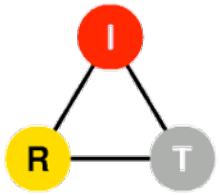
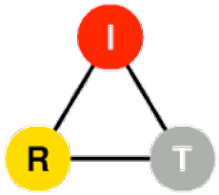


The Impact of SCTP on SIP Server Scalability and Performance

Kumiko Ono and Henning Schulzrinne

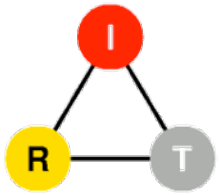
{kumiko, hgs}@cs.columbia.edu

CS@CU



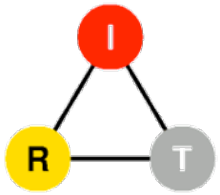
Outline

- What is SCTP?
- How does choosing SCTP as a transport protocol for SIP impact the SIP server?
  - Scalability
    - Number of sustainable SCTP associations compared with the number of TCP connections
  - Performance
    - Setup and transaction response times compared with TCP



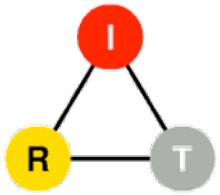
What is SCTP?

- Stream Control Transmission Protocol
  - RFC 4960 in 2007 (RFC 2960 in 2000)
  - Originally designed for carrying SS7 (Signaling Systems No.7) over IP
    - Reliability
    - Congestion control
    - Multi-streaming
    - Multi-homing
  - An alternative transport protocol for SIP
    - TCP, UDP or SCTP (RFC 4168 in 2005)



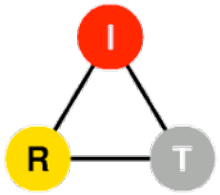
Comparison of transport protocols

	UDP	TCP	SCTP
Connection-oriented	No	Yes: create with a three-way handshake, and terminate with half-close. → SYN flooding attacks	Yes: create with a four-way handshake. No half-close in termination. → Resist SYN/INIT flooding attacks using cookies
large message >MTU	No: lean on IP fragmentation	Yes: segmentation	Yes: segmentation
Reliability	No: lean on App. features	Yes: support ack. , T.O., and re-transmission	Yes: support ack. , T.O., and re-transmission
Congestion control	No	Yes	Yes
Flow control	No	Yes	Yes
Message-oriented	Yes: preserve the boundary	No: byte-stream	Yes: preserve the boundary
Multi-homing	No	No	Yes: failover tolerant
Multi-streaming	No	No	Minimize head-of-line blocking



Outline

- What is SCTP?
- How does choosing SCTP as a transport protocol for SIP impact the SIP server?
  - Scalability
    - Number of sustainable SCTP associations compared with the number of TCP connections
  - Performance
    - Setup and transaction response times compared with TCP



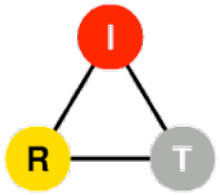
How does choosing SCTP impact SIP servers?

SCTP features

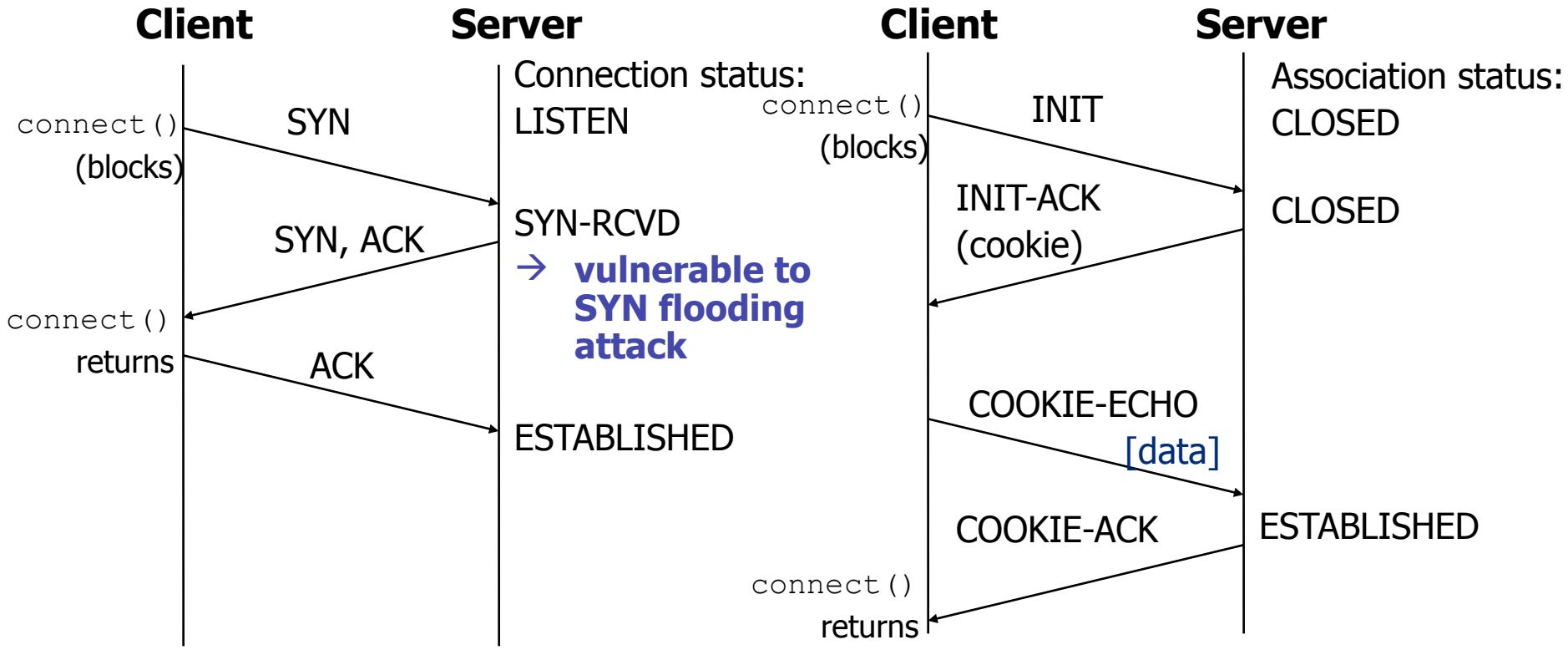
- Resist SYN flooding attacks
  - by cookies in the four-way handshake

Expected impacts

☹ More RTTs cause longer setup time, but **the piggyback setup option** could mitigate it.

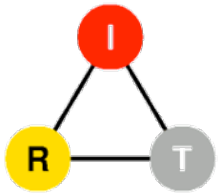


TCP vs. SCTP: Handshake to initiate a connection/association



TCP three-way handshake

SCTP four-way handshake [w/piggyback setup option]



How does choosing SCTP impact SIP servers?

SCTP features

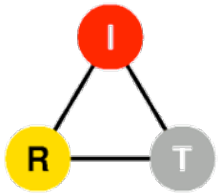
- Resist SYN flooding attacks
  - by cookies in the four-way handshake
- Minimizing HOL blocking
  - by multi-streaming
- Failover tolerance
  - by multi-homing
- Easier parsing
  - by preserving message boundaries

Expected impacts

☹ More RTTs cause longer setup time, but **the piggyback setup option** could mitigate that.

☹ More data structures make it less scalable, but **one-to-many style sockets** could mitigate that.

☺ Shorter transaction time

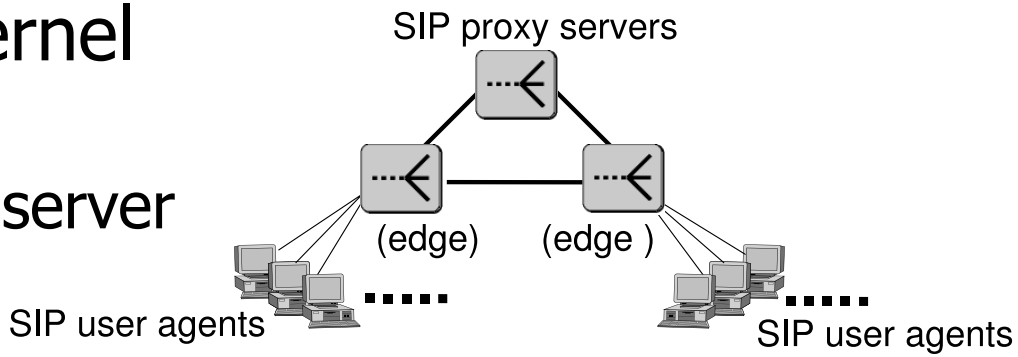


Outline

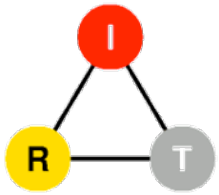
- What is SCTP?
- How does choosing SCTP as a transport protocol for SIP impact the SIP server?
  - Scalability
    - Number of sustainable SCTP associations compared with the number of TCP connections
  - Performance
    - Setup and transaction response times compared with TCP

SCTP measurement: Scalability

- Background:
  - available as a kernel module in Linux
    - can use btwn a server and clients

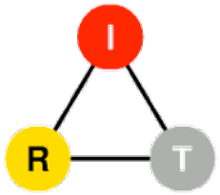


- Goals:
- to establish an upper limit of concurrent associations
  - SCTP one-to-one socket: TCP-like
  - SCTP one-to-many socket: UDP-like
- to clarify the effect of SCTP one-to-many sockets



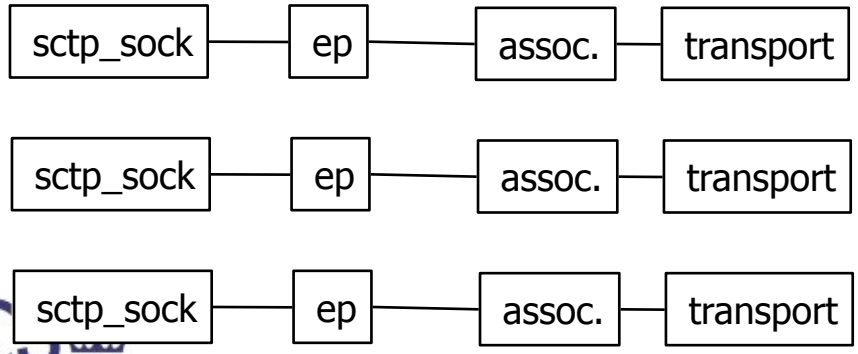
Measurement environment

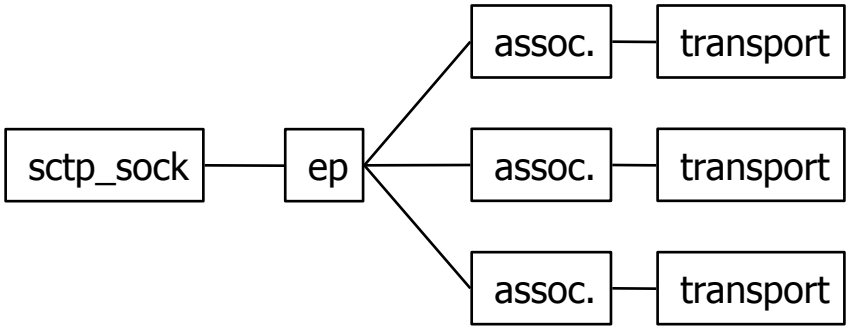
- Server: an echo server
  - CPU: Pentium IV, 3GHz (dual core) 32-bit
  - RAM: 4GB
  - OS: Linux 2.6.23 (default VM split, 1G/3G)
- Clients:
  - CPU: Pentium IV, 3GHz 32-bit
  - RAM: 1GB
  - OS: Redhat Linux 2.6.9
  - SCTP
    - enable a kernel module for SCTP
    - enable SCTP object count



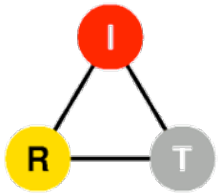
Echo server measurement: Number of sustainable assoc. for SCTP

- one-to-one socket
  - TCP-like
    - 1 socket : 1 assoc.
- one-to-many socket
  - UDP-like
    - 1 socket : N assoc.



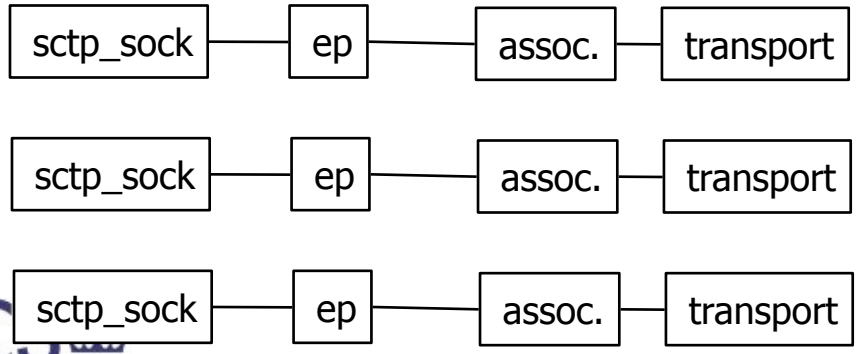


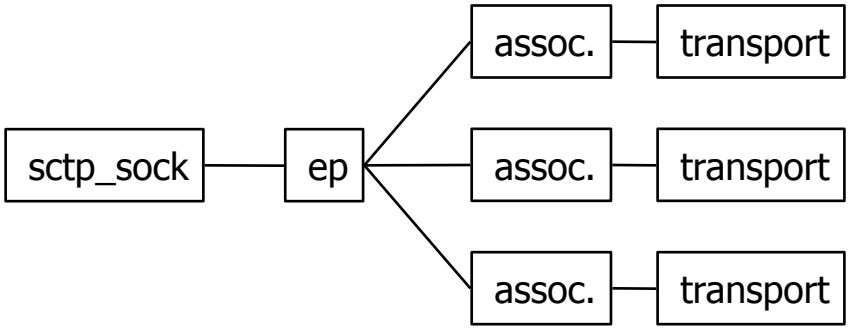
Overviews of the data structures for 3 SCTP associations



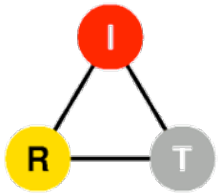
Echo server measurement: Number of sustainable assoc. for SCTP

- one-to-one socket
  - TCP-like
    - 1 socket : 1 assoc.
  - Upper limit
    - 74,000 assoc.
    - 11.1 KB/assoc
    - Ends by out-of-memory
    - [Ref] TCP connections: 419,000
- one-to-many socket
  - UDP-like
    - 1 socket : N assoc.
  - Upper limit
    - 90,000 assoc.
    - 8.9 KB/assoc
    - Ends by out-of-memory



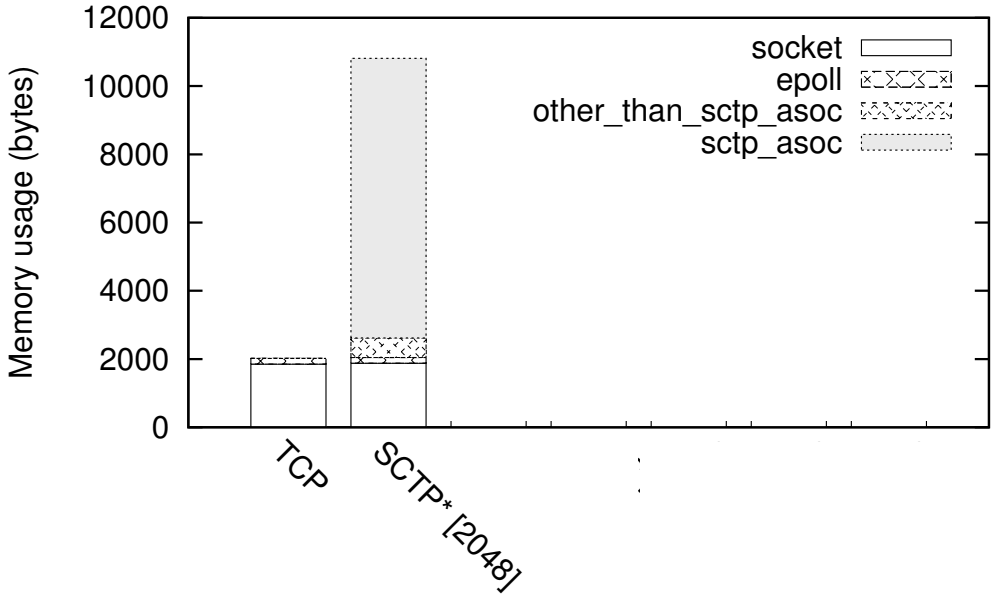


Overviews of the data structures for 3 SCTP associations



Detailed data structures: SCTP vs. TCP

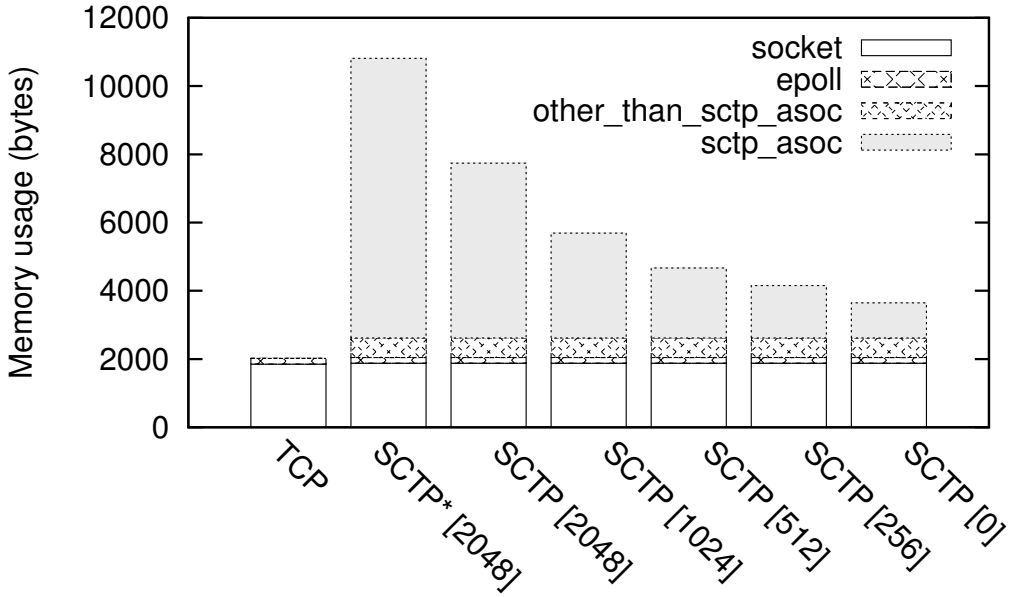
- The `sctp_association` data structure
  - dominates the memory usage of a SCTP socket: 5,120 bytes, but allocated at size-8192 slab object
  - The dominant sub member is `tsn_map` to trace received TSNs for unordered data delivery.



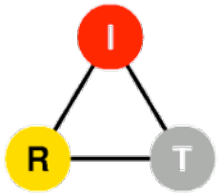
Protocol [SCTP_TSN_MAP_SIZE] *allocated from general purpose slab

Detailed data structures: SCTP vs. TCP

- The `sctp_association` data structure
  - dominates the memory usage of a SCTP socket: 5,120 bytes, but allocated at size-8192 slab object
  - The dominant sub member is `tsn_map` to trace received TSNs for unordered data delivery.



Protocol [SCTP_TSN_MAP_SIZE] *allocated from general purpose slab



How does choosing SCTP impact SIP servers?: Results

SCTP features

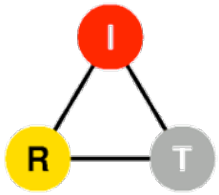
- Resist SYN flooding attack
  - by cookies in the four-way handshake
- Minimizing HOL blocking
  - by multi-streaming
- Failover tolerance
  - by multi-homing
- Easier parsing
  - by preserving message boundarie

Expected impacts

☹ More RTTs cause longer setup time, but **the piggyback setup option** could mitigate it.

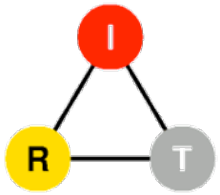
☹ More data structures make it less scalable, but **one-to-many style sockets** could mitigate it.

- Number of sustainable associations
  - One-to-one sockets: 74,000 (17%)
  - One-to-many sockets: 90,000 (21%)
  - [Ref] TCP connections: 419,000
- Improvable to up to 50% of TCP by adjusting the size of the `tsn_map`



Outline

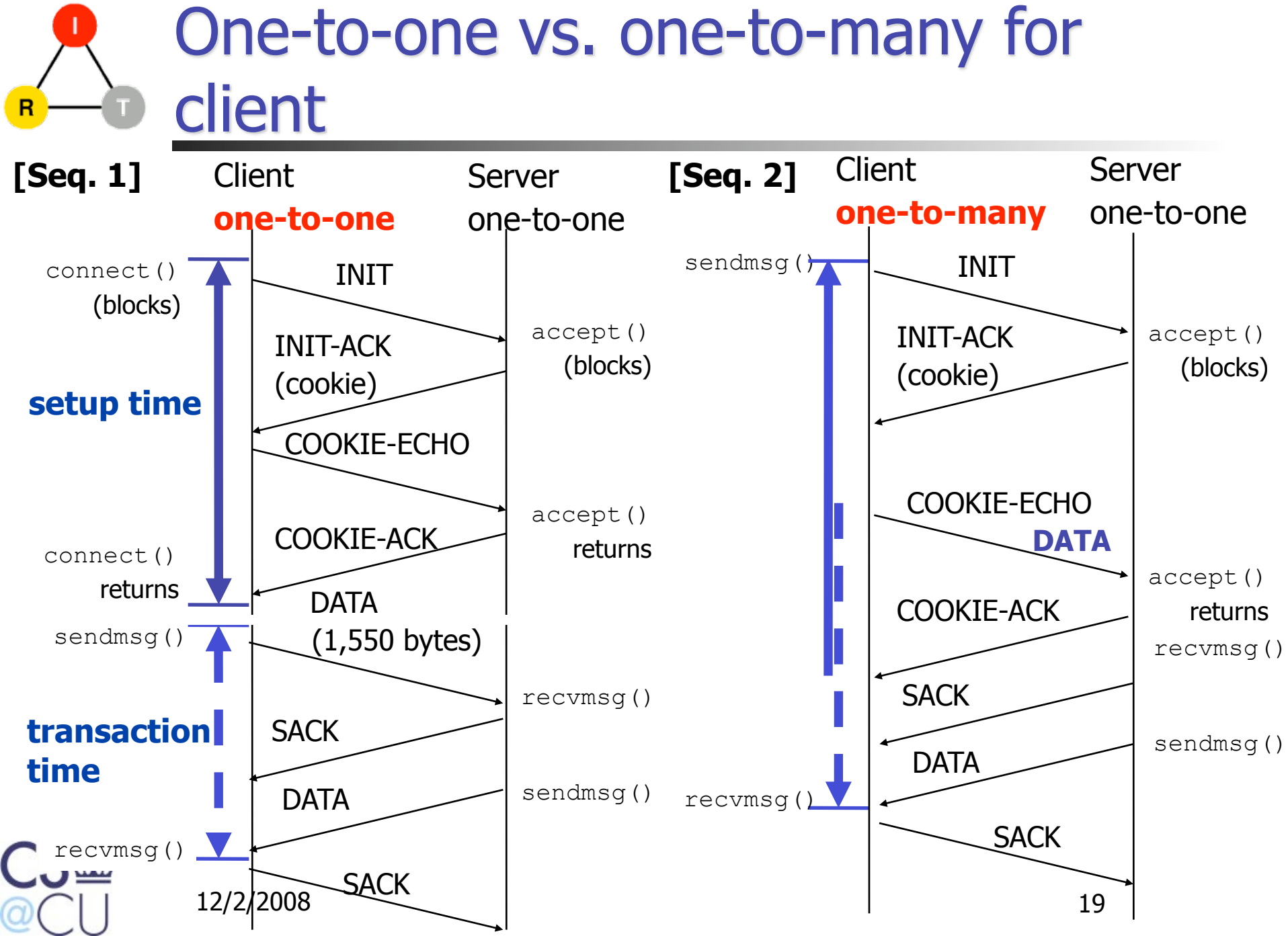
- What is SCTP?
- How does choosing SCTP as a transport protocol for SIP impact the SIP server?
  - Scalability
    - Number of sustainable SCTP associations compared with the number of TCP connections
  - Performance
    - Setup and transaction response times compared with TCP

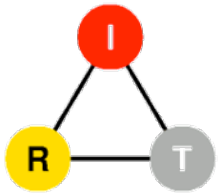


SCTP measurement: Performance

- Goals:
  - to clarify the effect of SCTP piggyback setup option
    - using an echo server
    - measuring the setup and transaction times
  - to clarify the effect of message-orientation
    - using a SIP front-end server, which focusing on message parsing

One-to-one vs. one-to-many for client

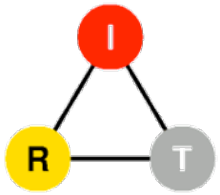




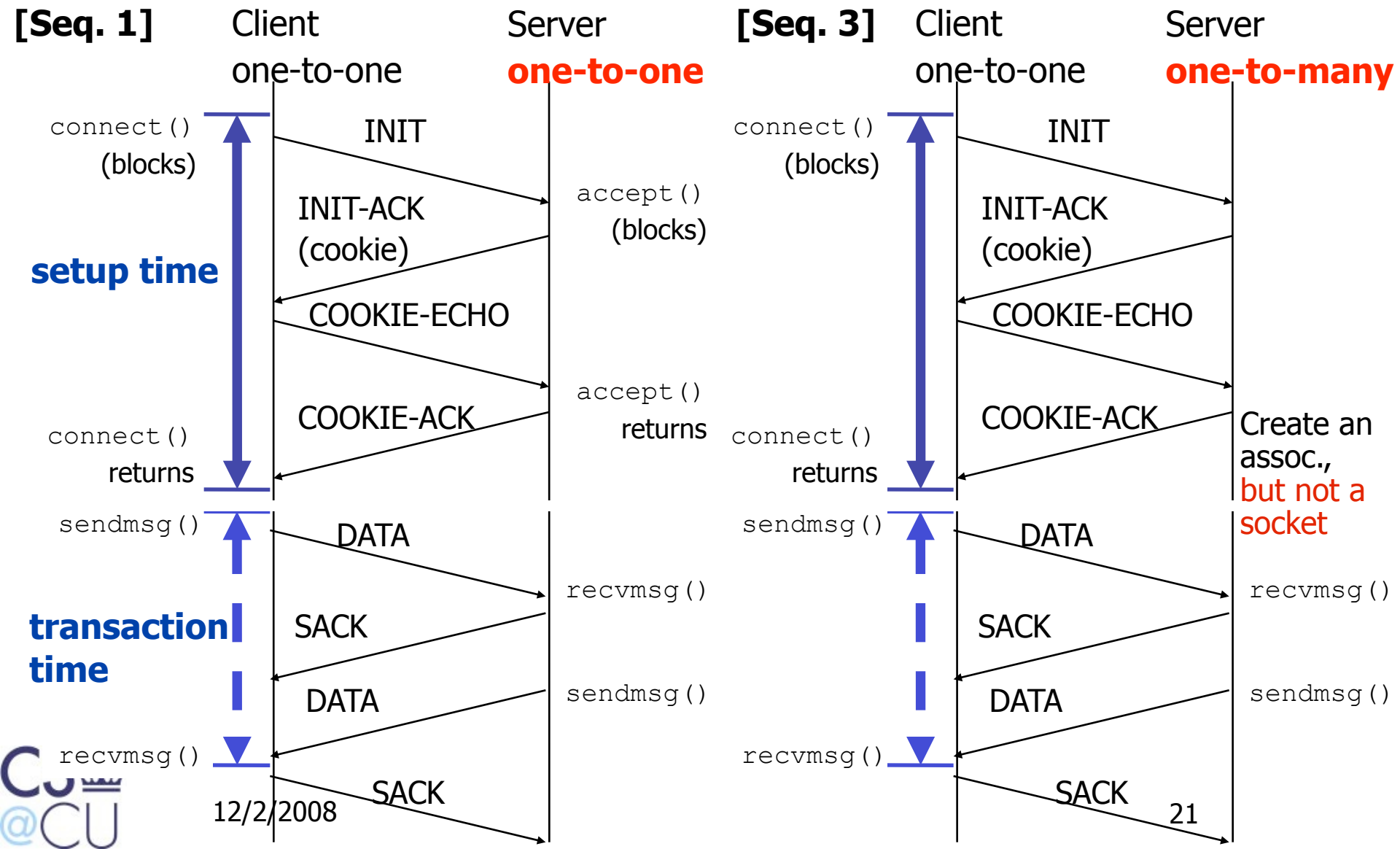
Results of setup and transaction time using echo server

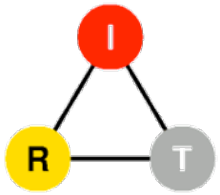
- SCTP one-to-one vs. TCP
  - Longer setup delay for SCTP by 0.23 ms than TCP
  - Piggyback setup of SCTP can slightly mitigate the delay in our environment.
    - Expensive cookie handling
  - Similar transaction time

Socket style at server	Setup type	Setup (ms)	Transaction (ms)	Total (ms)
SCTP one-to-one	regular	0.34	0.54	0.88
	piggyback	0.84		0.84
TCP		0.17	0.48	0.65



One-to-one vs. one-to-many for server

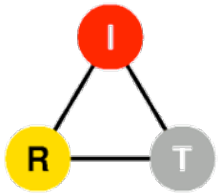




Results of setup and transaction time using echo server

- SCTP one-to-many sockets
  - Setup and transaction times do not remain constant, but linearly increase with the number of maintaining associations.

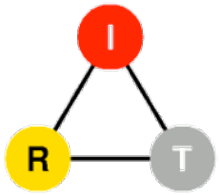
Socket style at server	Setup type	Setup (ms)	Transaction (ms)	Total (ms)
SCTP one-to-one	regular	0.34	0.54	0.88
	piggyback	0.84		0.84
SCTP one-to-many		0.38-170.91	0.65 – 34.14	1.03-205.05
TCP		0.17	0.48	0.65



Results of echo server measurement: Setup and transaction times

- SCTP one-to-many sockets
  - Setup and transaction times do not remain constant, but linearly increase with the number of maintaining associations.
    - Caused by linear search for an association corresponding to an endpoint.
    - Improvable by using a hash table lookup

Socket style at server	Setup type	Setup (ms)	Transaction (ms)	Total (ms)
SCTP one-to-one	regular	0.34	0.54	0.88
	piggyback	0.84		0.84
SCTP one-to-many		0.34	0.53	0.87
TCP		0.17	0.48	0.65



How does choosing SCTP impact SIP servers?: Results

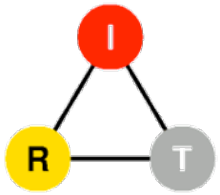
SCTP features

- Resist SYN flooding attack
  - by cookies in the four-way handshake
- Minimizing HOL blocking
  - by multi-streaming
- Failover tolerance
  - by multi-homing
- Easier parsing
  - by preserving message boundari

Expected impacts

☹ More RTTs cause longer setup time, but **the piggyback setup option** could mitigate it.

- The effect of piggyback setup is slight.
  - Smaller RTT is effective, but depends strongly on network conditions.
  - Expensive cookie handling
- Maintaining associations has no significant impact for SCTP after replacing a linear search with a hash table lookup.



Conclusion

- Using SCTP impacts scalability rather than performance.
  - Recommend to use the one-to-many sockets
    - Number of sustainable connections
      - 17-21% of TCP in the default configuration
      - up to 50% by adjusting the tsn_map size
  - Recommend to use the piggyback setup
    - Setup delay is longer than TCP by 0.17 ms, but would be useful in a wide area network.
- Need to mature implementation on Linux