An Extension to the Selective Acknowledgement (SACK) Option for TCP

draft-floyd-sack-00.txt

Sally Floyd, Jamshid Mahdavi, Matt Mathis,
Matthew Podolsky, and Allyn Romanow

IETF, Transport Area Working Group
November 1999

Outline of presentation:

- Review of RFC 2018, the SACK Option for TCP,
- Motivations for an extension to SACK.
- Details of the D-SACK (duplicate-SACK) extension.
- Uses of the D-SACK block.

Review of RFC 2018, the SACK Option for TCP:

- The TCP sender and receiver negotiate SACK capability.

- The TCP receiver uses the SACK option to acknowledge blocks of data not covered by the Cumulative Acknowledgement field.

- RFC 2018: ”The first SACK block ... MUST specify the contiguous block of data containing the segment which triggered this ACK, unless that segment advanced the Acknowledgment Number field in the header.”

- However, RFC 2018 does not specify the use of the SACK option when duplicate segments of data are received.

Motivations for an extension to SACK:

- The TCP sender could learn the order in which the receiver received packets, including duplicate packets.

- As a result, the TCP sender could detect (some) unnecessary retransmits, ACK loss, and packet replication within the network.

- This could be used to make TCP more robust to reordered packets, ACK loss, packet replication, and/or early retransmit timeouts.

- As a result, TCP could be more robust in environments with:
  - link-level retransmissions;
  - widely-varying roundtrip times;
  - loss on the return (ACK) path;
  - routing mechanisms that result in packet reordering.

The D-SACK extension to the SACK option:

- A D-SACK block reports a duplicate contiguous sequence of data received by the receiver in the most recent packet.

- Each duplicate contiguous sequence of data received is reported at most once.

- The left and right edges of the D-SACK block specify the duplicate contiguous sequence, following the SACK conventions.

The D-SACK extension to the SACK option, cont.:

- If the D-SACK block reports a duplicate contiguous sequence from a (possibly larger) block of data in the receiver's data queue above the cumulative acknowledgement, then the second SACK block in that SACK option should specify that (possibly larger) block of data.
  - E.g., the D-SACK block reports segment 5, and the second SACK block reports segments 5-7.

- Following the SACK blocks described above for reporting duplicate segments, additional SACK blocks can be used for reporting additional blocks of data, as specified in RFC 2018.

Examples of the D-SACK extension to the SACK option:

- The internet draft gives the following examples for using D-SACK:
  - Reporting a duplicate segment.
  - Reporting an out-of-order segment and a duplicate segment.
  - Reporting a duplicate of an out-of-order segment.
  - Reporting partial duplicate segments.

Interpreting the D-SACK block at the sender:

- Look at the first SACK block:

  – If the first SACK block is covered by the Cumulative Acknowledgement field, then it is a D-SACK block, and is reporting duplicate data.

  – Else, if the first SACK block is covered by the second SACK block, then the first SACK block is a D-SACK block, and is reporting duplicate data.

- Otherwise, interpret the SACK blocks using the normal SACK procedures.

Identifying a retransmit timeout due to ACK loss:

- This can be conclusively identified as an unnecessary retransmit time-out with D-SACK. This can not be so identified without D-SACK.

```
Transmitted     Received     ACK Sent
Segment         Segment      (Including SACK Blocks)

500-999         500-999      1000 (ACK dropped)
1000-1499       1000-1499    1500 (ACK dropped)
1500-1999       1500-1999    2000 (ACK dropped)
2000-2499       2000-2499    2500 (ACK dropped)
(timeout)
500-999         500-999      2500, SACK=500-1000
                                        --------
```

Identifying an early retransmission timeout:

- This can be conclusively identified as an unnecessary retransmit with D-SACK.

- Without D-SACK, *if* the sender received all of the duplicate acknowledgements sent by the receiver, and none were piggy-backed on data packets, then the sender could determine that *either* some data or ACK packet had been replicated in the network, or that an unnecessary retransmit timeout had occurred.

```
Transmitted    Received     ACK Sent
Segment        Segment      (Including SACK Blocks)

500-999        (delayed)
1000-1499      (delayed)
1500-1999      (delayed)
2000-2499      (delayed)
(timeout)
500-999        (delayed)
               500-999      1000
1000-1499      (delayed)
               1000-1499    1500
...
               1500-1999    2000
               2000-2499    2500
               500-999      2500, SACK=500-1000
                                       --------
               1000-1499    2500, SACK=1000-1500
                                       ---------
               ...
```

Identifying a false retransmit due to reordering:

• With D-SACK, the TCP sender knows that either the retransmitted packet was duplicated in the network, or the packet was unnecessarily transmitted.

• Without D-SACK, the TCP sender knows that either *some* packet was duplicated in the network, the ACK was duplicated in the network, or the retransmitted packet was unnecessarily transmitted.

Identifying a false retransmit due to reordering, cont.:

```
Transmitted     Received      ACK Sent
Segment         Segment       (Including SACK Blocks)

500-999         500-999       1000
1000-1499       (delayed)
1500-1999       1500-1999     1000, SACK=1500-2000
2000-2499       2000-2499     1000, SACK=1500-2500
2500-2999       2500-2999     1000, SACK=1500-3000
1000-1499       1000-1499     3000
                1000-1499     3000, SACK=1000-1500
                                         ---------
```

Identifying data packet replication in the network:

• Without D-SACK, the TCP sender can not distinguish between the replication of a data packet and the replication of an ACK packet.

```
Transmitted     Received     ACK Sent
Segment         Segment      (Including SACK Blocks)

500-999         500-999      1000
1000-1499       1000-1499    1500
                (replicated)
                1000-1499    1500, SACK=1000-1500
                                        ---------
```