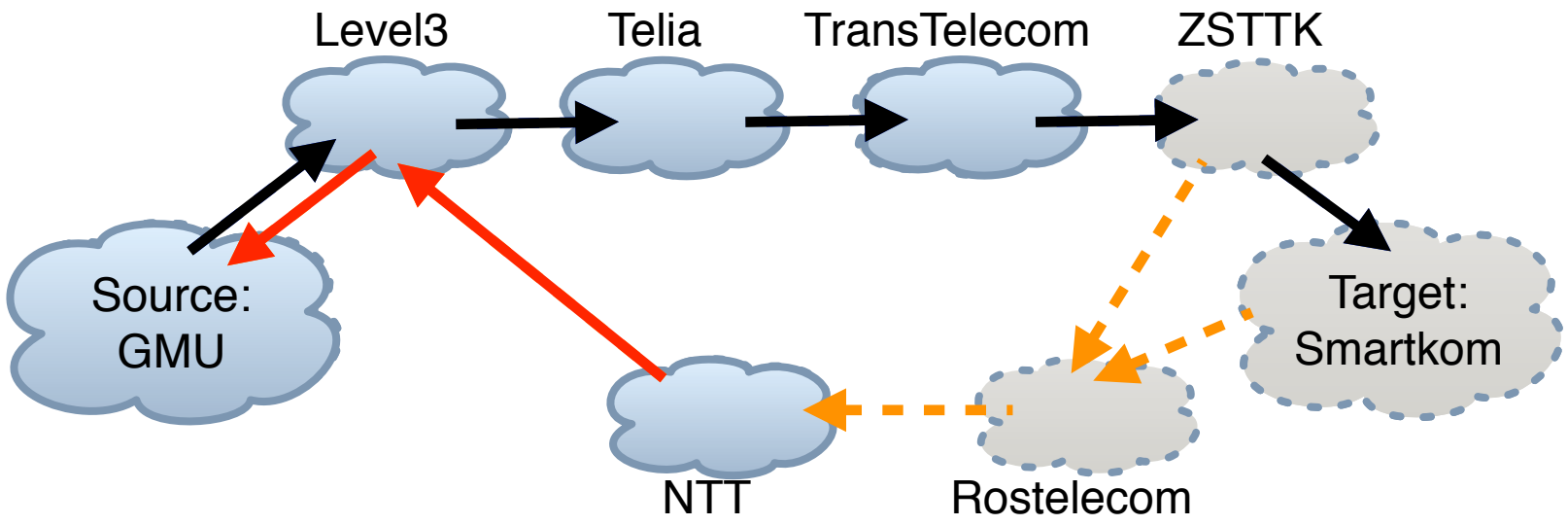
During outage:



- Forward path works
- Rostelcom is not forwarding traffic towards GMU

How does **LIFEGUARD** locate a failure?

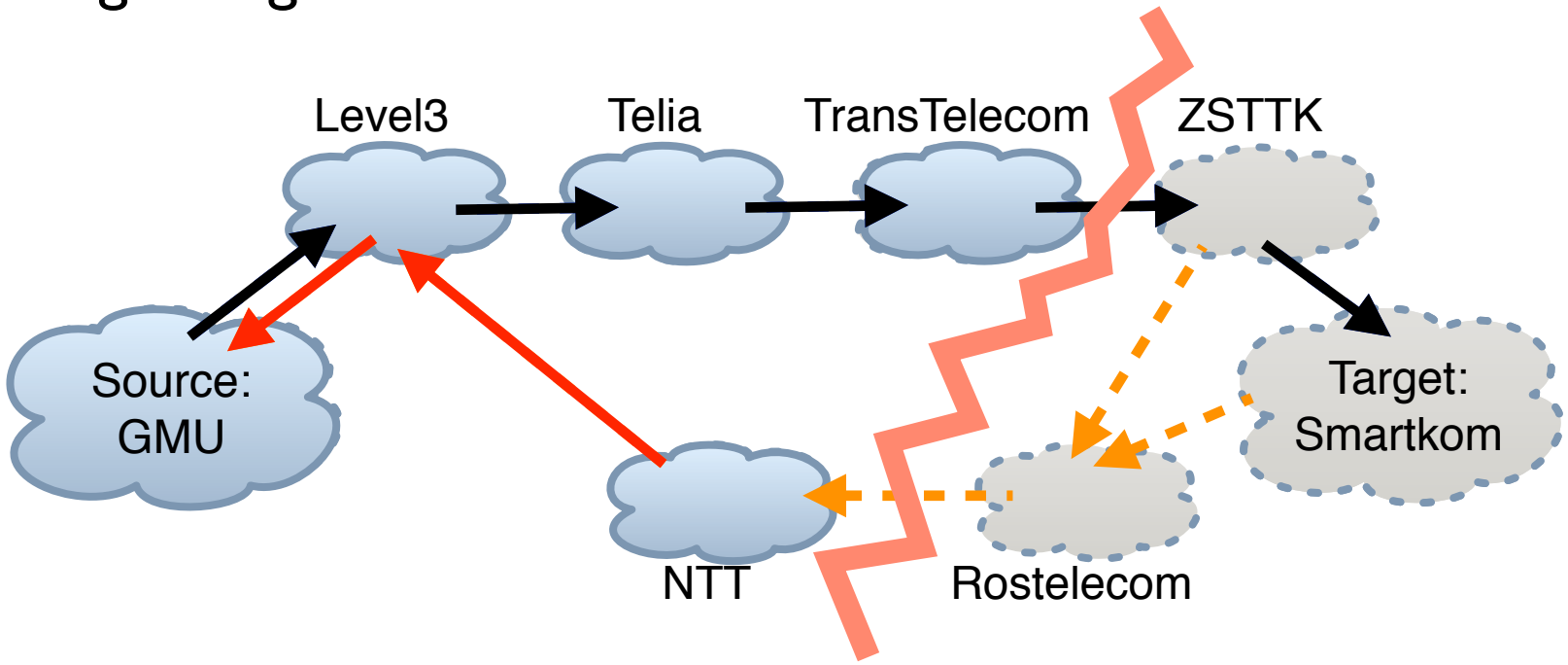
During outage:



- Forward path works
- Rostelcom is not forwarding traffic towards GMU

How does **LIFEGUARD** locate a failure?

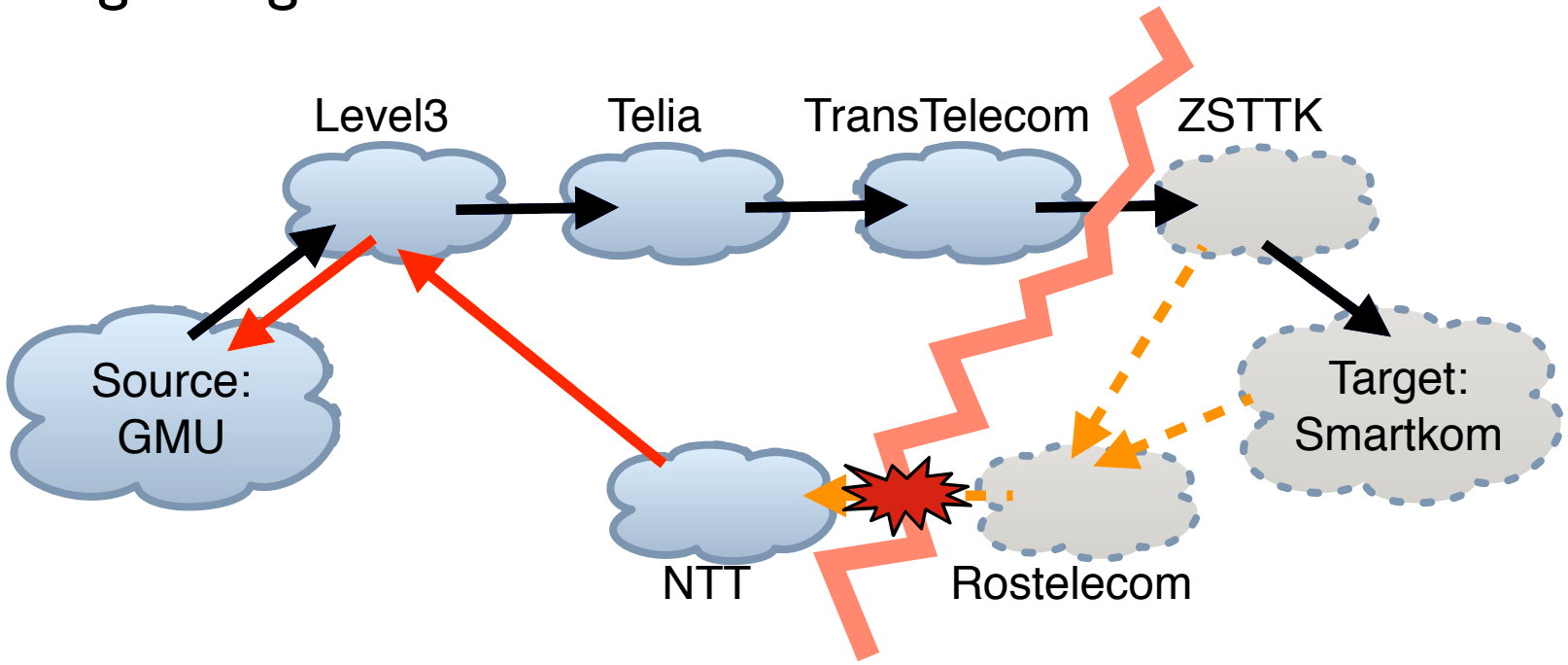
During outage:



- Forward path works
- Rostelcom is not forwarding traffic towards GMU

How does **LIFEGUARD** locate a failure?

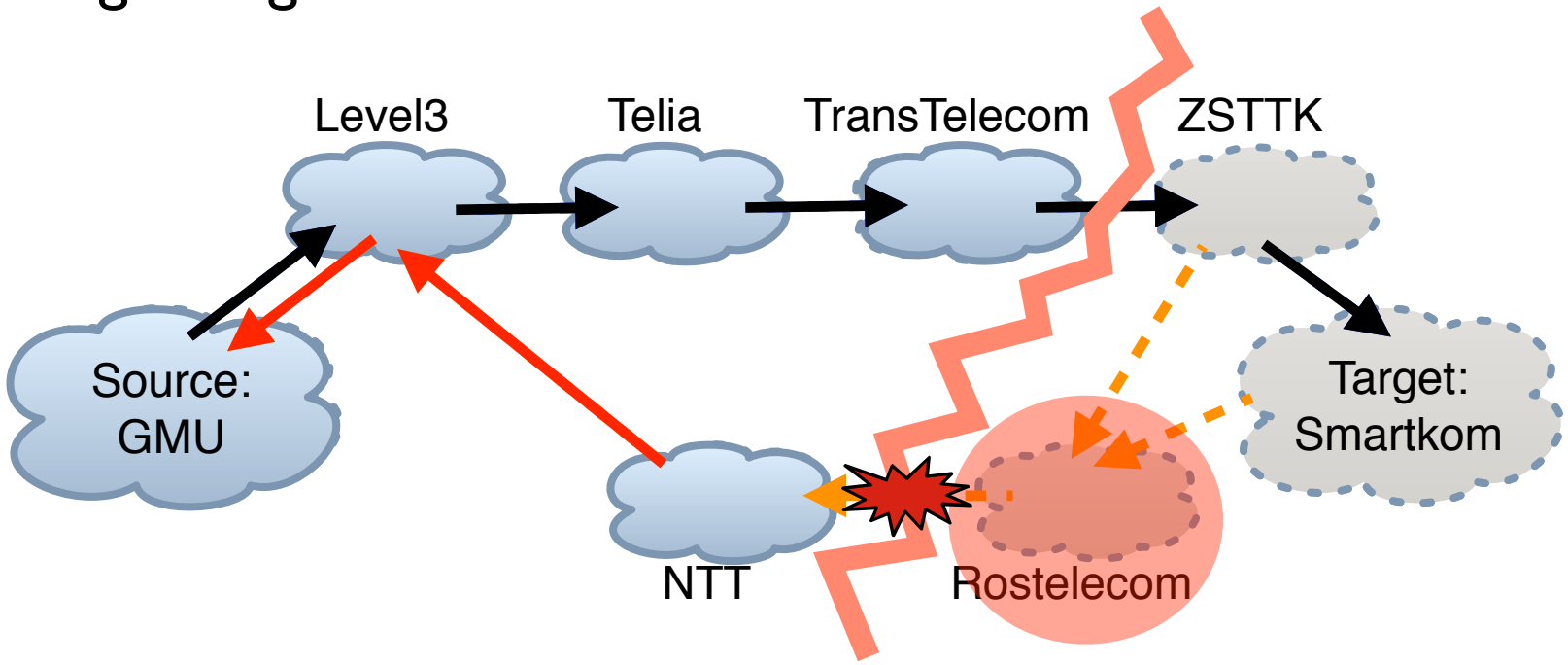
During outage:



- Forward path works
- Rostelcom is not forwarding traffic towards GMU

How does **LIFEGUARD** locate a failure?

During outage:



- Forward path works
- Rostelcom is not forwarding traffic towards GMU

How **LIFEGUARD** Locates Failures

LIFEGUARD:

1. Maintains background historical atlas
2. Isolates direction of failure, measures working direction
3. Tests historical paths in failing direction in order to prune candidate failure locations
4. Locates failure as being at the horizon of reachability

Our Approach and Outline

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

Our Approach and Outline

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

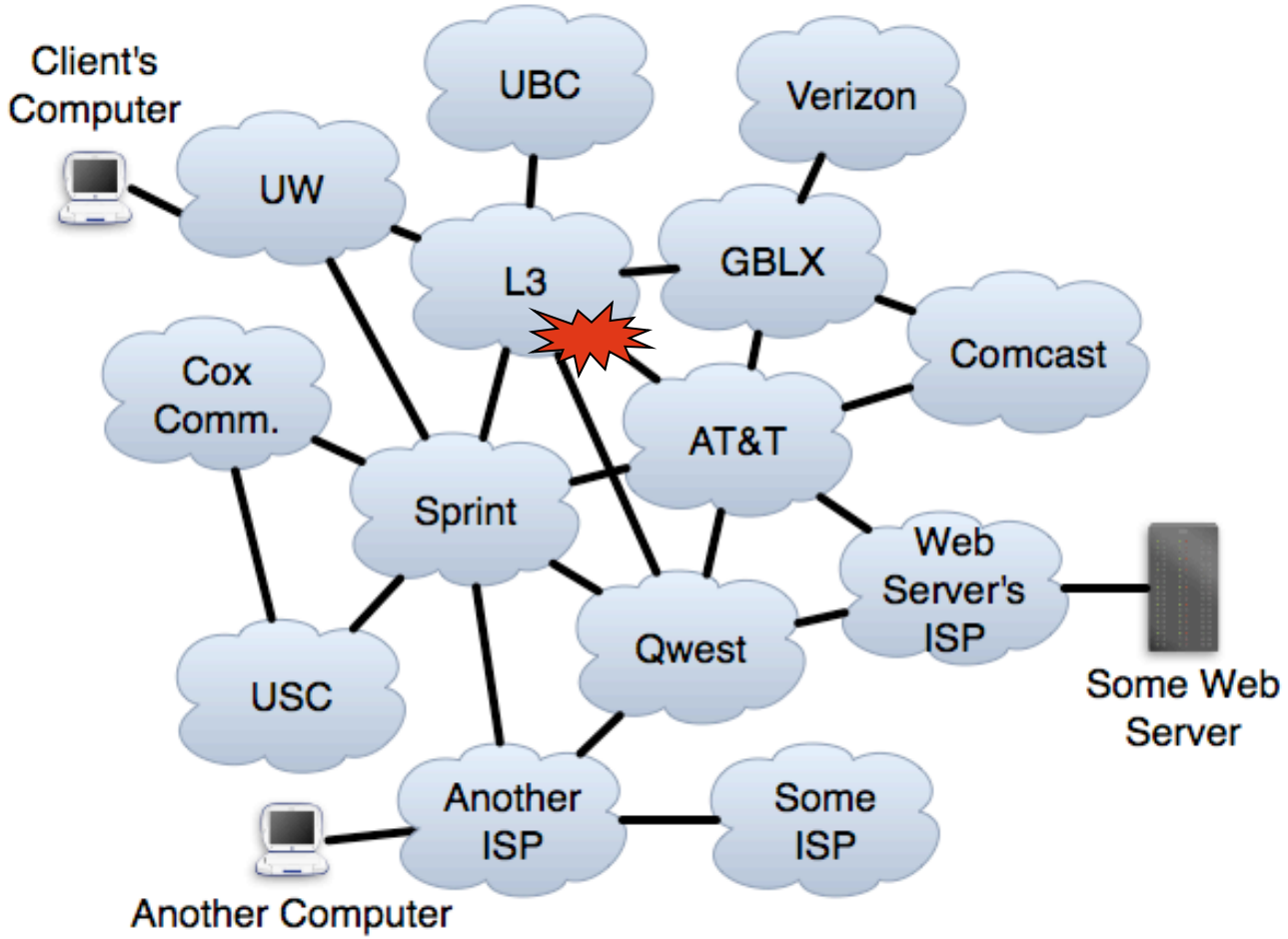
- Enable content / service providers to repair persistent routing problems affecting them, regardless of which ISP is causing them

Setting

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

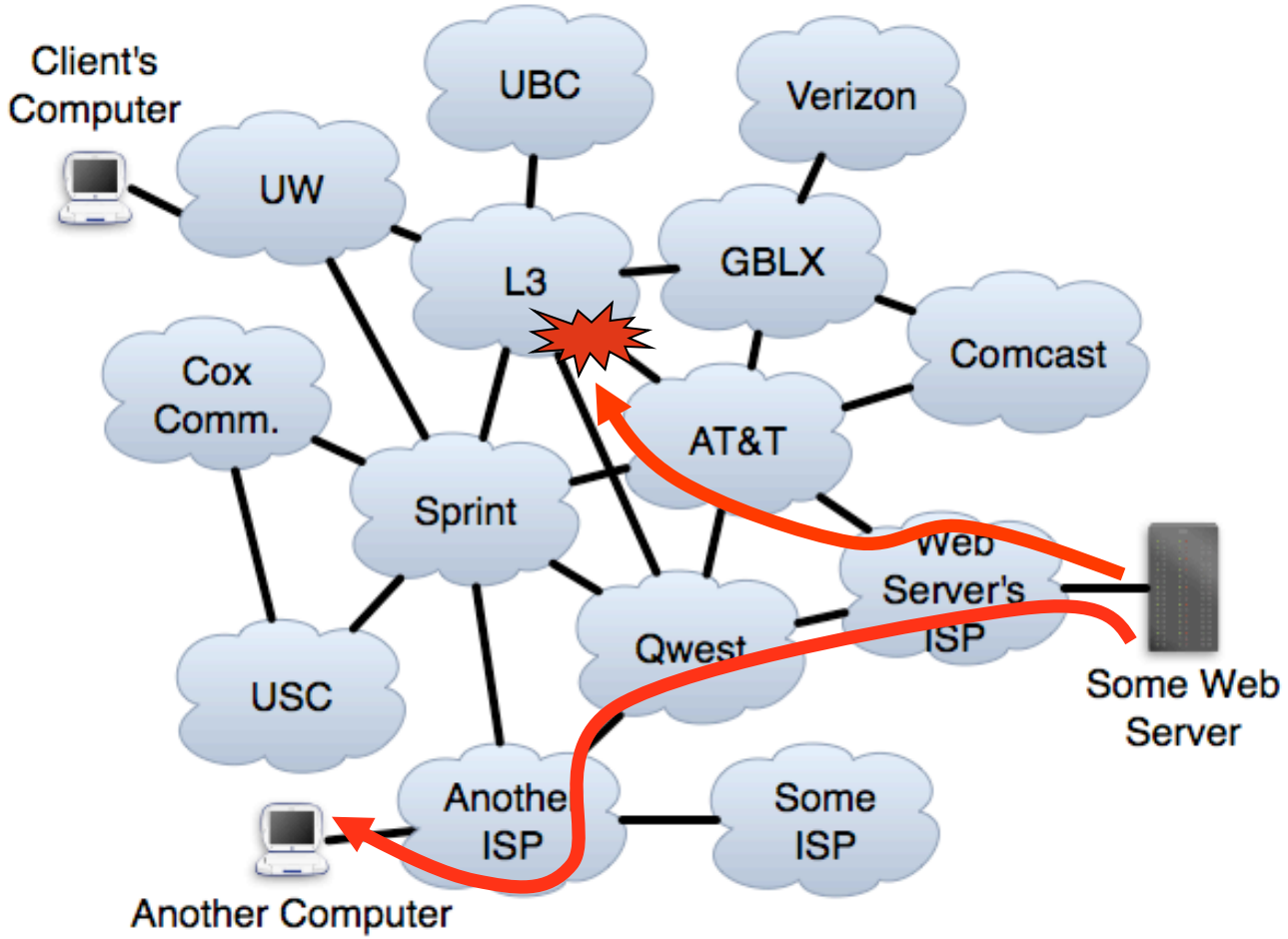
- We use BGP-Mux to speak BGP to the real Internet: 5 US universities as providers

Self-Repair of Forward Paths



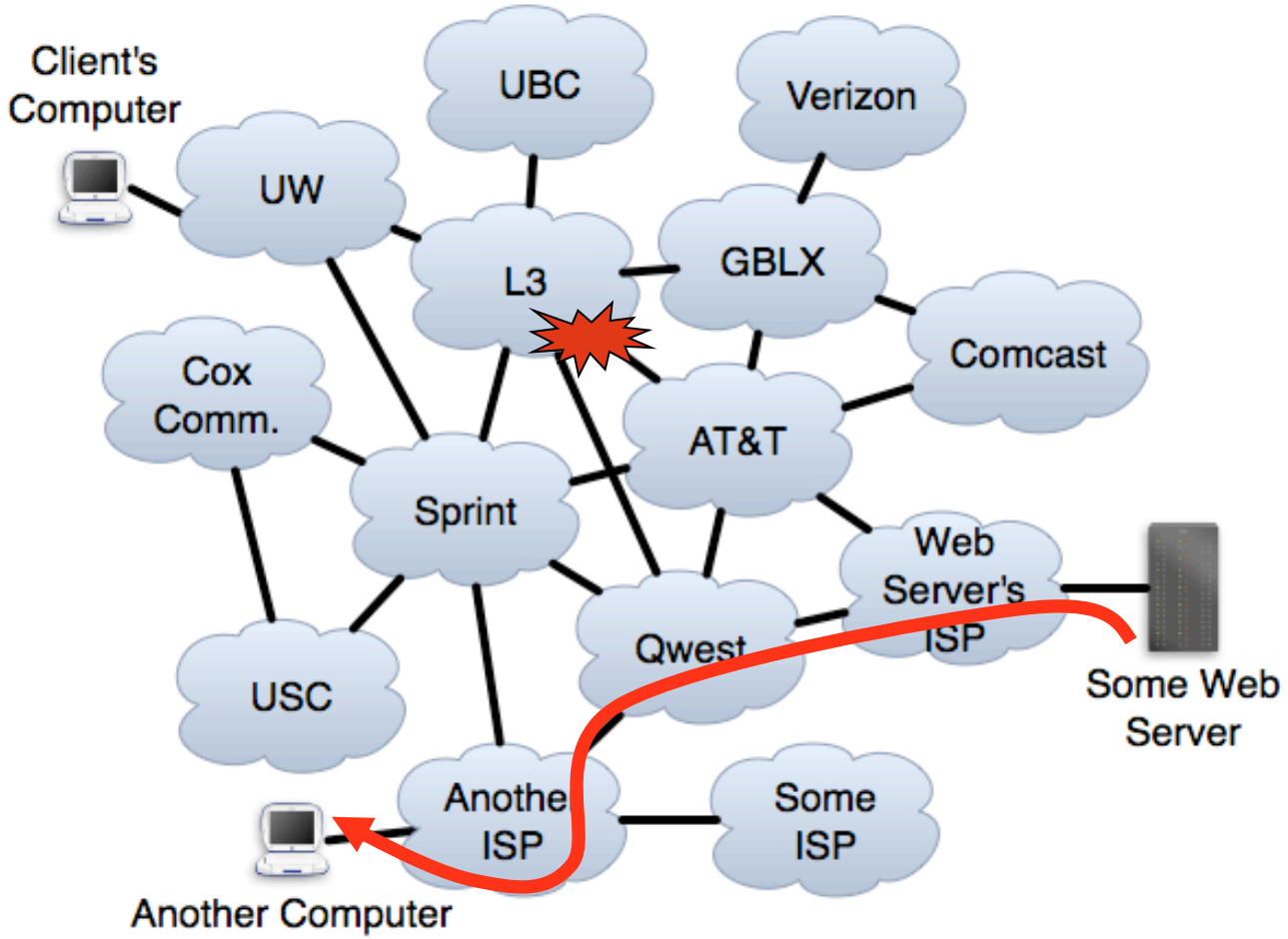
Straightforward: Choose a path that avoids the problem.

Self-Repair of Forward Paths



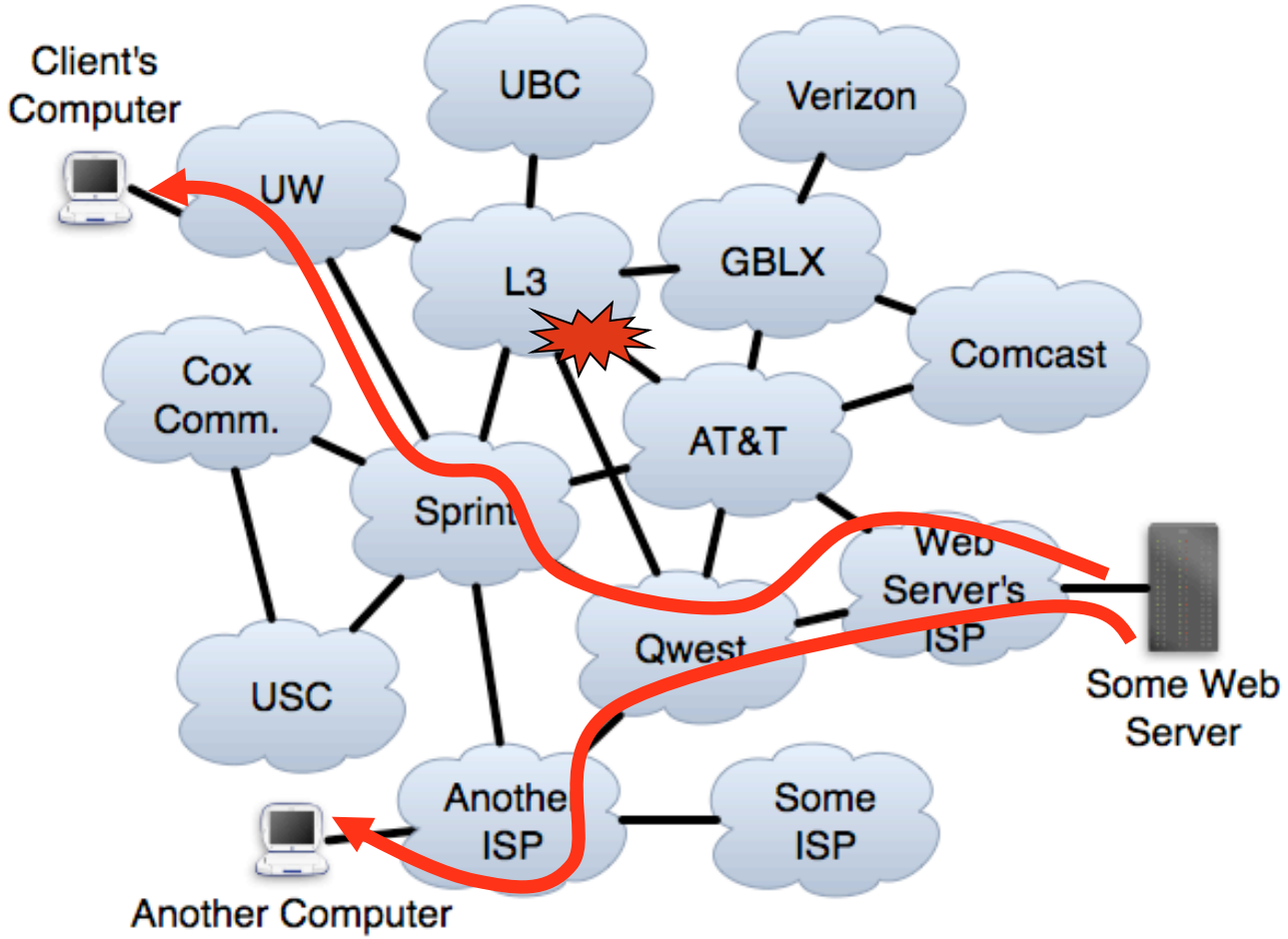
Straightforward: Choose a path that avoids the problem.

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.

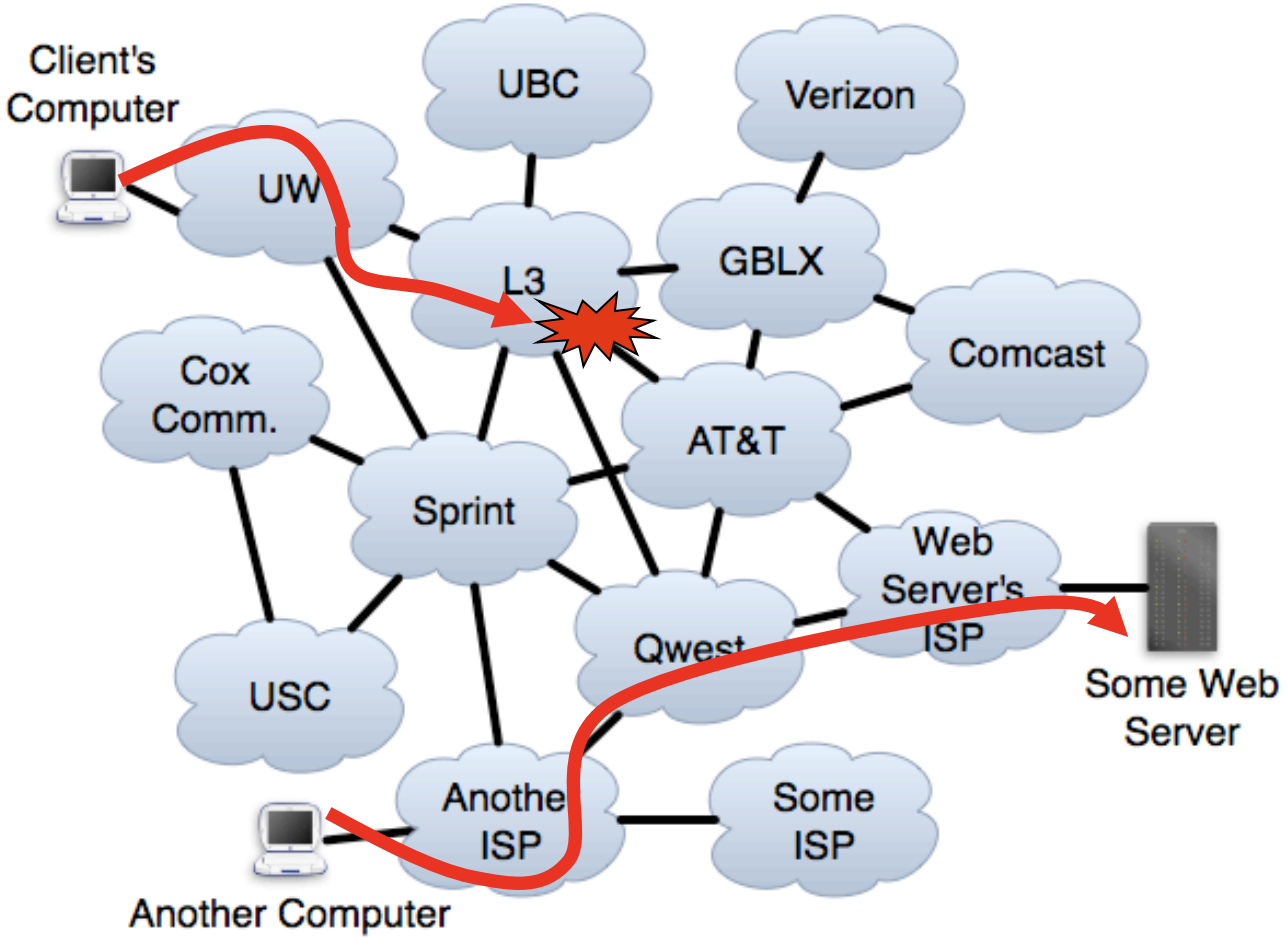
A Mechanism for Failure Avoidance

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

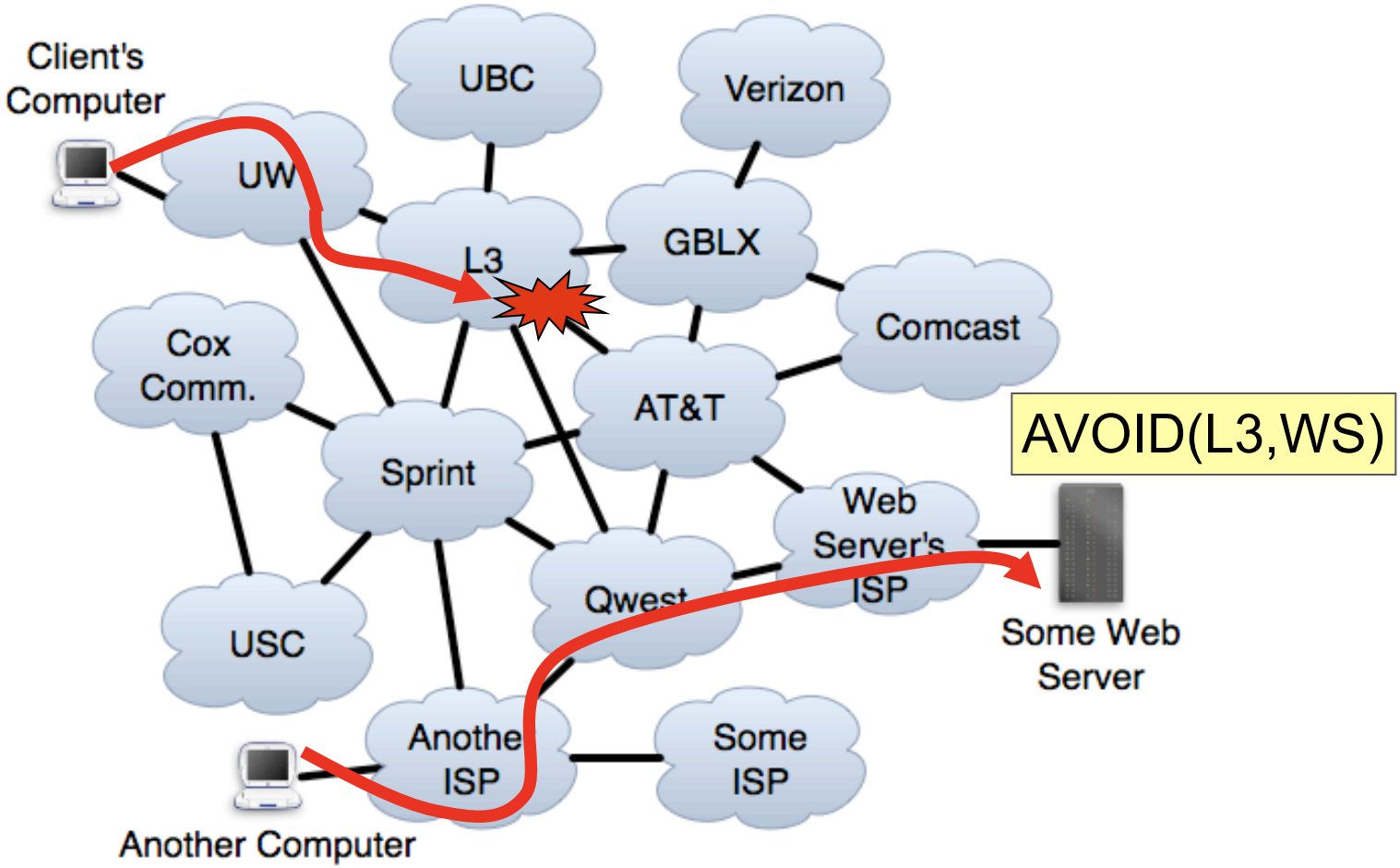
Reverse path: Want others to choose paths to my prefix P that avoid ISP or ISP-ISP link X

- 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

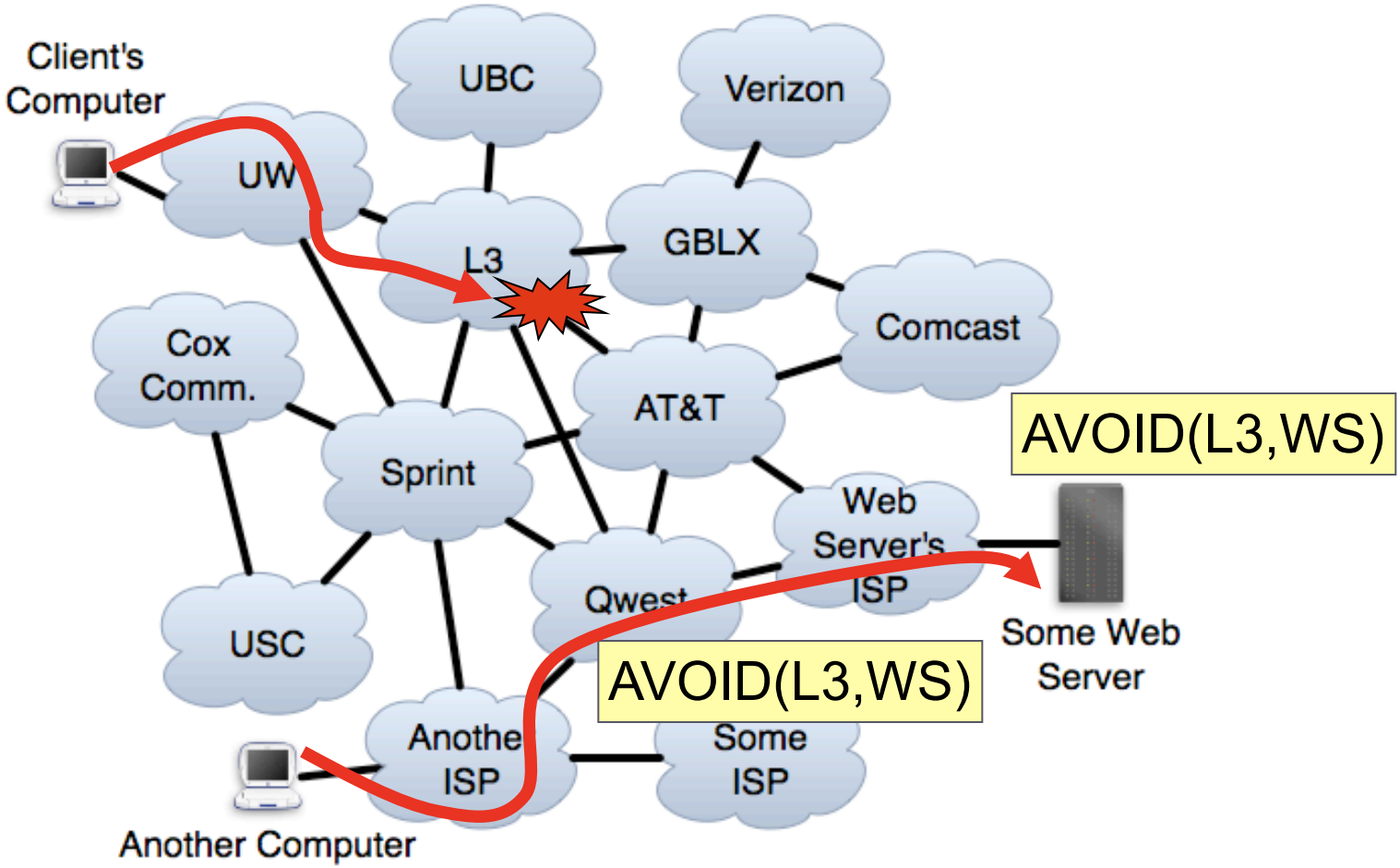
Ideal Self-Repair of Reverse Paths



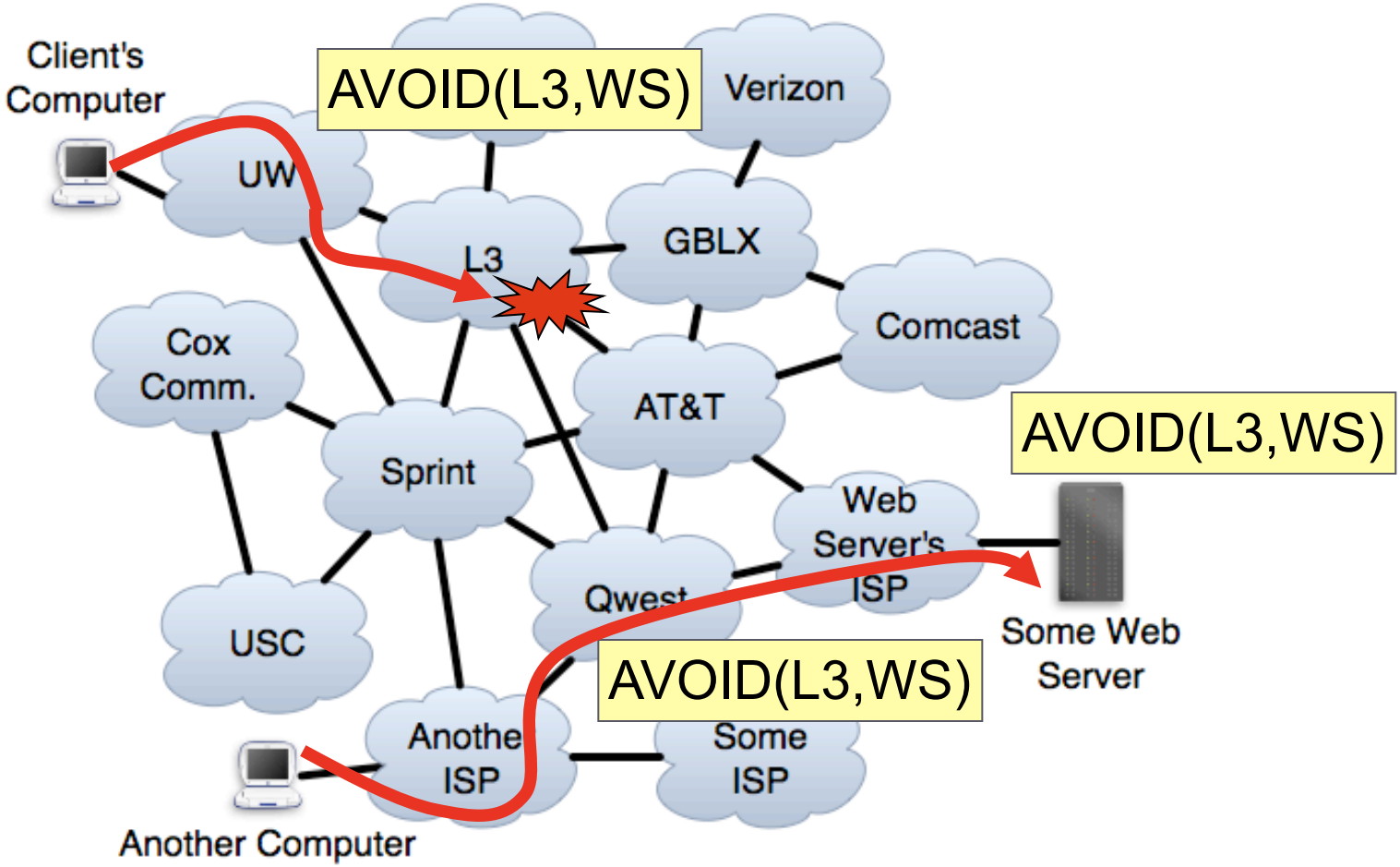
Ideal Self-Repair of Reverse Paths



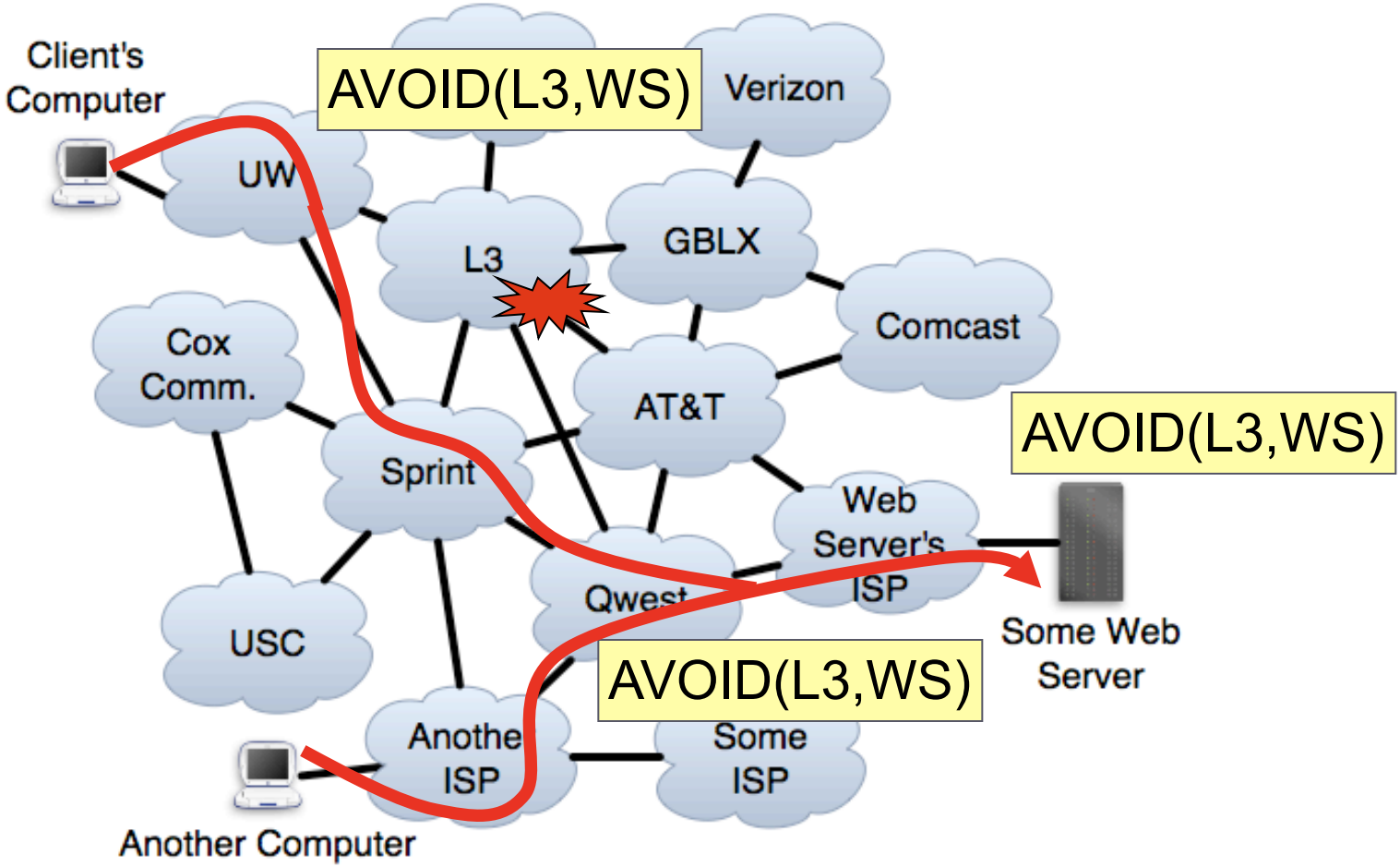
Ideal Self-Repair of Reverse Paths



Ideal Self-Repair of Reverse Paths



Ideal Self-Repair of Reverse Paths



Do paths exist that AVOID problem?

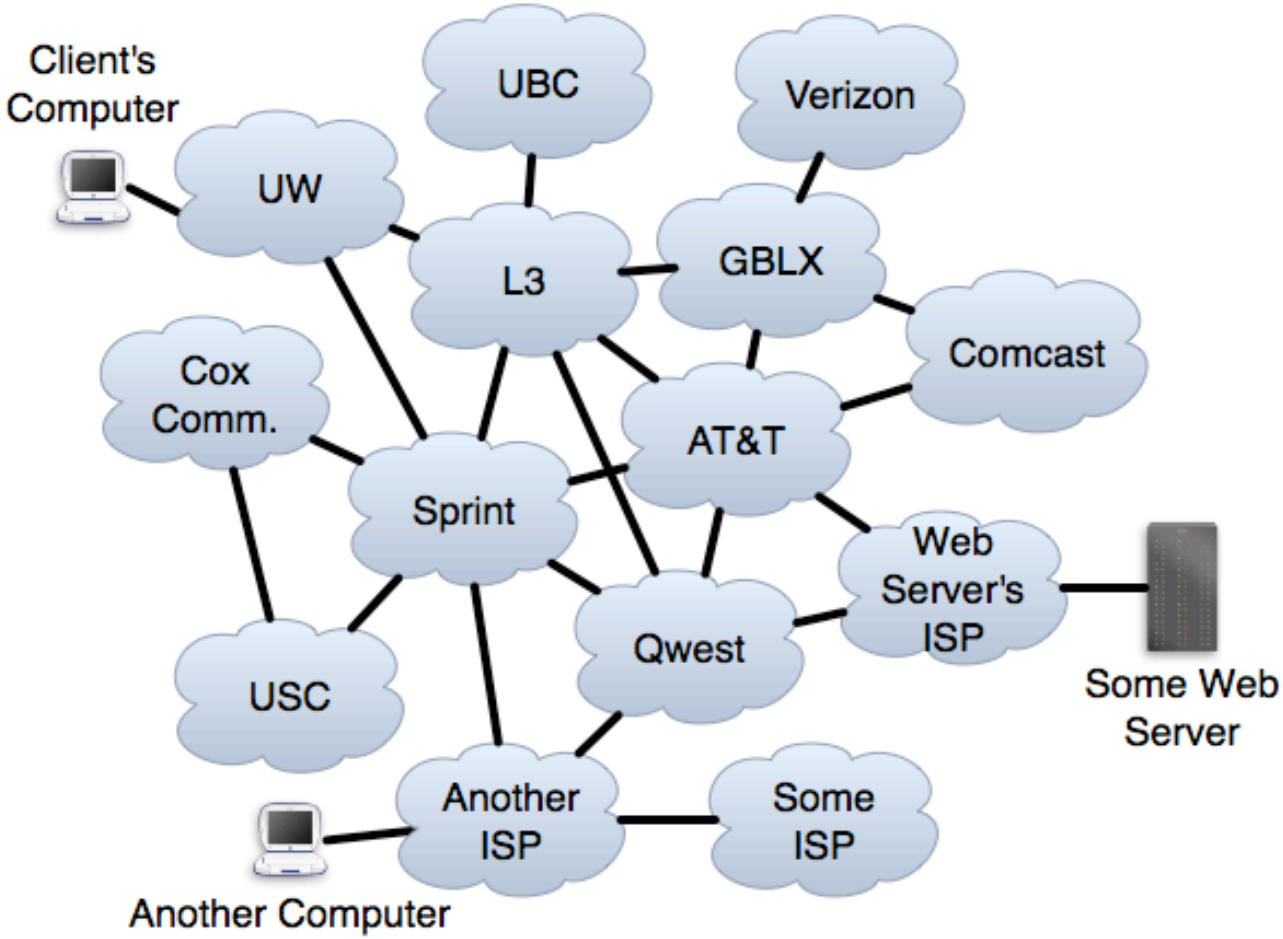
LIFEGUARD repairs outages by instructing others to avoid particular routes.

Q: Do alternative routes exist?

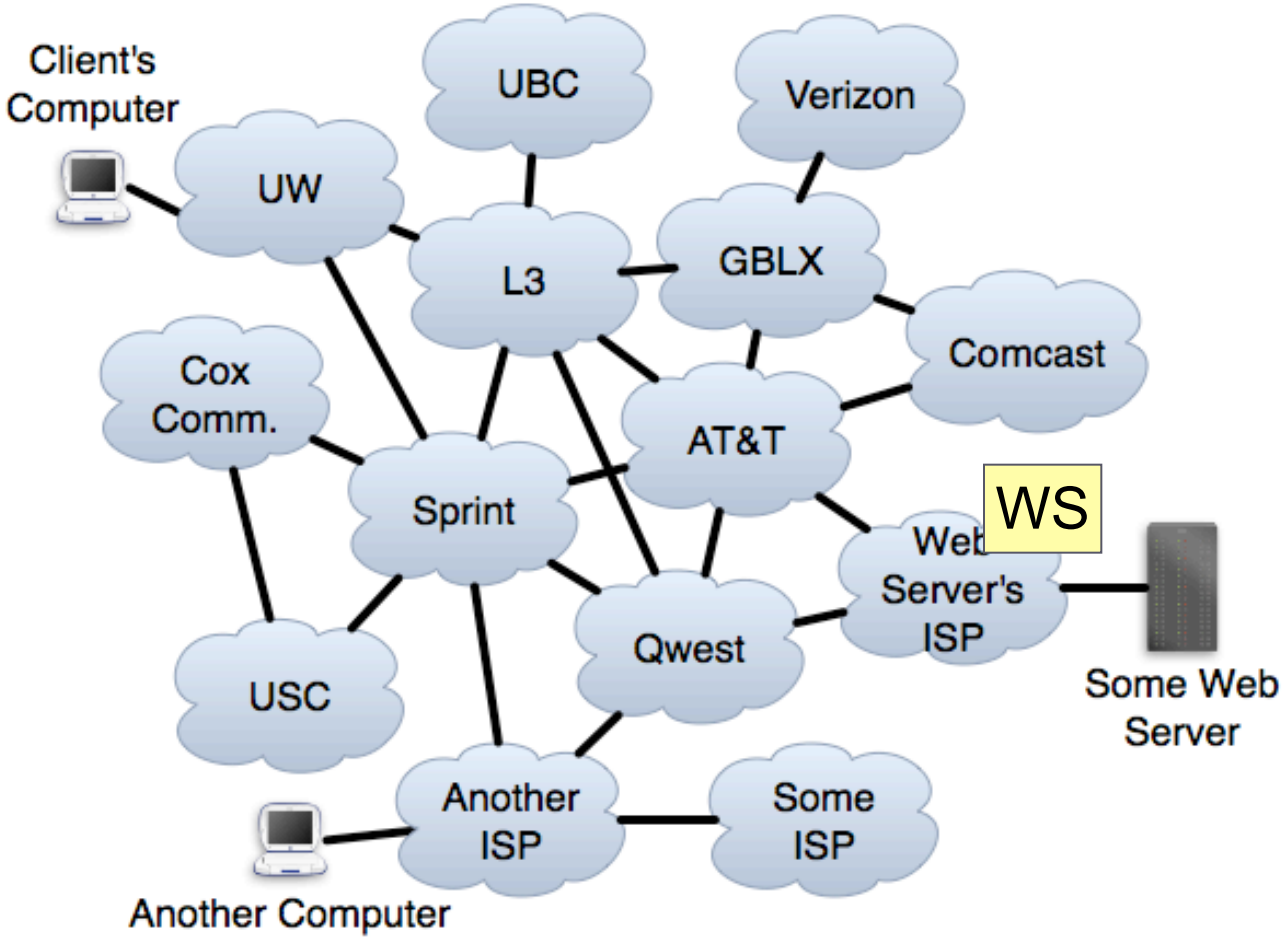
A: Alternate policy-compliant paths exist in 90% of simulated AVOID(X,P) *announcements.*

- Simulated 10 million AVOIDs on actual measured routes.

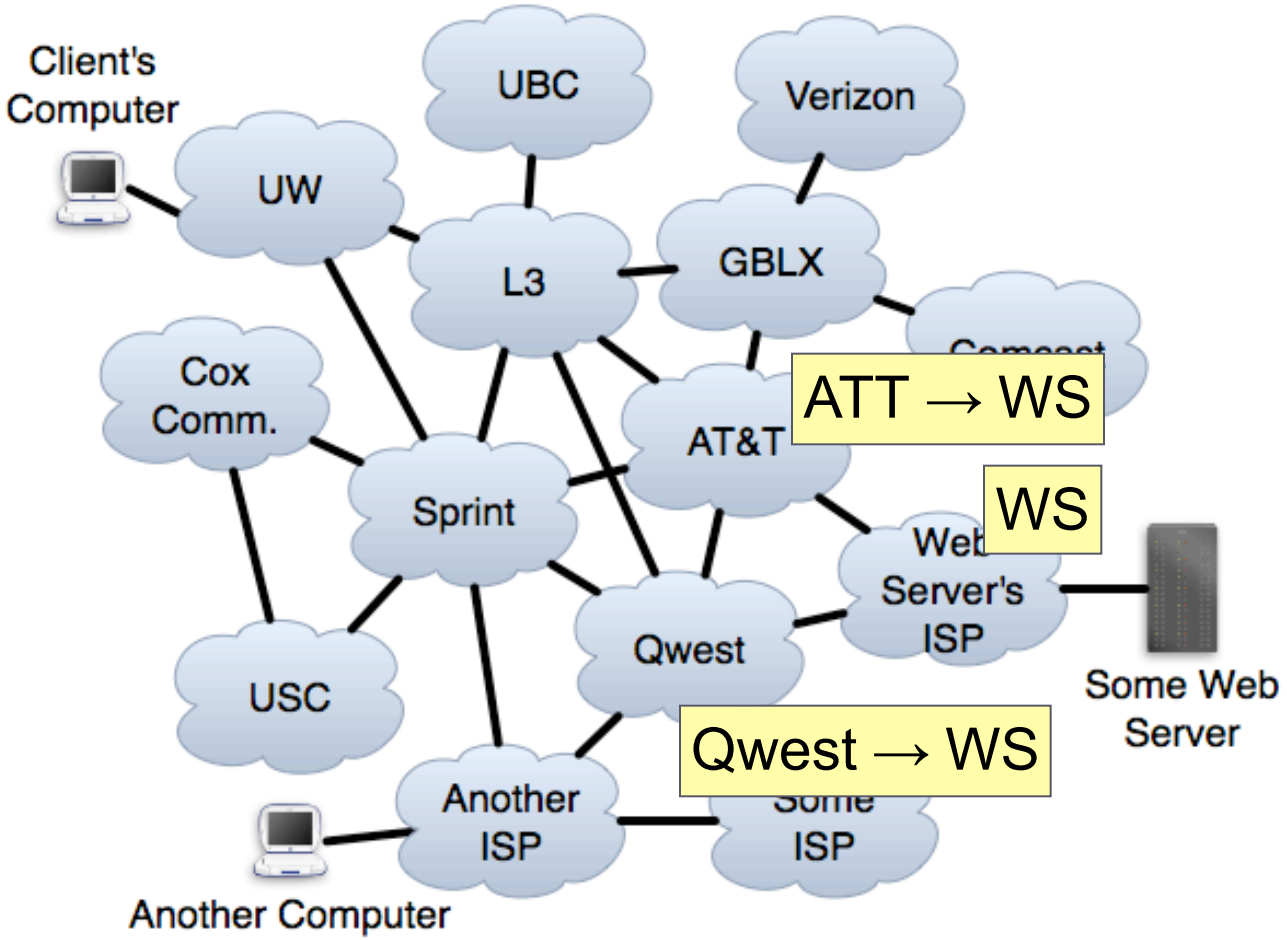
Practical Self-Repair of Reverse Paths



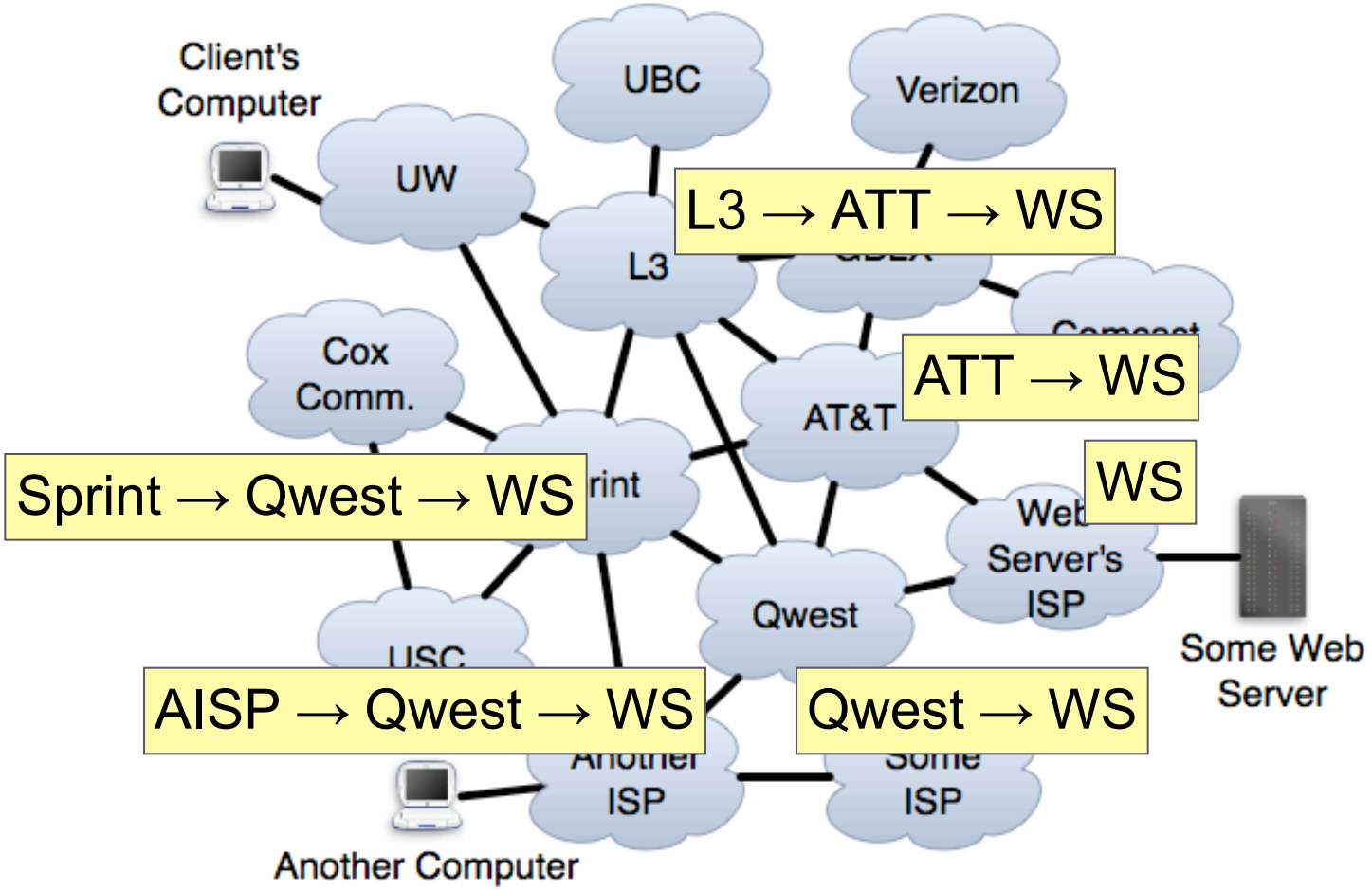
Practical Self-Repair of Reverse Paths



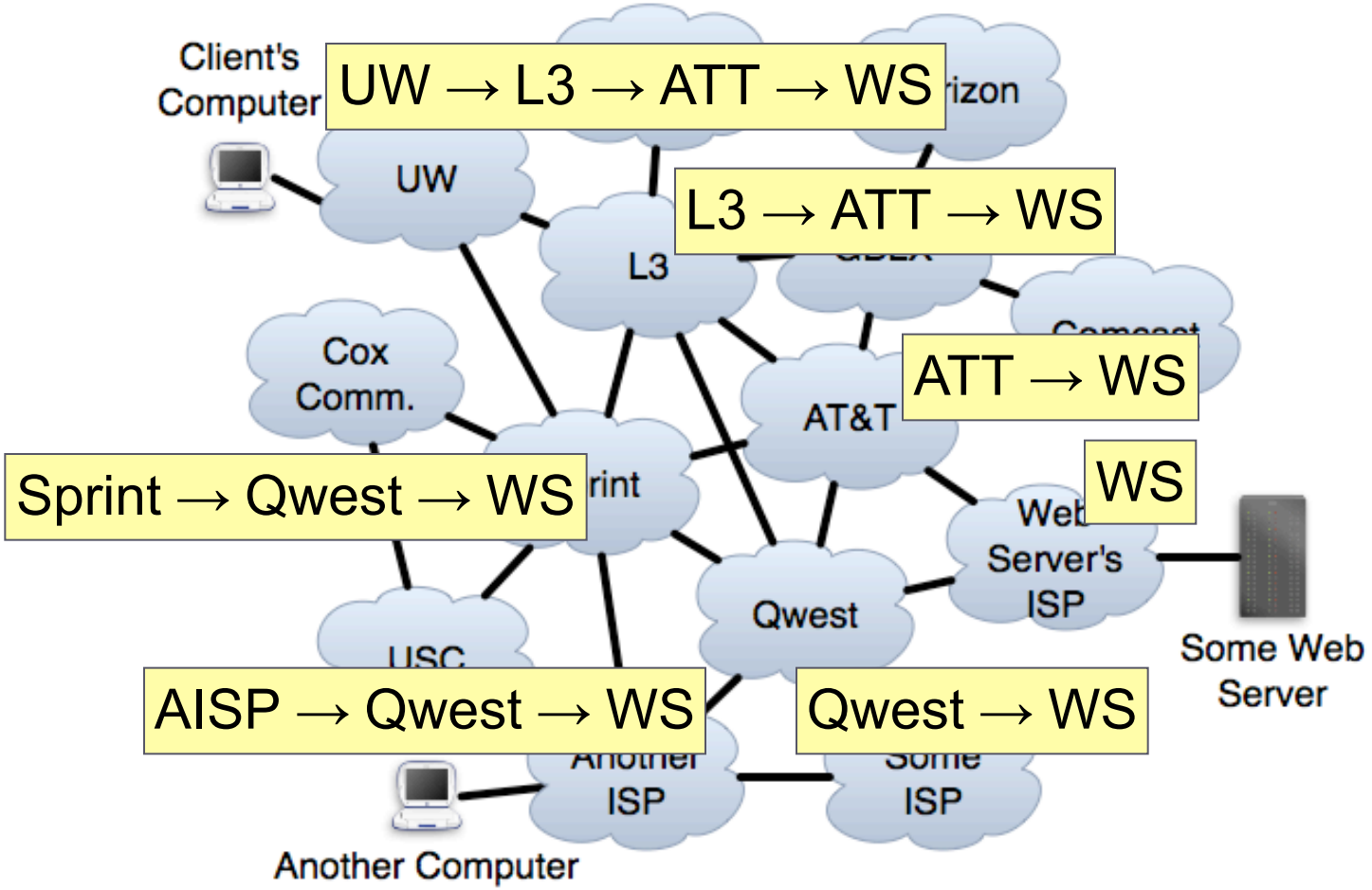
Practical Self-Repair of Reverse Paths



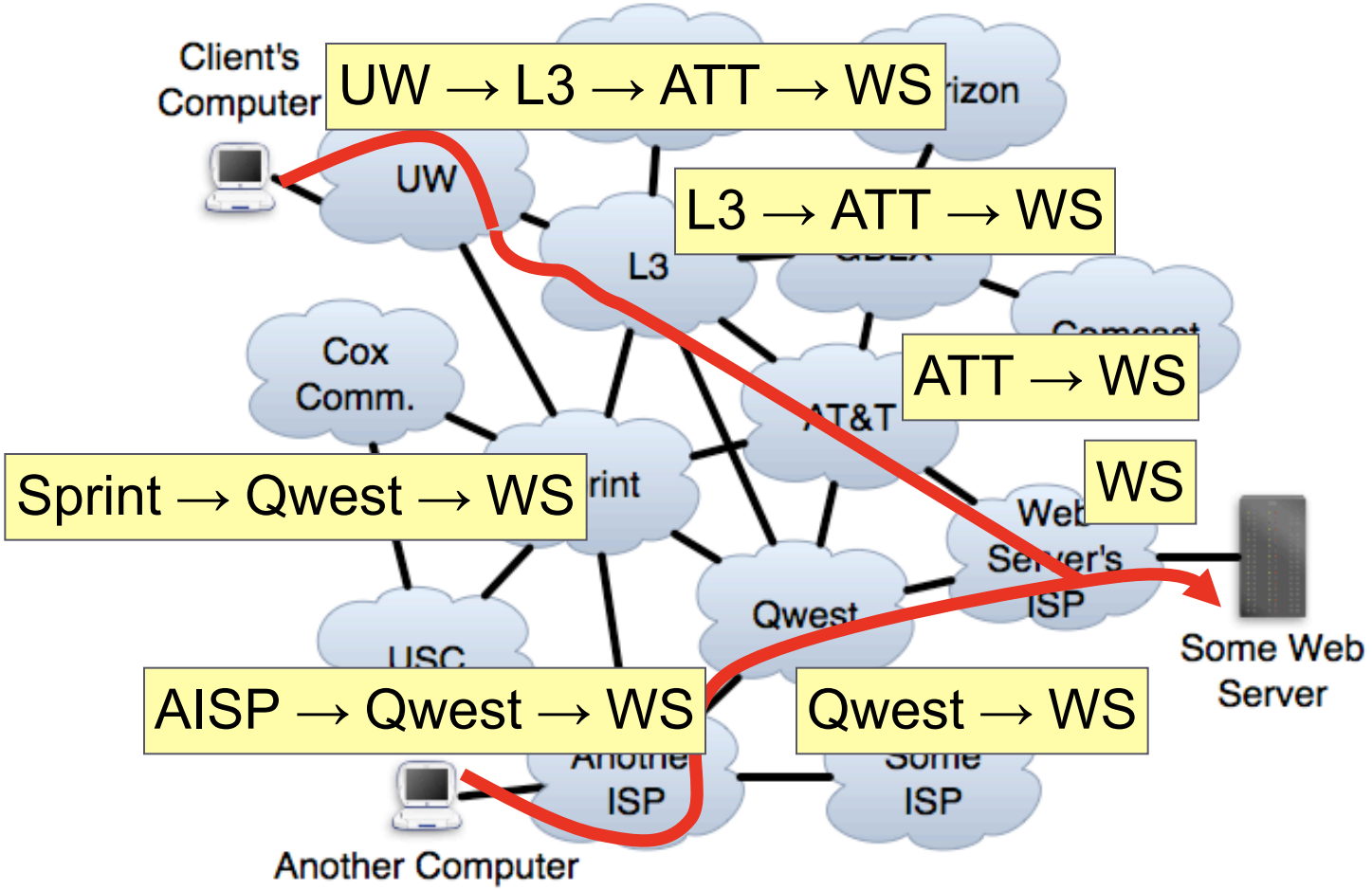
Practical Self-Repair of Reverse Paths



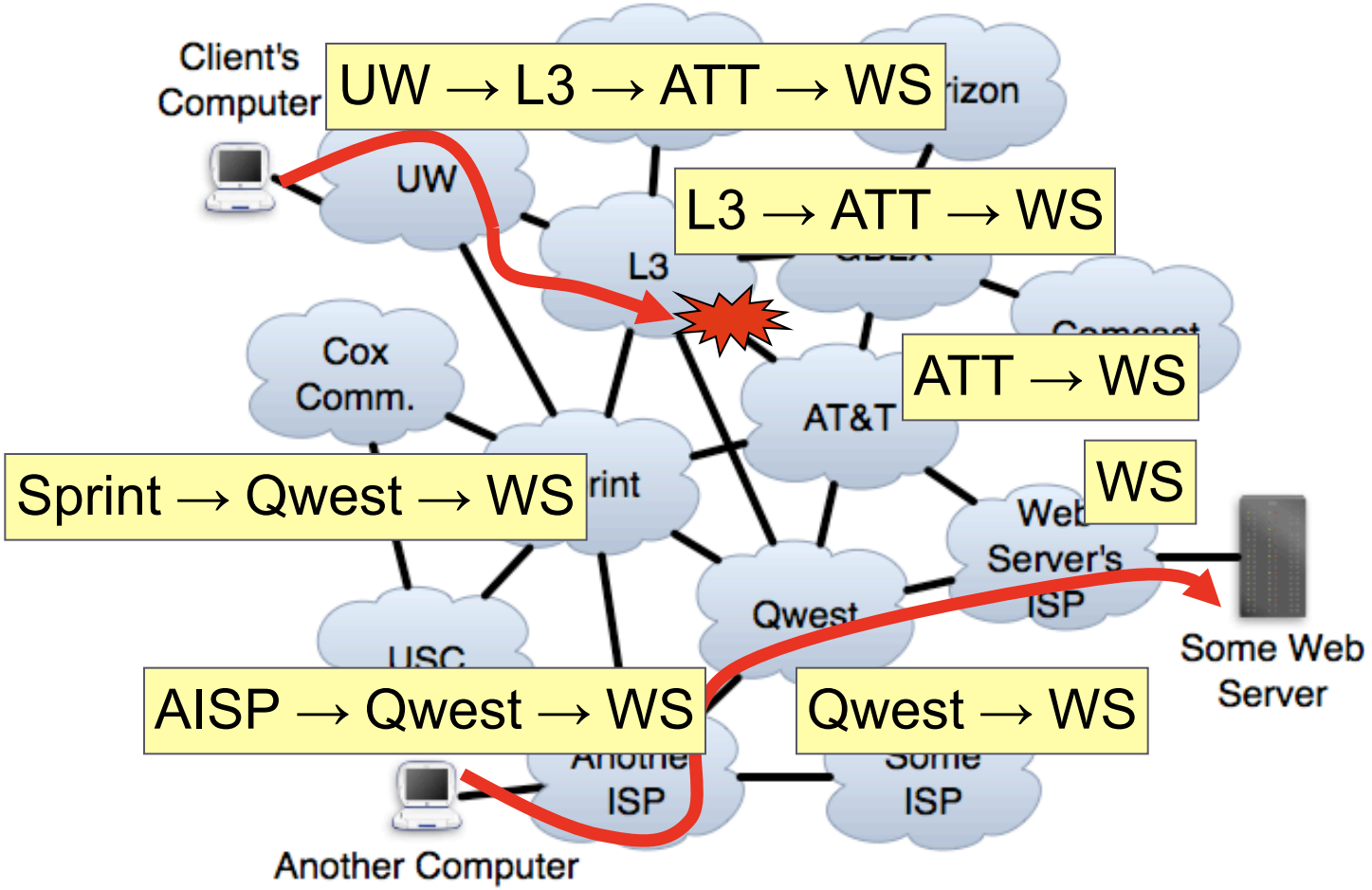
Practical Self-Repair of Reverse Paths



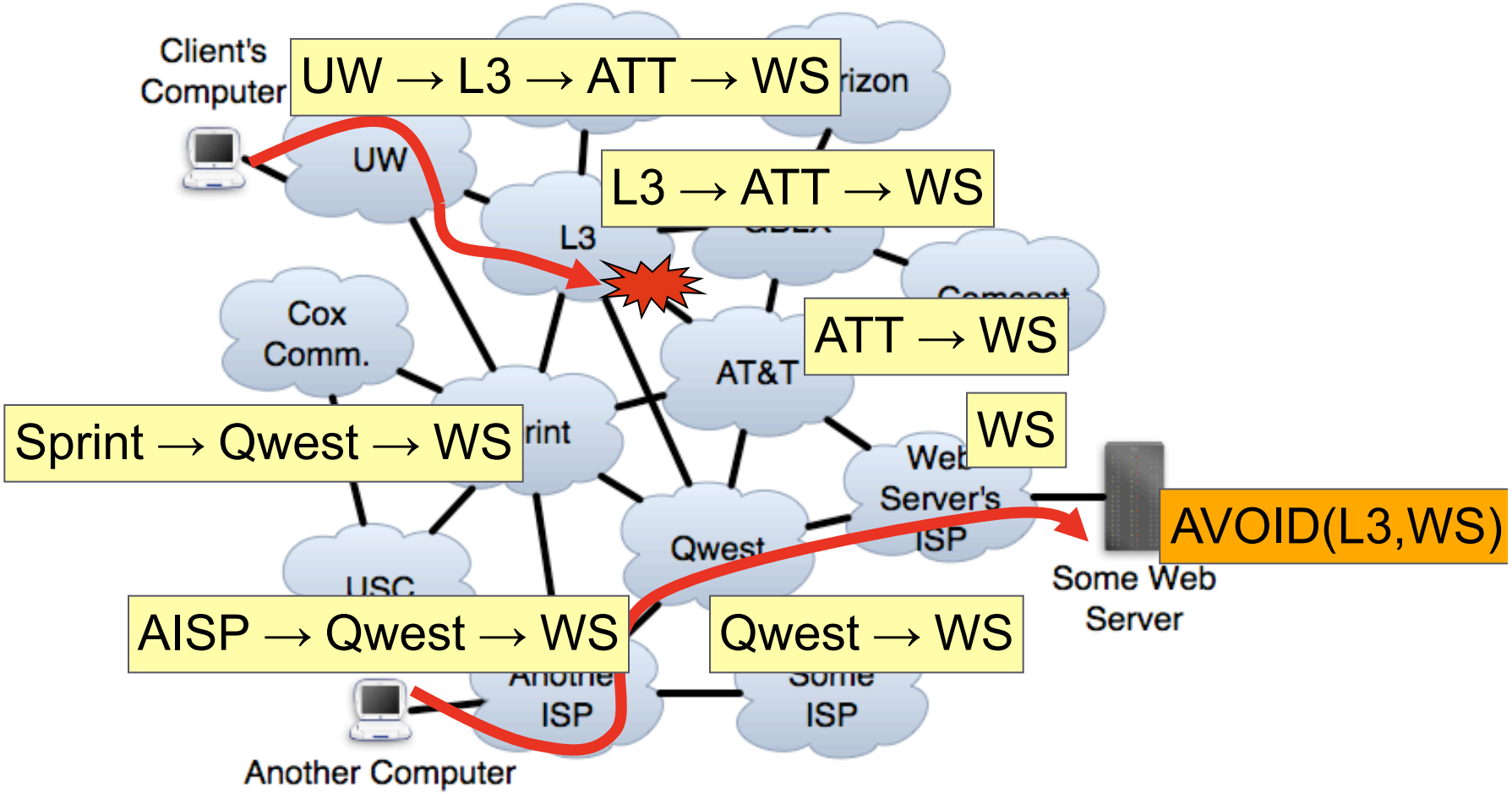
Practical Self-Repair of Reverse Paths



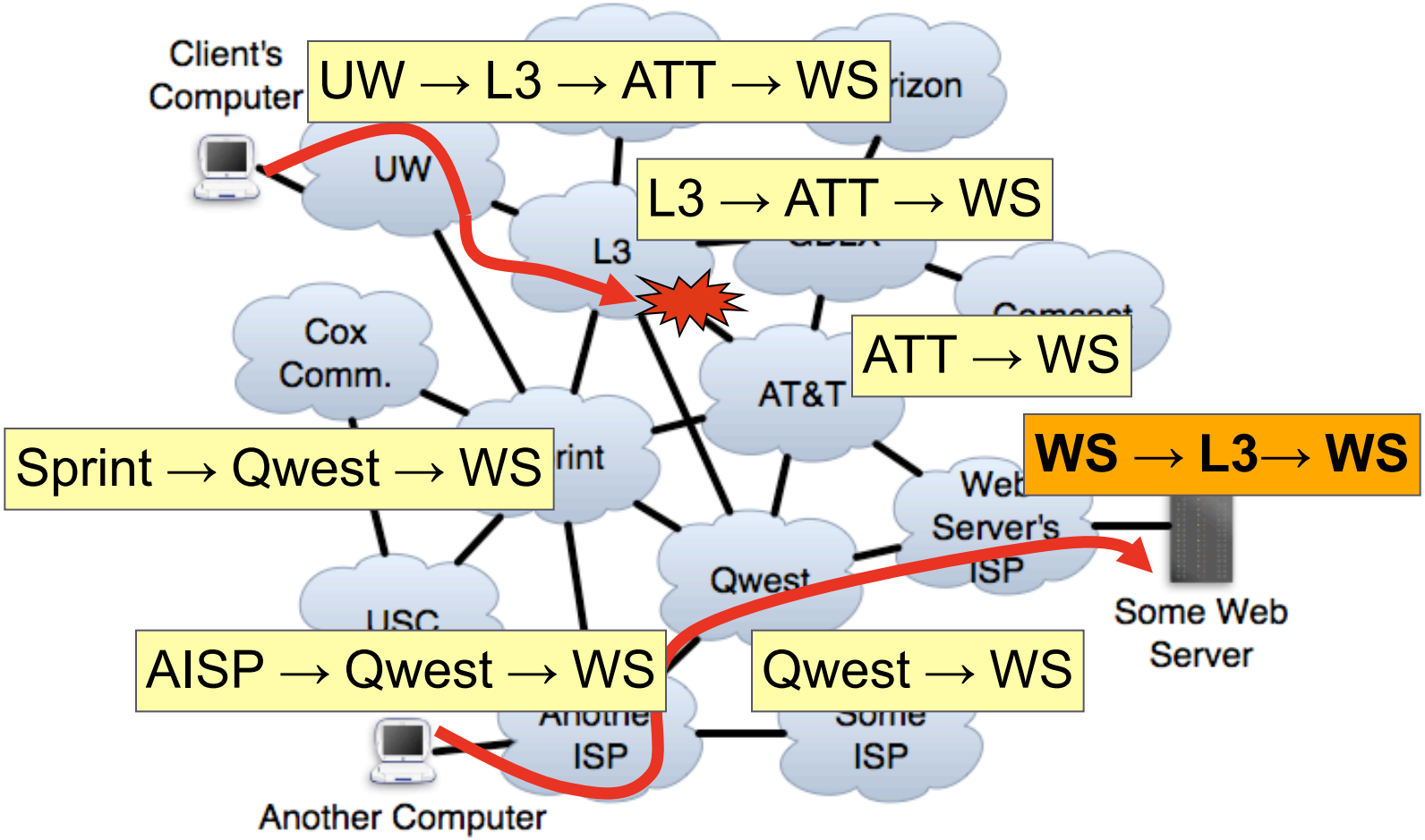
Practical Self-Repair of Reverse Paths



Practical Self-Repair of Reverse Paths

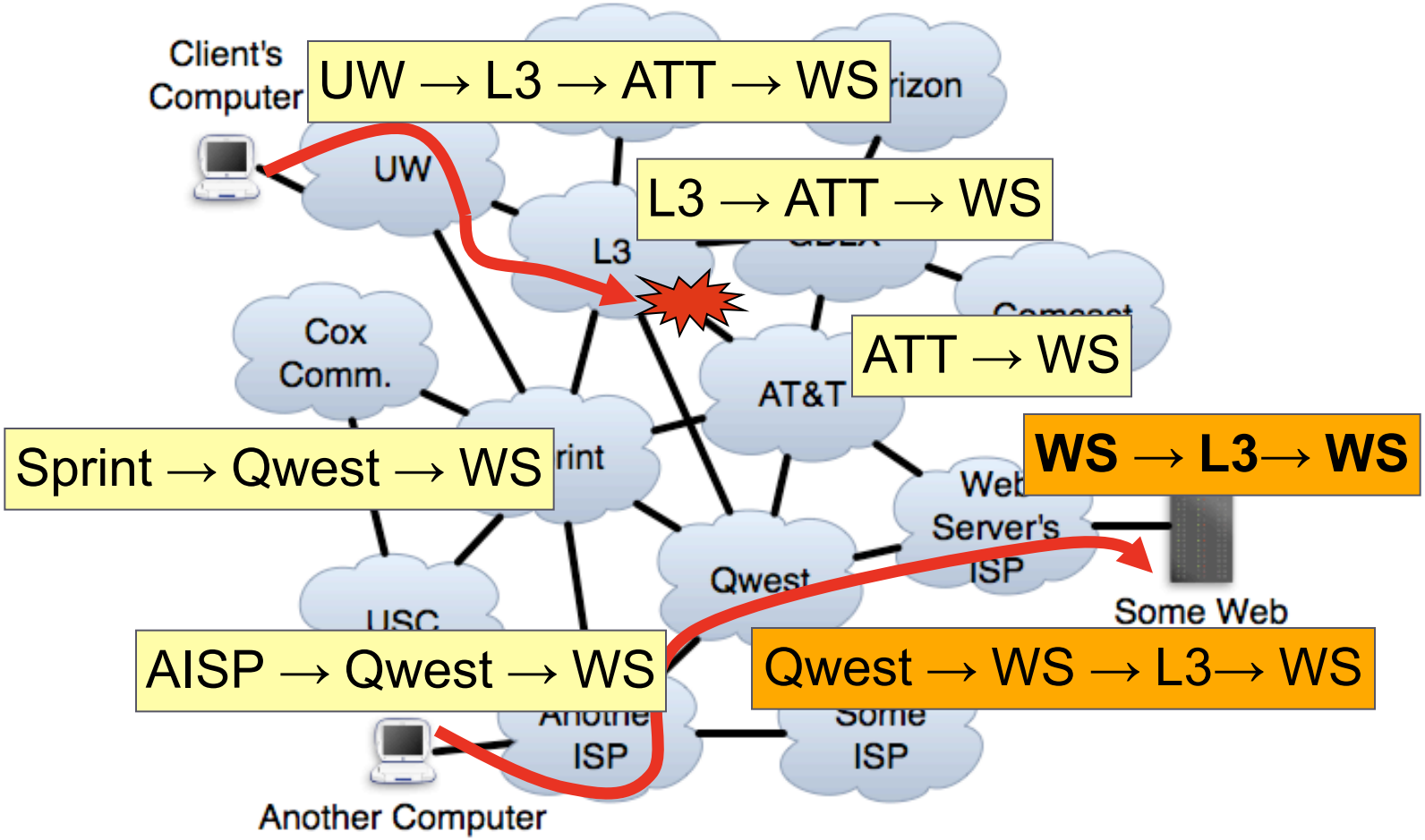


Practical Self-Repair of Reverse Paths



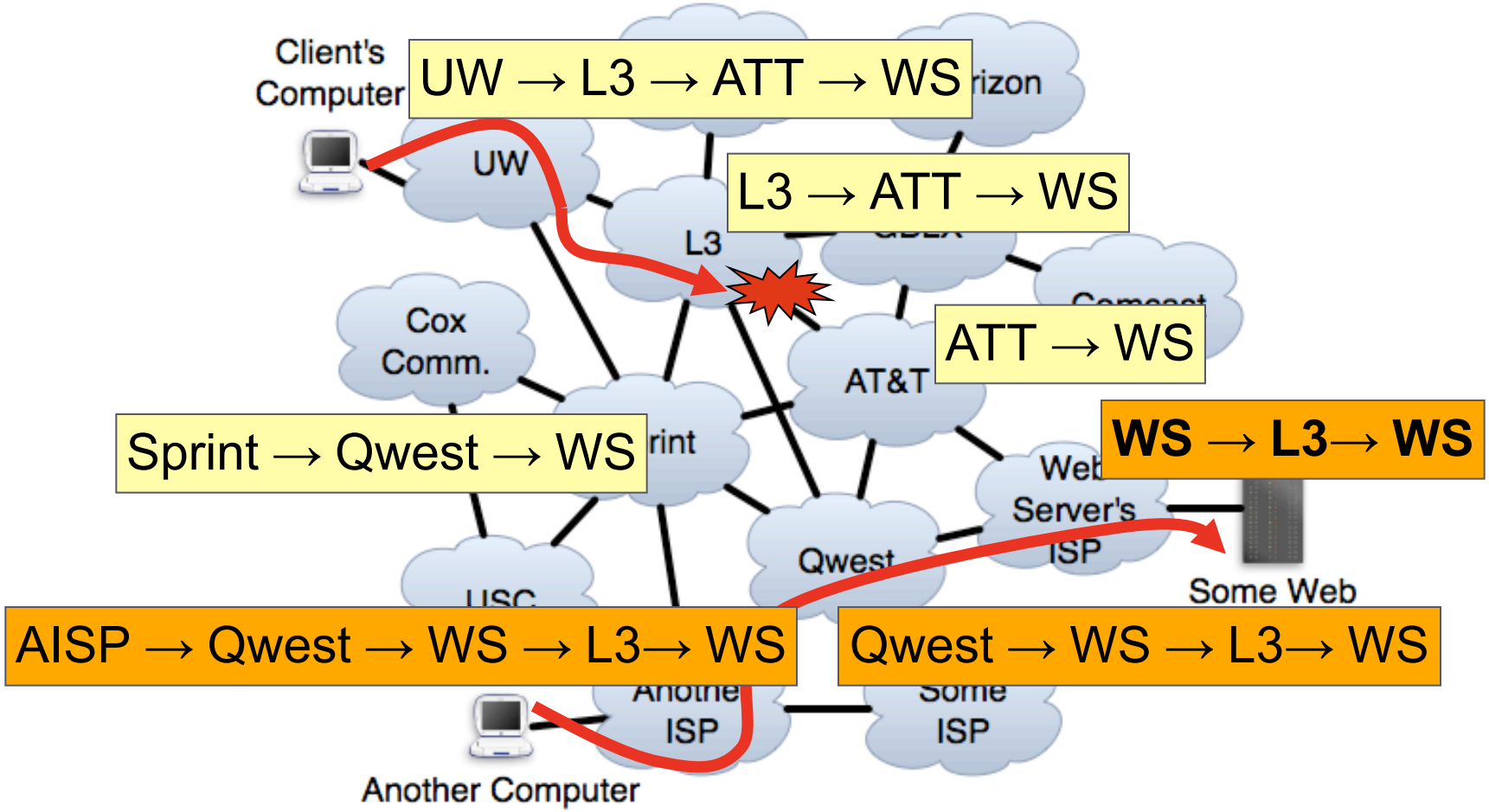
BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



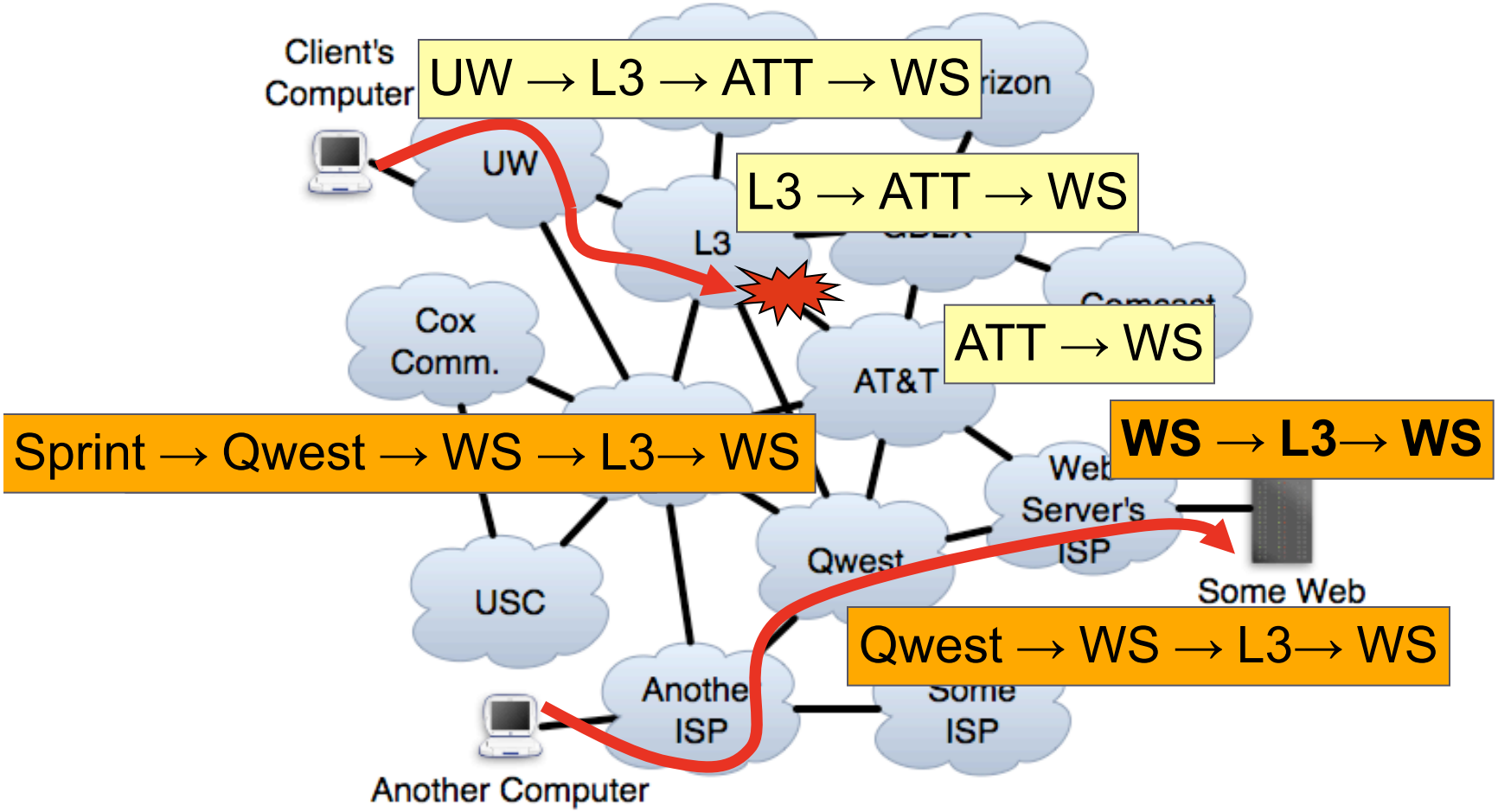
BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



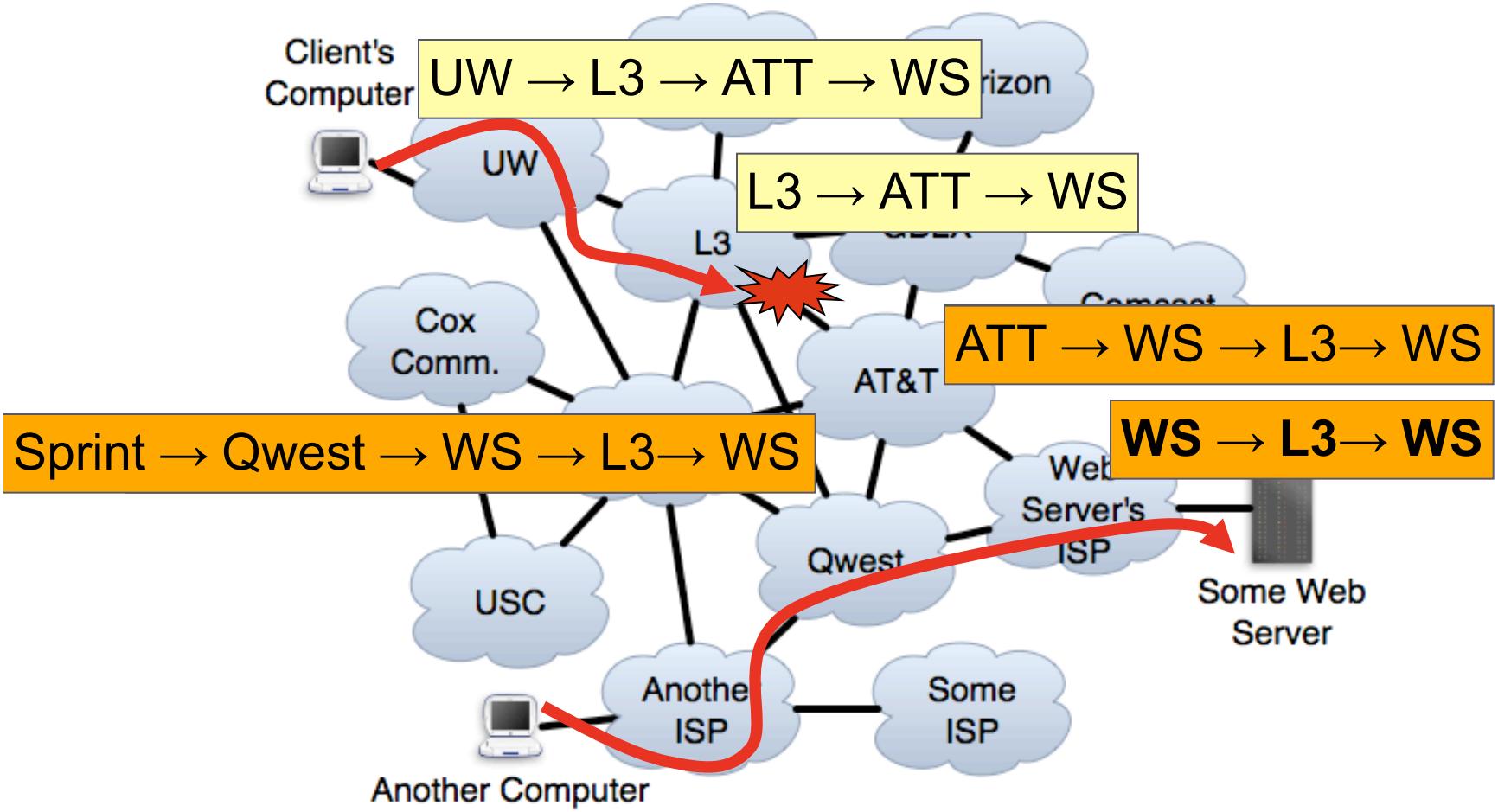
BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



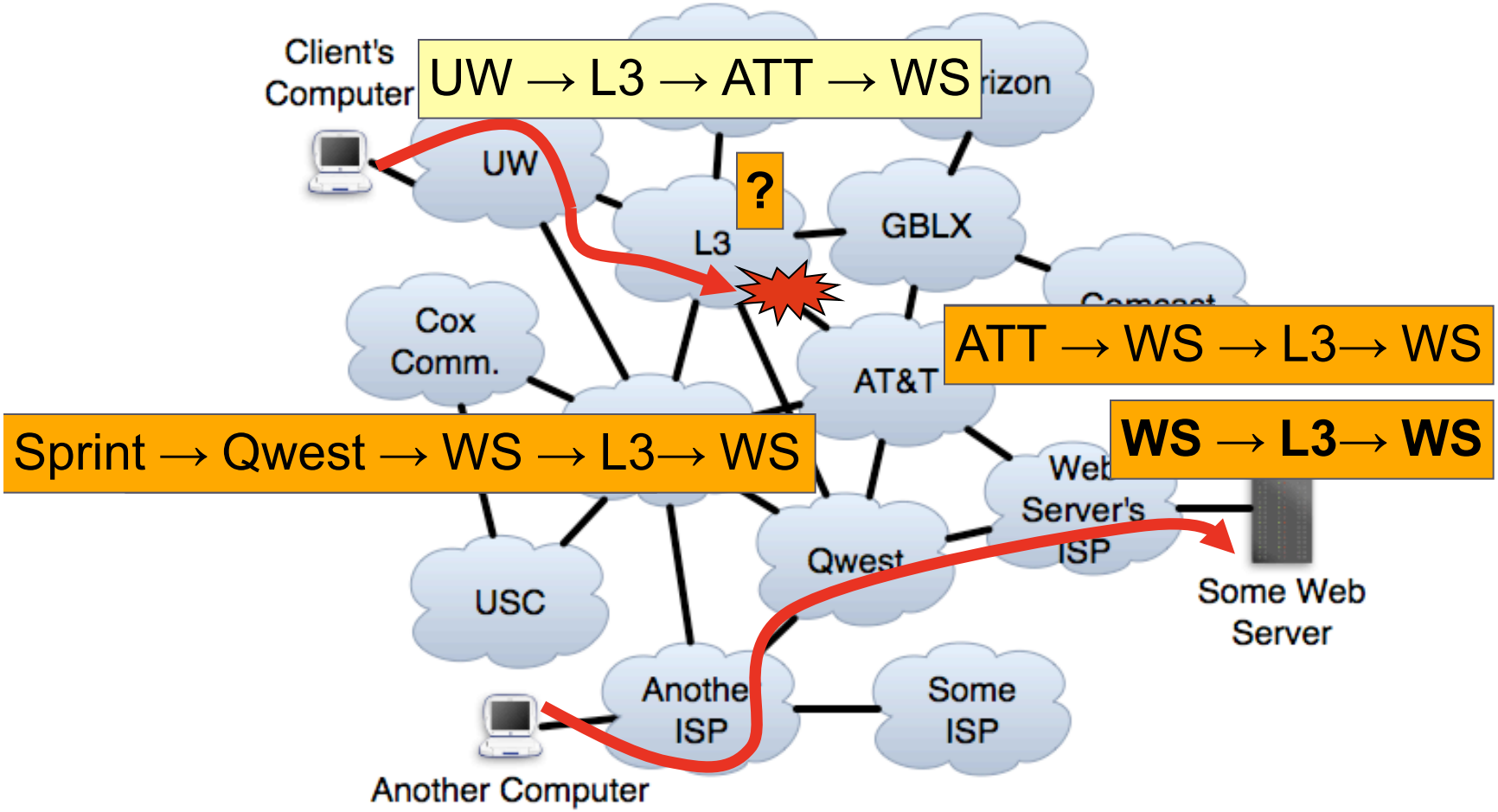
BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



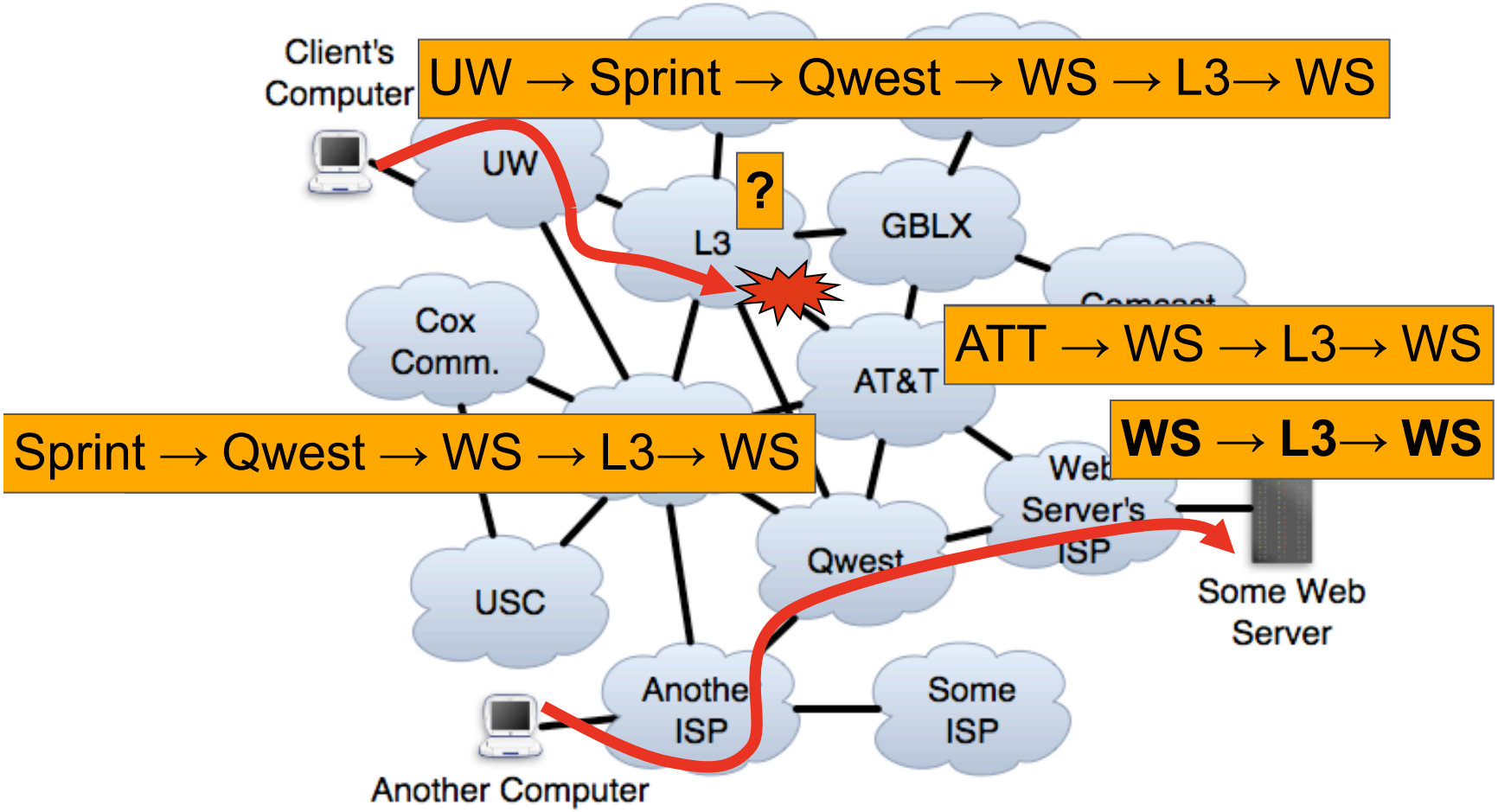
BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



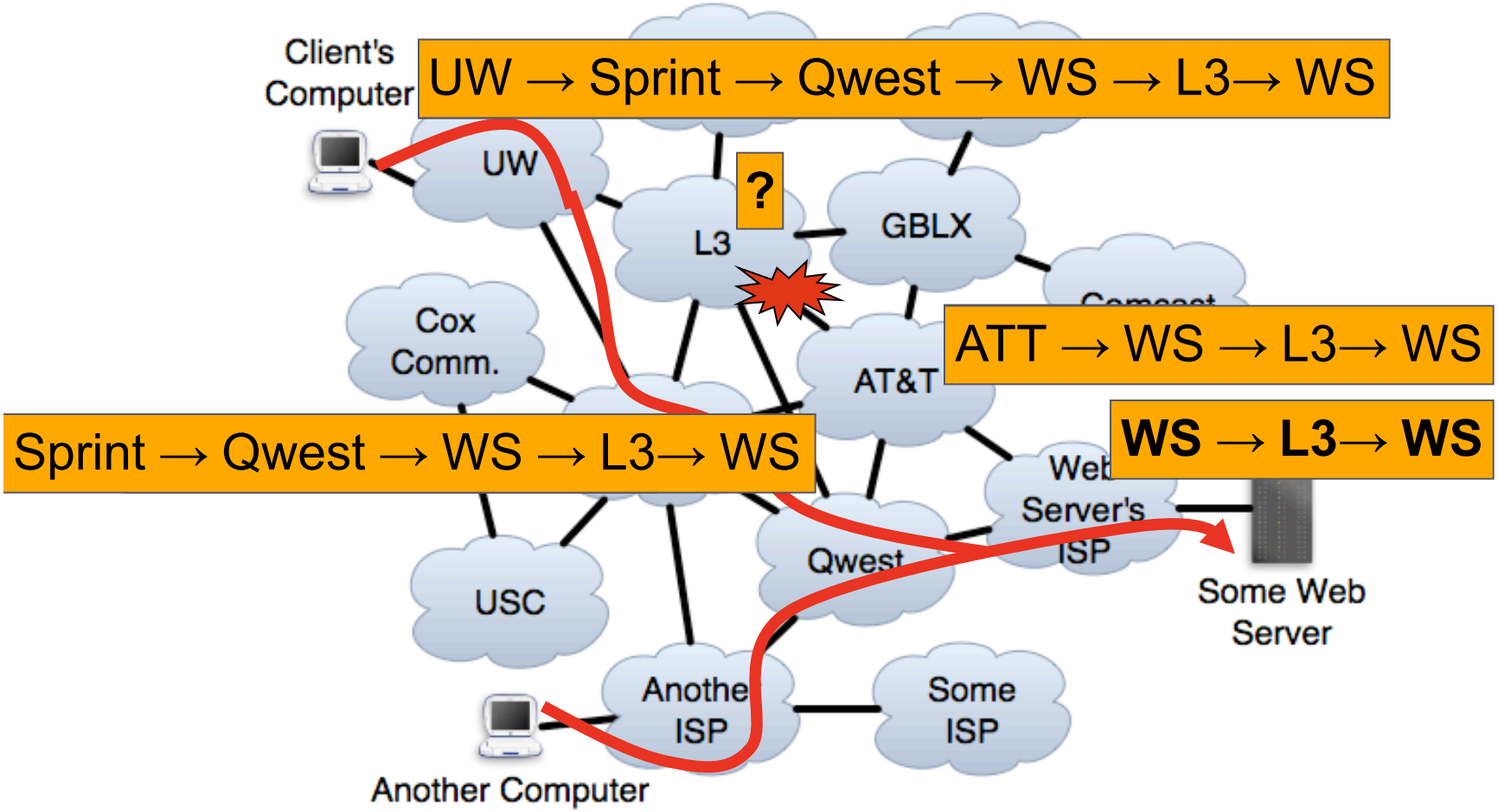
BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



BGP loop prevention encourages switch to working path.

Practical Self-Repair of Reverse Paths



BGP loop prevention encourages switch to working path.

Stuff I Don't Have Time to Talk About

Results from real poisonings

- Poisoning in the wild / poisoning anomalies
- Case study of restoring connectivity

Making poisoning flexible

- Monitoring broken path while it is disabled
- Allowing ISPs w/o alternatives to use disabled route

LIFEGUARD's scalability

- Overhead and speed of failure location
- Router update load if many ISPs deploy our approach

Alternatives to poisoning

- Compatibility with secure routing (BGPSEC, etc.)
- Comparing to other route control mechanisms

Can poisoning approximate AVOID effects?

LIFEGUARD's poisoning repairs outages by disabling routes to induce route exploration.

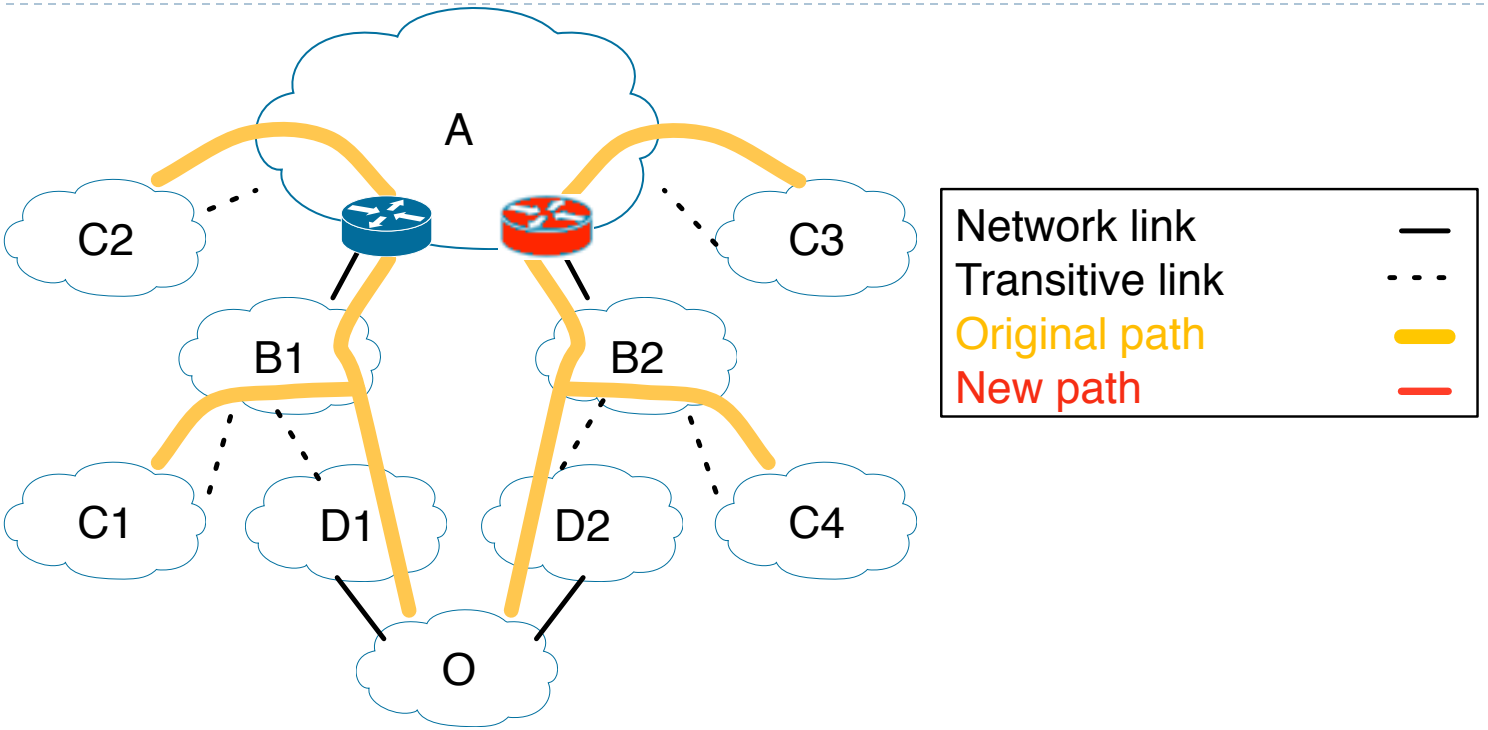
Q: Does poisoning disrupt working routes?

A: No. As I will describe:

(a) Under certain circumstances, we can disable a link without disabling the full ISP.

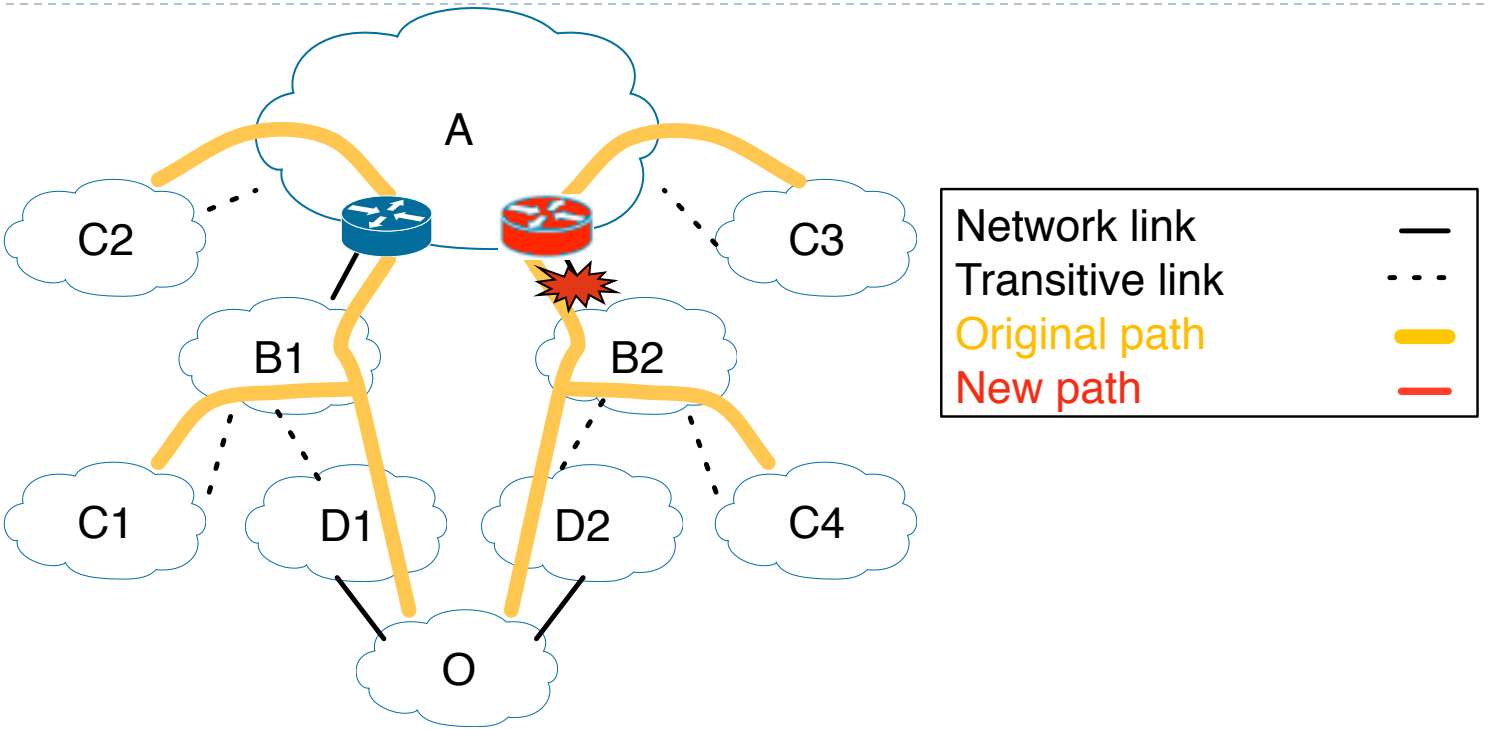
(b) We can speed BGP convergence by carefully crafting announcements.

What if some routes in an ISP still work?



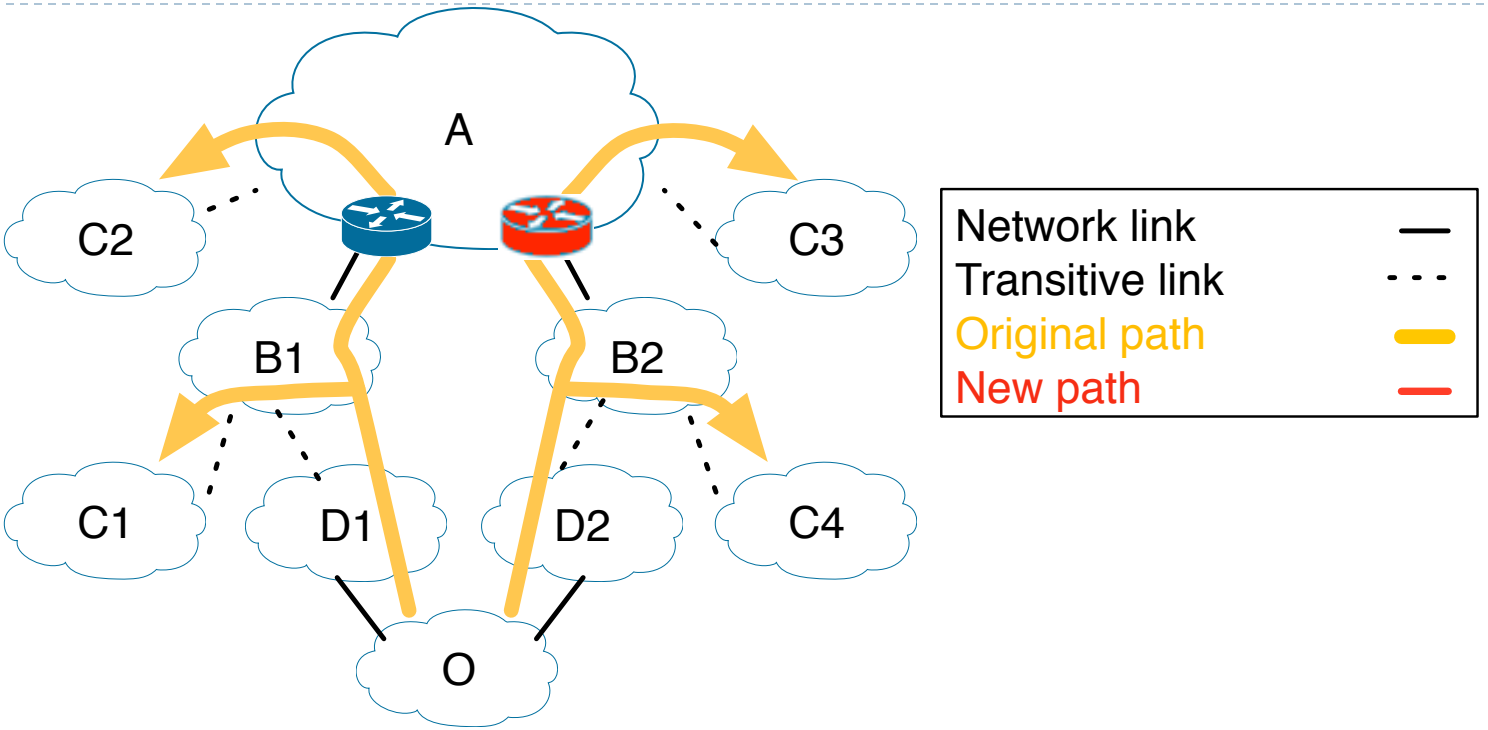
- We only want ***C3*** to change its route, to avoid ***A-B2***

What if some routes in an ISP still work?



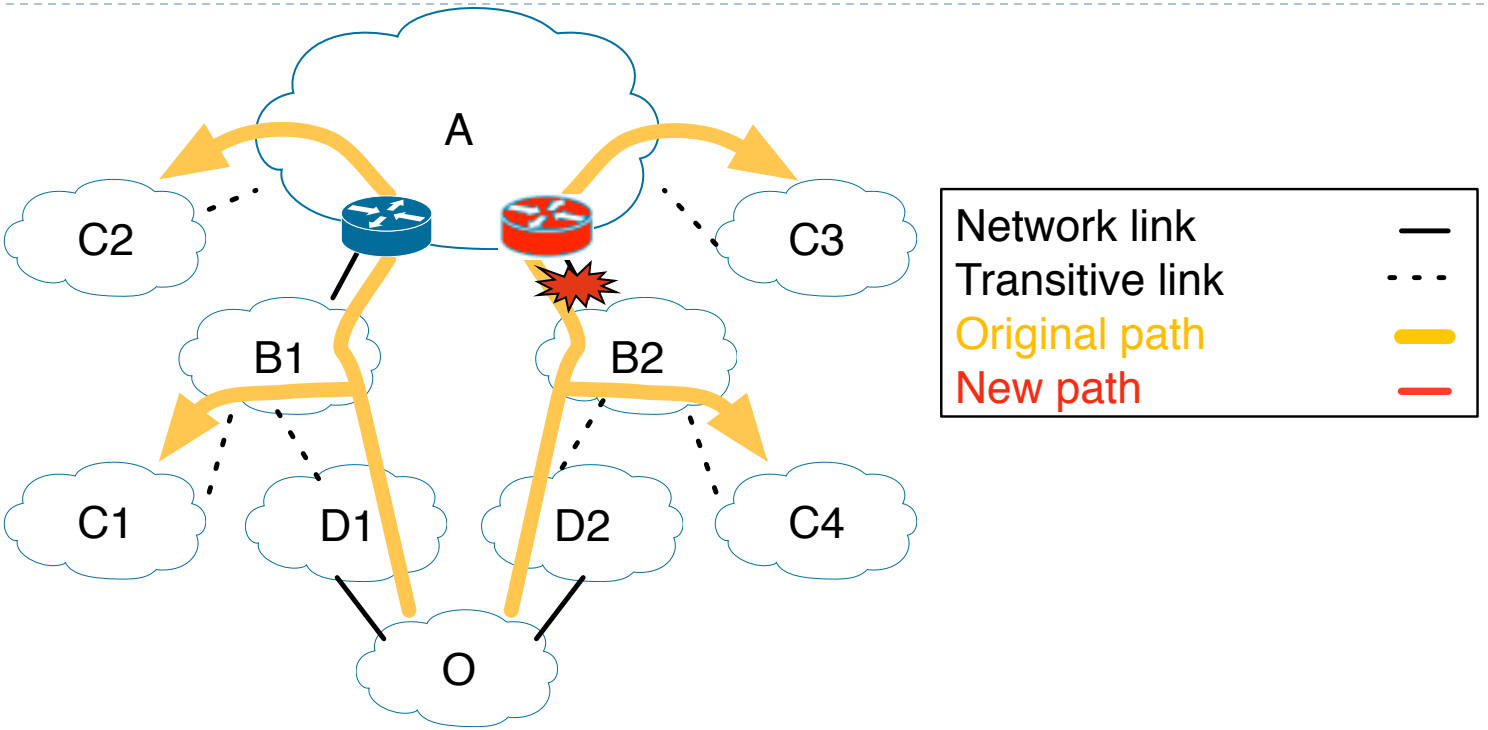
- We only want ***C3*** to change its route, to avoid ***A-B2***

What if some routes in an ISP still work?



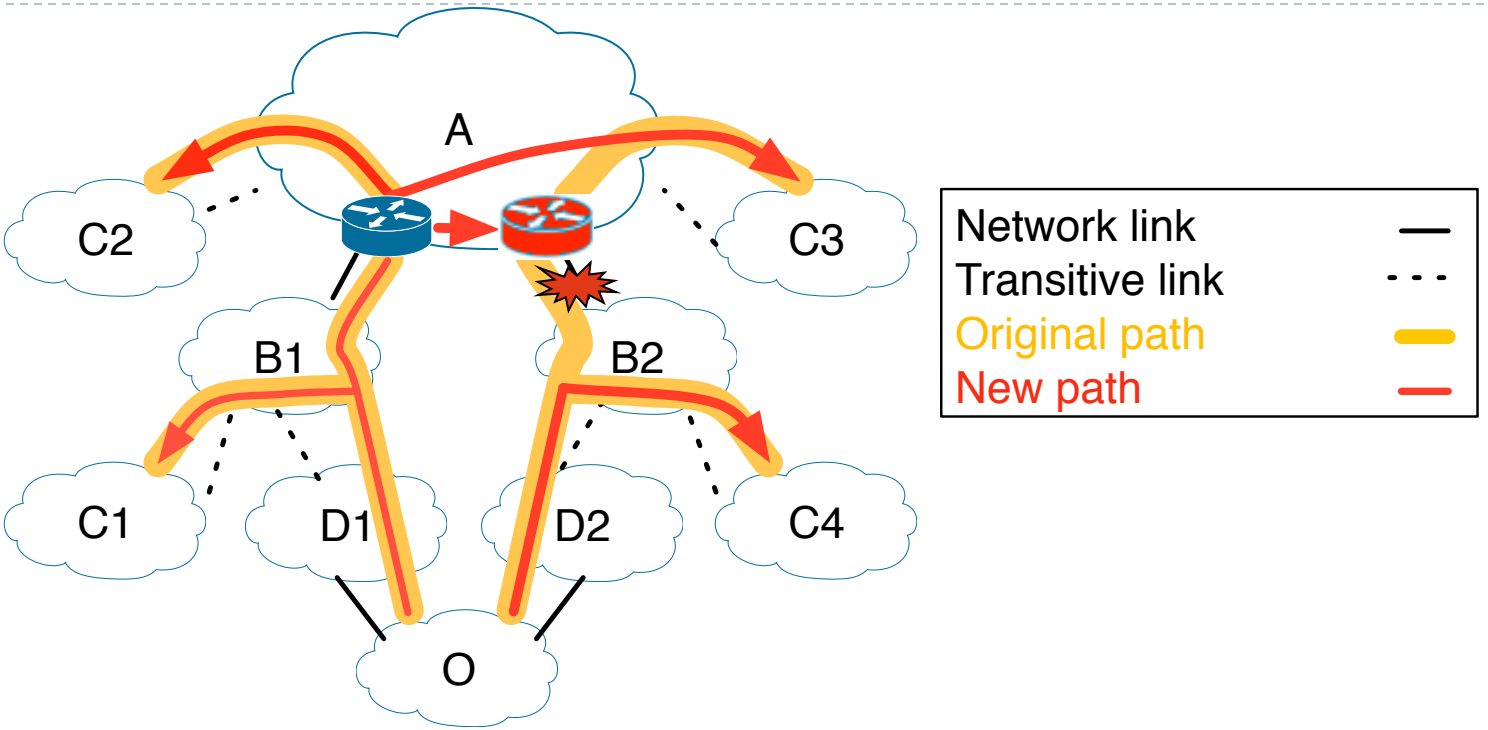
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Forward direction is easy: choose a different route

What if some routes in an ISP still work?



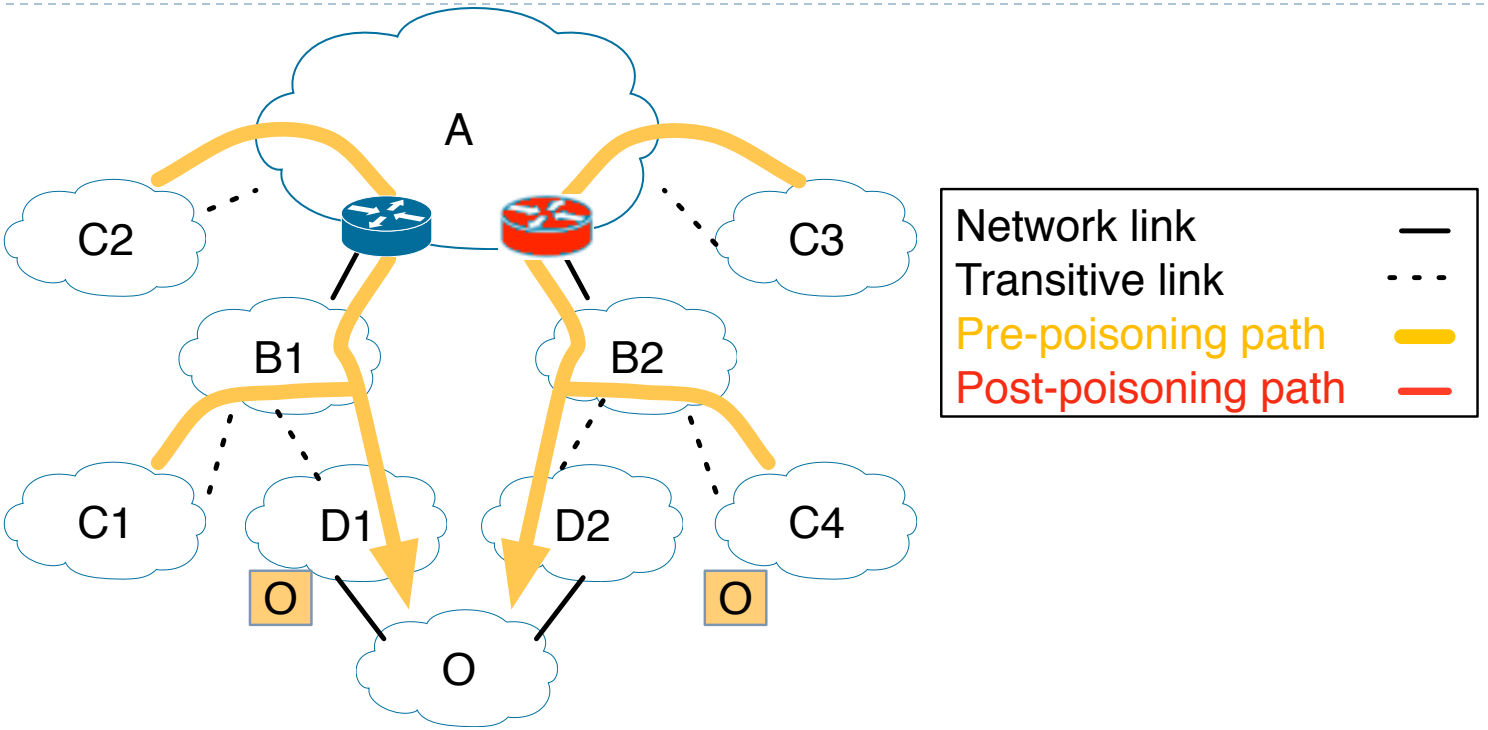
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Forward direction is easy: choose a different route

What if some routes in an ISP still work?



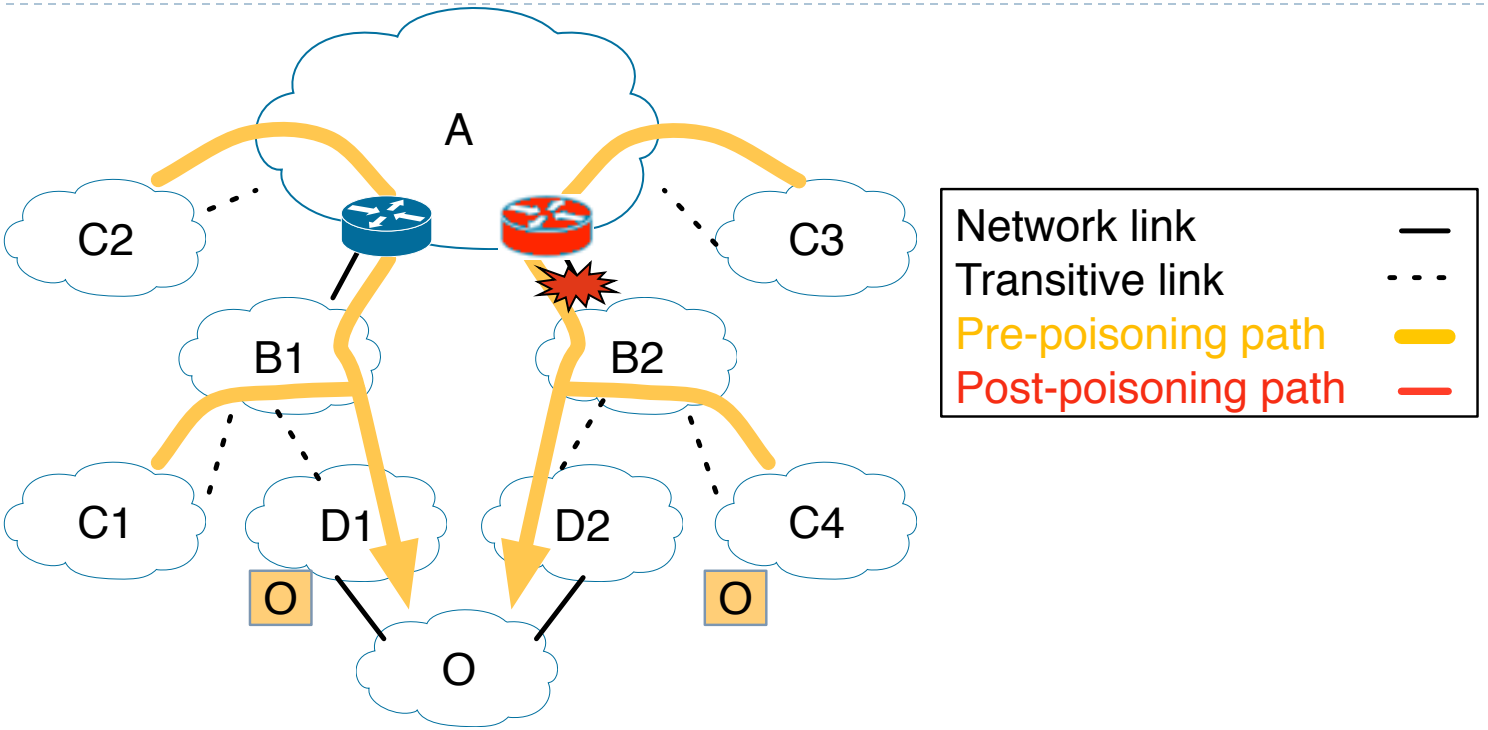
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Forward direction is easy: choose a different route

What if some routes in an ISP still work?



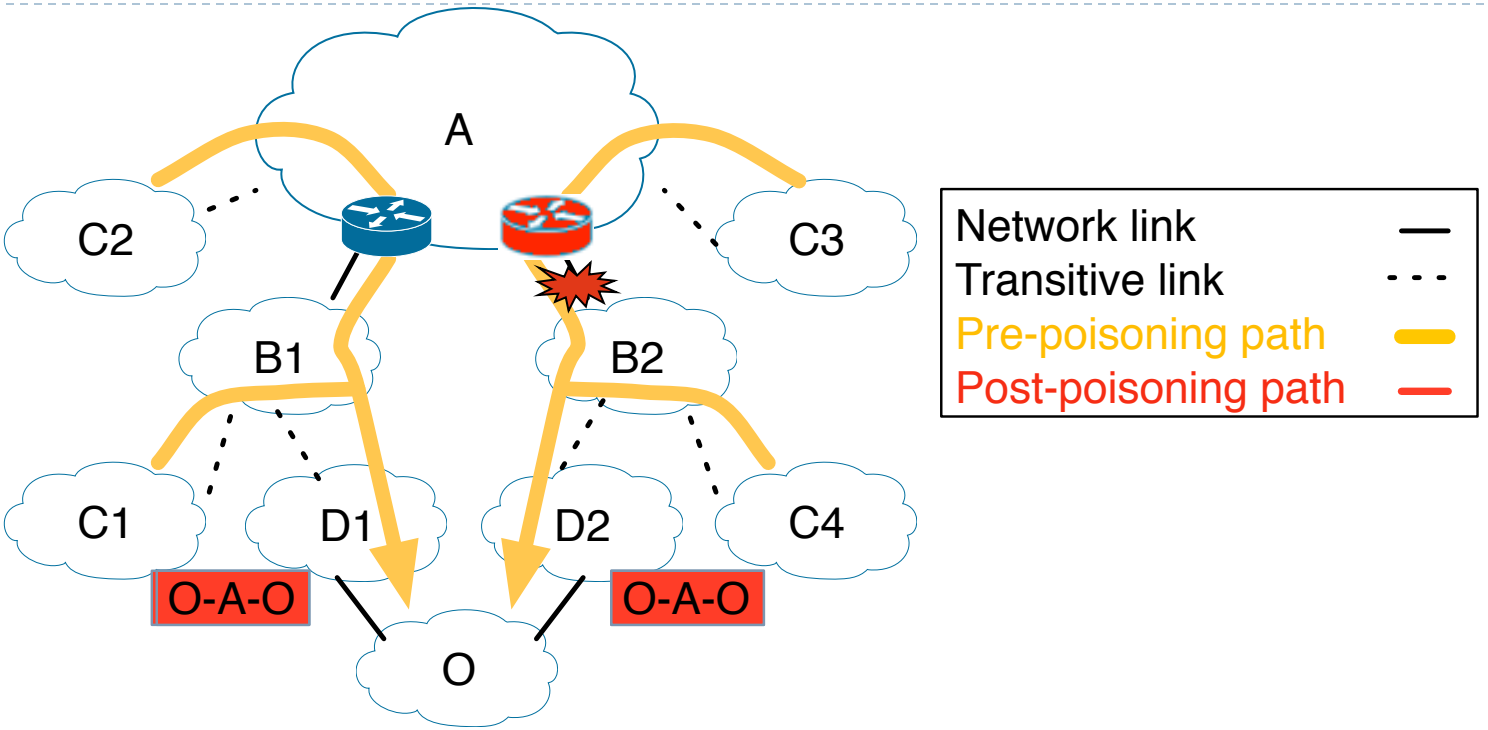
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP

What if some routes in an ISP still work?



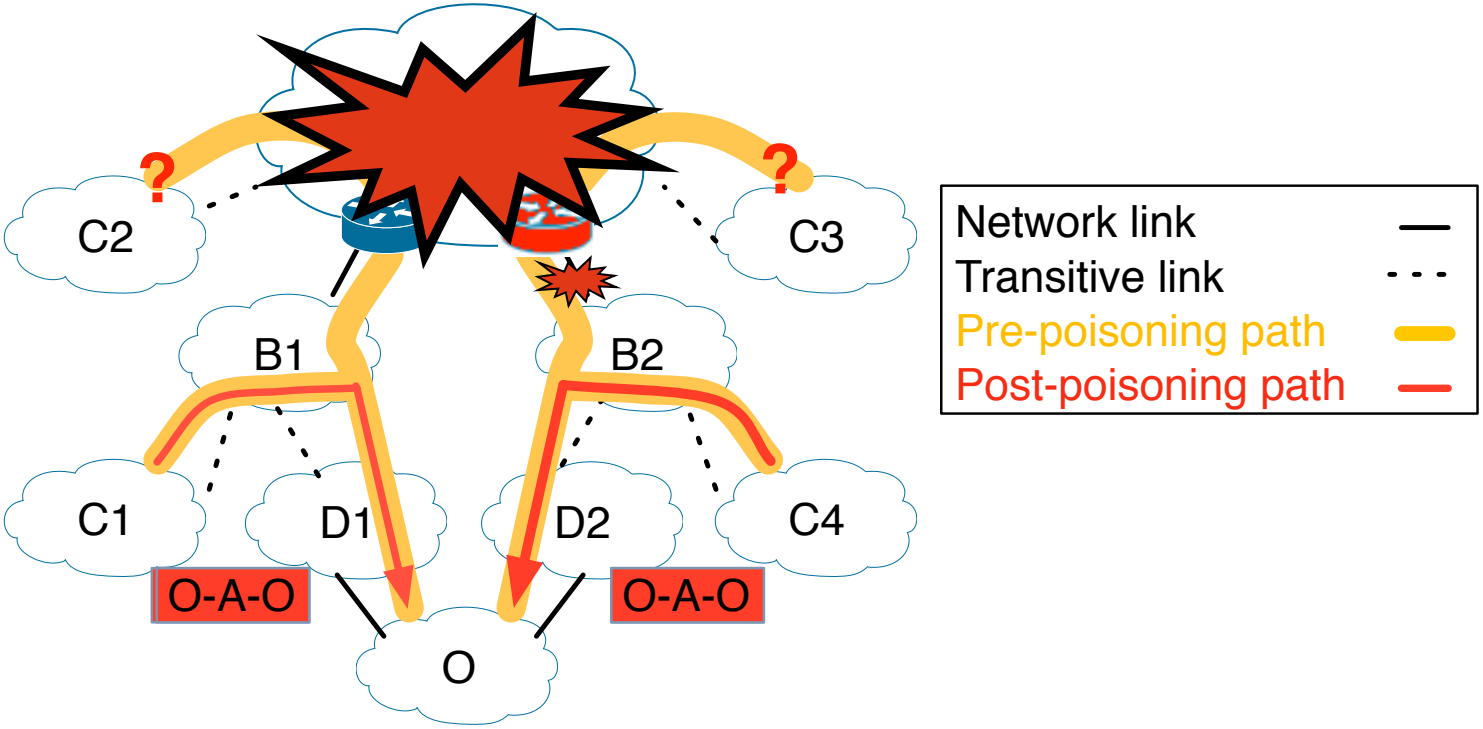
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP

What if some routes in an ISP still work?



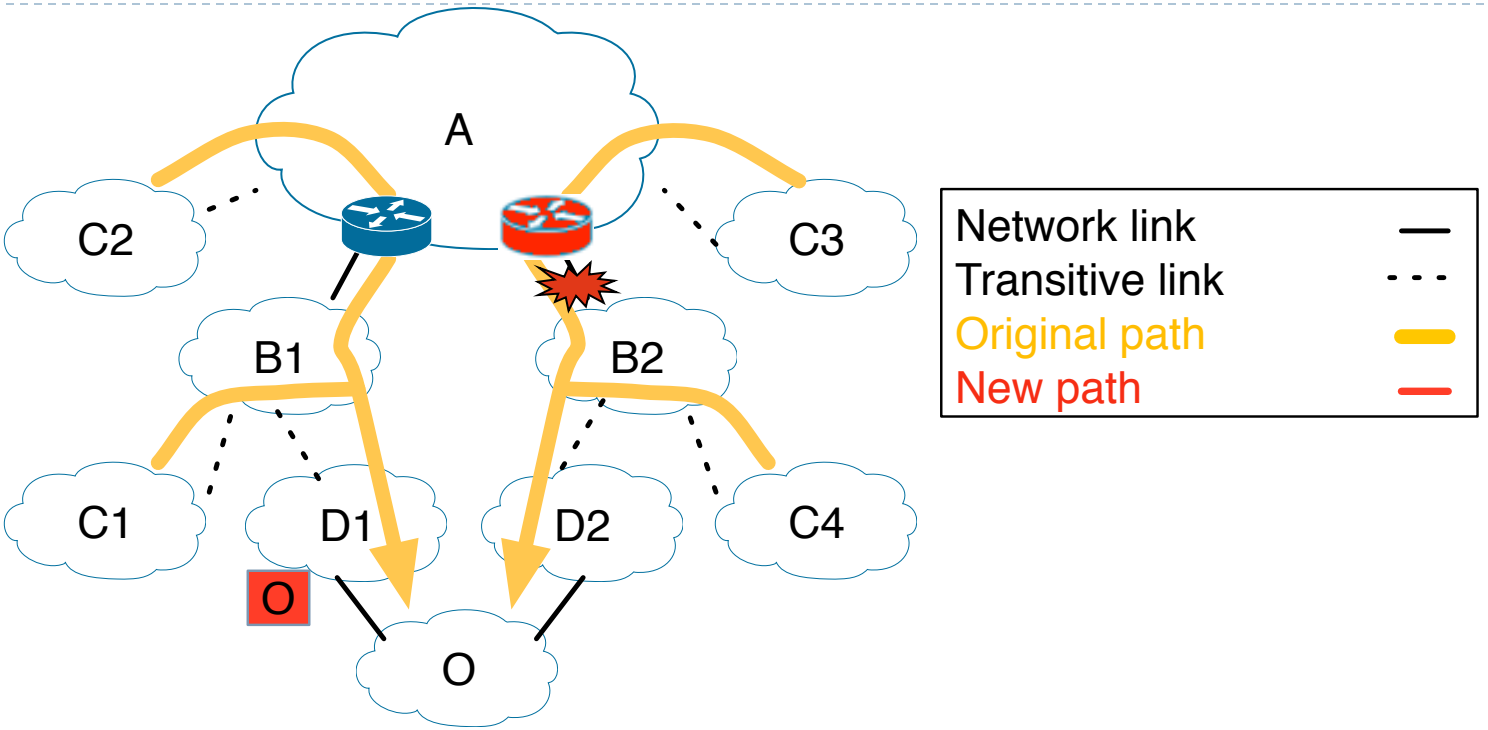
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP

What if some routes in an ISP still work?



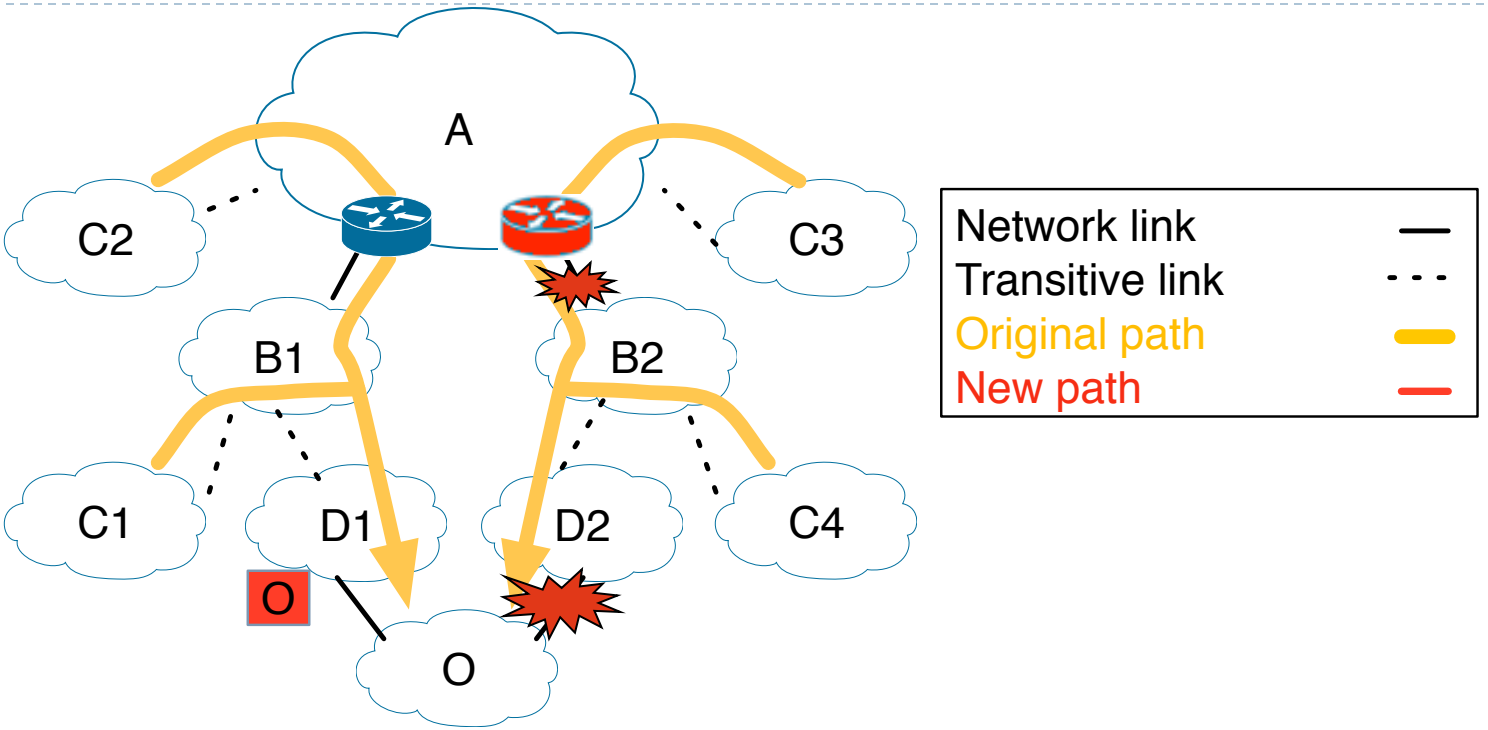
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP

What if some routes in an ISP still work?



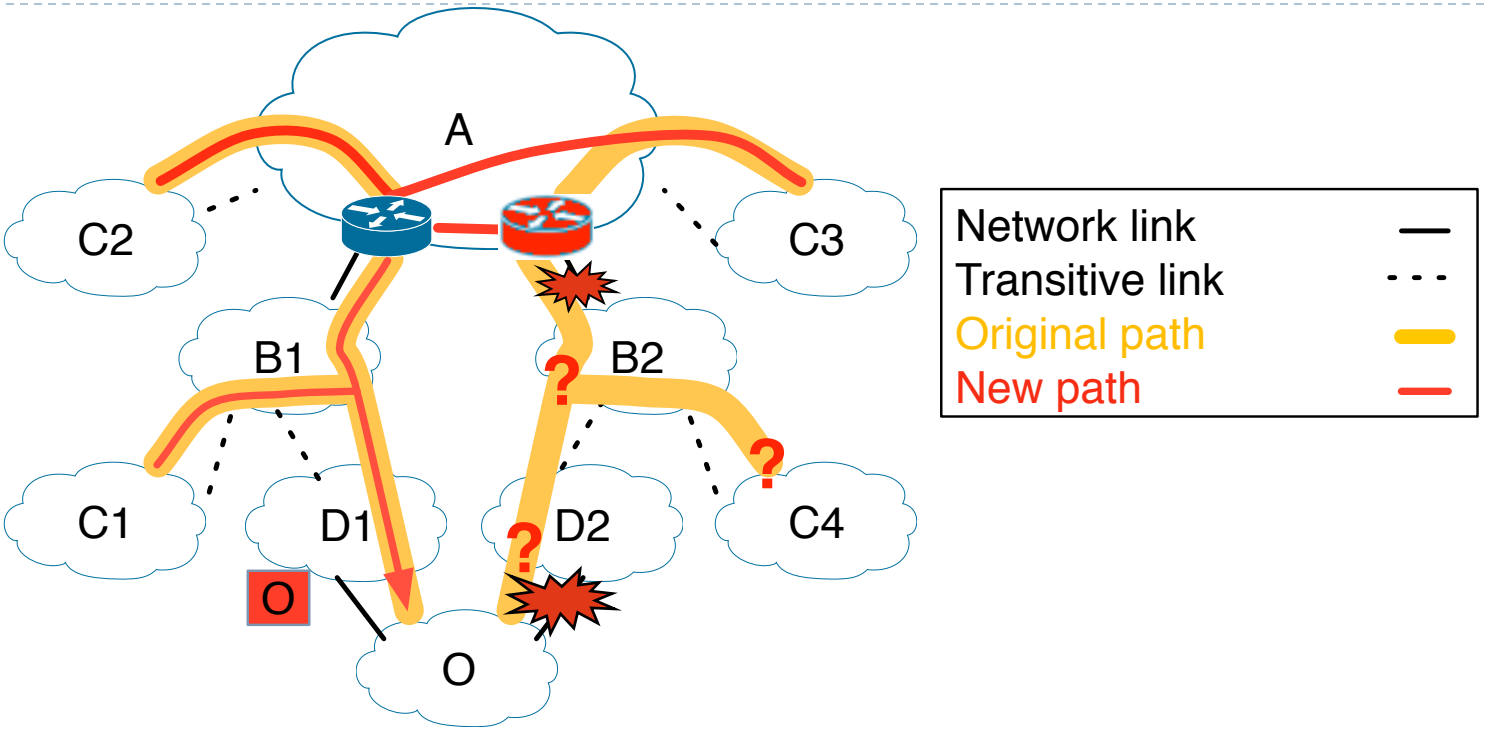
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP
- Selective advertising via just ***D1*** is also blunt

What if some routes in an ISP still work?



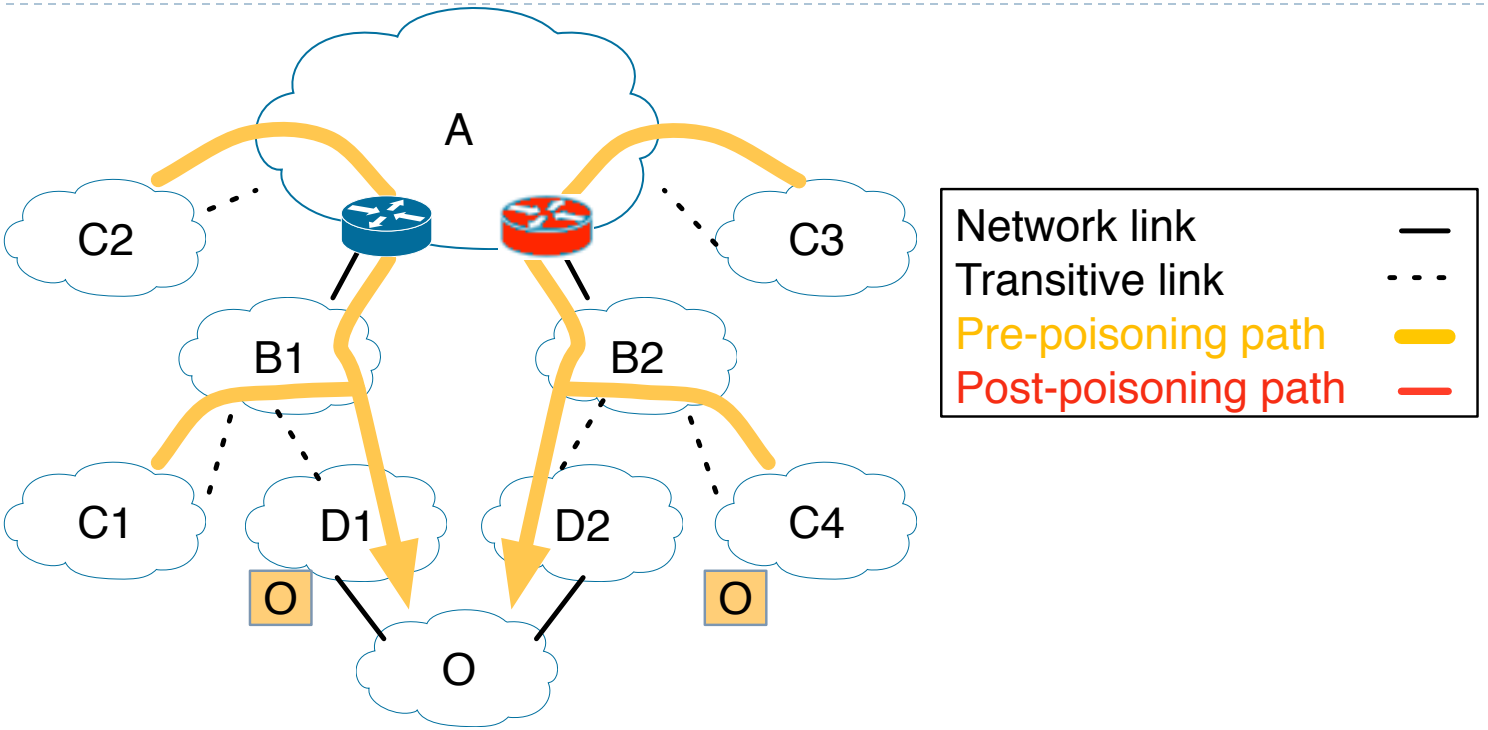
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP
- Selective advertising via just ***D1*** is also blunt

What if some routes in an ISP still work?



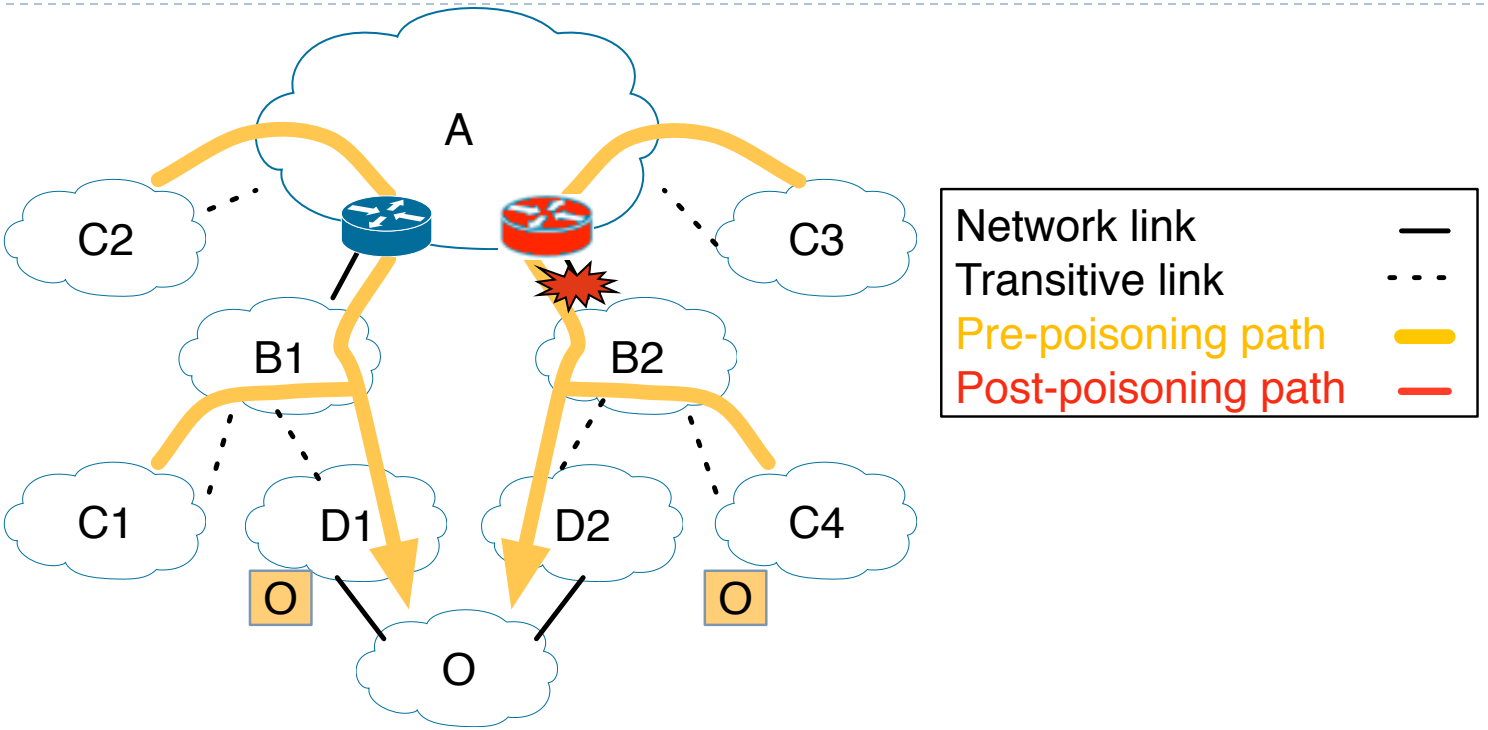
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP
- Selective advertising via just ***D1*** is also blunt

What if some routes in an ISP still work?



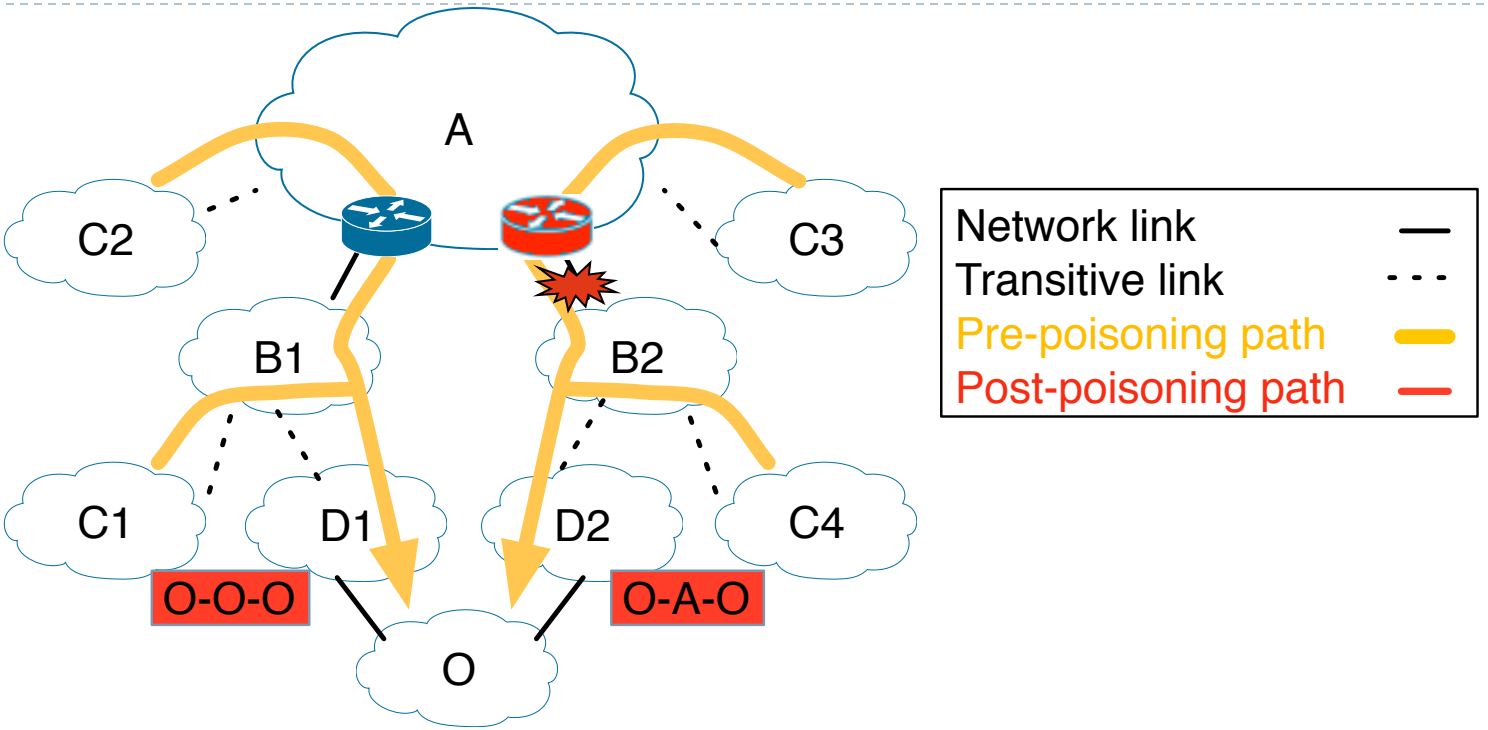
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP
- If ***D1*** and ***D2*** (transitively) connect to different PoPs of ***A***, selectively poison via ***D2*** and not ***D1***

What if some routes in an ISP still work?



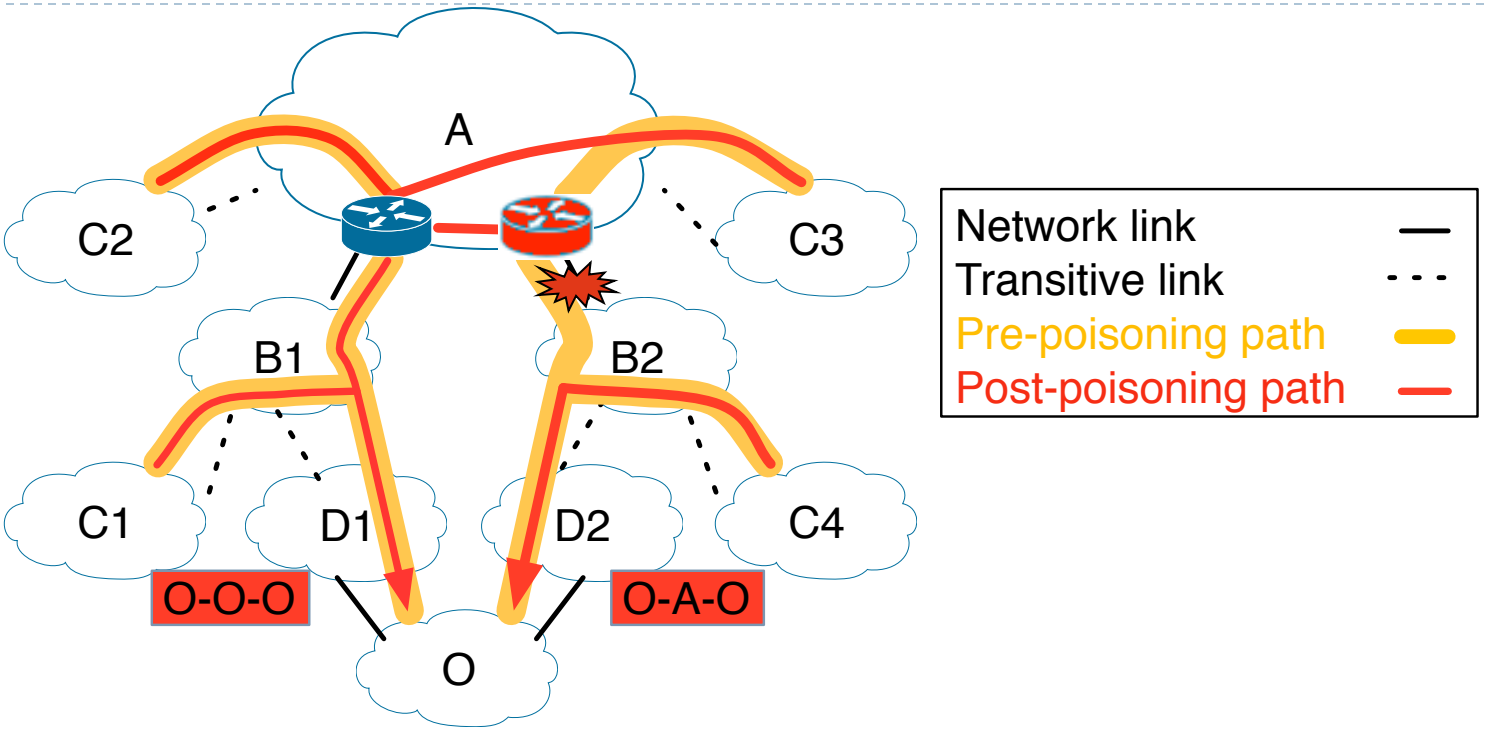
- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP
- If ***D1*** and ***D2*** (transitively) connect to different PoPs of ***A***, selectively poison via ***D2*** and not ***D1***

What if some routes in an ISP still work?



- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP
- If ***D1*** and ***D2*** (transitively) connect to different PoPs of ***A***, selectively poison via ***D2*** and not ***D1***

What if some routes in an ISP still work?



- We only want ***C3*** to change its route, to avoid ***A-B2***
- Poisoning seems blunt, disabling an entire ISP
- If ***D1*** and ***D2*** (transitively) connect to different PoPs of ***A***, selectively poison via ***D2*** and not ***D1***

Can poisoning approximate AVOID effects?

LIFEGUARD's poisoning repairs outages by disabling routes to induce route exploration.

Q: Does poisoning disrupt working routes?

A: No. As I will describe:

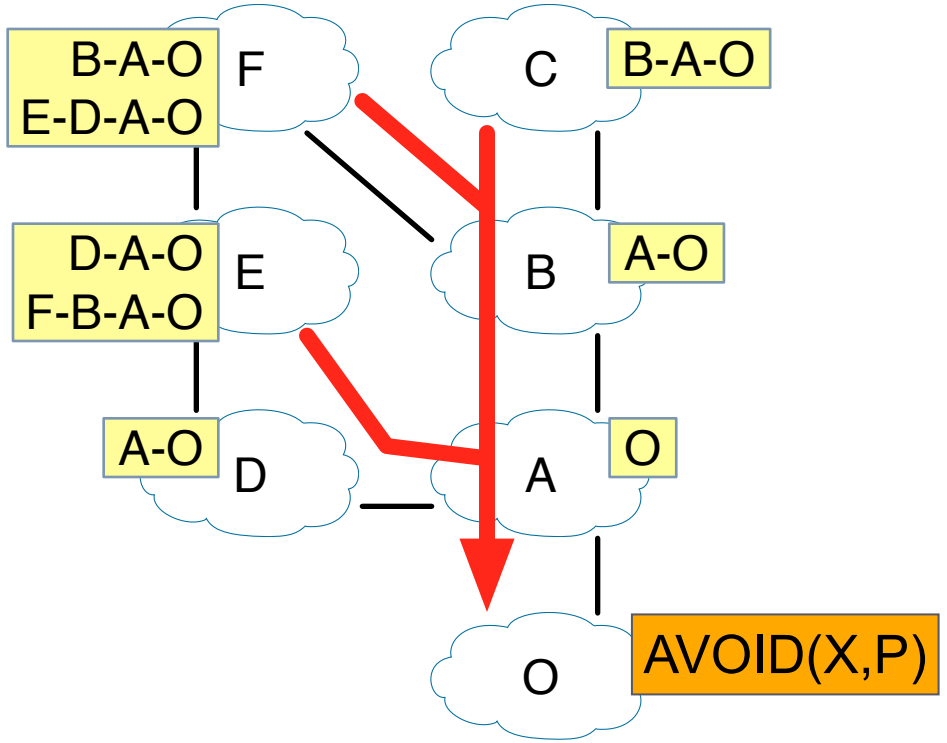
(a) "Selective poisoning" can avoid 73% of links without disabling entire AS.

- Real-world results from 5 provider BGP-Mux testbed

(b) We can speed BGP convergence by carefully crafting announcements.

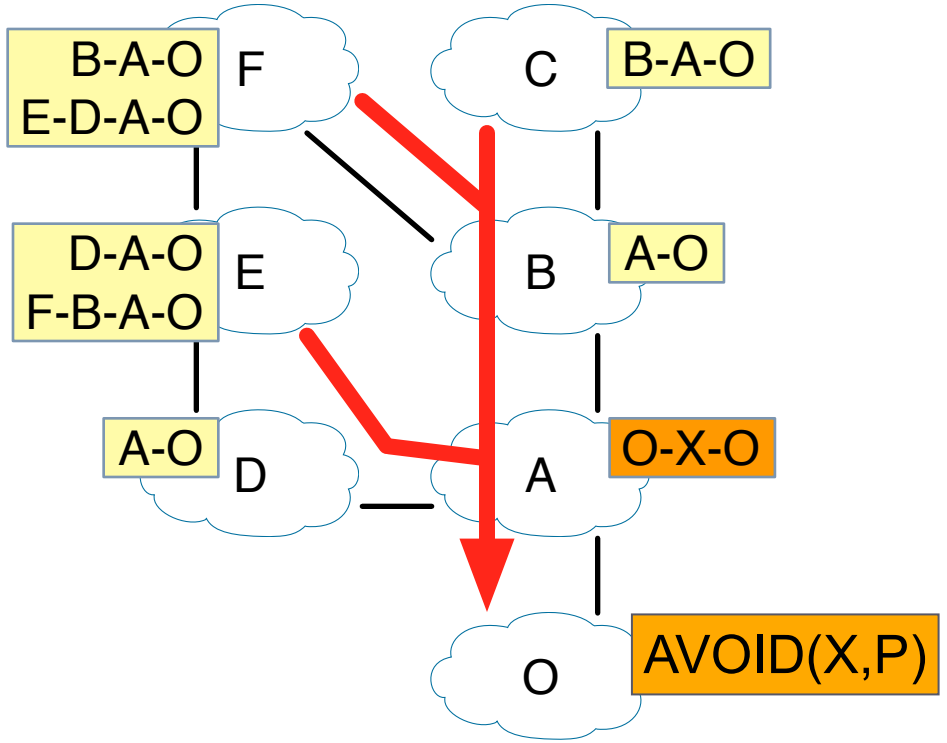
Naive Poisoning Causes Transient Loss

- 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



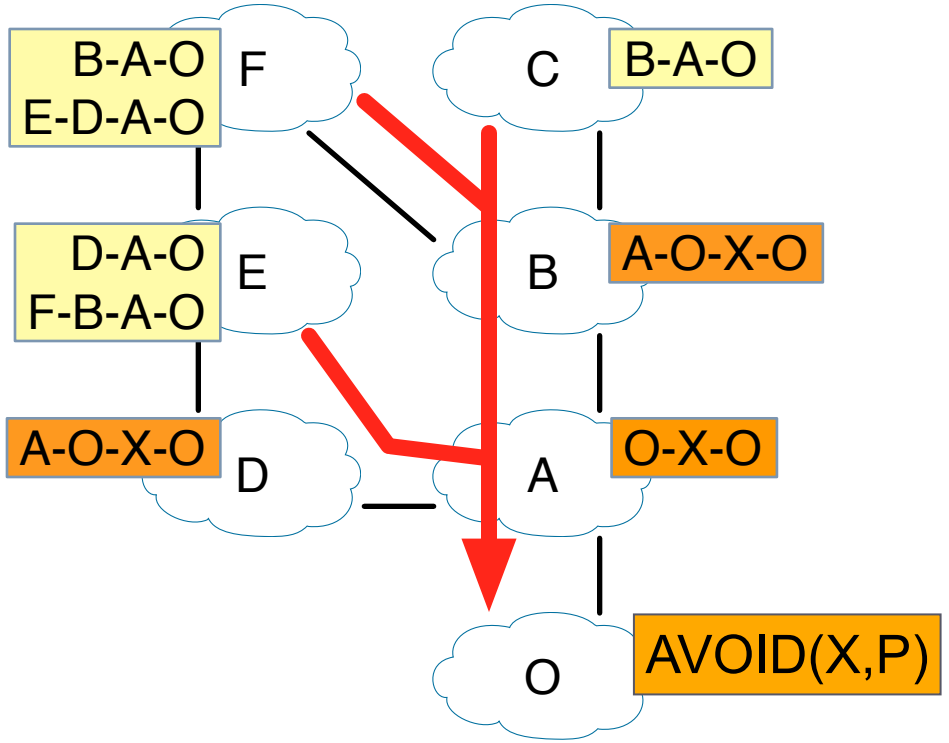
Naive Poisoning Causes Transient Loss

- 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



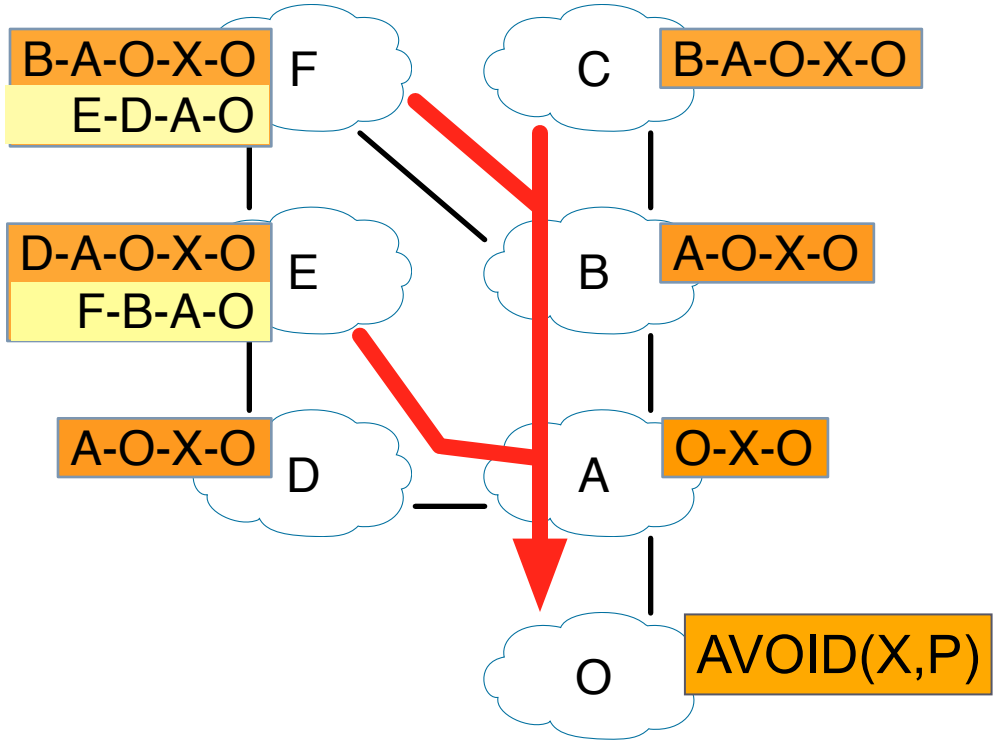
Naive Poisoning Causes Transient Loss

- 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



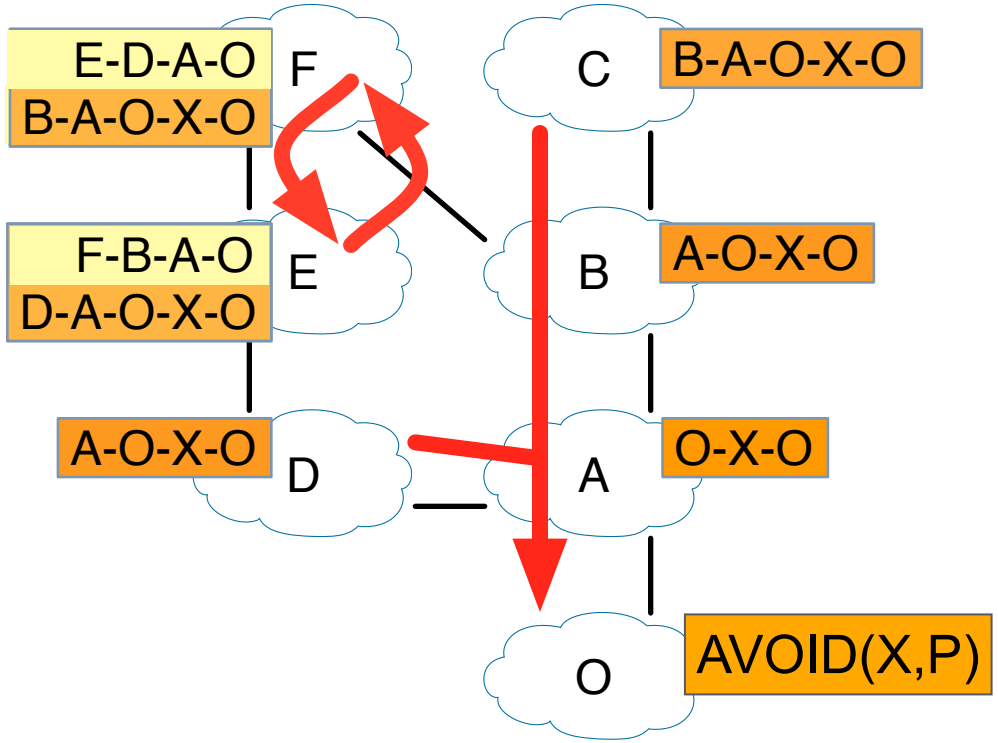
Naive Poisoning Causes Transient Loss

- 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



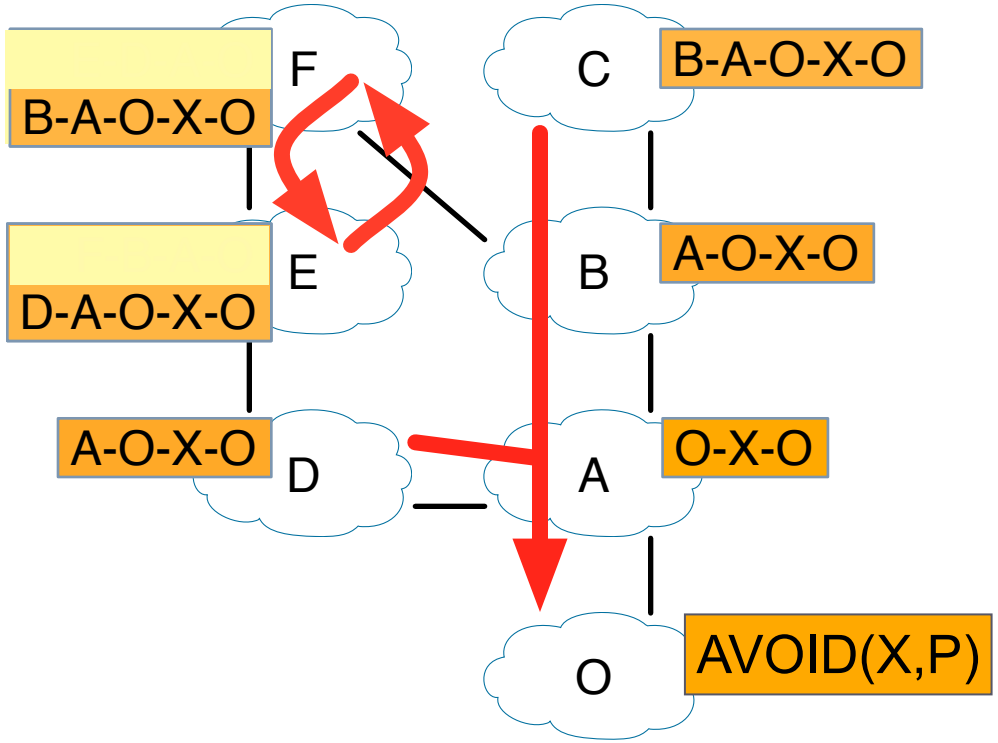
Naive Poisoning Causes Transient Loss

- 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



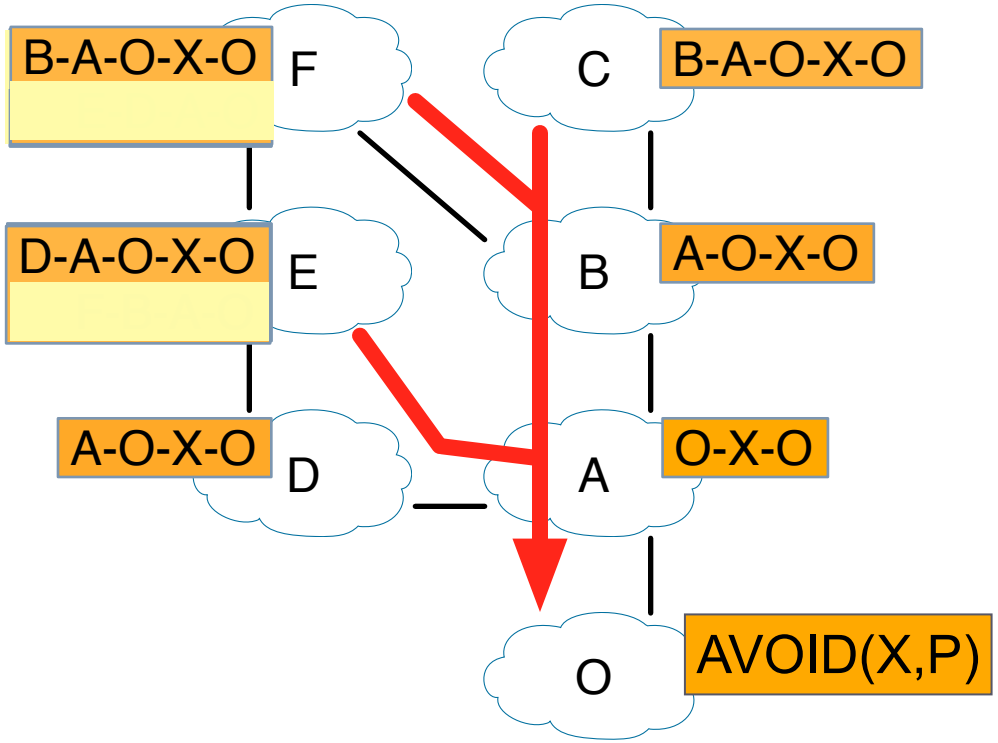
Naive Poisoning Causes Transient Loss

- 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



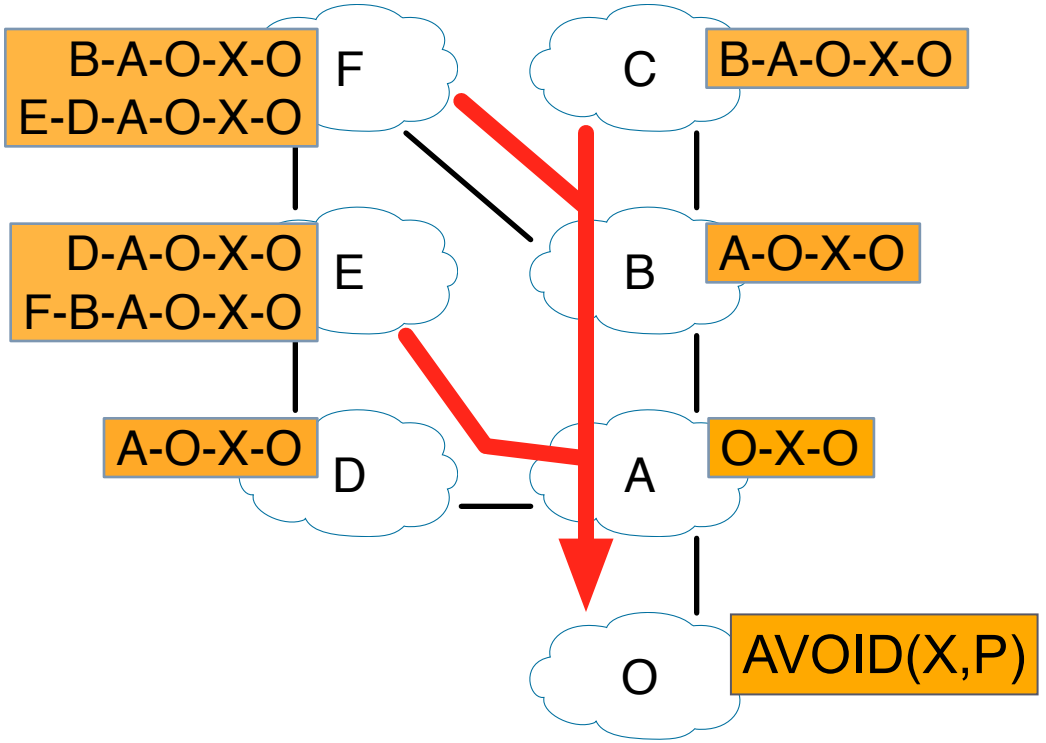
Naive Poisoning Causes Transient Loss

- 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



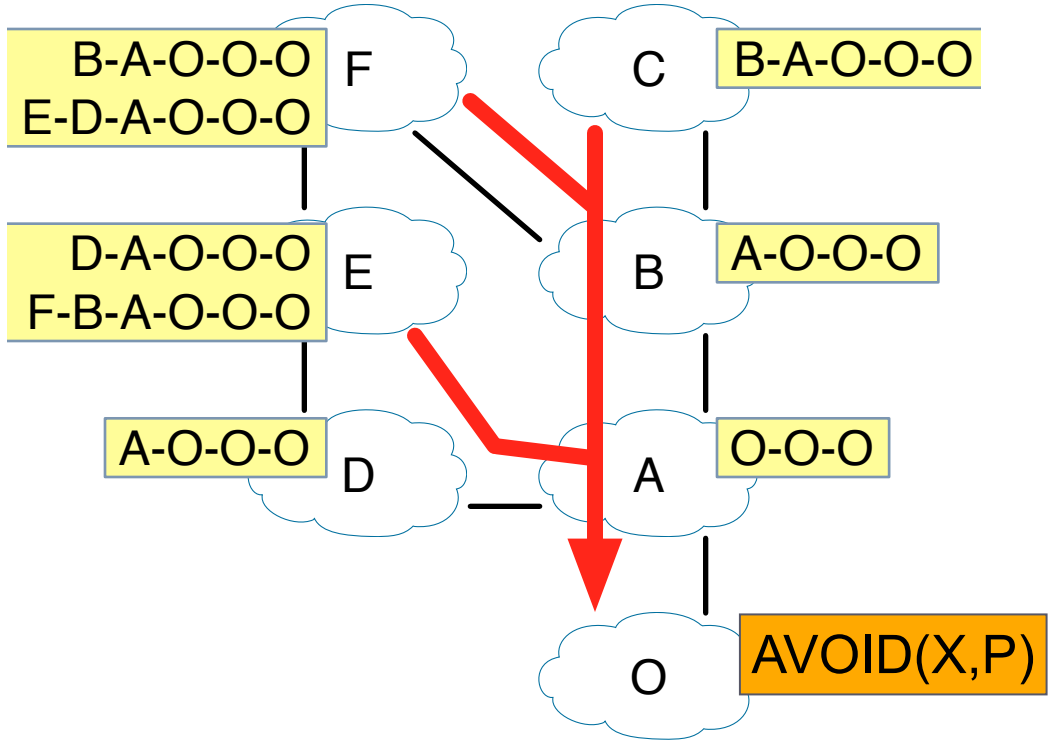
Naive Poisoning Causes Transient Loss

- 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



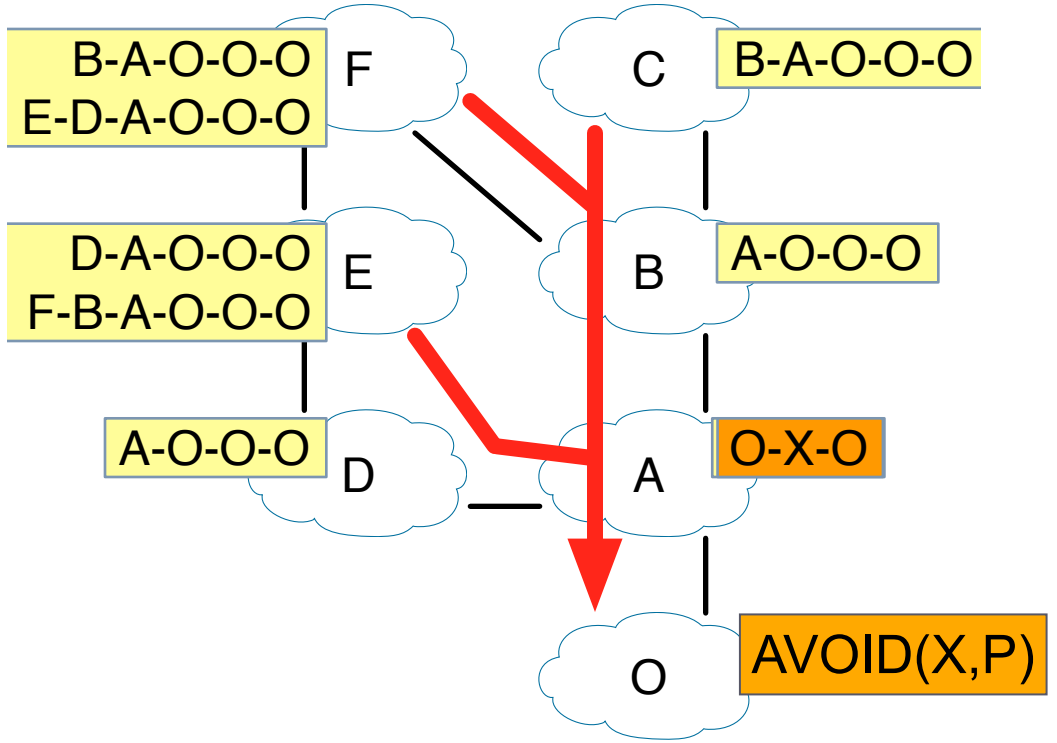
Prepend to Reduce Path Exploration

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



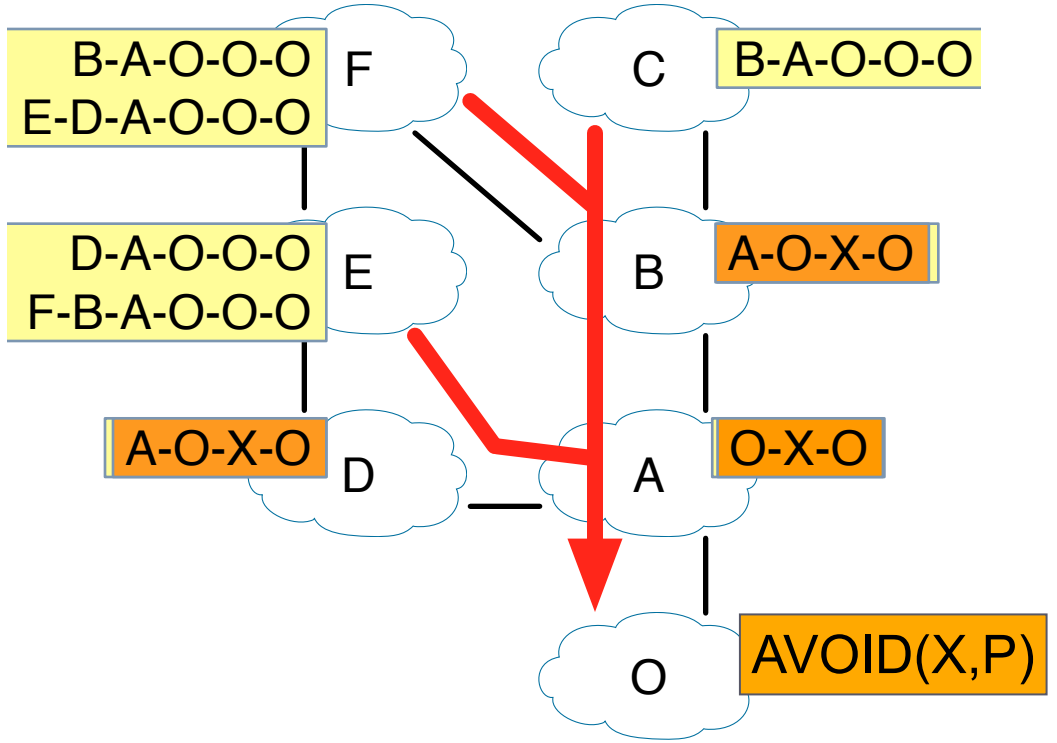
Prepend to Reduce Path Exploration

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



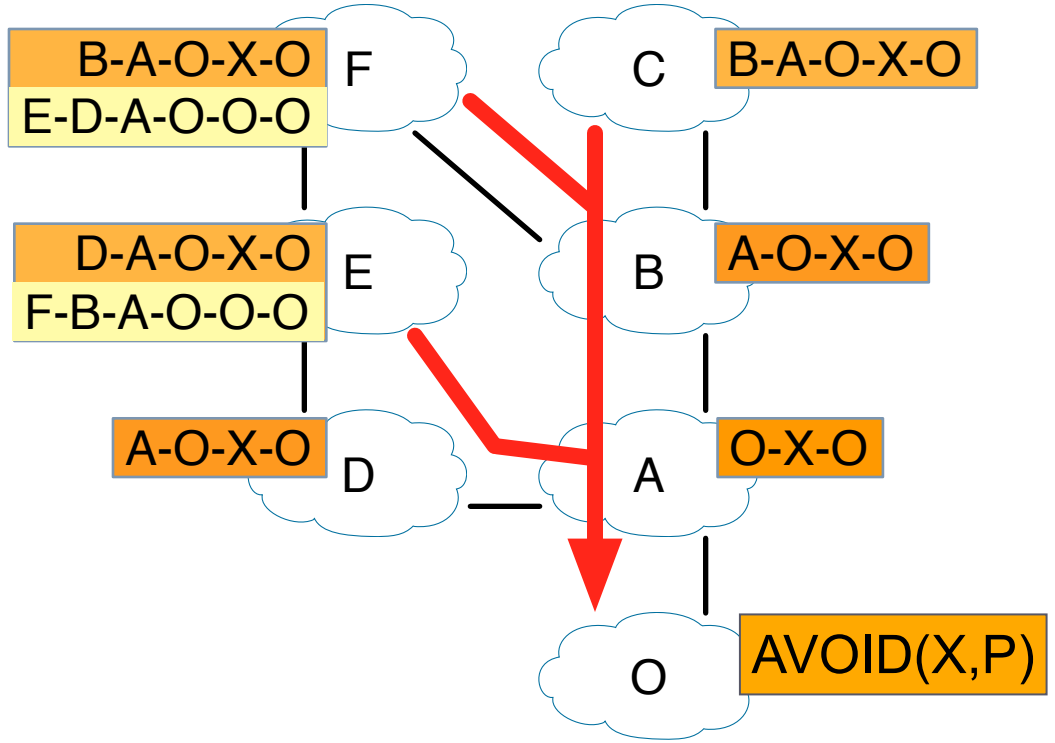
Prepend to Reduce Path Exploration

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



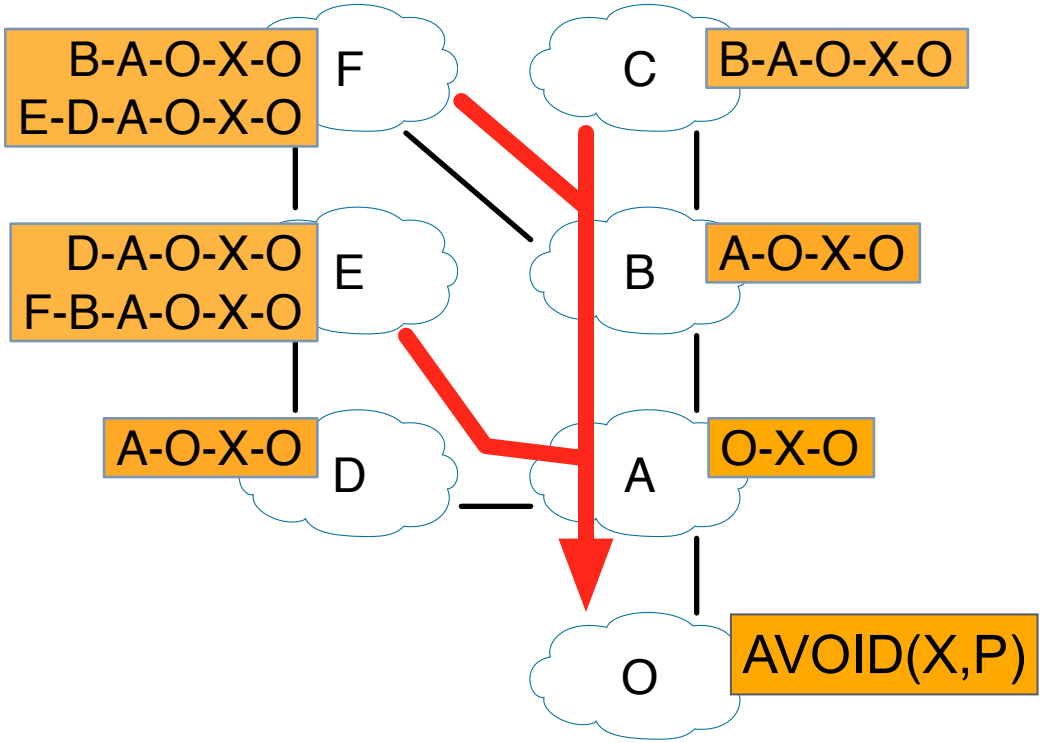
Prepend to Reduce Path Exploration

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

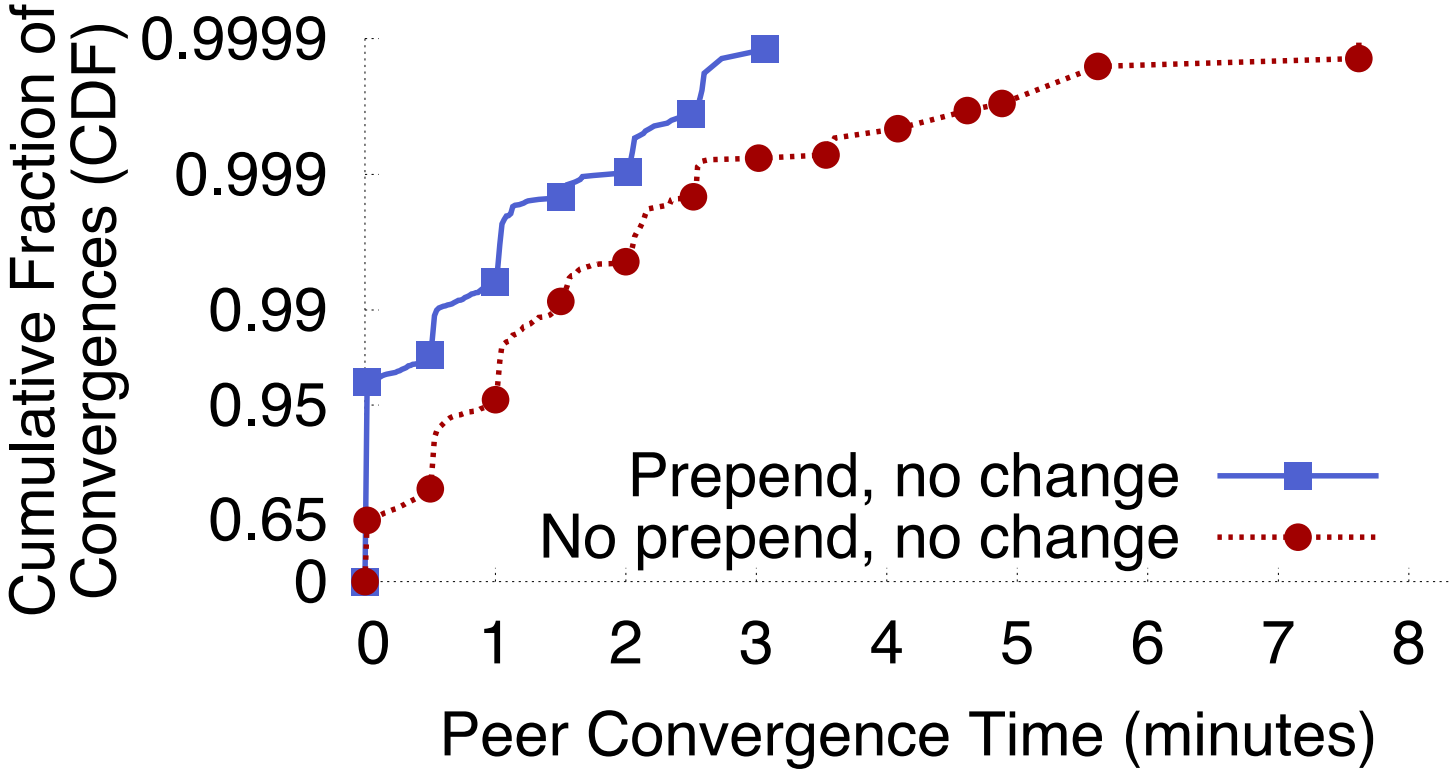


Prepend to Reduce Path Exploration

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



Prepending Speeds Convergence



- 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

Conclusion

- We increasingly depend on the Internet, but availability lags
- Much of Internet unavailability due to long-lasting outages

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

- Location challenge: Find problem, given unidirectional failures and tools that depend on connectivity
  - Use reverse traceroute, isolate directions, use historical view
- Avoidance challenge: Reroute without participation of transit networks
  - BGP poisoning gives control to the destination
  - Well-crafted announcements ease concerns