A Platform for Unobtrusive Measurements on PlanetLab

Rob Sherwood Neil Spring

University of Maryland
http://www.cs.umd.edu/projects/sidecar

Need for Measurements

Measurements benefit many applications:

- Performance optimization [OASIS]
- Overlay construction [i3]
- Network diagnosis [PlanetSeer],[CoMoN],[iPlane]

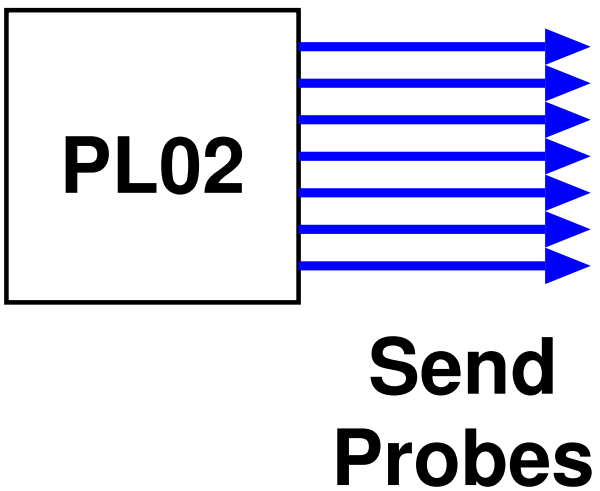
Grand Challenge:

- Collect a "Day in the Life"
  - CSTB *Looking Over the Fence* report

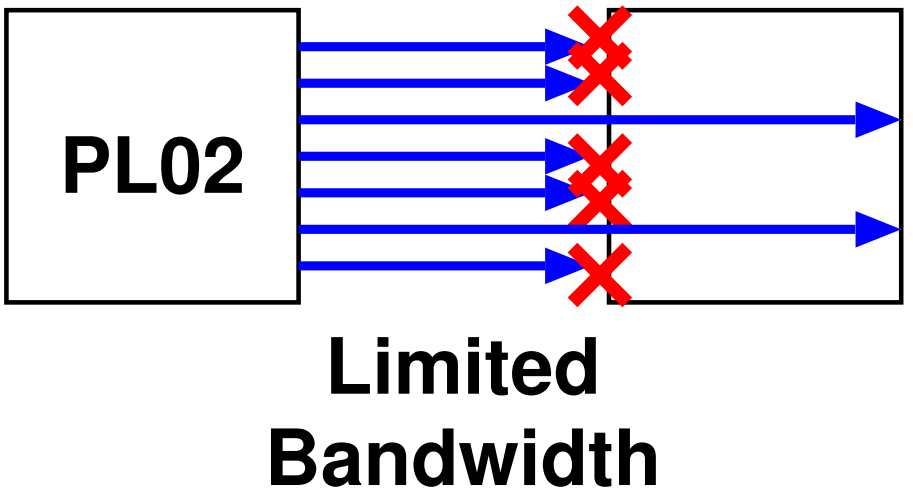
Why Not Just Measure Everything?

for src in PlanetLab **do**
  ssh to src
  for dst in All Addresses **do**
    Traceroute dst
    Ping dst
    Pathchar dst
    Other measurements ...

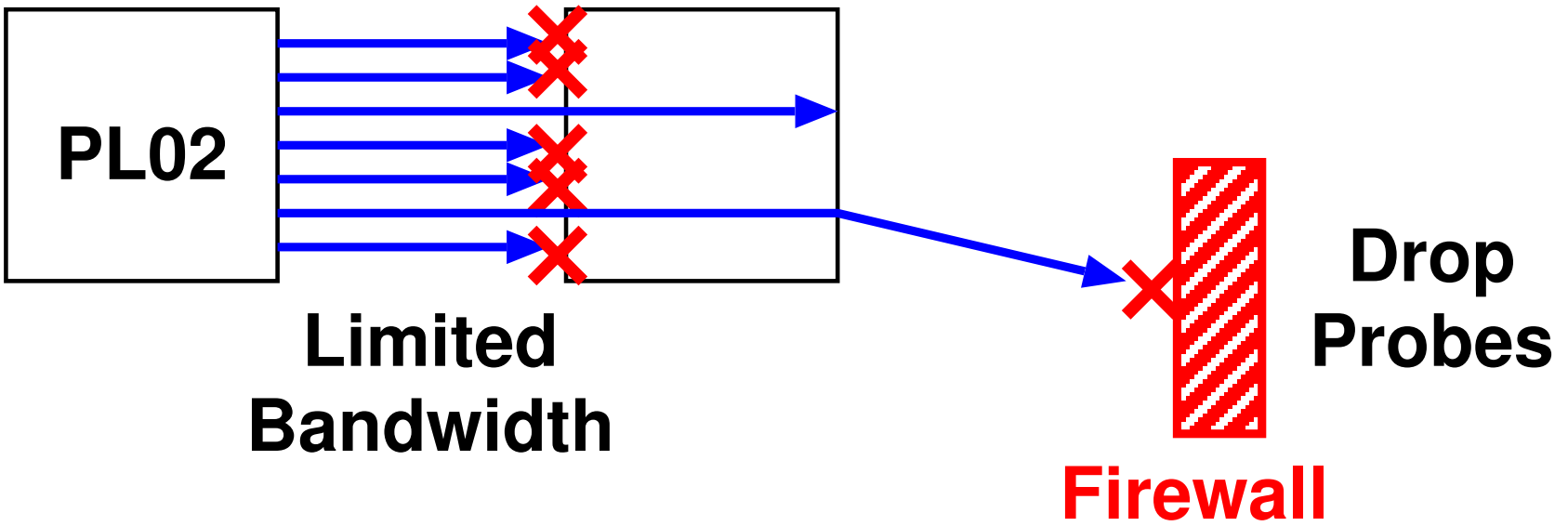
Why Not Just Measure Everything?



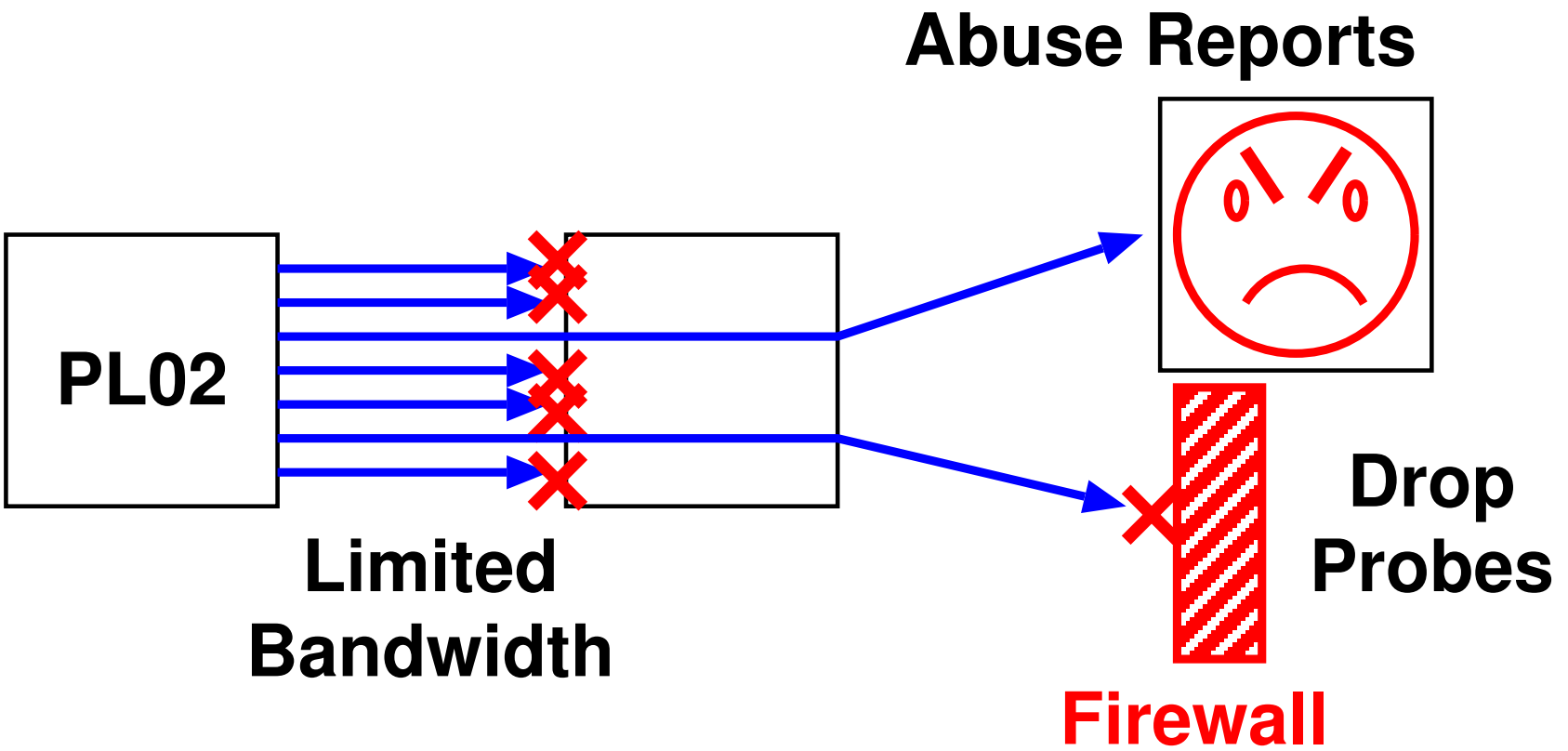
Why Not Just Measure Everything?



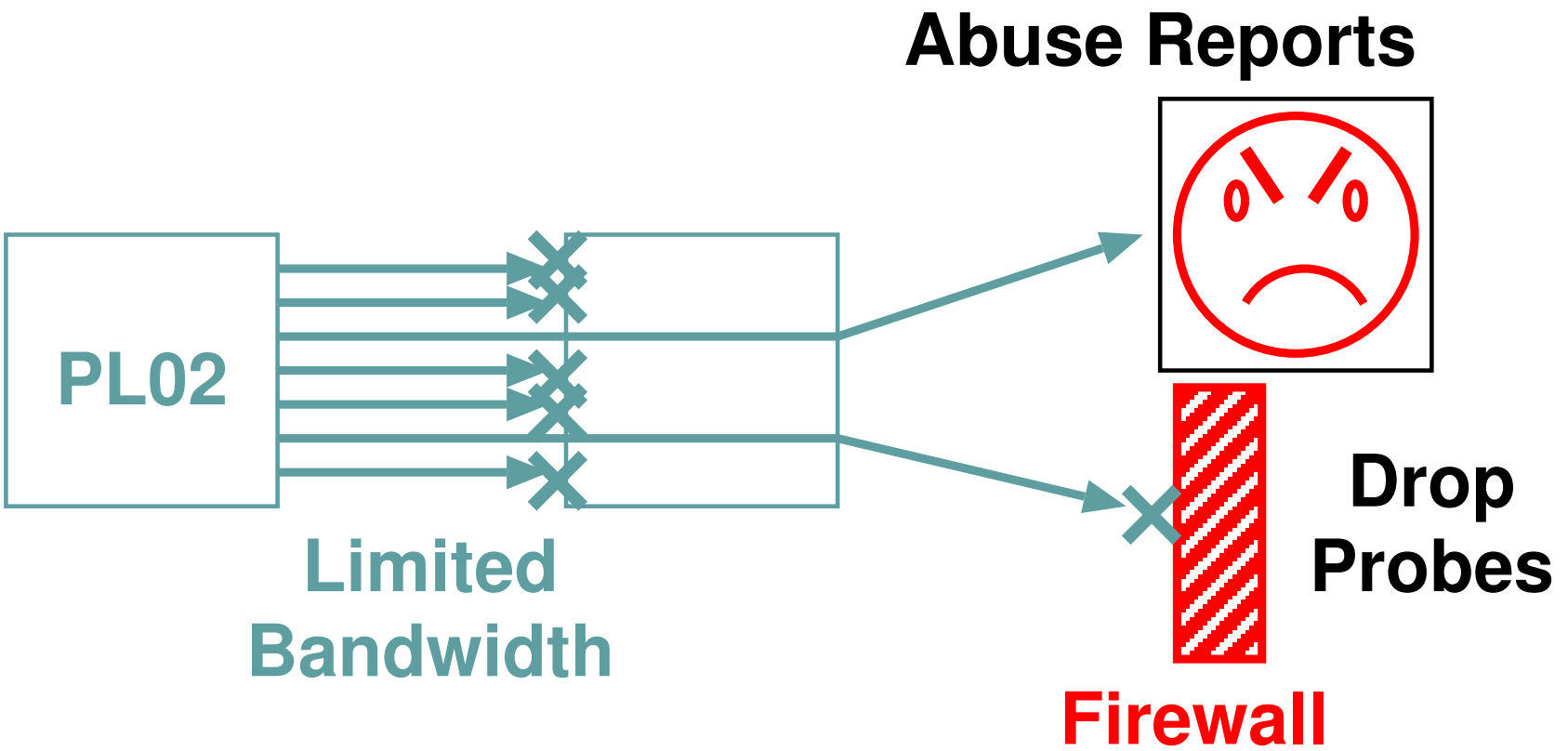
Why Not Just Measure Everything?



Why Not Just Measure Everything?



Why Not Just Measure Everything?



Abuse Reports

- Measurement traffic exceptional
- Exceptional == suspicious
- Reports of network abuse are handled with care
  - Thank you Mark Huang and PL Staff!
- Curtails experiment scope

Measurement Platform: Sidecar

Inject probes into normal traffic:

- Probes are retransmissions
- Avoids abuse report
- Allows firewall/NAT traversal

General measurement platform:

- Latency — Sideping
- Bottleneck location — Artrat
- Topology — Passenger [IMC06]

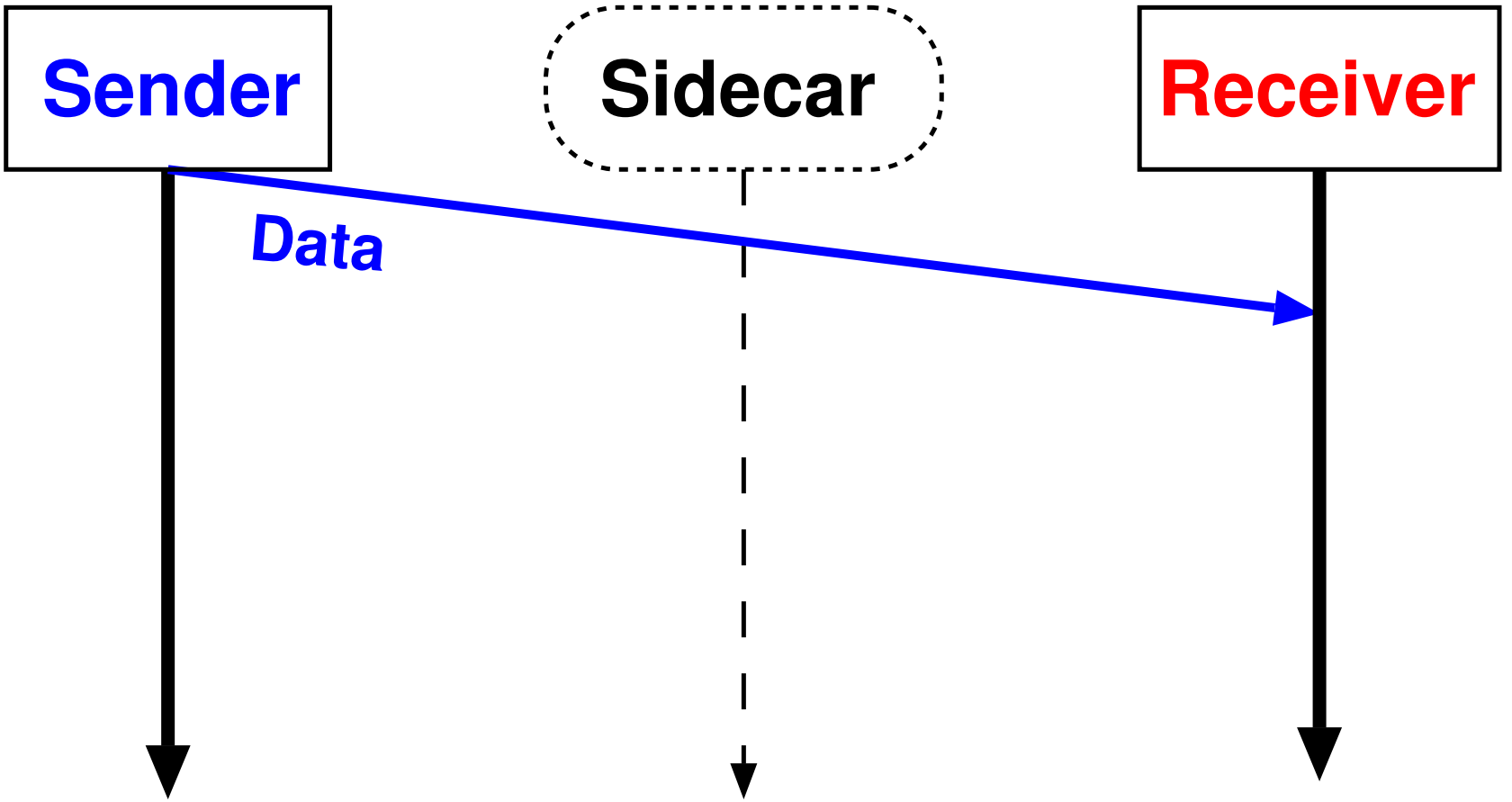
Talk Overview

- How Sidecar works
- Learning from Sidecar
- What Sidecar can do

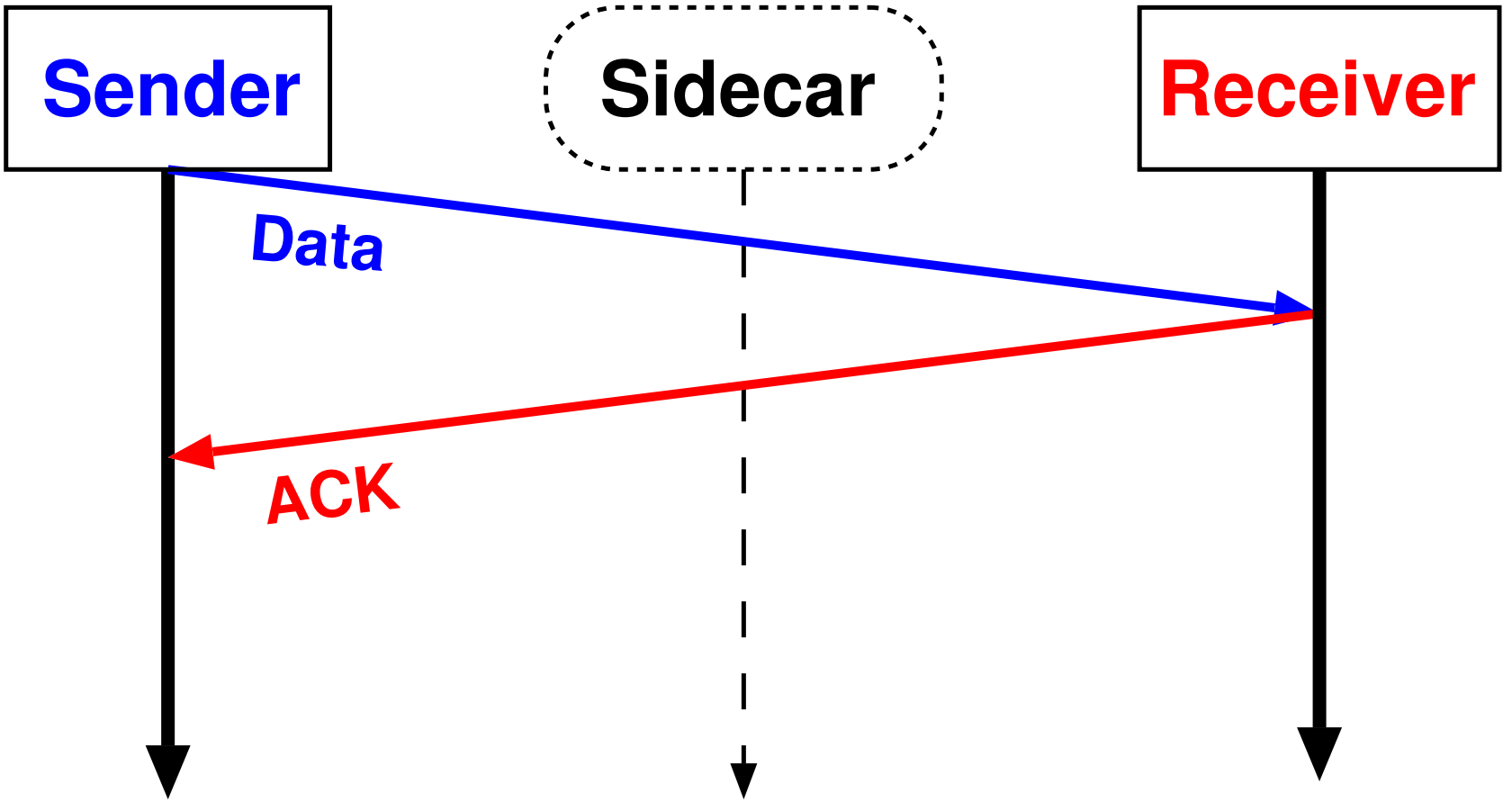
Talk Overview

- How Sidecar works
- Learning from Sidecar
- What Sidecar can do

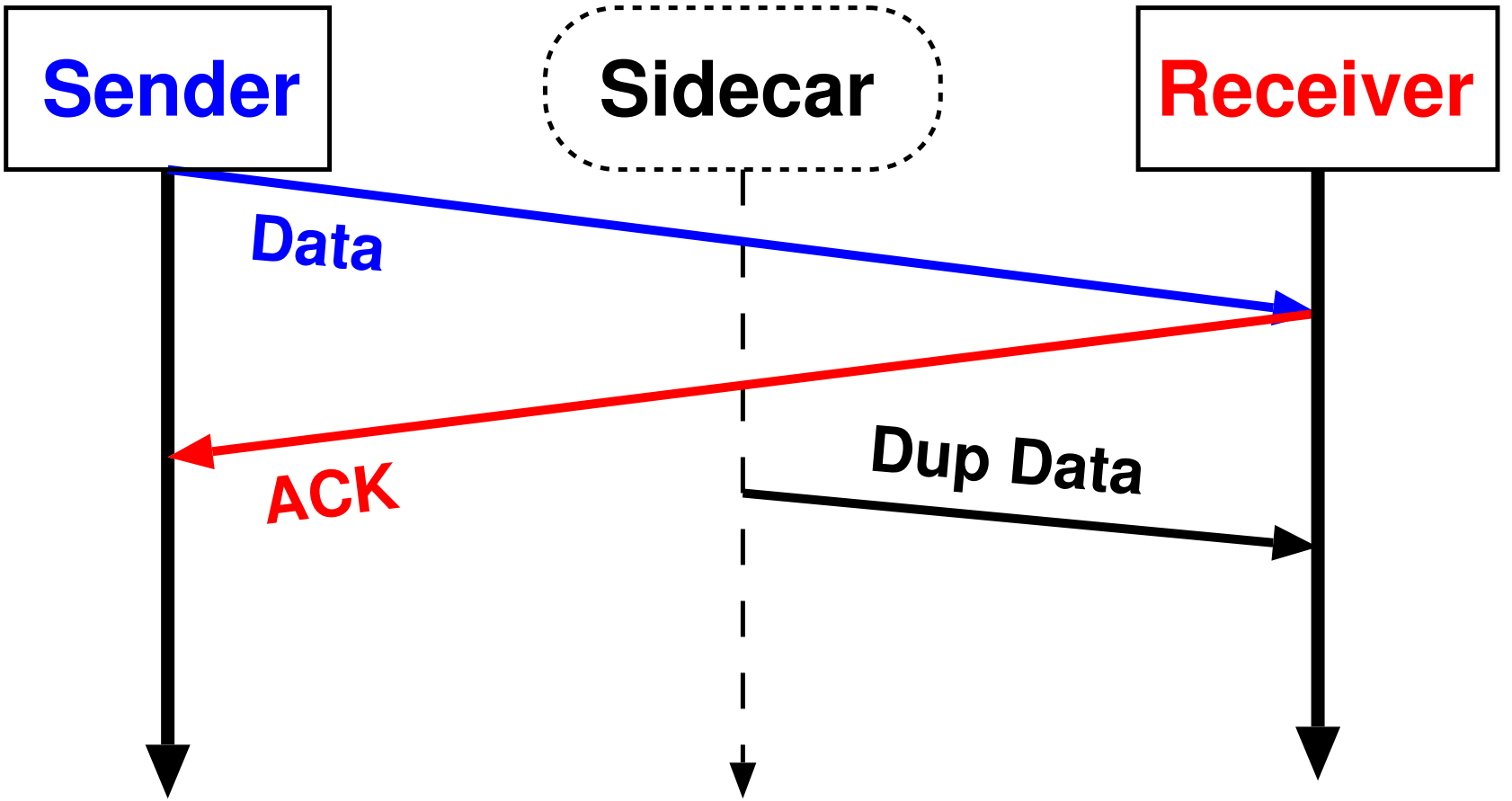
Sidecar Probes



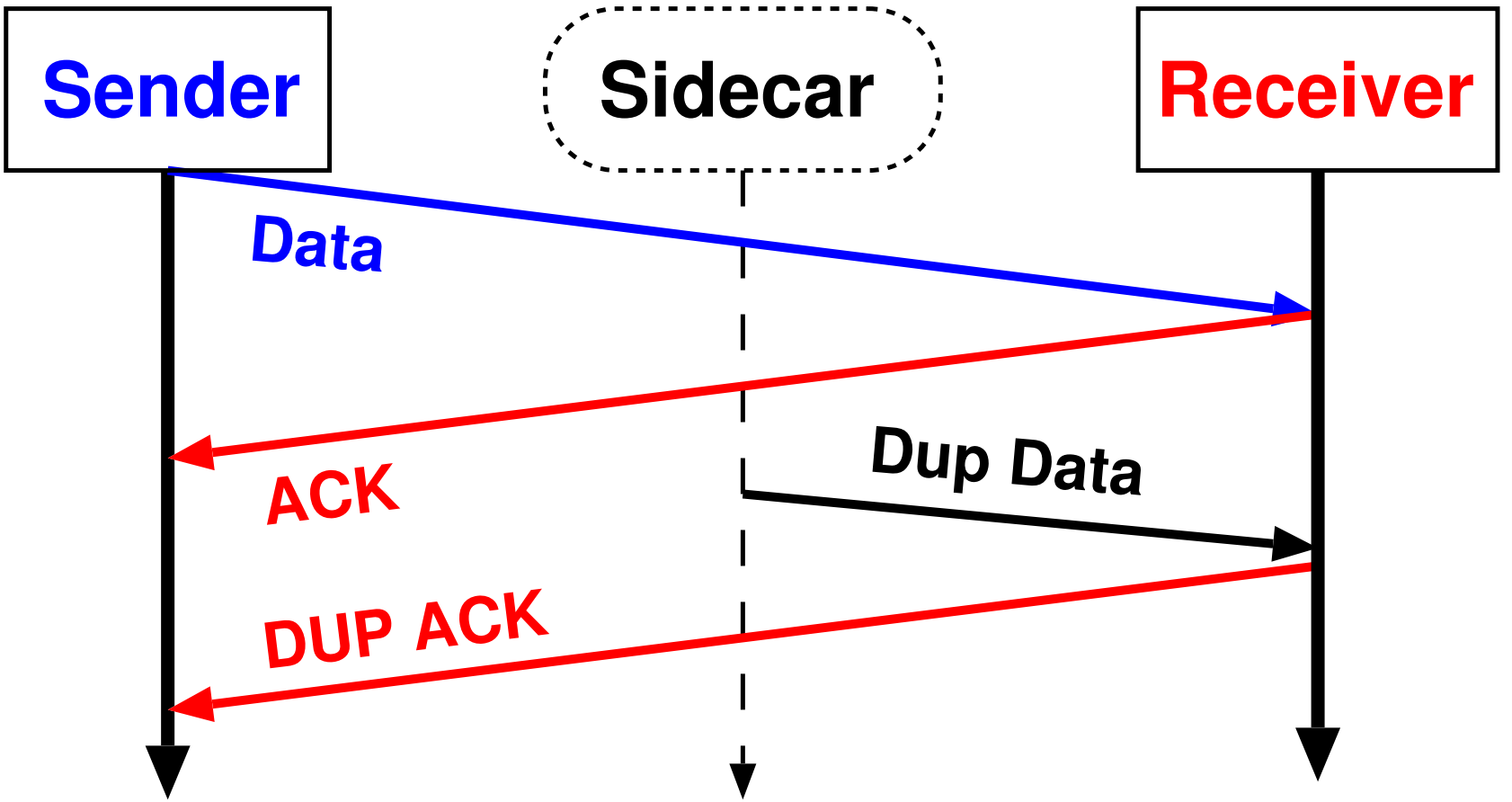
Sidecar Probes



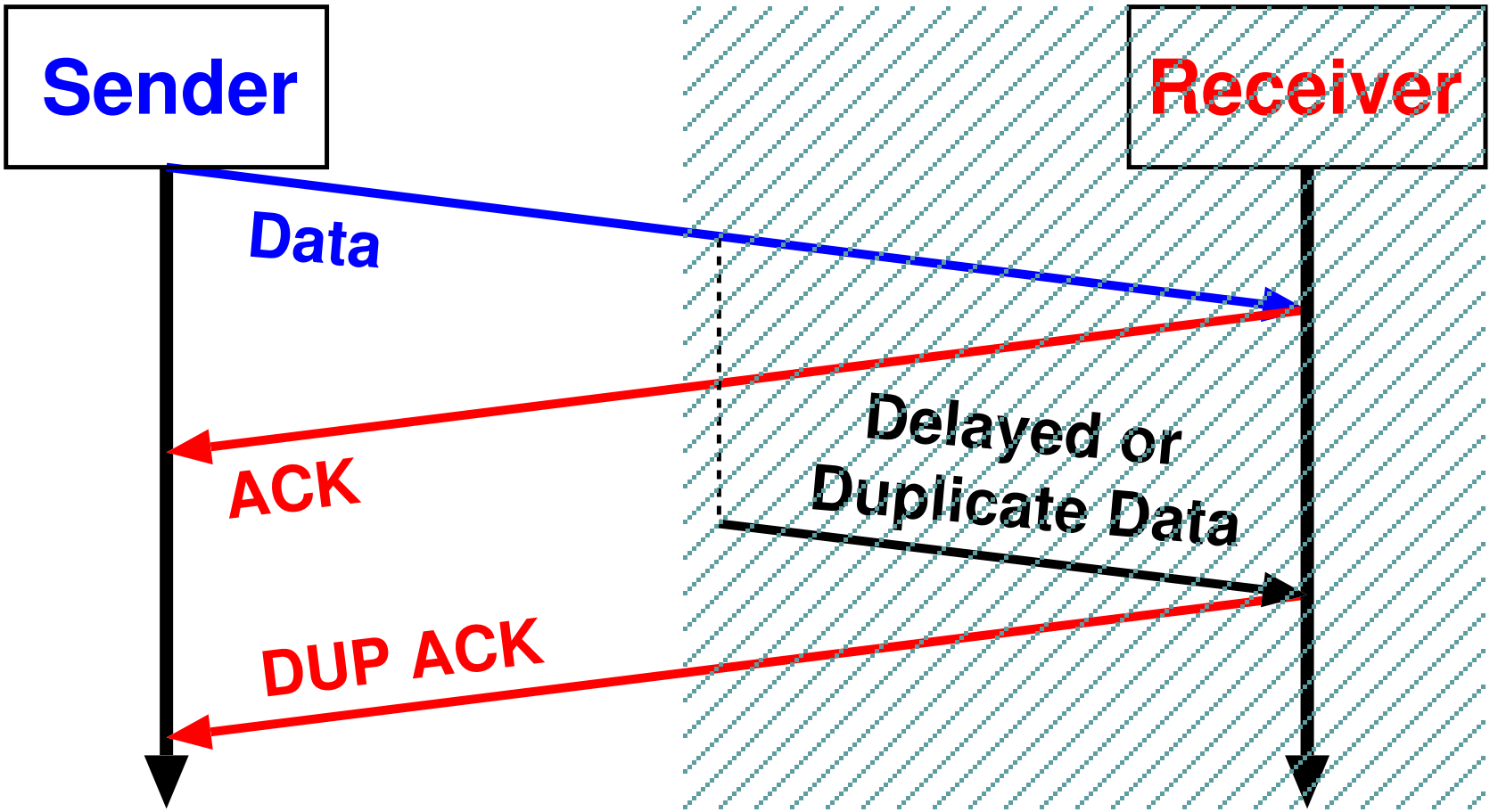
Sidecar Probes



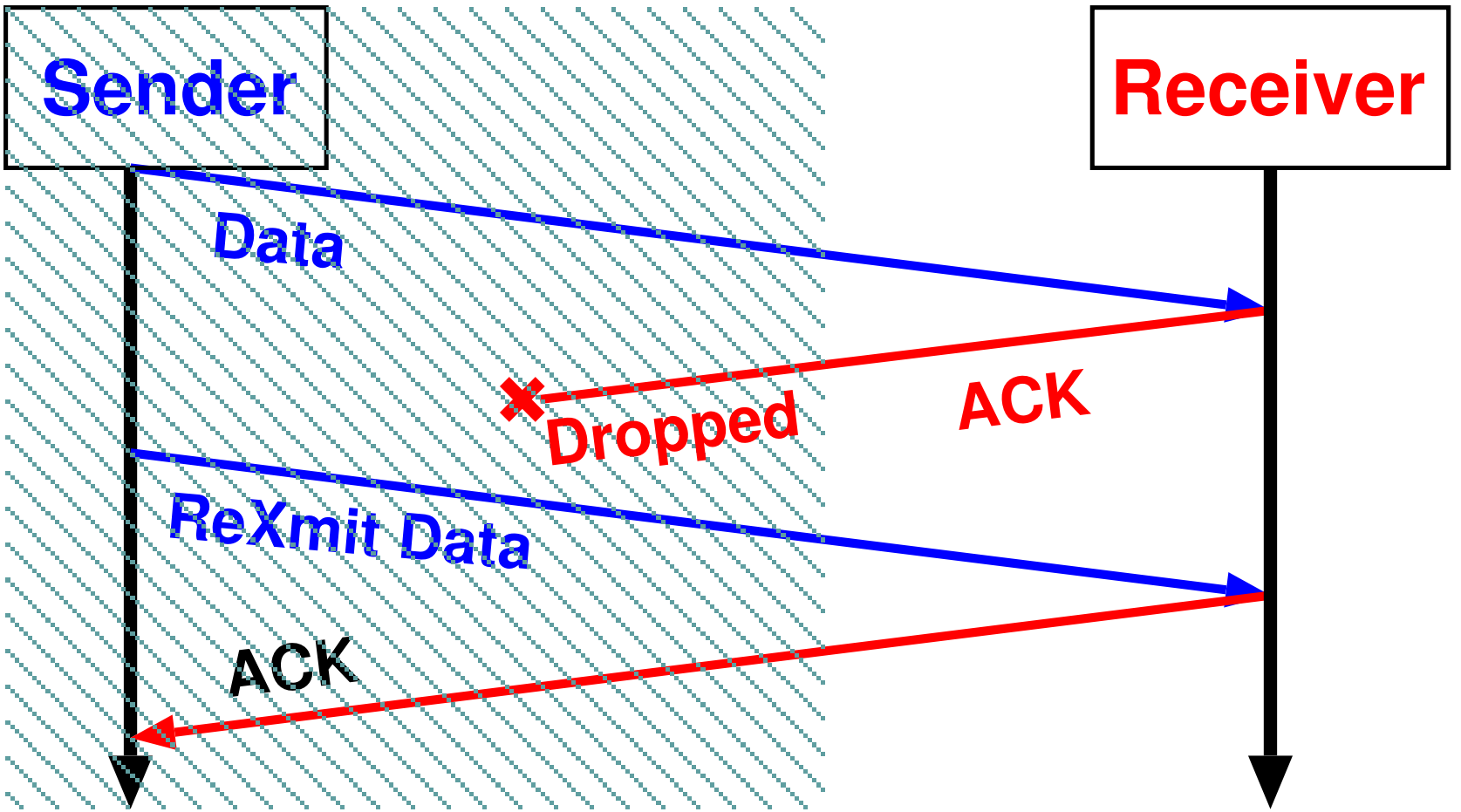
Sidecar Probes



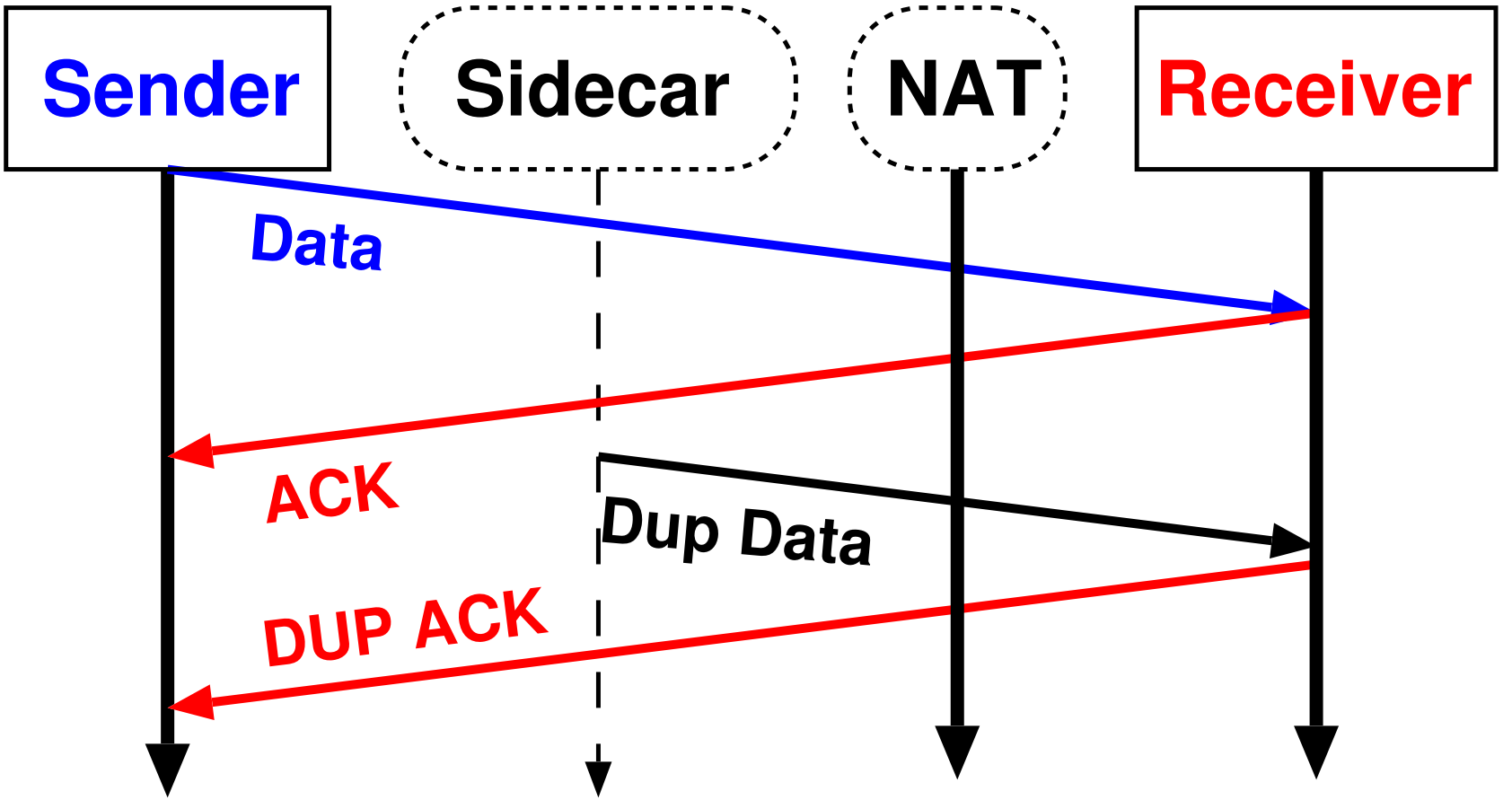
Sidecar Probes: Sender's View



Sidecar Probes: Receiver's View



Firewall and Nat Traversal

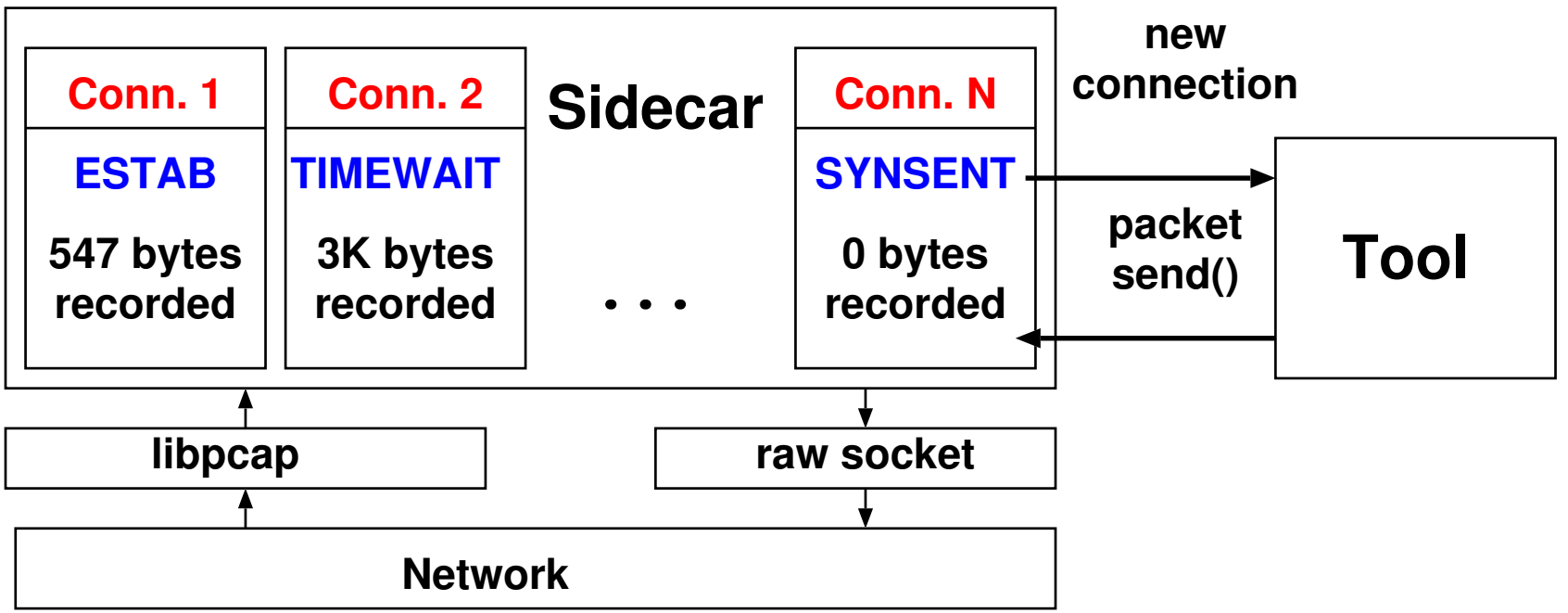


Sidecar Probes

- Probes are retransmissions
  - Requires no end-point support
  - Send probes when connection is idle
- Modify probes for specific measurement
  - Reduce TTL
  - Send probes in train
  - Add IP options
- Can send probes after connection closes
  - Receiver in TIME_WAIT state

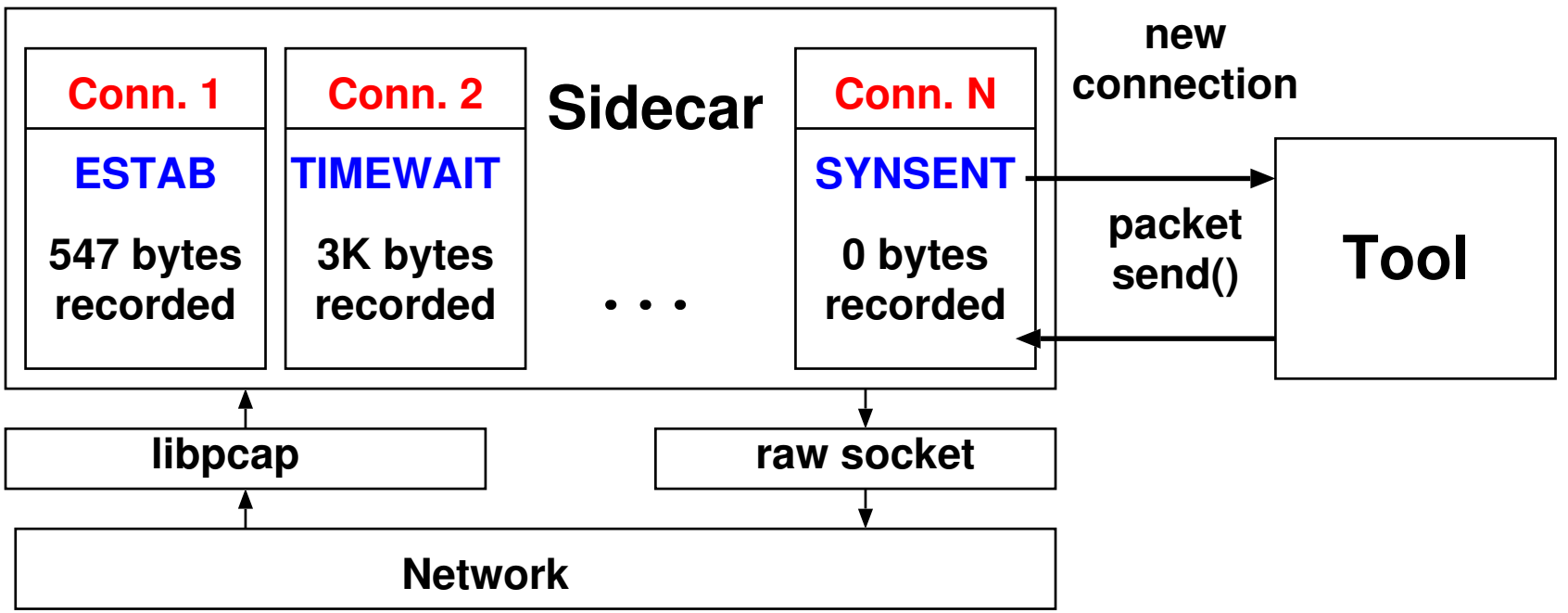
Sidecar API

- libpcap filter: "tcp port 80"



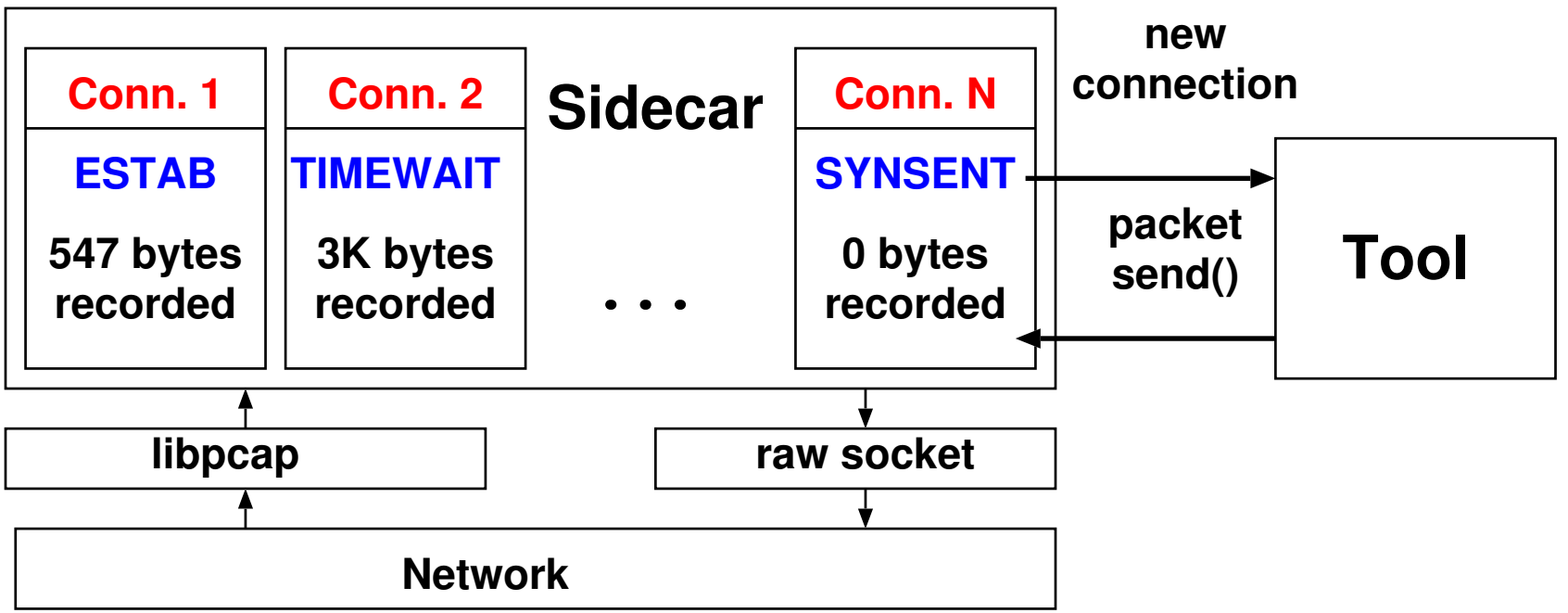
Sidecar API

- Tracks connection state and data



Sidecar API

- Applications register callbacks
  - Events: new connection, probe returned, idle, close, timeout



Example Code for Sidecar Tool

1. sc_register_connection(connectCB);
2. sc_init("tcp port 80");
3. connectCB(conn *c){
   sc_register_idle(c,idleCB);
   sc_register_in(c,inCB);}
4. idleCB(conn *c){send_probe(c);
   sc_register_timeout(c,timeoutCB) }
5. inCB(conn *c, packet *p){
   print "Response@"+calcRTT(p)}

Talk Overview

- How Sidecar works
- Learning from Sidecar
- What Sidecar can do

What We Learned: Overview

- Sidecar generates no abuse reports
- Generate traffic carefully
- Clocks are not accurate
- Causally related packets are reordered
- Firewalls unset DF bit
- IO systems calls lag

No Abuse Reports

- Probed all traffic to CoDeeN clients
- Experiment ran for 1 week
- Sidecar traceroute to each client
- 13.4M hosts probed
- No abuse reports generated

Generate Traffic Carefully 2/6

- Instrument custom web crawler with Sidecar probes
  - 168K web servers $\times$ PL Nodes
  - Caused ten abuse reports
  - . . . but from web crawler, not Sidecar
- Correct User-Agent, Virtual hosts
- Crawlers synchronized $\longrightarrow$ traffic spikes

N1: AAABBCD
N2: AAABBCD
N3: AAABBCD
N4: AAABBCD

Generate Traffic Carefully 2/6

- Instrument custom web crawler with Sidecar probes
  - 168K web servers $\times$ PL Nodes
  - Caused ten abuse reports
  - . . . but from web crawler, not Sidecar
- Correct User-Agent, Virtual hosts
- Crawlers synchronized $\longrightarrow$ traffic spikes

N1: ABCD
N2: ABCD
N3: ABCD
N4: ABCD

Generate Traffic Carefully 2/6

- Instrument custom web crawler with Sidecar probes
  - 168K web servers $\times$ PL Nodes
  - Caused ten abuse reports
  - . . . but from web crawler, not Sidecar
- Correct User-Agent, Virtual hosts
- Crawlers synchronized $\longrightarrow$ traffic spikes

N1: ABCD
N2: BACD
N3: ADBC
N4: DCAB

Generate Traffic Carefully 2/6

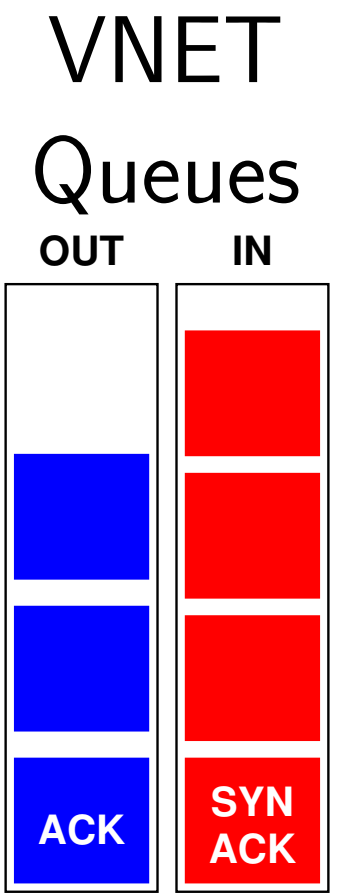
- Instrument custom web crawler with Sidecar probes
  - 168K web servers $\times$ PL Nodes
  - Caused ten abuse reports
  - . . . but from web crawler, not Sidecar
- Correct User-Agent, Virtual hosts
- Crawlers synchronized $\longrightarrow$ traffic spikes
- Application logs

Clock Irregularities

- Clocks would change rate, jump backwards
  - Similar to [Myths05]
- PlanetLab nodes have diverse hardware
- Future work: add RDTSC sanity check
  - Signal tool that clock jumped

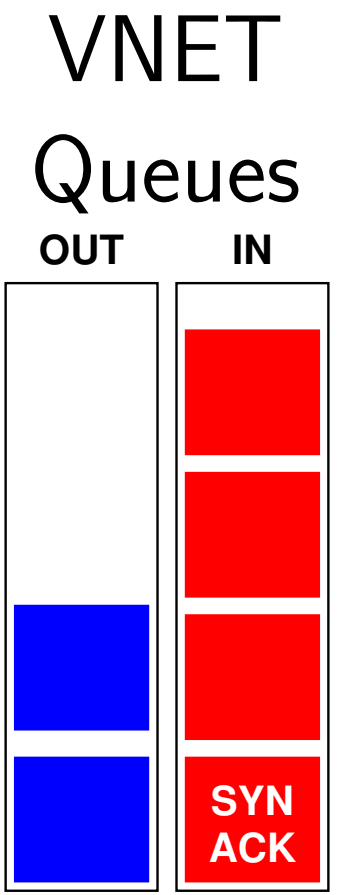
Causal Packets Reordered 4/6

- VNET + libpcap interaction caused causally related packets to arrive out-of-order
  - e.g., ACK before SYN|ACK
  - Replicated by other researchers
- Time stamps correct
- Made state transition logic more robust
- Future: patch VNET code



Causal Packets Reordered 4/6

- VNET + libpcap interaction caused causally related packets to arrive out-of-order
  - e.g., ACK before SYN|ACK
  - Replicated by other researchers
- Time stamps correct
- Made state transition logic more robust
- Future: patch VNET code



Firewalls Unset DF Bit

- Recursive packet train measurements overflowed libpcap [Sigcomm04]
- Tried to use DF bit to ignore payload
  - Some firewalls unset DF on incoming packets
- Implications for MTU discovery measurements

Lag in IO Systems Calls 6/6

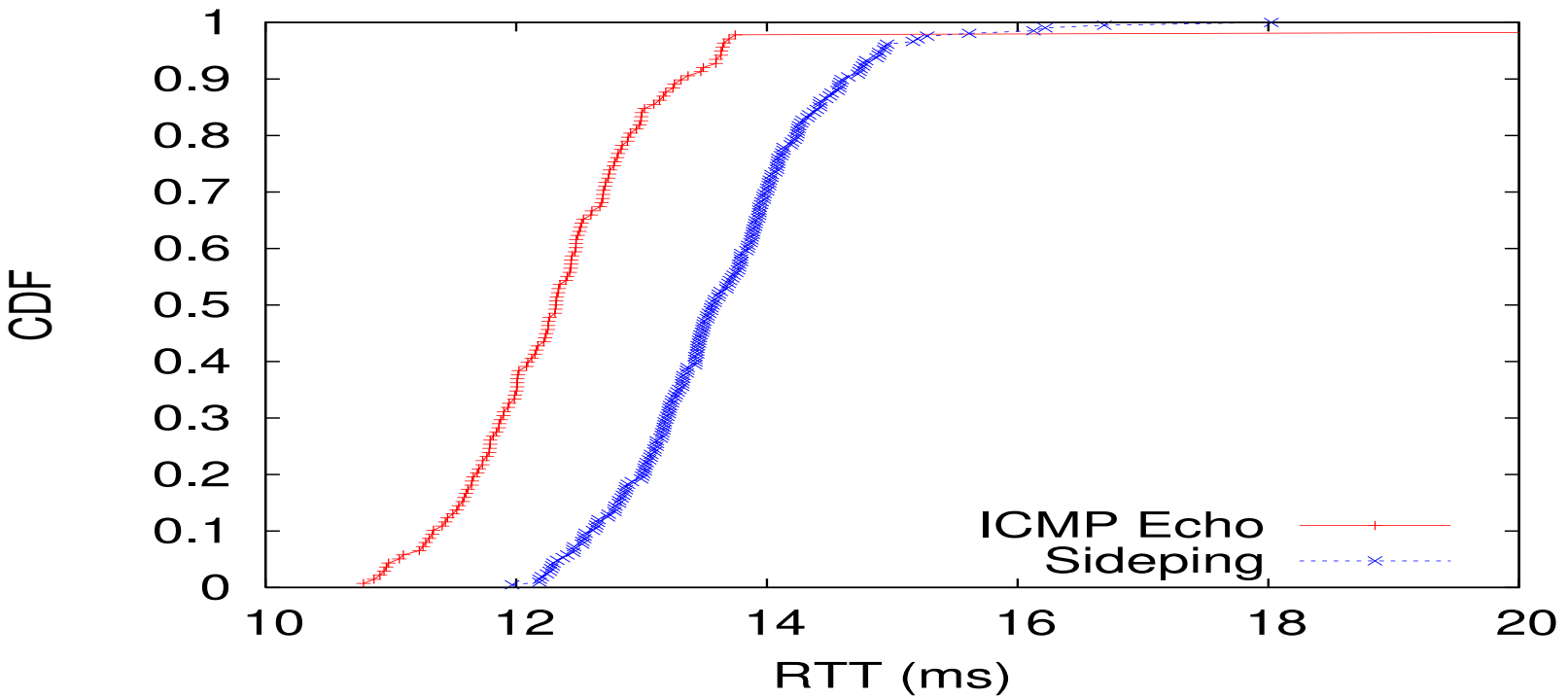
- Many concurrent writes
- *strace -T* showed that *open()* and *write()* calls took 1-3 seconds to return
- Intermittent; could not diagnose

Talk Overview

- How Sidecar works
- Learning from Sidecar
- What Sidecar can do

Sideping

- 24/482 PL nodes drop ICMP Echo
  - All nodes allow Sidecar probes
- Sideping traverses firewalls and NATs
  - Exposes higher latency extra hop

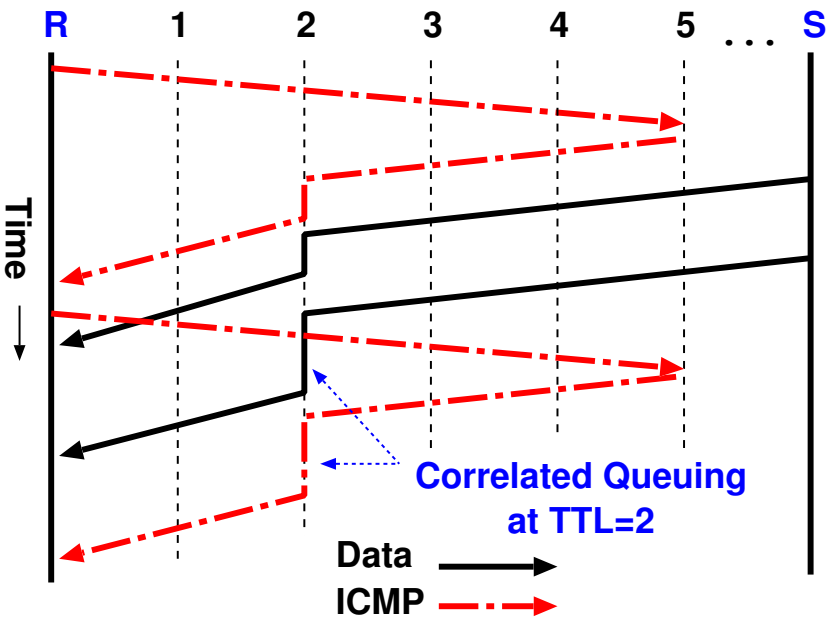


Artrat

- Artrat: Active Receiver-side TCP Rate Analysis Tool
- Locate bandwidth bottleneck from receiver
- Sanity check PlanetLab experiment conditions

Artrat: Technique

- Use IP timestamp option with ICMP echo to measure queuing delay
- Router with highest correlated delay is bottleneck



Conclusion

- Sidecar is a technique and API/package for unobtrusive probing
- Probes caused no* abuse reports
- Symbiotic relationship between service and measurement projects
  - Measurements $\Leftrightarrow$ application traffic
- Download API and tools from

  `http://www.cs.umd.edu/projects/sidecar`

Bibliography 1/2

1. [OASIS]
   http://oasis.coralcdn.org/
2. [i3] http://i3.cs.berkeley.edu/
3. [PlanetSeer]
4. [CoMoN]
   http://comon.cs.princeton.edu/
5. [iPlane]
   http://iplane.cs.washington.edu/

Bibliography 2/2

1. [IMC06] "Touring the Internet in a TCP Sidecar"
   Rob Sherwood, Neil Spring.
2. [Myths05] "Using PlanetLab for Network Research: Myths, Realities, and Best Practices." Neil Spring, Larry Peterson, Andy Bavier, and Vivek Pai
3. [Sigcomm04] "Locating Internet Bottlenecks: Algorithms, Measurements and Implications". Ningning Hu, Li Erran Li, Zhuoqing Morley Mao, Peter Steenkiste, Jia Wang