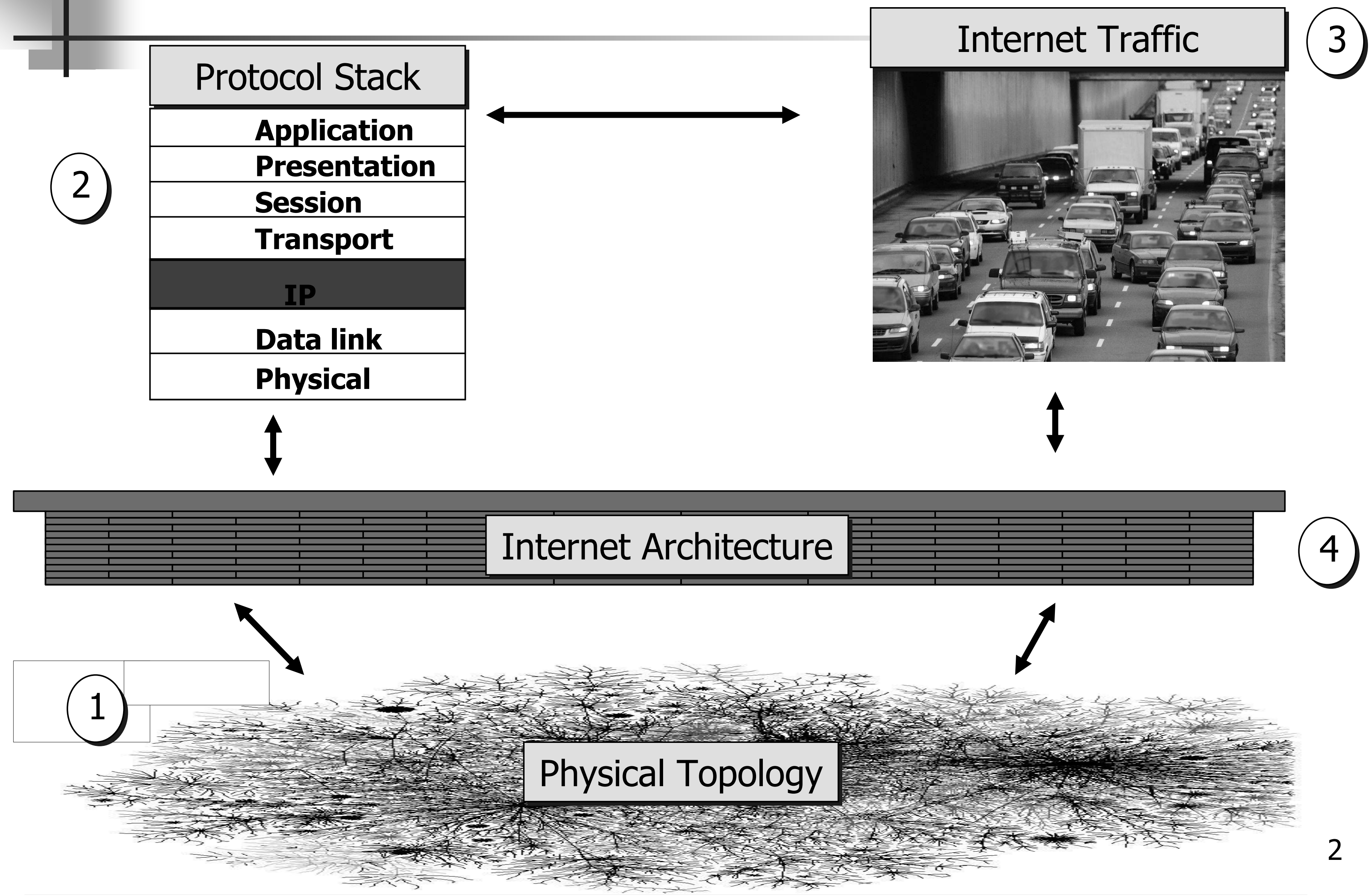
Measuring the Evolution of Transport Protocols in the Internet

Alberto Medina
Mark Allman
Sally Floyd

The Internet

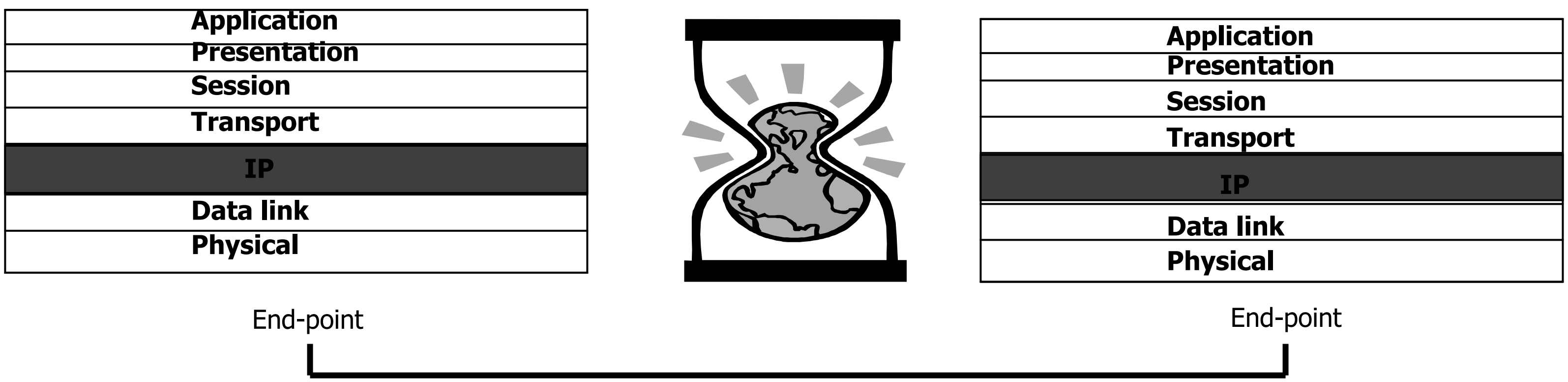


TCP Evolution

- Congestion Control behavior
  - No proper congestion control => congestion collapse
- Deployment, correctness of transport mechanisms
  - Assess correctness and behavior of newer additions
- Dynamics: Theory vs. Practice
  - Differences between <u>protocol specs</u> (theory) and their <u>implementation and its environment</u> (practice)
- Network Modeling
  - Aim at improving accuracy of network models

Network Evolution

- Hourglass Model



- End-to-end principle
  - “Some functions can only be implemented completely and correctly end-to-end, with the help of the end points”
- Study effect of middleboxes on these principles
  - firewalls, load balancers, NATs, …

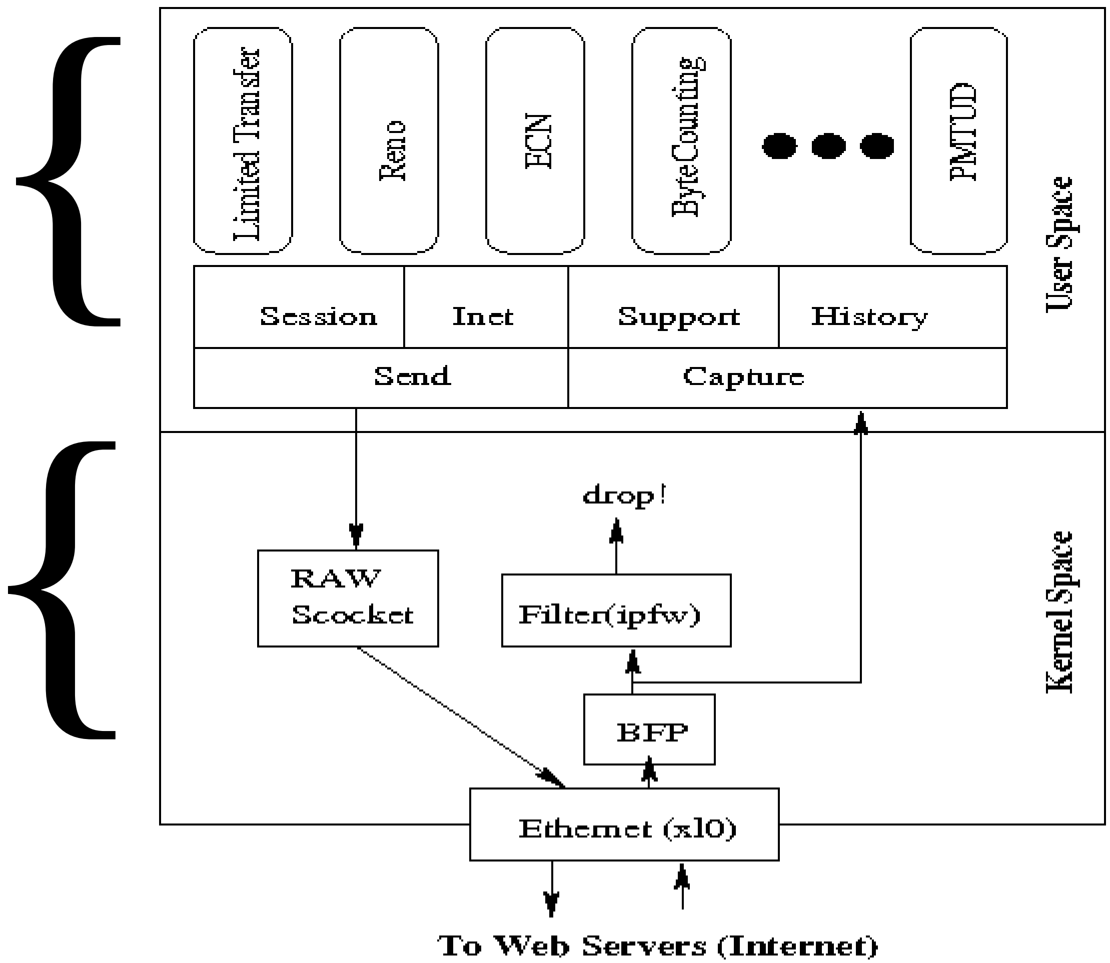
Experimental Platform

- Measuring TCP implementations
  - Passive measurements of web clients
  - Active measurements
    - Web server mechanisms
    - Interactions with environment
- Active measurements requirements
  - Measure in-the-field Web servers
  - Employ only conformant TCP traffic
  - Unilateral control at measurement side
- Employ "undercover" web clients…

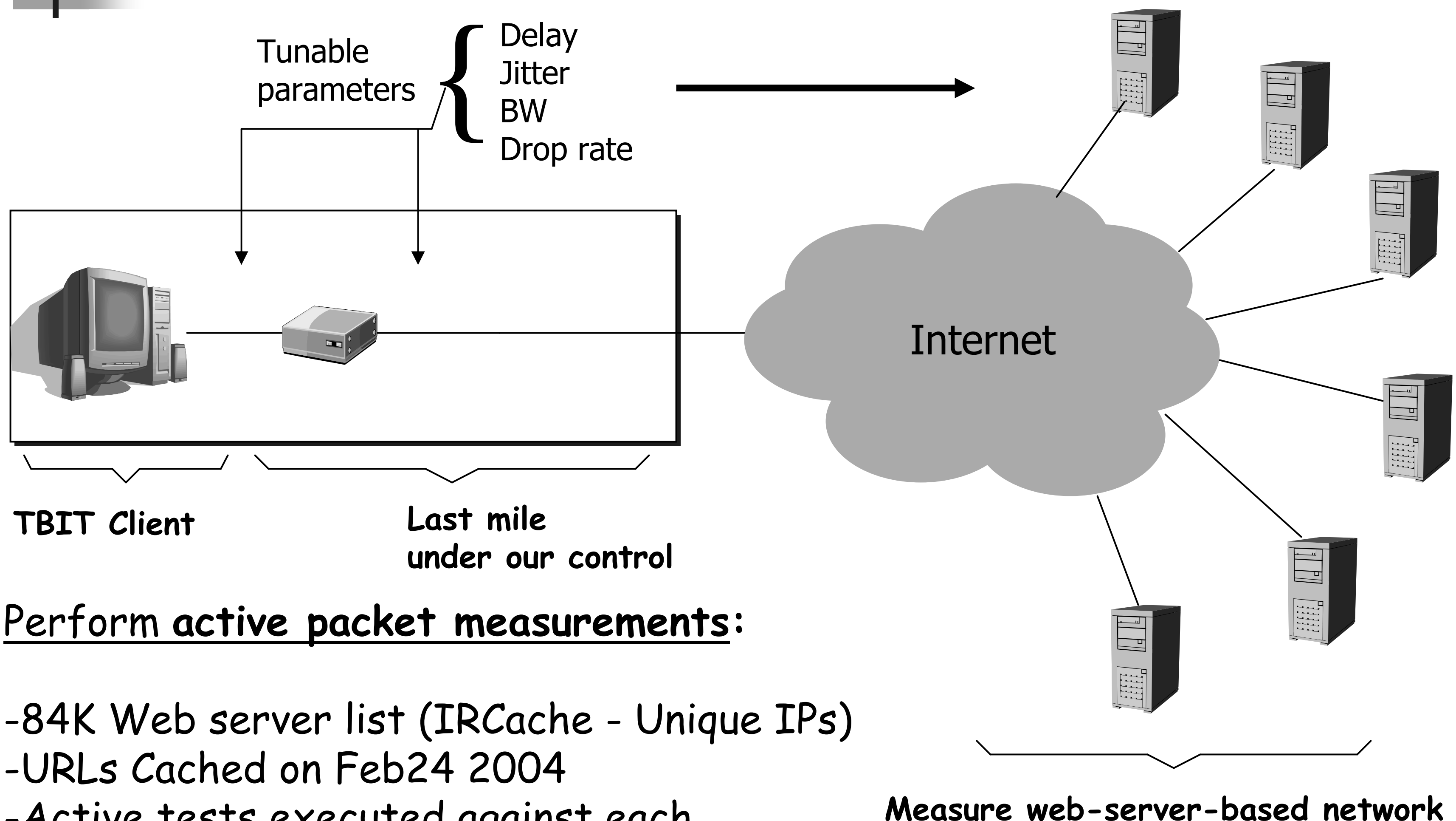
Undercover Web Clients: TBIT

TBIT

FreeBSD
Linux



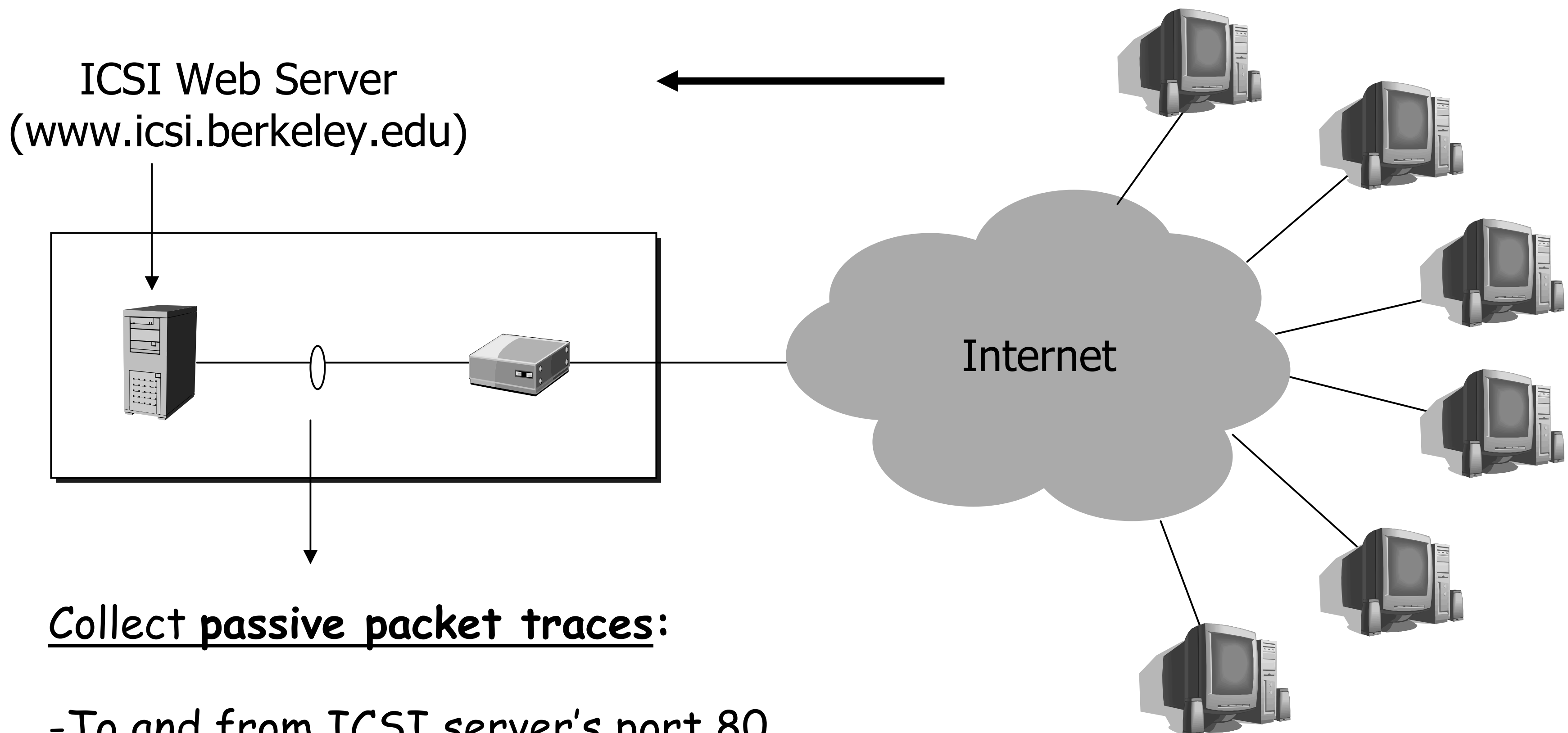
Experimental Platform: Server Side



Perform **active packet measurements**:

-84K Web server list (IRCache - Unique IPs)
-URLs Cached on Feb24 2004
-Active tests executed against each server in URL list

Experimental Platform: Client Side



Collect **passive packet traces**:

-To and from ICSI server's port 80
-Two-week collection time (Feb24-Mar10)
-206K Connections observed
-28K Clients (e.g. IP addresses)

MEASUREMENTS

Deployment evolution

- TCP stacks Deployment
- SACK Info Processing
- SACK Info Generation
- D-SACK
- Byte Counting
- Limited Transmits
- Window-Scale Option
- Window Halving
- Cong. Window Buildup

Internet Architecture Evolution

- ECN Deployment
- Path MTU Discovery
- Use/Abuse of IP Opts
- Use/Abuse of TCP Opts
- Middlebox Discovery

Tracking Changes

- Reordering, drops
- MSS Values
- ICW Values/Perf.
- RTO Values
- Redirections

Talk Outline

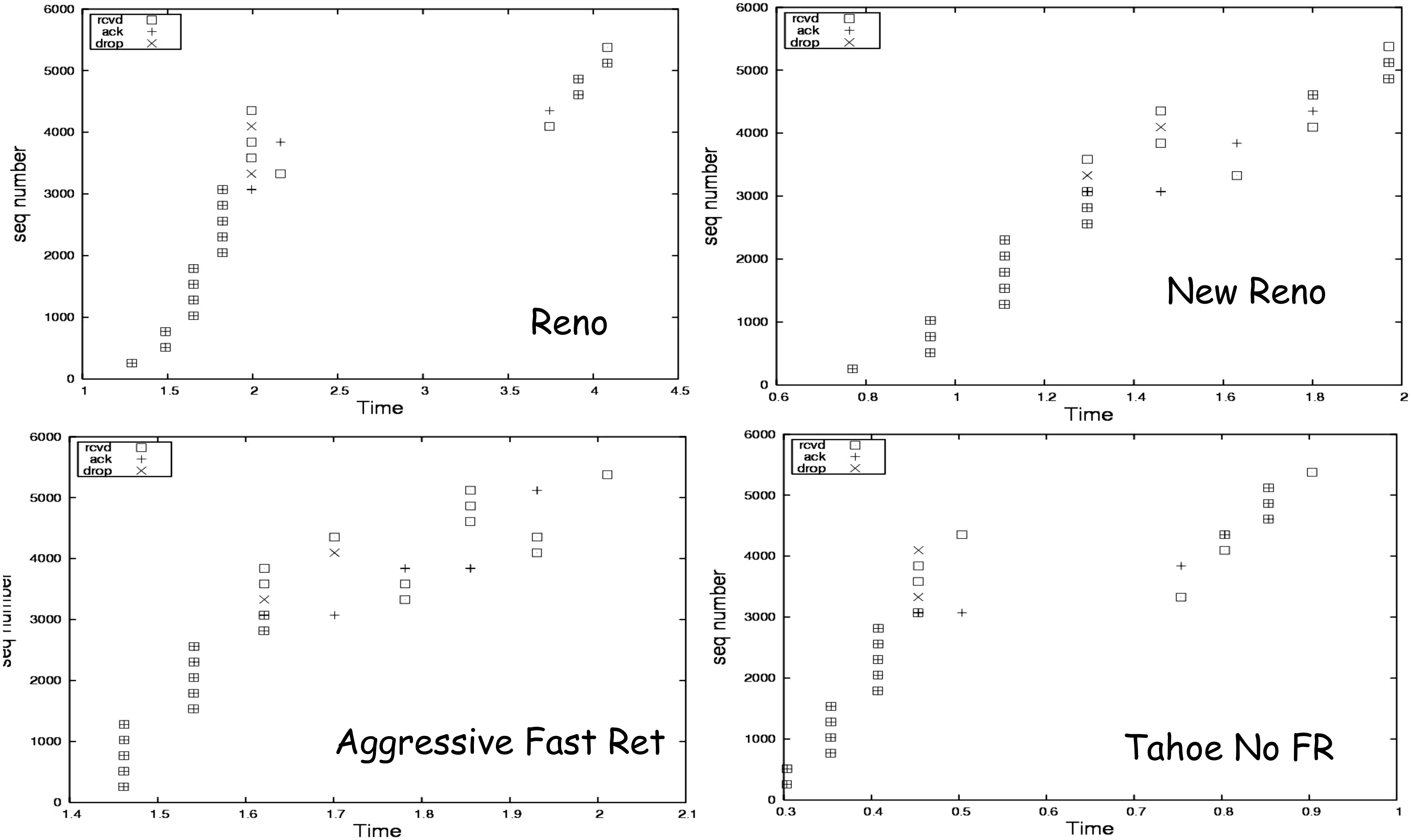
- ✓ Motivation
- ✓ Measurement Platform
- Active Measurements
  - Deployment of Transport Mechanisms
  - Middleboxes and Transport Protocols
- Summary of Results
  - Including client-side
- Conclusions
- Future Work

Deployment of Transport Mechanisms

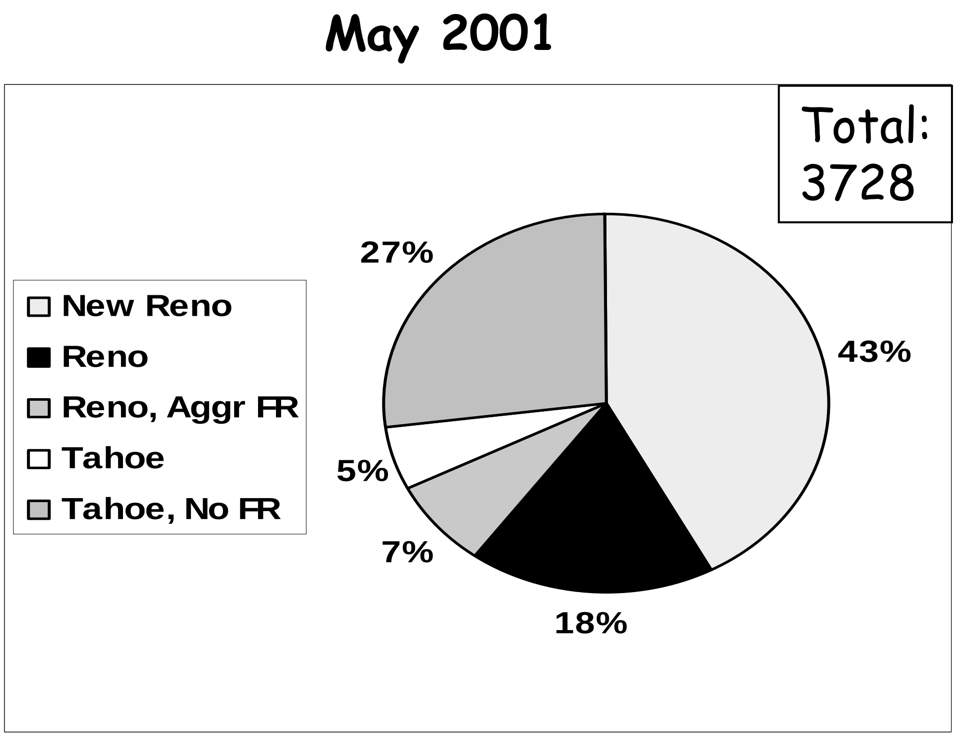
Test: Assess Deployment of TCP stacks

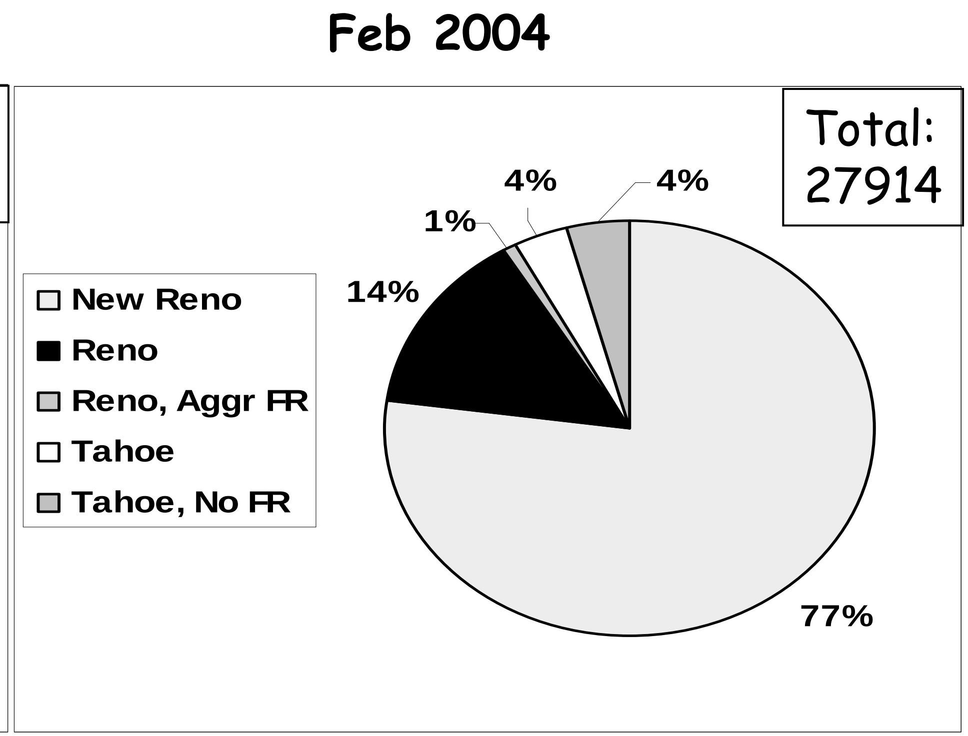
- Establish connection with Web server
  - Use small MSS
  - Restrict Congestion window to 5 segments
- Request web page
- Receive and ACK incoming packets, but…
  - Drop packet 13^{th}
  - Receive and ACK packets 14^{th} and 15^{th}
  - Drop Packet 16^{th}
  - Continue download until receiving packet 25^{th}

TCP Behaviors observed



TCP Deployment Results



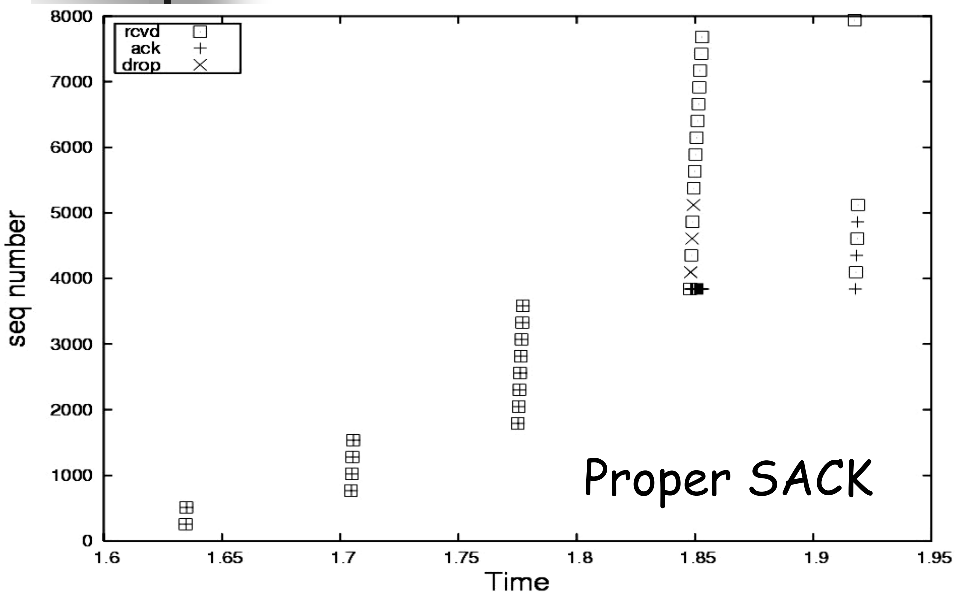


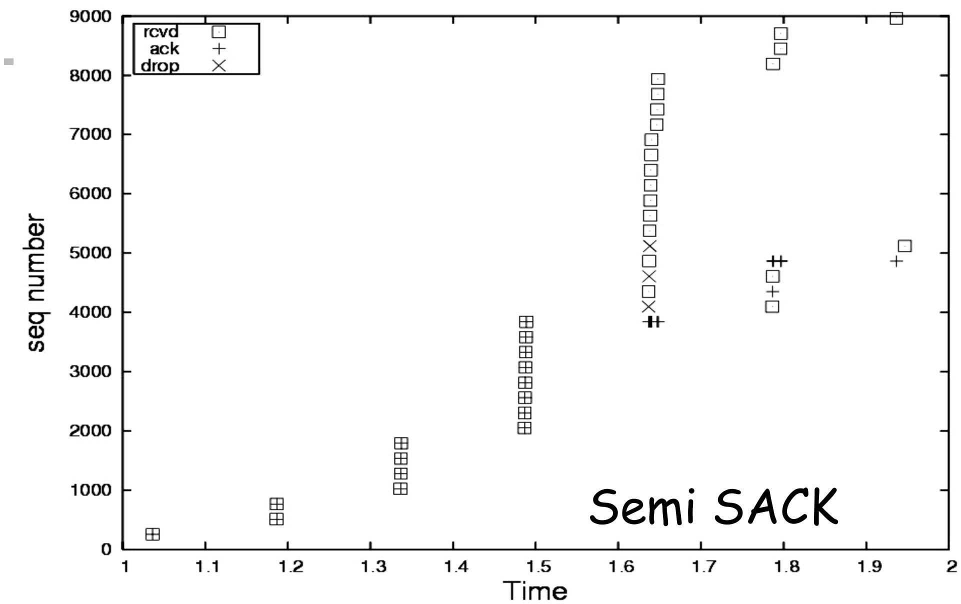
- Deployment of New Reno increased significantly
- Buggy Tahoe without Fast Retransmit decreased
- Network simulations should use New Reno TCP

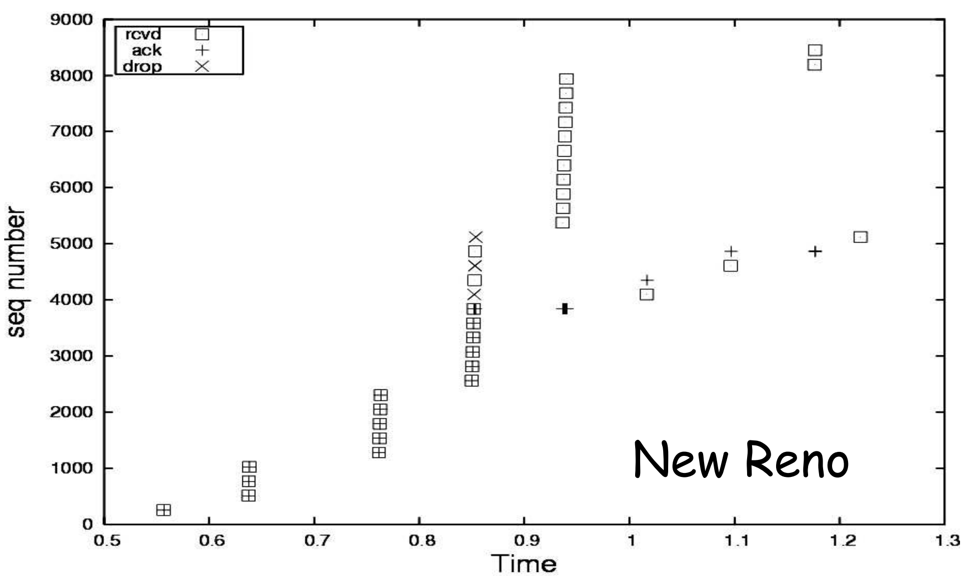
Test: Assesing SACK Behaviors

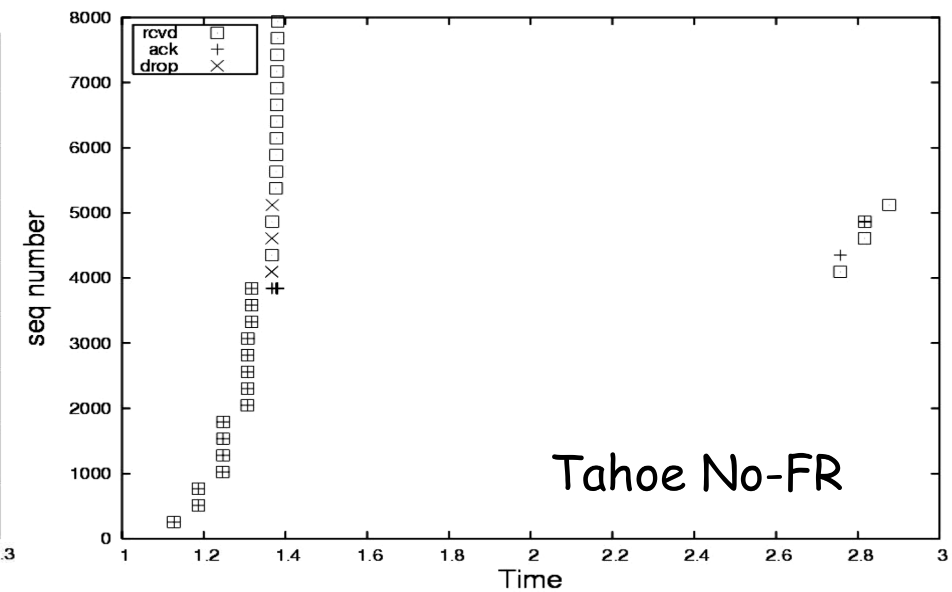
- Negotiate SACK-enabled connection
- If server not capable => NO SACK
- Request web page download
- Receive and ACK incoming packets but…
  - Drop packets 15th, 17th, and 19th
  - Continue receiving and ACKing packets normally (sending appropriate SACK blocks for "drops")
  - Observe retransmission behavior
  - Terminate test, close connection

SACK Behaviors observed



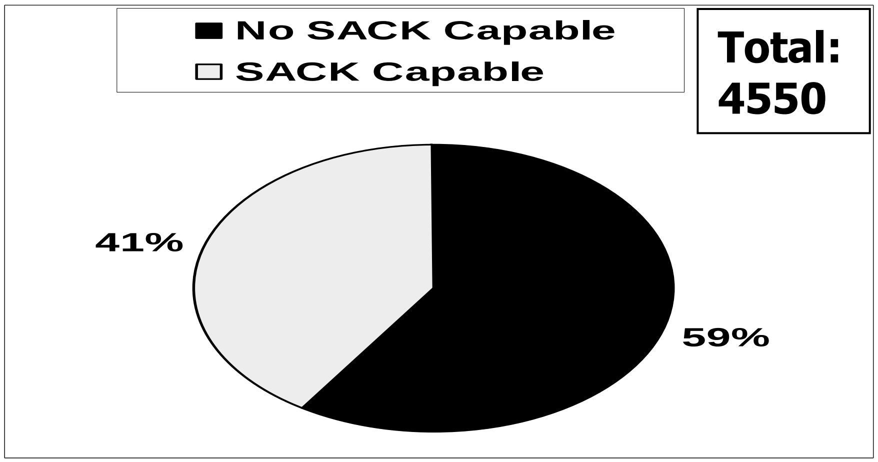


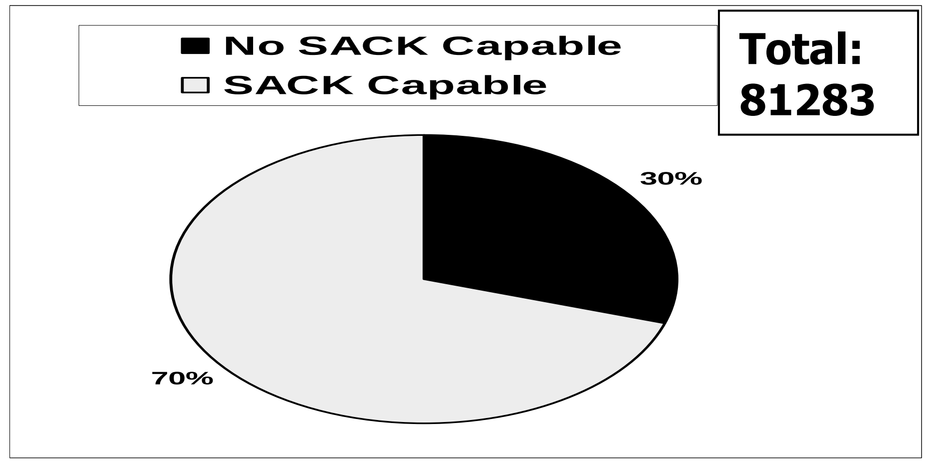


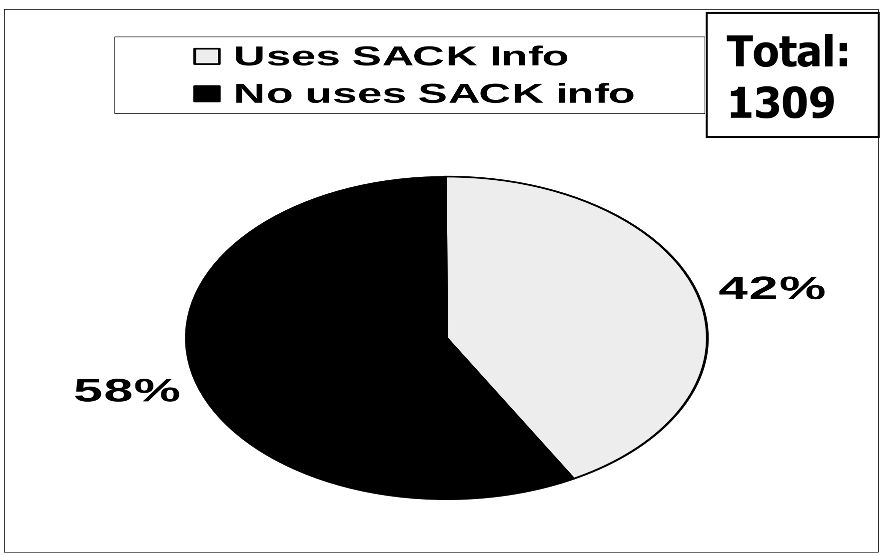


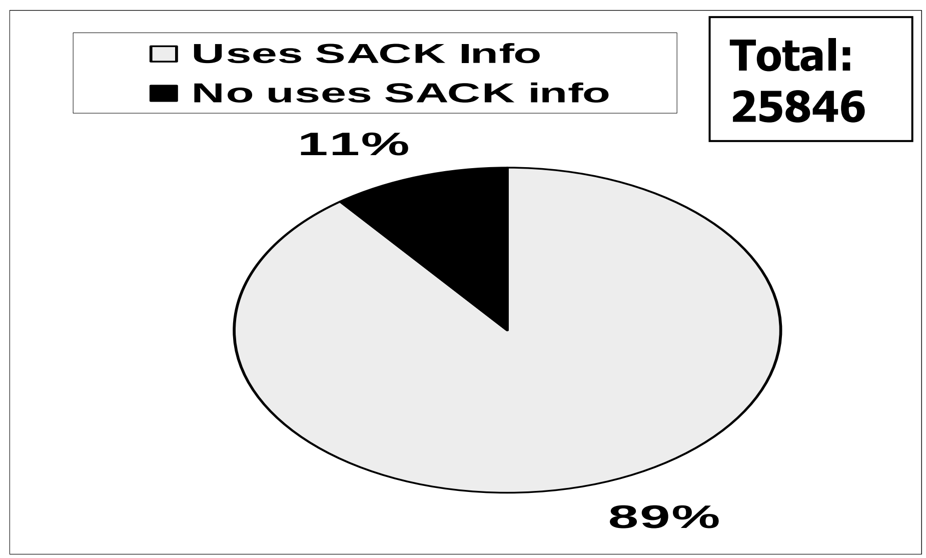
Sack Sender test: Results

May 2001 | **Feb 2004**



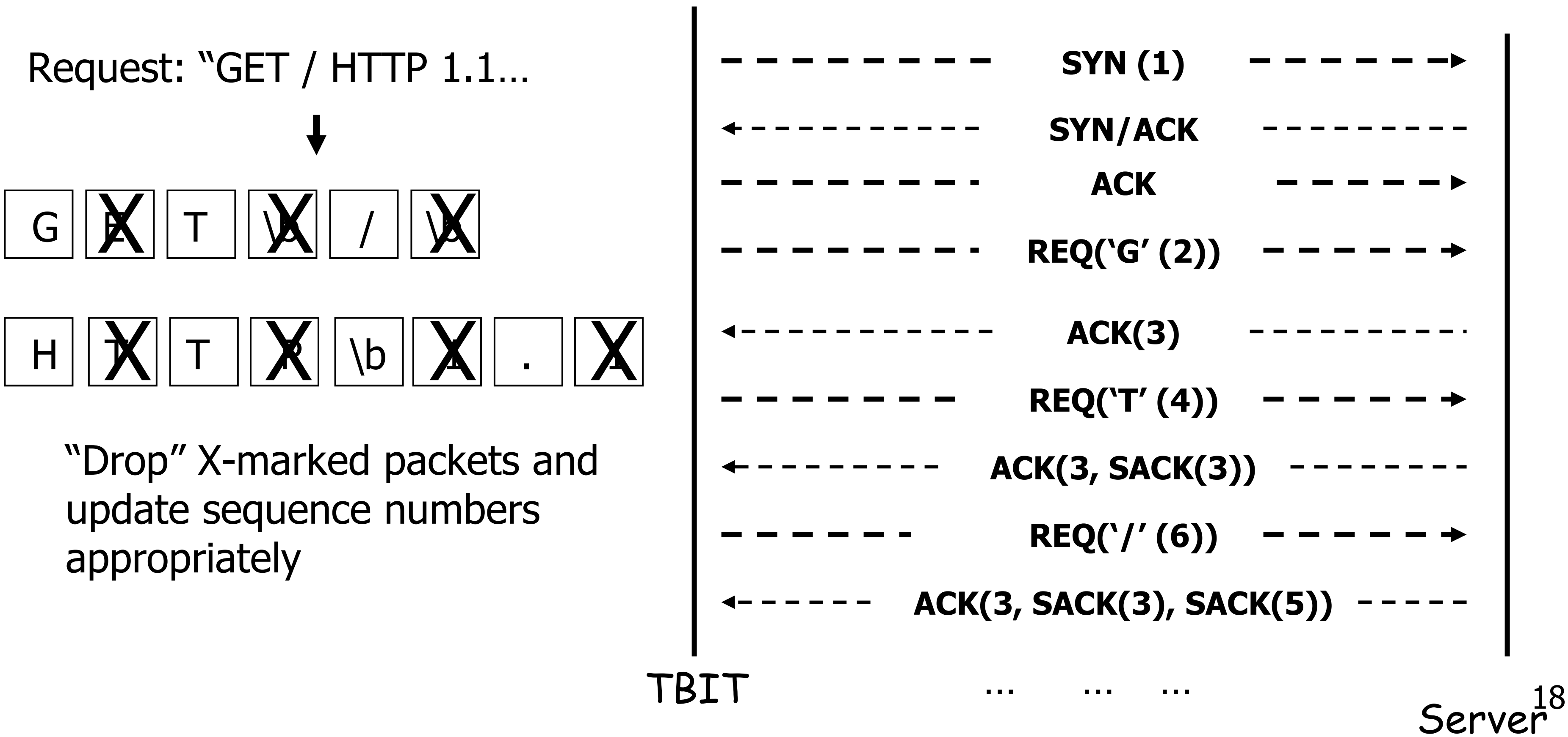






Generation of SACK Information

- Do servers generate accurate SACK information?
- Test:



Sack Receiver Test: Results

Type of Server	Servers (%)
Total servers	84394
I. Not SACK-Capable	24361 (28.8%)
II. SACK blocks OK	54650 (64.7%)
III. Shifted SACK blocks	346 (0.5%)
I.V. Errors	5037 (6%)

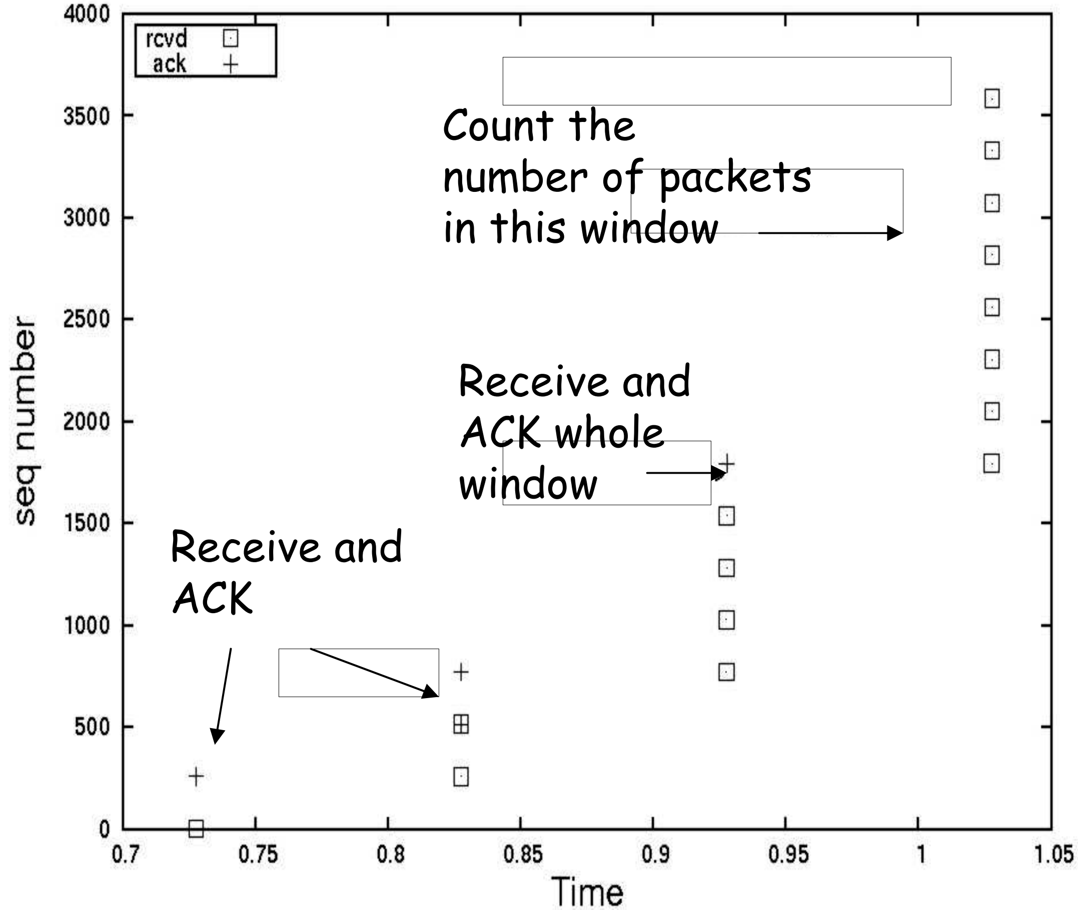
- Shifting blocks could have been caused by:
  - NATs, Fingerprint scrubbers,…
- Such middlebox interactions affect any TCP-based communication

Test: Appropriate Byte Counting (ABC)

- TCP Congestion Control
  - Slow start: increase CWND by one segment for each received ACK
  - Congestion avoidance: increase CWND by 1/MSS for each received ACK
- Drawbacks
  - Delayed ACKs reduce CWND opening rate
  - Mis-behaving receivers may induce servers to open CWND too fast
- ABC Proposal
  - Increase CWND based on bytes ACKed by incoming ACKs, instead of based on number of ACKs received

ABC Test: Example for ICW = 1

- Receive and ACK packets 1, 2 and 3
- Wait for window of 4 packets to arrive
- ACK whole window
- Count number of packets received in next window



ABC Test: Results

Slow Start Behavior	Number (%)
Total number of servers	44579
I. Classified Servers	23170 (52%)
I.A. Packet Counting	15331 (51.9%)
I.B. ABC	65 (0.1%)
II. Unknown behavior	288 (0.6%)
III. Errors	21121 (47%)

- Notice a 5-year old proposed mechanism addressing (1) performance concerns and (2) security issues and yet, not being deployed!

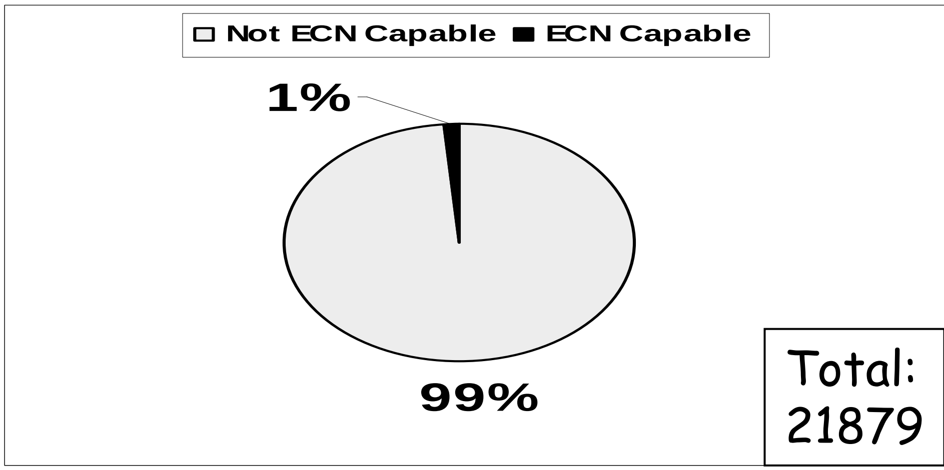
Middleboxes and Transport Protocols

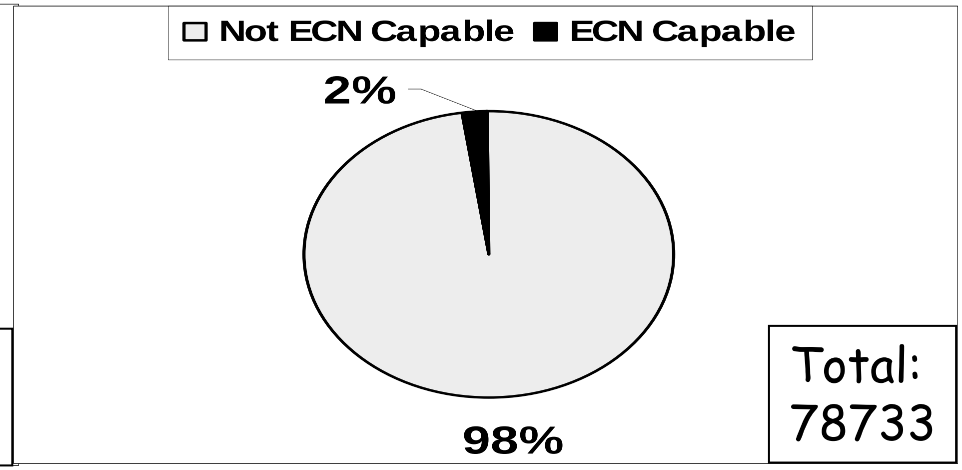
ECN Capabilities

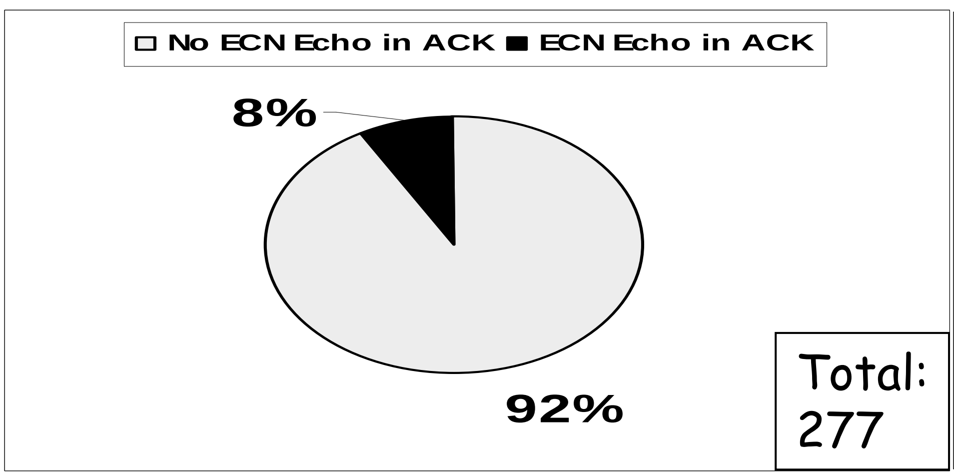
- **ECN**: Explicit Congestion Notification
  - Allows routers to notify congestion to end nodes
- **TCP**: 2-way handshake
  - Active end: send ECN-Setup SYN (ECN_ECHO, CWR)
  - Passive end: send ECN-Setup SYN/ACK (ECN_ECHO)
- **IP**: 2-bit ECN field in IP header => 4 ECN CPs
  - 00: Not ECT
  - 01: ECT(1) – Sender is ECN capable
  - 10: ECT(0) – Sender is ECN capable
  - 11: Congestion Experienced (CE)

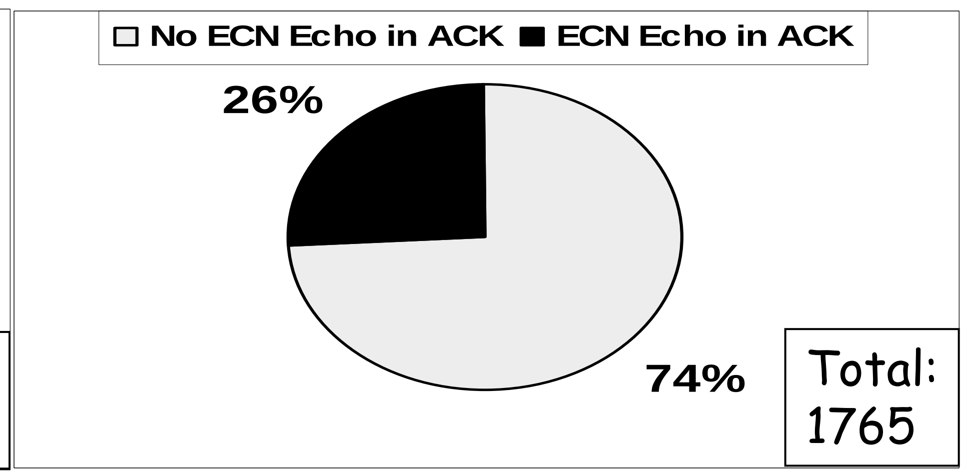
ECN Test: Results

May 2001 / **Feb 2004**



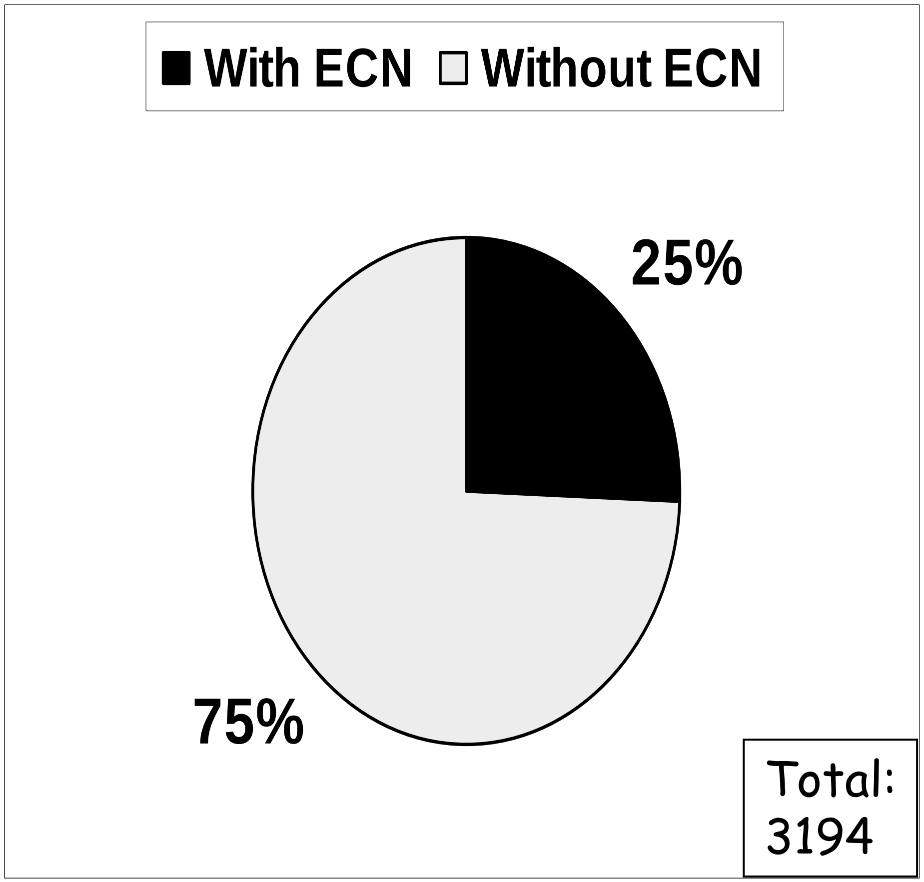






ECN Test: Results (2)

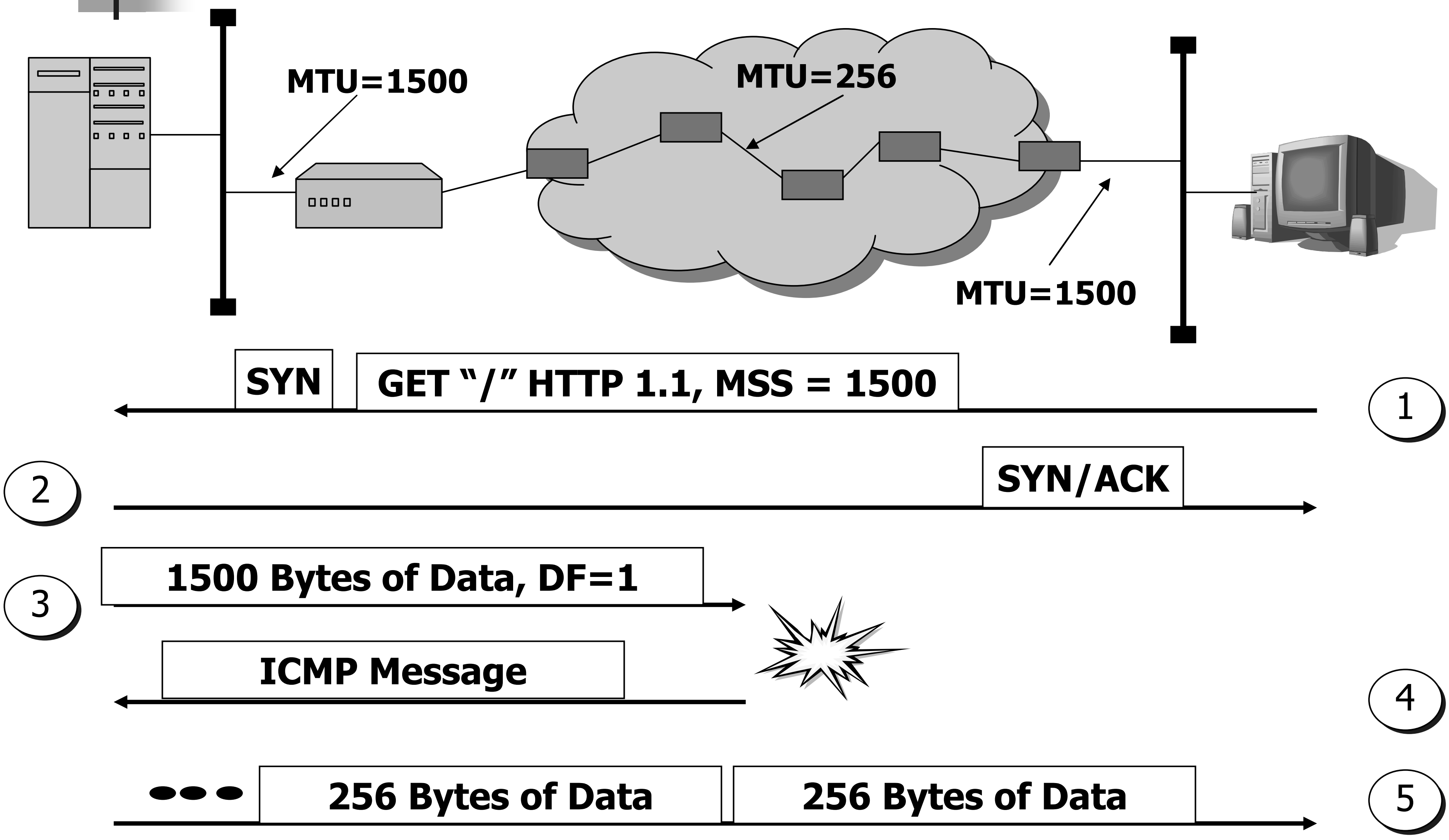
Blocking ECN (3194 Conns)



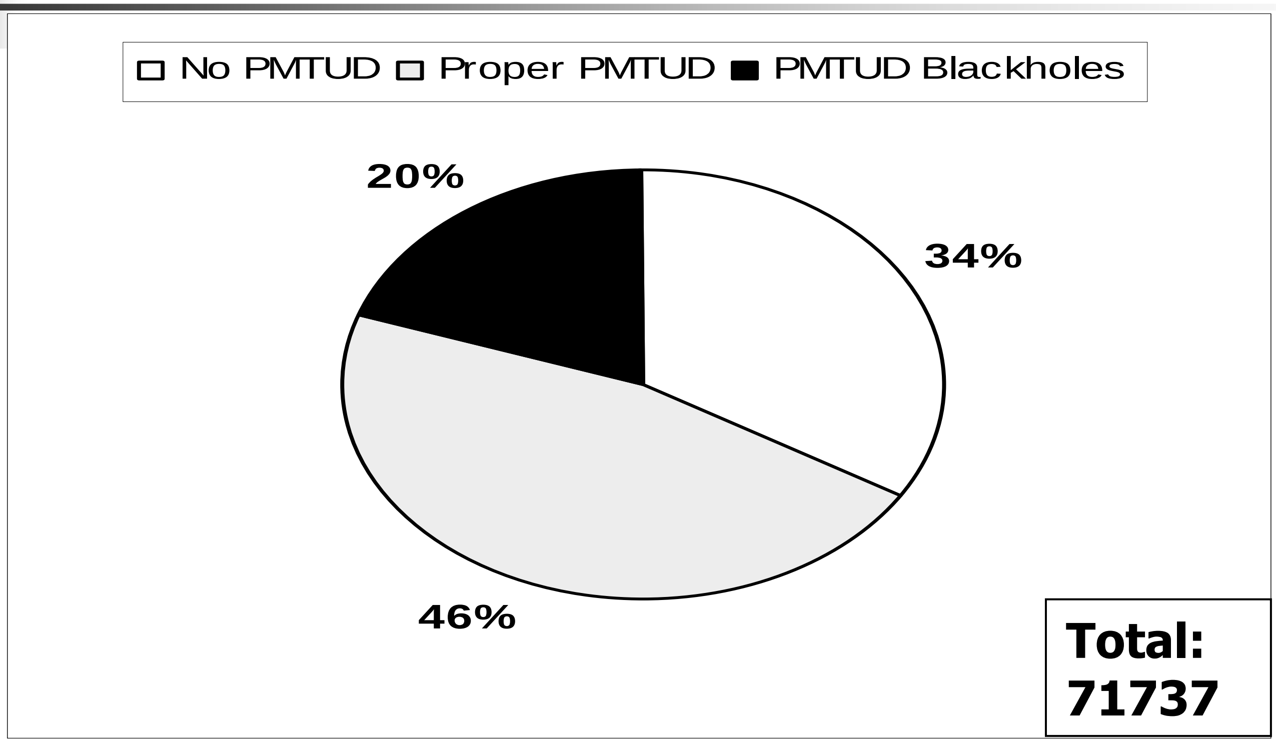
ECN Code Points (1765)

ECN CPs data pkts	Number (%)
Received pkts w/ ECT 00 (Non-ECT)	758 (42%)
Received pkts w/ ECT 01 (ECT(1))	0 (0%)
Received pkts w/ ECT 10 (ECT(0))	1167 (66%)
Received pkts w/ ECT 11 (CE)	0 (0%)
Received both pkts with ECT00 & ECT 10	174 (10%)

Path MTU Discovery (PMTUD)



PMTUD Test: Results

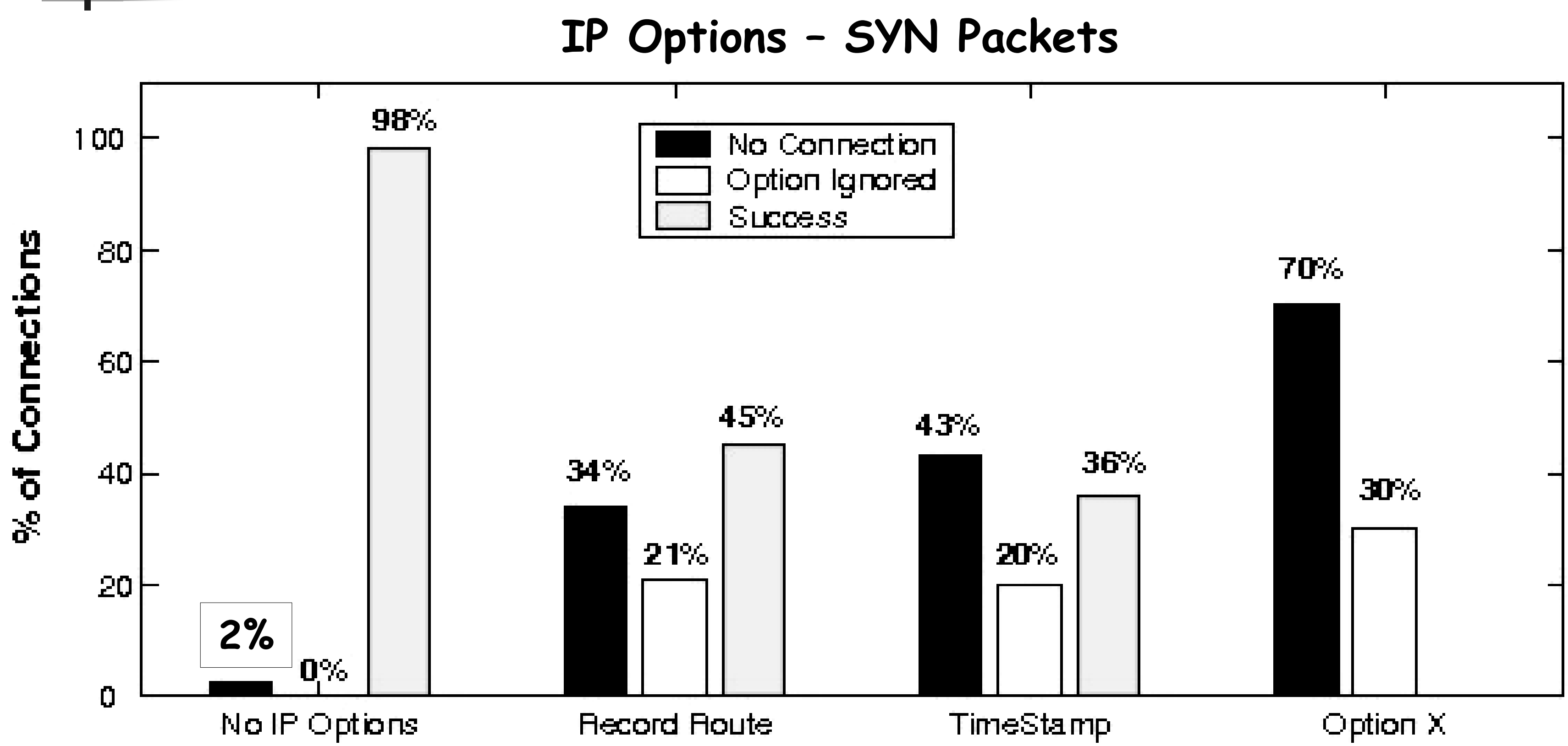


- Observed a non trivial number of black holes
- No hope for new ICMP-based proposed mechanisms
  - Explicit corruption notification
  - Handoff notification

Interference with TCP/IP Options

- TCP/IP options
  - Allow encoding additional information at end of packets
- Several concerns raised about using IP Options
  - Overhead, misalignment problems, DoS attacks
- Solutions to concerns
  - Range from OS patches to dropping "offending" packets
- Issue concerns protocol designers
  - Use of unused TCP/IP options in new proposals
  - Ex: QuickStart (QSR) IP Option
- TCP/IP Options tests
  - Evaluate connections with SYN-packet TCP/IP options
  - Evaluate connections with Mid-Stream TCP/IP options

IP Options Test: Results



- Severe interference with known and unknown IP options
- Negative results for of new IP-option-based mechanisms

Summary of Results: TCP Evolution

TCP Mechanism	**Conclusion**
TCP Cong. Ctrl	~ 2/3 use **<u>NewReno</u>** => Use it in ns
Loss Recovery	**<u>SACK-cap Prevalence</u>**: ~ 2/3 servers, 9/10 clients Most claiming SACK, do SACK properly **<u>SACK info</u>**: (mostly) correct
DSACK	~ ½ of SACK-capable servers, send D-SBs
ABC	Not deployed
LT	Not fully deployed (~1/4 of servers)
MSS	Most clients use ~ 1.4K bytes Most servers accept << 1.4K bytes
RTO	Many servers use RTO < 1s

Summary of Results: TCP Evolution (2)

TCP Mechanism	Conclusion
ICW	Many ICW = 1, Most used ICW = 2-4 Some gain from larger ICWs No changes for reordering and losses
Window Scaling	Most servers support WS (shift count=0)
Window Halving	Most servers do proper window halving Some servers use CWND without caring for RWND
Window Buildup	Most servers do no increase cwnd if not used
Advertised Window	Most clients surveyed advertise 64KB windows Many clients advertise 8KB and 16KB
ECN	Very few servers using ECN (~2.3%) 1% Increase since 2001

Summary of Results: Network Evolution

Behavior	Conclusion
SACK	Small number of cases, web servers and clients receive SACK blocks with incorrect sequence numbers
ECN	Roughly 1% of refused connections
PMTUD	< ½ servers PMTUD-capable Likely routers/middleboxes blocking ICMP messages for 1/6 of the servers
IP Options	Many failures (1/3) when IP RR or TS SYN options used Majority of failures (70%) if unknown IP option used
TCP Options	More resilient and tolerant than unknown options
Reordering	Significant small-scale reordering

Conclusions

- Achieved set goals
  - Tracked deployment of transport mechanisms
  - Evaluated transport-network interactions
- Competition of interests complicates deployment priorities
  - ABC not implemented, LT implemented
  - PMTUD failing, no ECN deployment,…
- Pinpointed specific cases exemplifying how evolving network challenges end-to-end principle
  - Fundamental design principles of Internet have changed
  - Current network needs to evolve towards new reality

Future Work

- Further TCP in-the-field behavior
  - Restart behavior after an idle period; Backoff behavior
  - Behavior in other environments (p2p, wireless,…)
- Study other protocols and mechanisms
  - <u>Existing</u>: UDP, FTP, HTTP, RTP, …
  - <u>New</u>: AQM, High-Speed TCP, SCTP, DCCP,…
  - TCP in other environments (P2P, web caching,…)
- Further exploration of Middlebox behavior/impact
  - Many open questions (e.g. How about PEPs?)
  - Detecting middleboxes
- Continuous Monitoring Platform
- Active measurements of client behaviors
- Unilaterally-controlled Active Measurements

Contact & Information

- People:
  - Alberto Medina: medina@icir.org
  - Sally Floyd: floyd@icir.org
  - Mark Allman: mallman@icir.org
- Software and data
  - http://www.icir.org/tbit

icir

i.c.s.i. center for internet research