AS Interfaces

Katerina Argyraki

with

Petros Maniatis, Timothy Roscoe, Scott Shenker

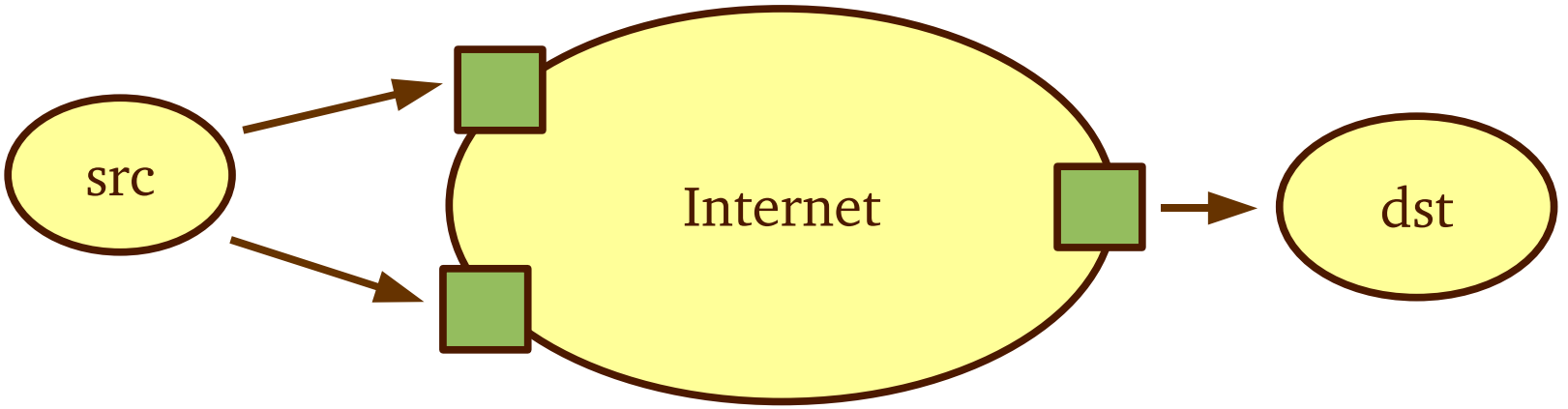
Internet path control

- Route control
  - where my outgoing traffic goes
- Path identification + filtering
  - where my incoming traffic is coming from
- Accountability
  - who did what to my traffic

It's an old story

- Loose source record route IP option
  - source specifies router-level path
  - receiver learns router-level path
- Didn't go anywhere
  - forwarding overhead in routers
  - security considerations

The Internet, viewed from the edge



- No control beyond first/last hop
- No transparency = no accountability

It's a black box

Indirect path control

- Probing to reverse-engineer structure/failures
  - traceroute, network tomography
  - accountability??
- Overlays to affect outgoing/incoming path
  - Skype, Prolexic, RON, SOS
  - critical applications??
- At the mercy of ISPs

End systems are seeking better path control

The ISP viewpoint

- Probing is dangerous
  - can reveal vulnerabilities
  - business policies
- Overlay traffic is undesirable
  - does not generate revenue
  - can interfere with traffic engineering

End systems already have too much path control

The ISP viewpoint (2)

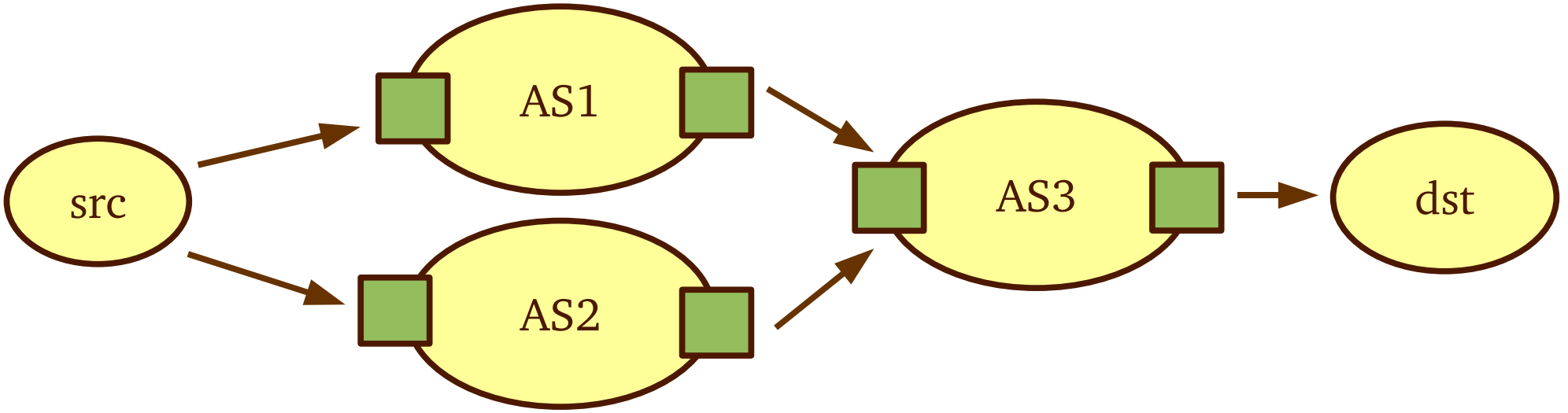
- Customers have come to expect it
  - traceroute = health monitor
  - net neutrality

...but ISPs can't just get rid of it

The right path-control balance?

- Useful visibility + control for the edge
  - monitor ISP performance
  - localize/adapt to failures, DDoS attacks
- Respecting ISP privacy + business model
  - keep internal structure opaque
  - absolute control over routing policies

ASes as first-class Internet objects



- ASes export checkpoints
  - points of explicit visibility and control

Expose Internet view as graph of ASes

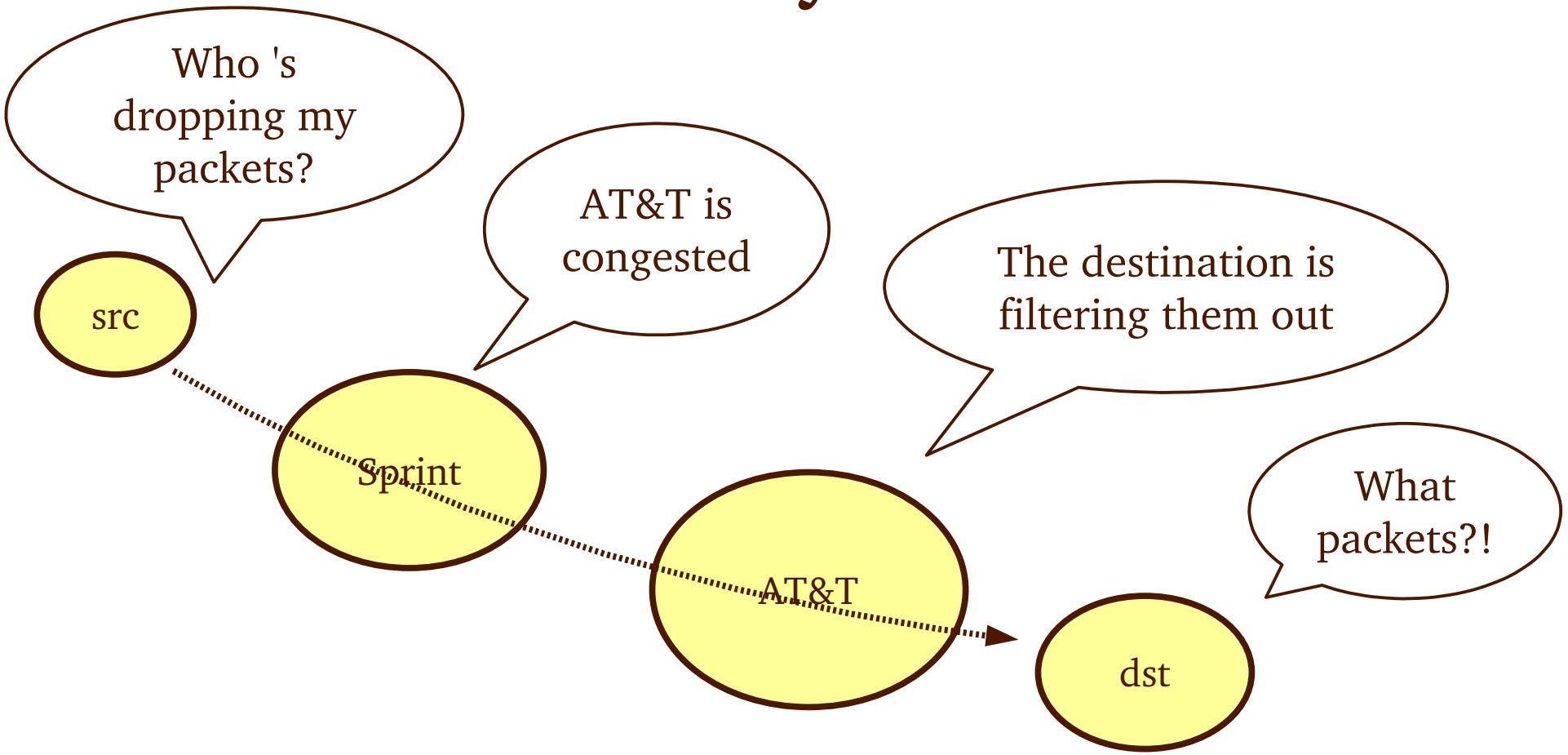
A basic AS interface

- *report(aggregate, attribute)*
- *forward(aggregate, nextHop)*
- *mark(aggregate, offset, attribute)*
- *drop(aggregate, lastHop)*

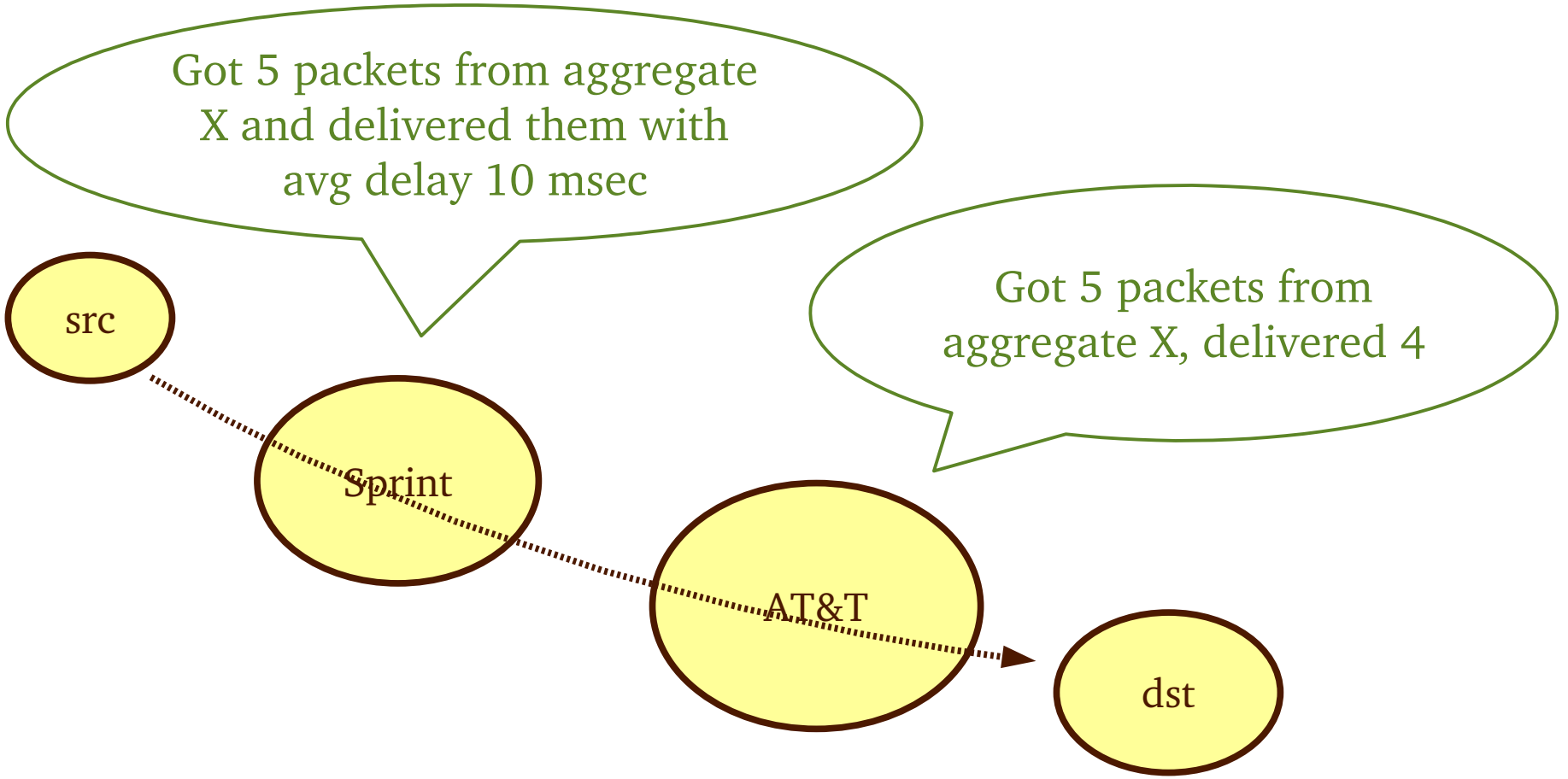
A basic AS interface

- *report(aggregate, attribute)*
- *forward(aggregate, nextHop)*
- *mark(aggregate, offset, attribute)*
- *drop(aggregate, lastHop)*

Accountability interface



Accountability interface



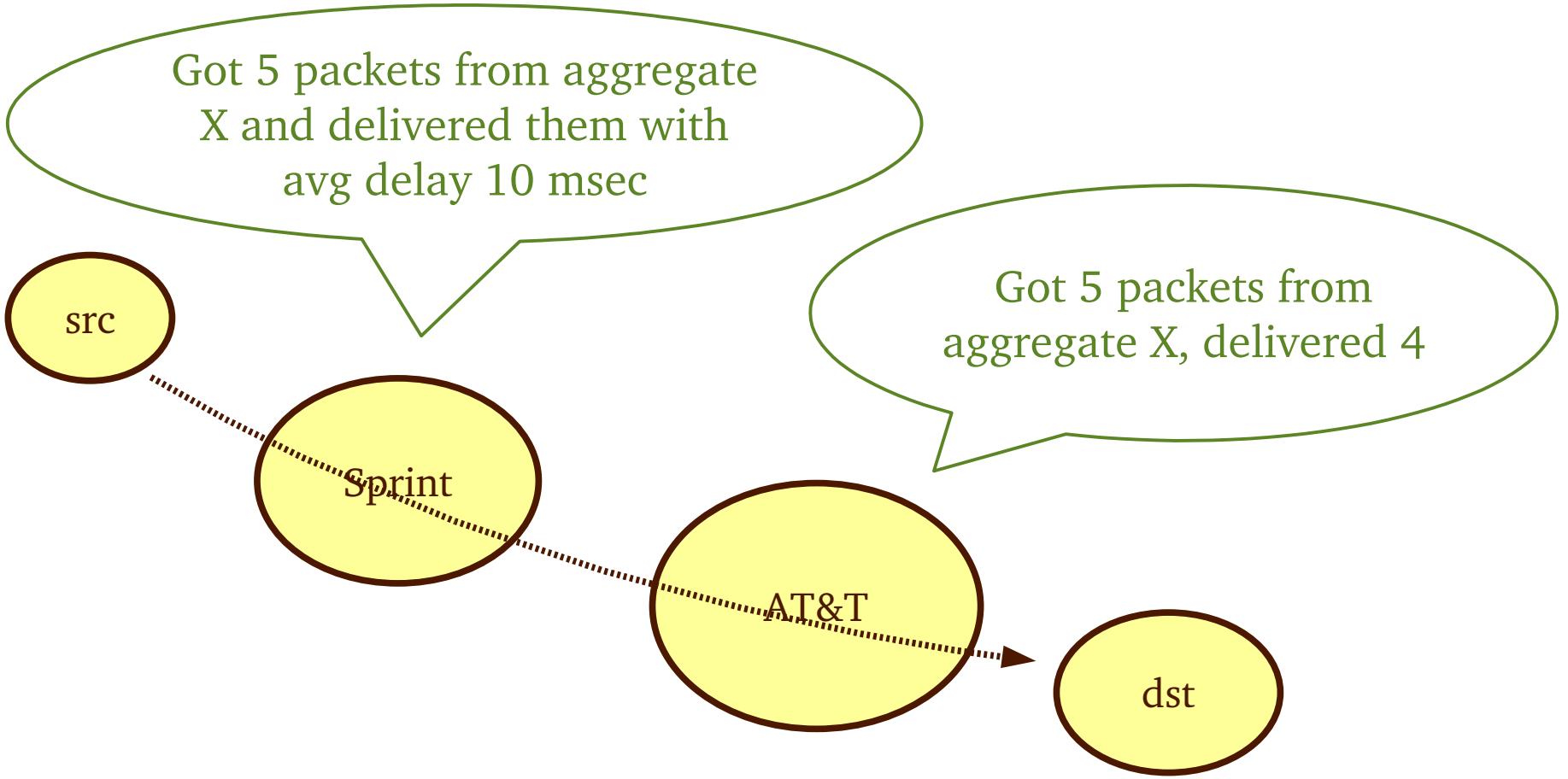
ASes report on their own performance

But the Internet is best effort

- Best effort = no a priori guarantees
- Accountability = after-the-fact info
- Helps make the best of best-effort service
  - edges can adapt to network conditions

Accountability != QoS

Accountability interface



ASes report on their own performance

Questions + challenges

- Traffic-aggregate definition
  - packets, TCP flows…
- Statistics
  - number of packets, time-related statistics, ??
- Fault (and lie) tolerance
  - otherwise as useful as current SLAs
- Implementing statistics collection
  - reasonable hardware requirements, scalability

Threat model

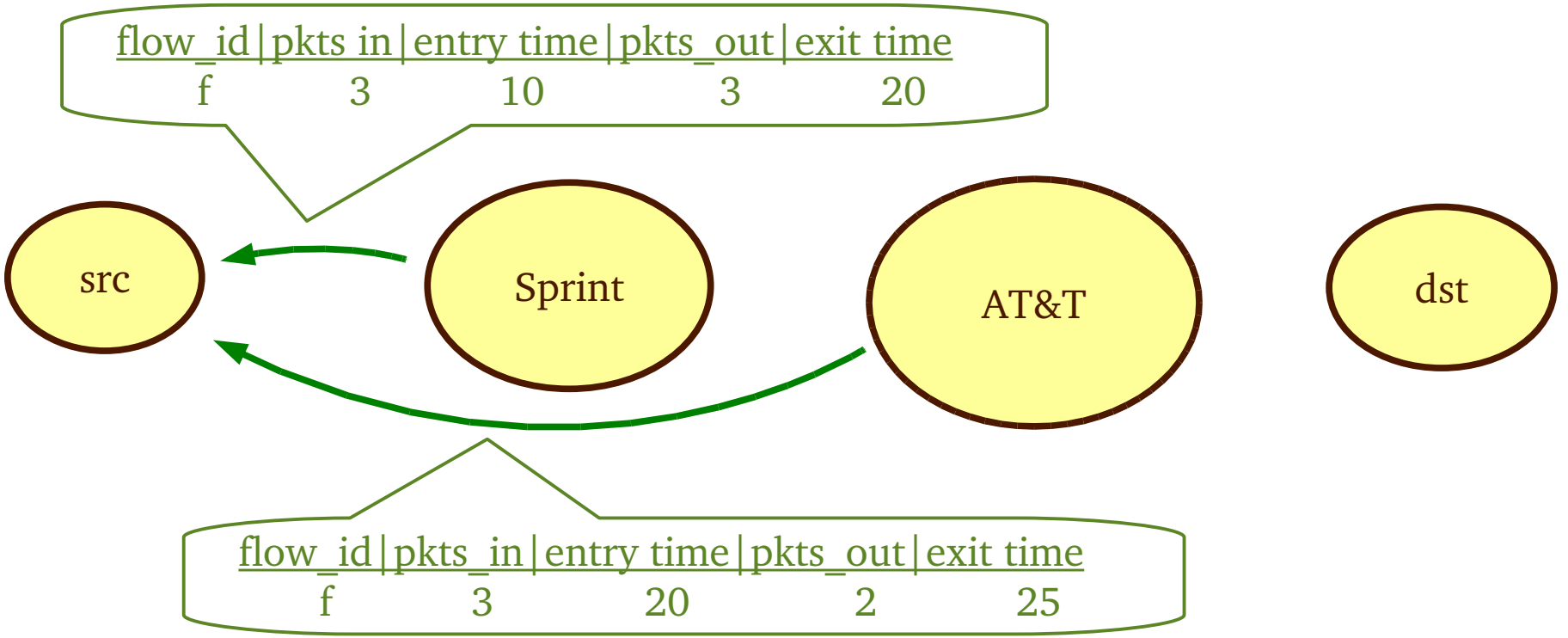
- Off-path lies
  - malicious nodes pretend they are transit Ases
  - report spurious feedback to confuse source
- On-path lies
  - transit ASes exaggerate their performance
- Feedback corruption
  - transit ASes modify other AS feedback

Example: accountability for TCP flows

- Statistics:
  - number of packets that entered, exited each AS
  - average entry and exit time
  - next and previous checkpoint

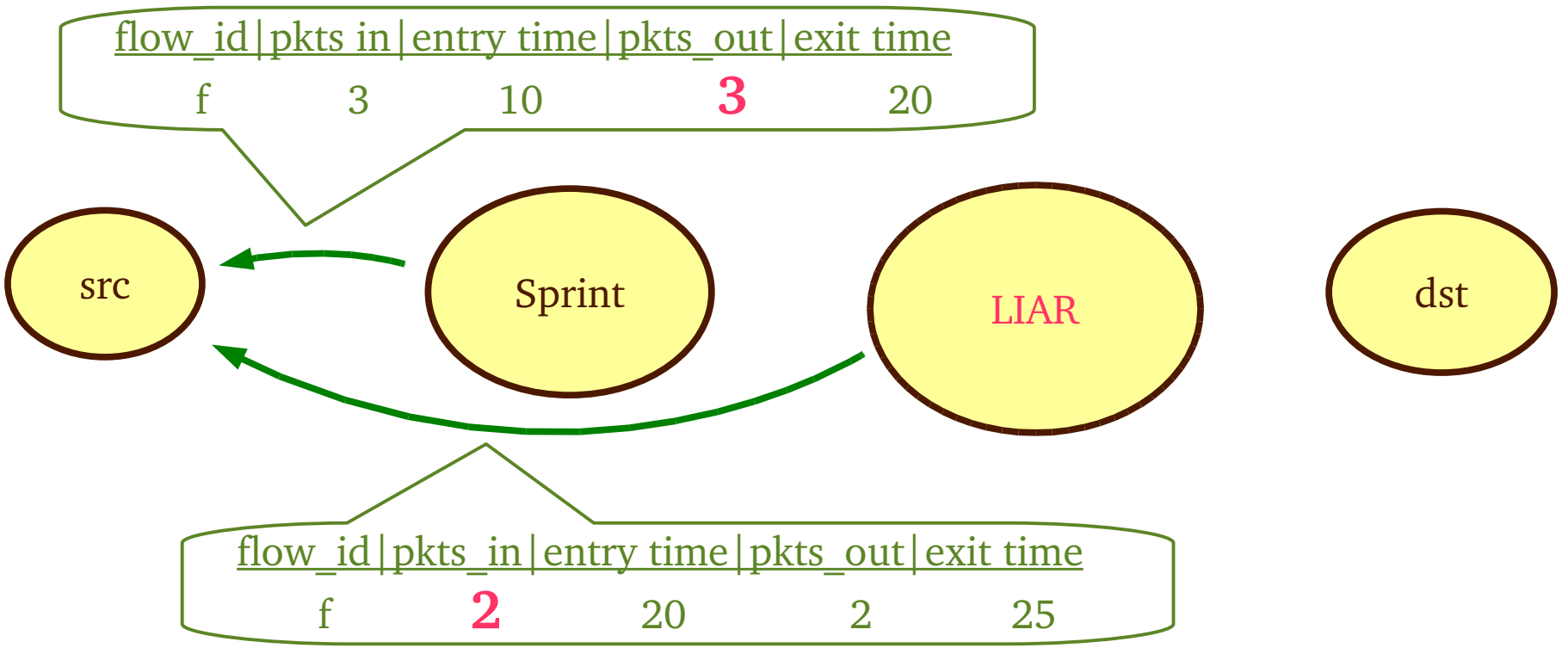
- Threat model
  - on-path lies only
  - no feedback corruption

Example: accountability for TCP flows

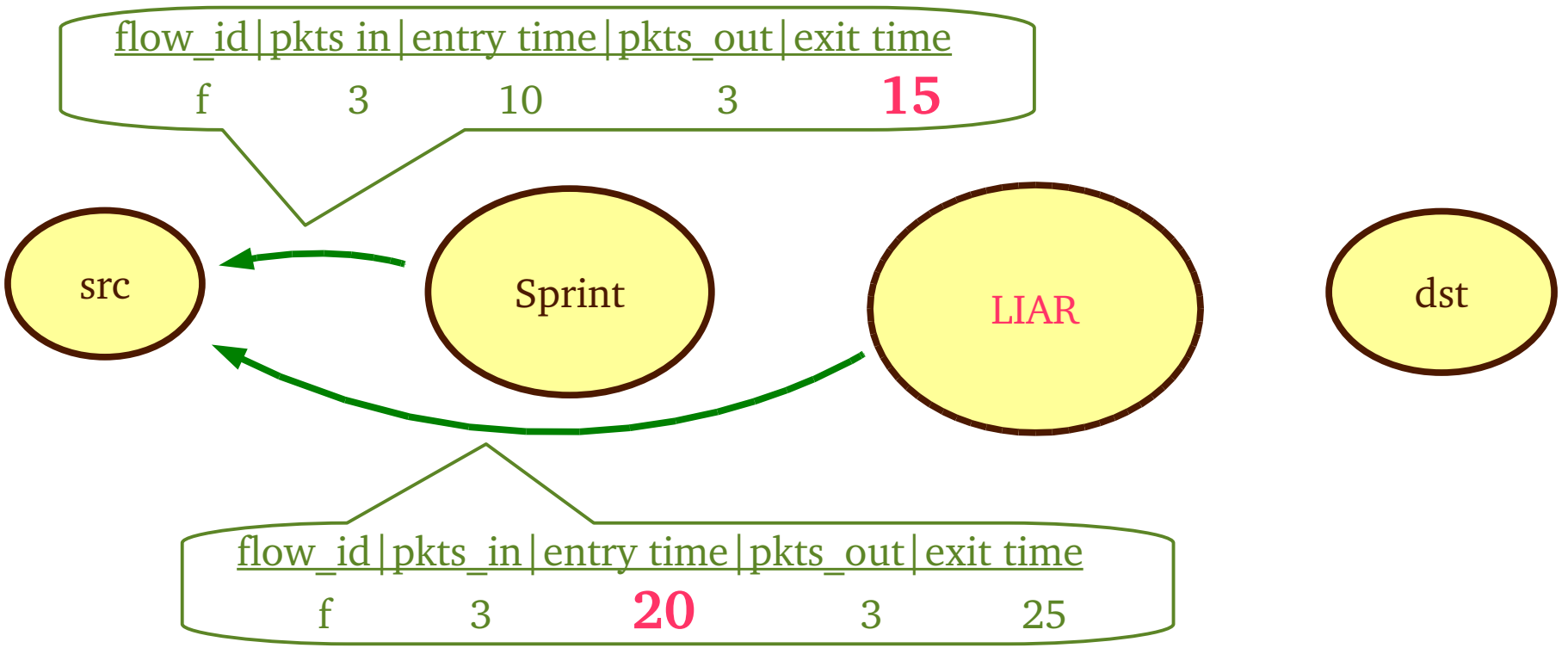


- Source learns loss + avg delay per AS

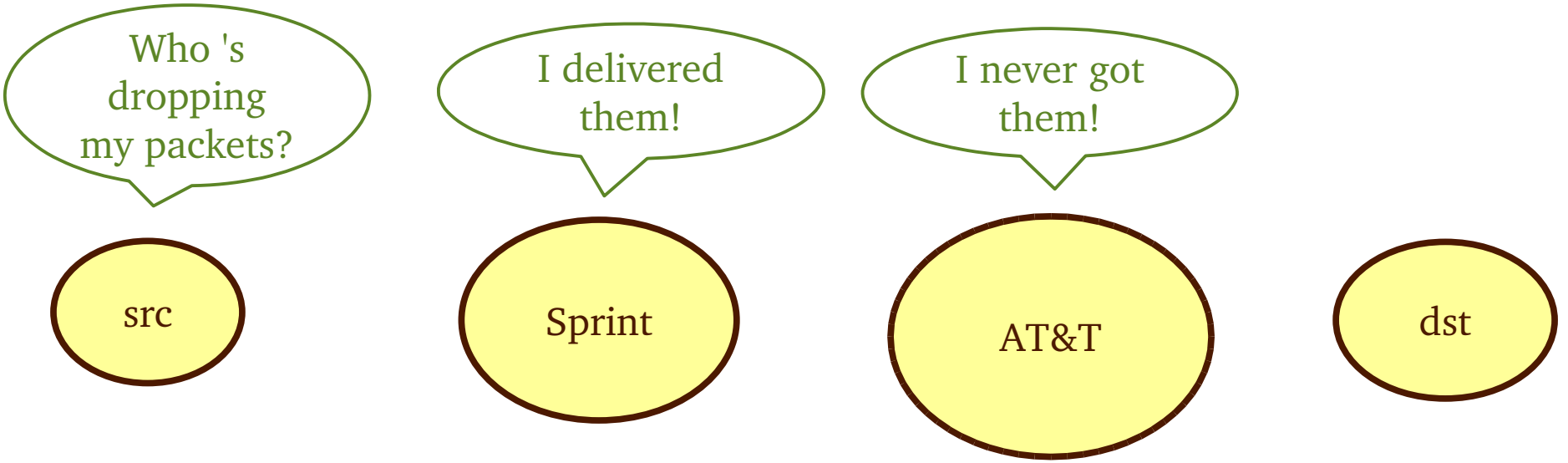
Example: accountability for TCP flows



Example: accountability for TCP flows



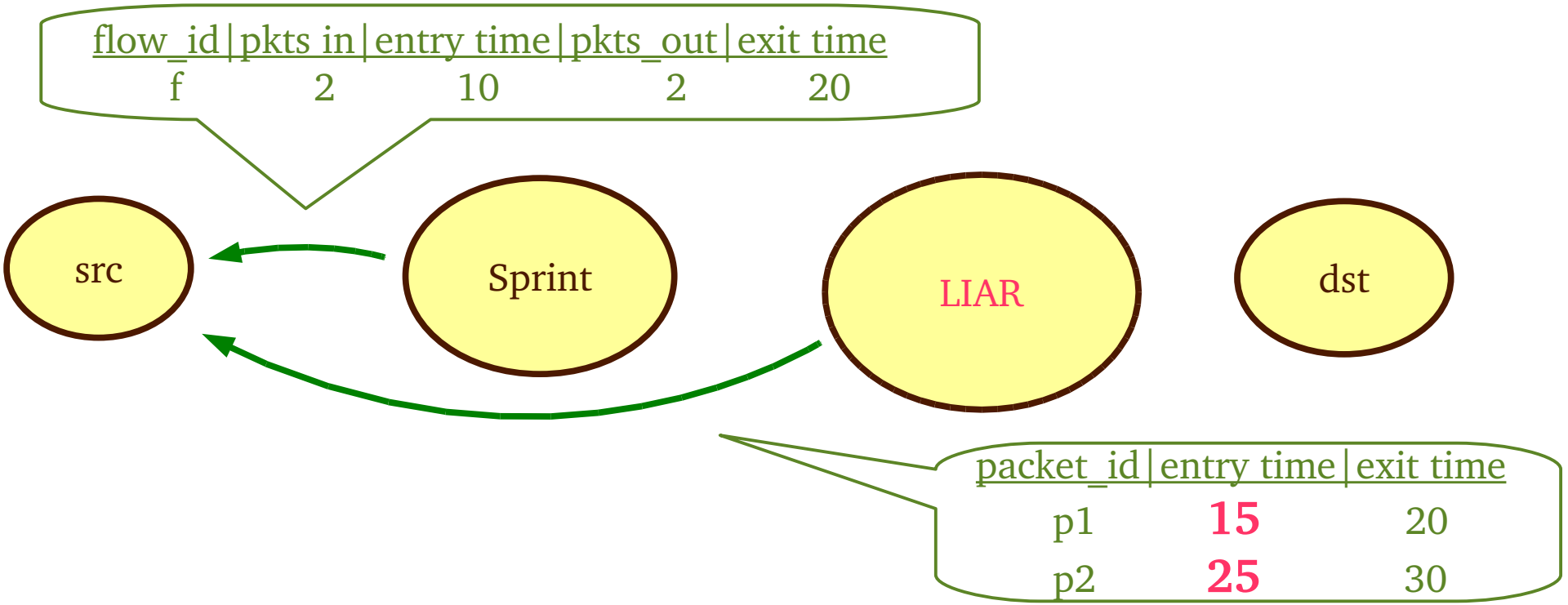
Lie tolerance



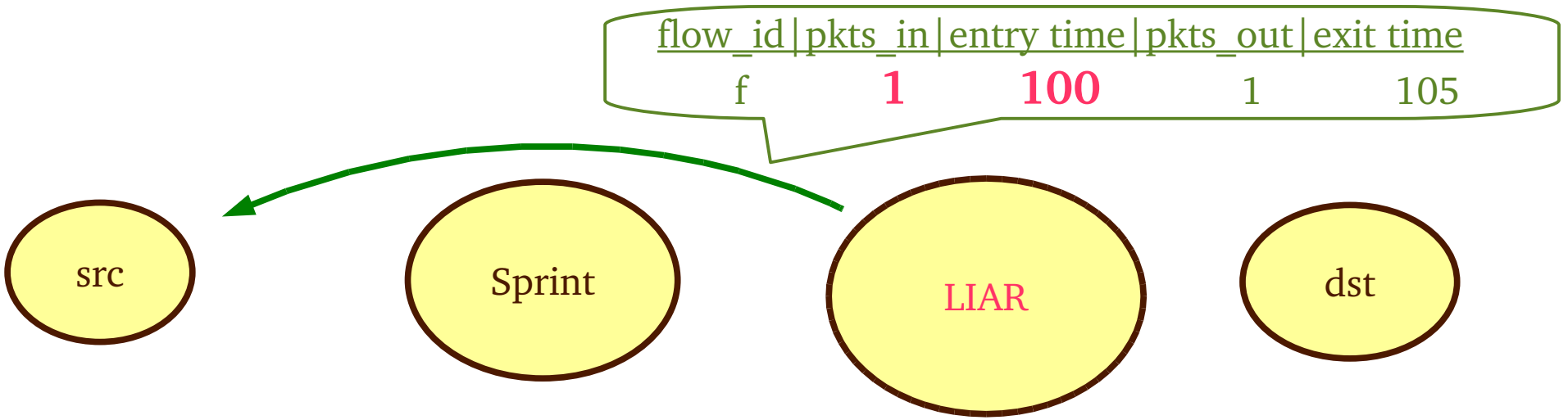
- Lie tracked down to inter-AS link
- Lying AS exposed to the peer it implicated

Lies manifest as feedback inconsistencies

Can we catch *all* lies?



Can we catch *all* lies?



Lie tolerance

- AS specifies performance bounds
- Peers can lie within bounds
- Tighter bounds = fewer lies
  - but more overhead

The more you tell, the safer you are

TCP-flow statistics collection

- Line-speed header inspection, flow-id lookup
  - NetFlow already does that
- Challenge: match entry-exit point statistics
  - loss affects delay statistics
  - multi-path flows

Conclusion

- In search of the right path-control balance
  - visibility + control for end systems
  - privacy and flexibility for ISPs
- Expose ASes as first-class Internet objects
- Define explicit AS interfaces
  - ISPs choose what visibility/control they export

Better both for end systems and ISPs