Robust TCP Stream Reassembly In the Presence of Adversaries

Sarang Dharmapurikar and
Vern Paxson
Washington Univ.
UC Berkeley
Usenix Security 2005
Presented by N. Sertac Artan

Motivation

- TCP Reassembly is hard
- It is even harder in the presence of adversaries
- Benign TCP traffic's out-of-sequence behavior is not completely characterized

Contribution of the paper

- Characterization of TCP traffic for out-of-sequence behavior
- A TCP reassembly architecture that is designed for the worst case, and optimized for the common case.
- High-speed, in-line network element to enable high-speed intrusion prevention
- Deployed at a site's gateway

Problem Definition

- A hole/gap can result from packet loss or reordering
- Must buffer out-of-order packets until the hole is filled
- Only after the buffer is filled the buffered packets can be inspected
- How much buffer is needed?
- How many connections need such buffers?
- How long does a hole persist?
- Should the hardware immediately forward out-of-order packets to the receiver, or only after they have been inspected in correct order?
- What if an adversary tries to overflow the buffers?
- What should be done if a buffer overflows?

Discussion on Out-Of-Order Packet Arrival

- In SIGCOMM'97 Paxson observed the relation of
  - packet loss and
  - Reordering
- Result
  - Many connections are loss free
  - Losses come in bursts
  - Packet reordering varies between 0.6% and 2% of all the packets → Exception
- Data is now out of date.

Discussion on Out-Of-Order Packet Arrival

- In Trans. Networking, Bennett *et. al.* reported 90% of reordering in TCP packets!
- Result
  - Packet reordering is not a pathological behavior
  - It is normal behavior
- Older router architecture
  - Packet-level-parallelism instead of connection-level parallelism

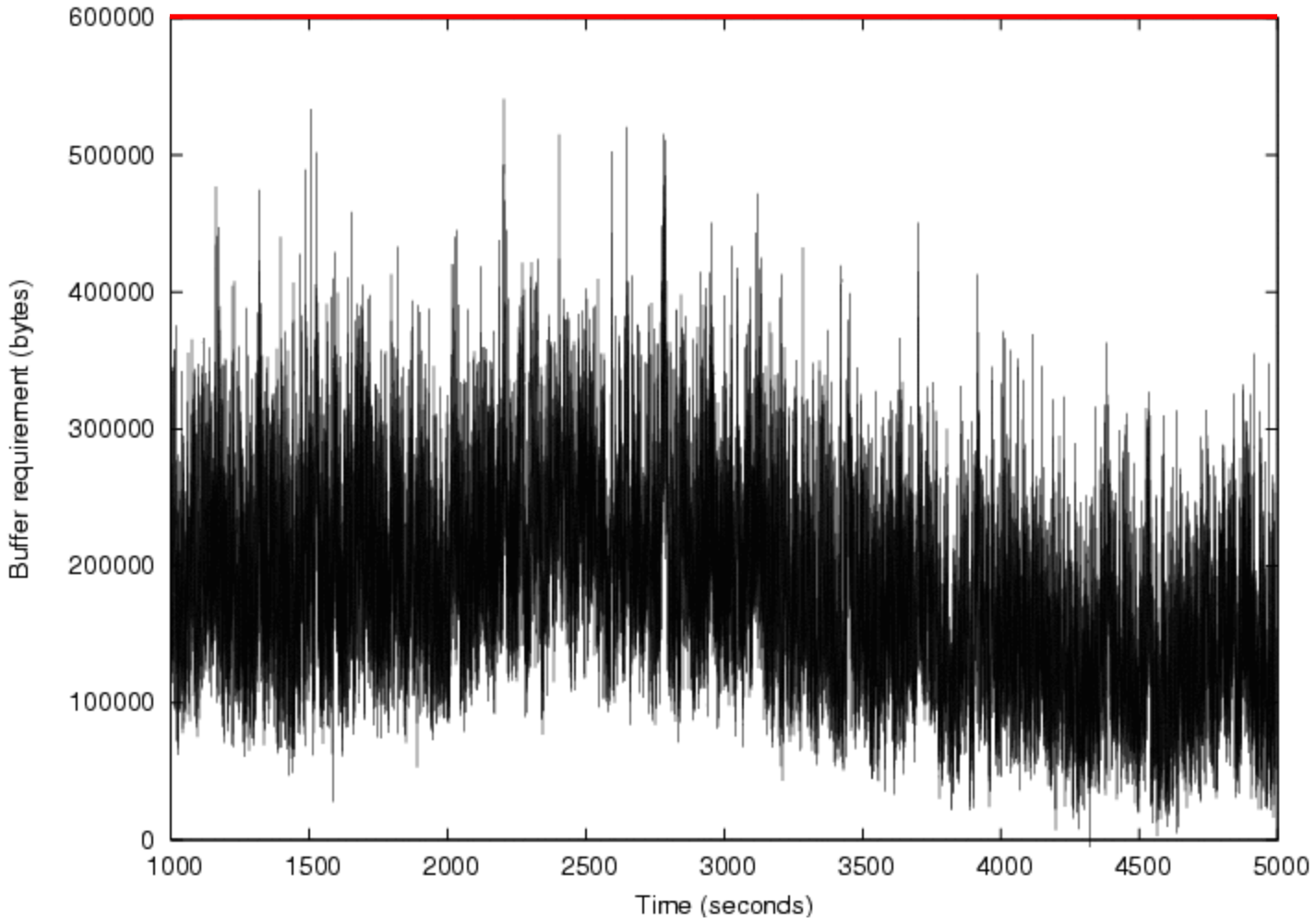
Discussion on Out-Of-Order Packet Arrival

- In SIGCOMM'02, Bellardo and Savage
- Result
  - No relevance between the forward and reverse paths in terms of packet reordering
  - Reordering rates increase when the spacing between packets decreases.
  - Out-of-order packets are an exception

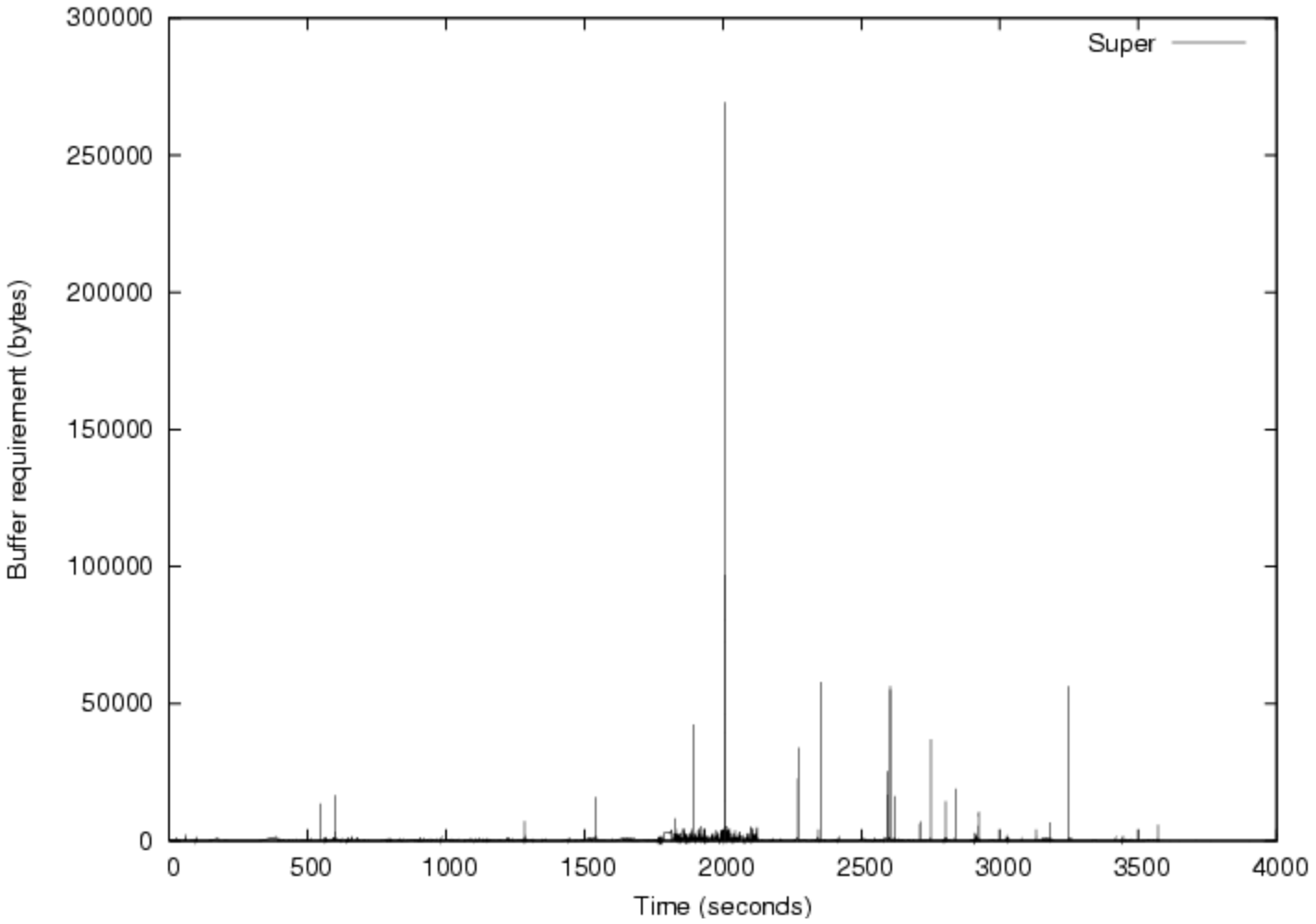
Datasets

	$Univ_{sub}$	$Univ_{19}$	Lab_{lo}	Lab_2	*Super*	*T3*	*Munich*
Trace duration (seconds)	303	5,697 / 300*	3,602	3,604	3,606	10,800	6,167
Total packets	1.25M	6.2M	1.5M	14.1M	3.5M	36M	220M
Total connections	53K	237K	50K	215K	21K	1.04M	5.62M
Connections with holes	1,146	17,476	4,469	41,611	598	174,687	714,953
Total holes	2,048	29,003	8,848	79,321	4,088	575K	1.88M
Max buffer required (bytes)	128 KB	91 KB	68 KB	253K	269 KB	202 KB	560KB
Avg buffer required (bytes)	5,943	2,227	3,111	13,392	122	28,707	178KB
Max simultaneous holes	15	13	9	39	6	94	114
Max simultaneous holes in single connection	9	16	6	16	6	85	61
Fraction of holes with < 3 packets in buffer	90%	87%	90%	87%	97%	85%	87%
Fraction of connections with single concurrent hole	96%	98%	96%	97%	97%	95%	97%
Fraction of holes that overlap hole on another connection of same *external* host (§ 5.1)	0.5%	0.02%	0.06%	0.06%	0%	0.46%	0.02%

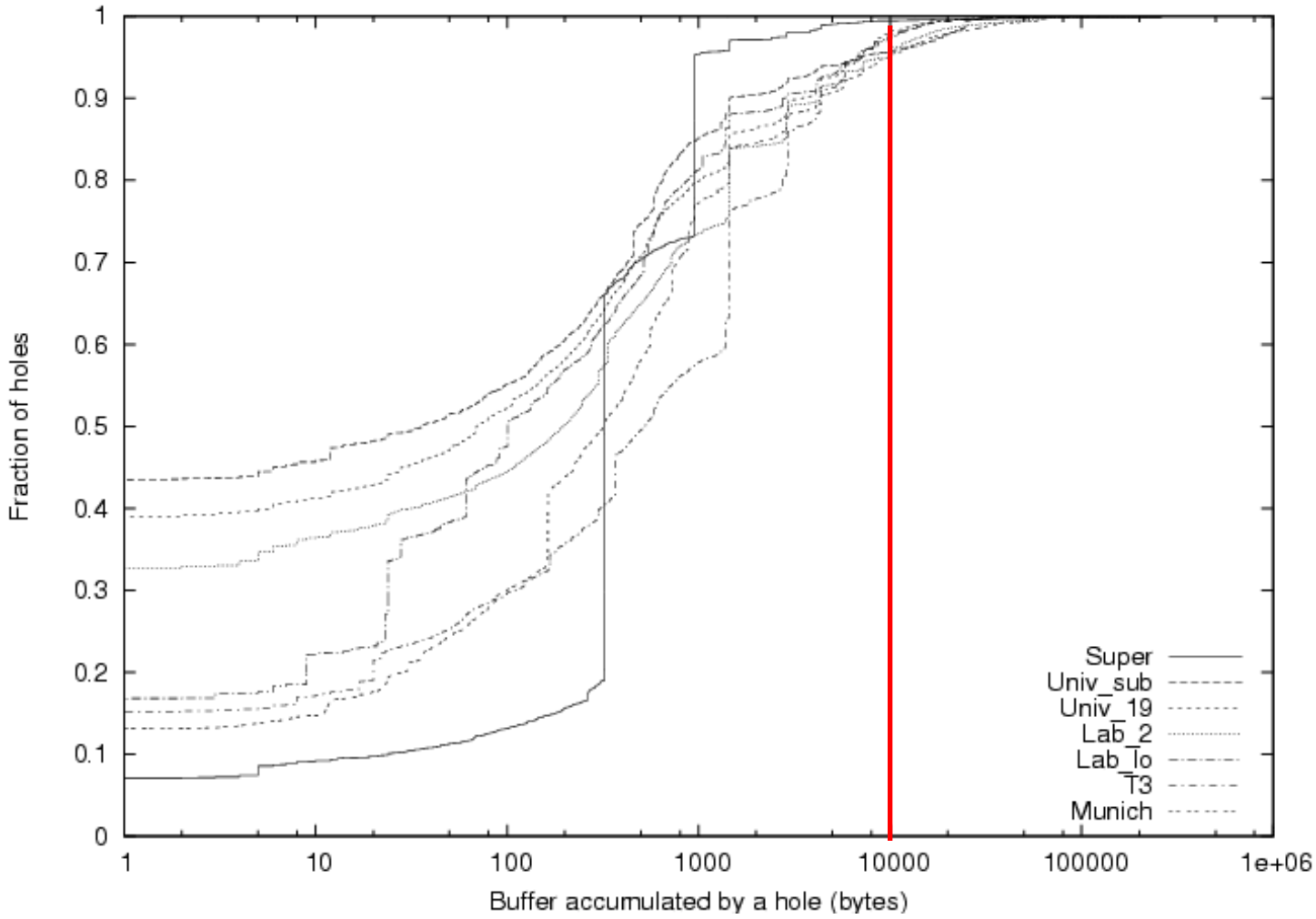
Munich Trace Buffer Occupancy



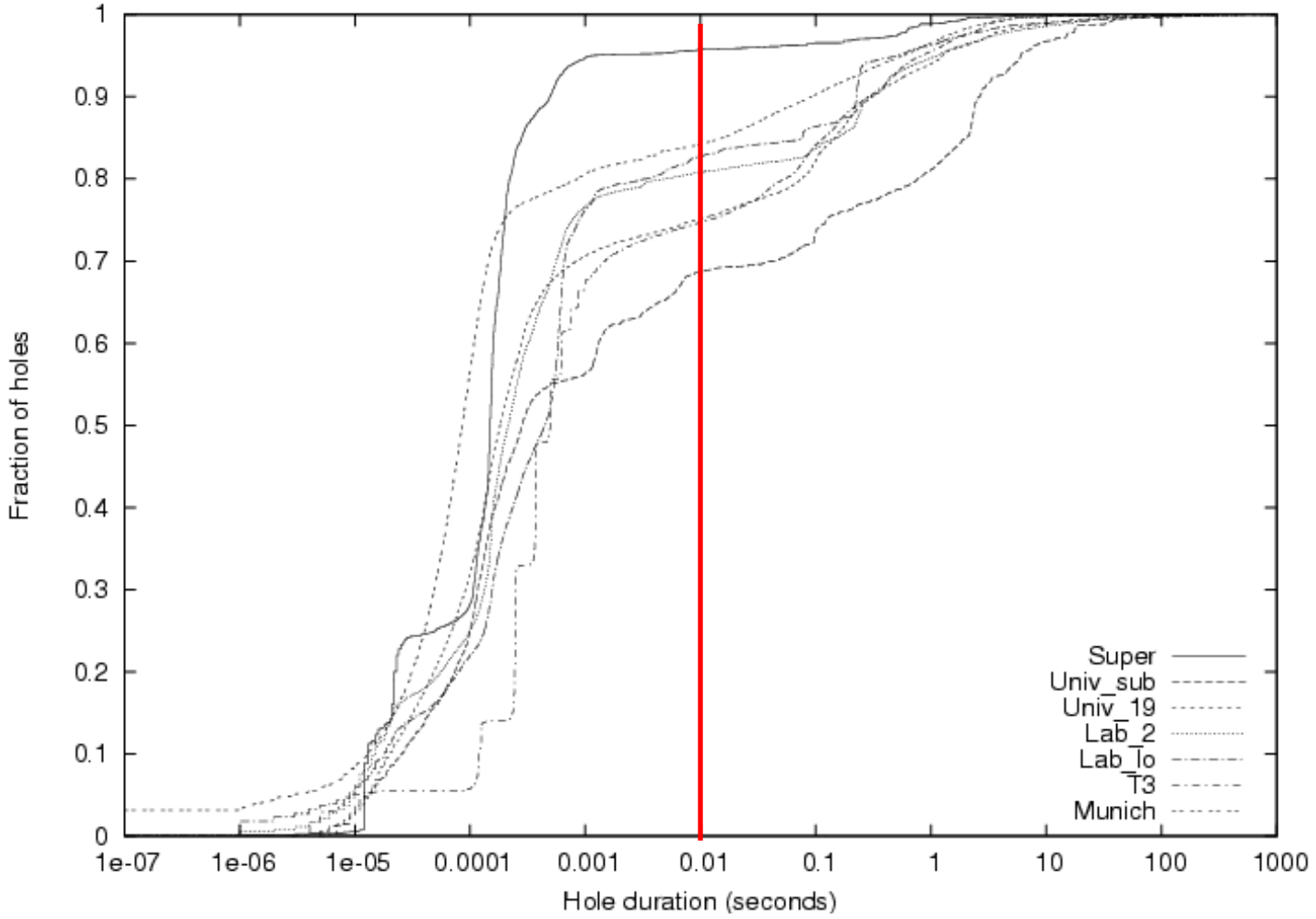
Super Trace Buffer Occupancy



Buffer Occupancy



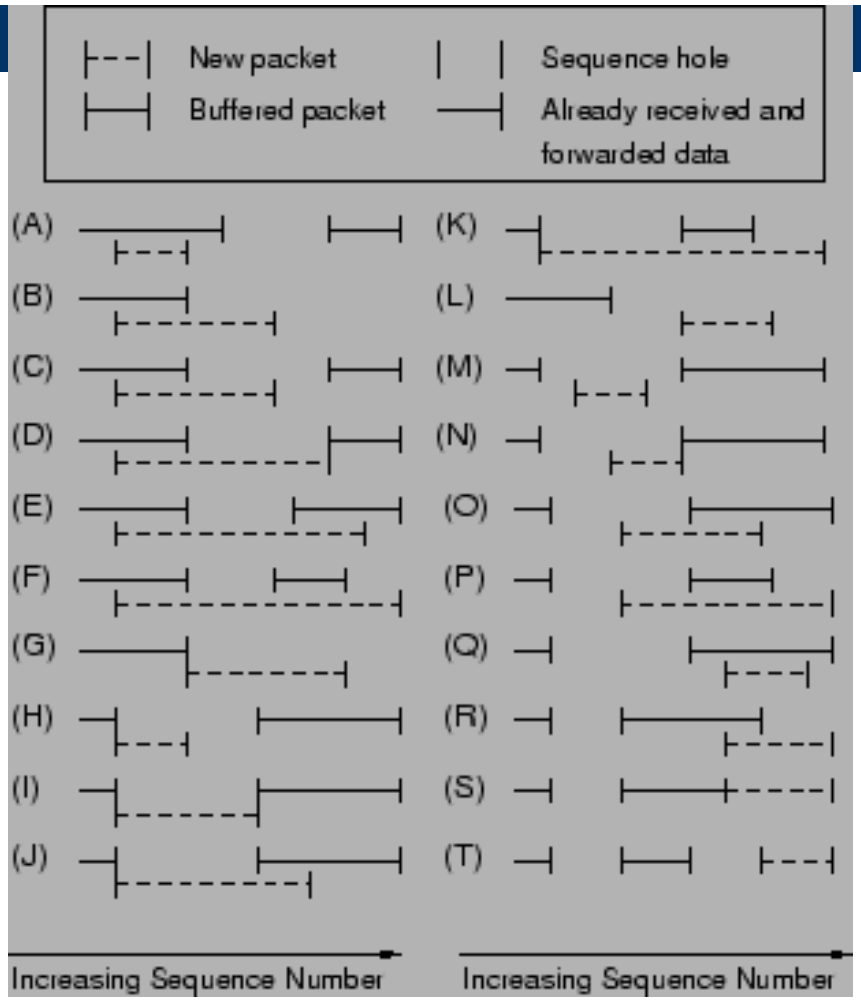
Hole duration



Discussion on Out-Of-Order Packet Arrival
Measurements of the current proposal

- Packet reordering in TCP traffic affects 2-3% of the overall traffic
- Nearly arbitrary permutations of sequence hole creation
- Buffer occupancy remains below 600 KB
- Most holes have a lifetime of less than 0.01 seconds, average is less than 1 ms
  - This suggests they are due to packet reordering and not due to packet loss
  - In the latter case the hole will persist at least for RTT
  - Packet reordering is possibly due to multi-pathing

Hole creation

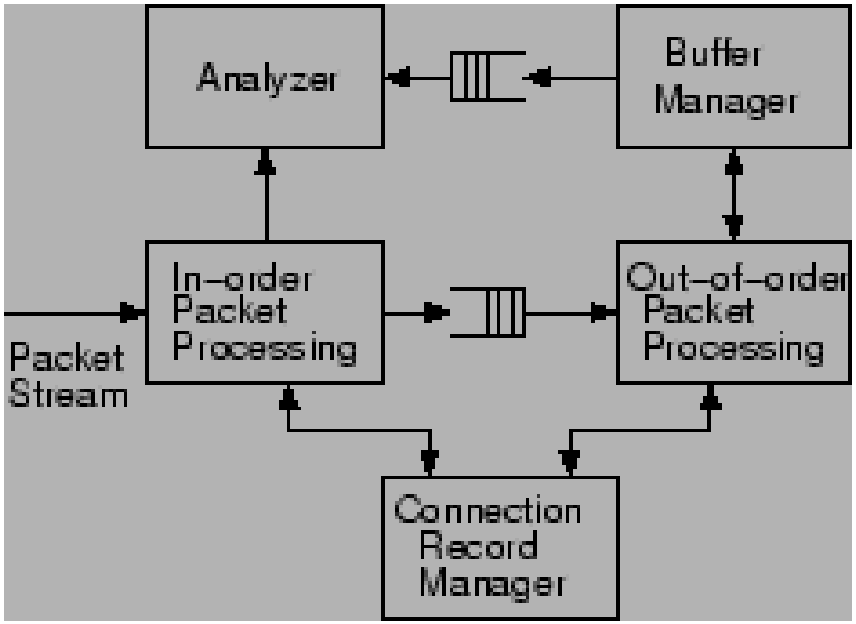


Hole creation possibilities

- Very long-lived holes
- Holes that accumulate large amounts of buffer
- Large number of simultaneous holes in a connection
- Simultaneous holes in both directions of a connection
- High-rate of hole creation
- Most cases above are highly rare
- Highly dominant case is a single, short lived hole in just one direction of the connection

System Architecture

- Design for common case, optimize for worst case
- An inline-system
  - Can drop packets,
  - Kill connections
- Connection record
  - Hash table
- Out-of-order packet buffer
- Buffer size do not exceed 64 KB



Out-of-order Processing

- An arriving packet can plug a hole partially or completely
  - It can close the hole from the beginning, from the end or from the middle
  - If closes from the middle the hole becomes two holes
- If there is an overlap, discard the overlapping data from the new packet

Dealing with an adversary

- Possible targets
  - Connection state memory
  - Out-of-order packet buffer

Out-of-order Processing: To forward or not to forward?

- Should we forward out-of-sequence packets as they arrive, or hold on to them until they become in sequence?
  - Not forwarding these packets may affect TCP performance significantly
- In their initial design they always forward packets
  - Deliver in-order packets immediately to byte-analyzer
  - Retain copies of out-of-order packets for later delivery
- When a packets plugs a hole from the beginning
  - The packet can be immediately inspected and forwarded
- If the packet closes a hole completely
  - All the packets in the buffer can be delivered to the byte-analyzer

Defense against attacks to the connection state memory

- Given 512 MB memory, 16M 32-byte connection records can be kept
- If the memory is full, the non-established connections will be evicted first
  - This can be done by randomly selecting a connection and evicting it, if it is not established
  - Under flooding conditions, the table will be heavily loaded with non-established connections

Defense against attacks to the packet buffer

- For straightforward attacks
  - By design each connection is limited to a single hole
  - Per connection buffer is limited
- Overflow the buffer by creating multiple connections each with one hole and limited buffer size
  - It is rare for a single external client host to have multiple connections with holes, concurrently
  - So, allow one connection with a hole / client

What if there is a distributed attack?

- If a deterministic eviction policy is used, the adversary may take advantage of this
- Instead use random eviction
- What are we going to do about the connection?

Eviction and Connection Termination

- If the packet is evicted packet has reached to receiver, without inspection:
  - Potential evasion
  - An adversary first sends out-of-order packets to cause the device to flush without analyzing them
  - The receiver has already received the packets
  - Then fills the sequence hole with packets completing the attacks
  - Is terminating this connections be a good solution?

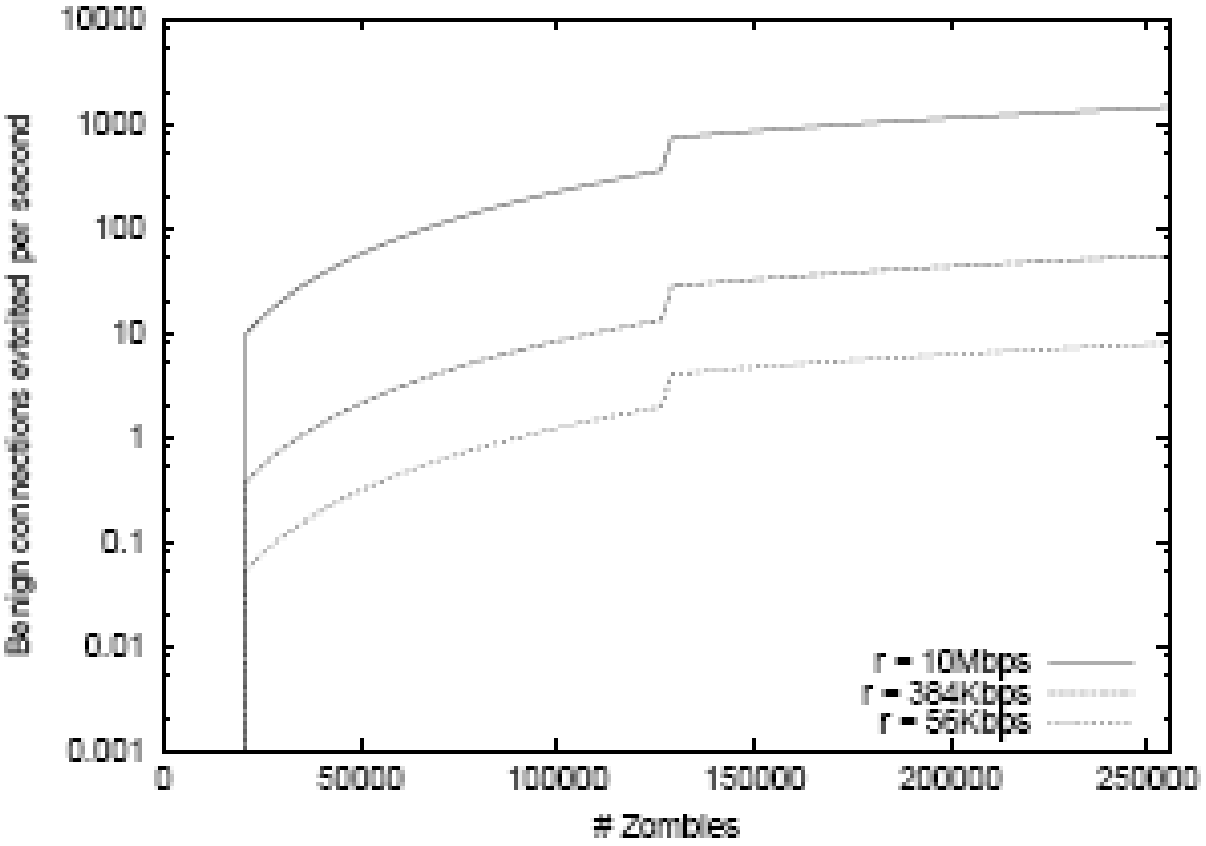
Connection Termination

- Terminating each evicted connection may cause a high-price for legitimate packets
- If upon buffering an out-of-order packet if the we do not also forward it to the receiver, then we do not need to terminate the connection upon eviction
- So, we force the sender to retransmit the out-of-order packets
- The multi-path problem still holds

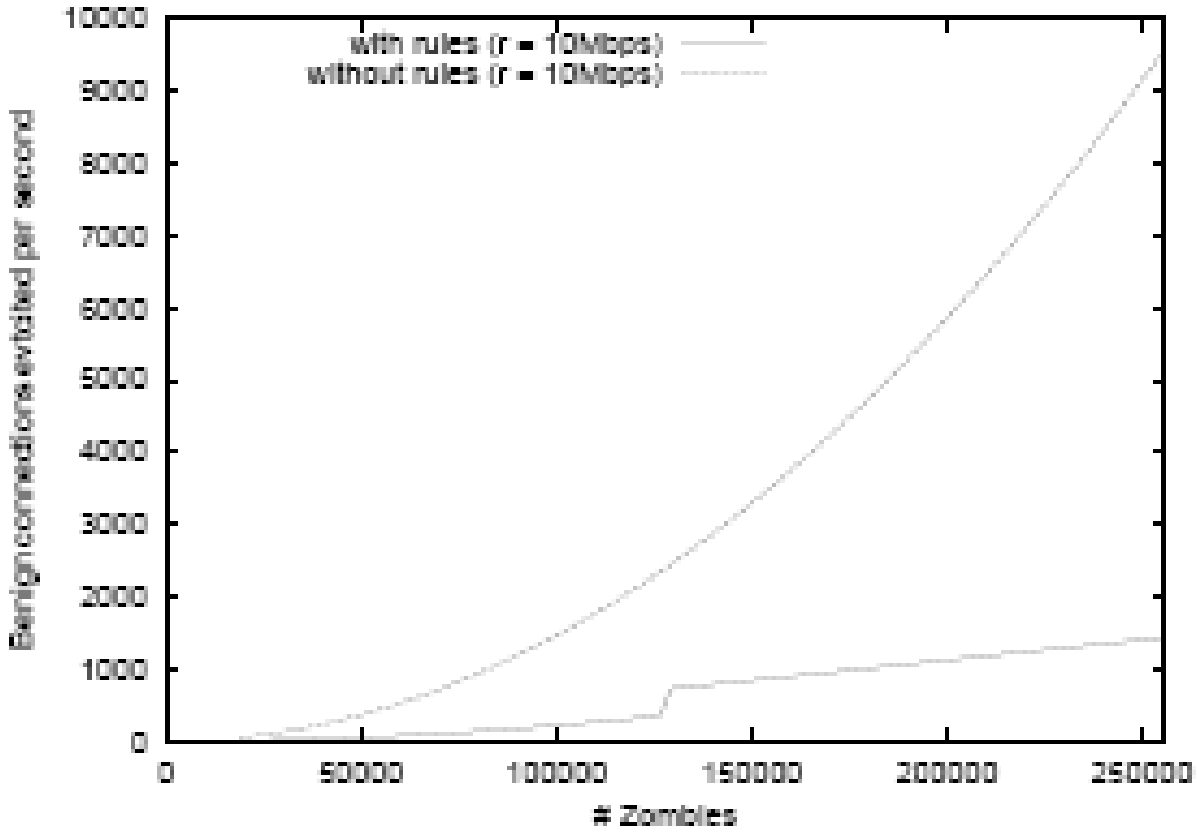
Connection Termination (cont'd)

- How will "not forwarding" affect performance?
- Triggering a duplicate-ACK that results in fast retransmission requires at least 3 duplicate-ACK packets.
- If receiver receives fewer than 3 out-of-order packets, it will not trigger fast retransmission
- Always buffer without forwarding the first 2 out-of-order packets
- When third packet arrives release all packets
- Before the third packet arrives evicting the packets does not require performance degradation

Benign Connection Eviction



Connection Eviction



Conclusion

- The authors characterized out-of-sequence packet behavior
- They show that most common case is short-lived one hole in one direction of the connection
- They designed a robust system against adversaries against TCP Reassembly

Q & A

Thank you

Backup Slides

Case of missing packet due to alternate path routing

- Missing packet will never arrive to the reassembly system
- Its ACK can be seen going in the opposite direction
- Close the hole if such ACK is seen
- Possible evasion!