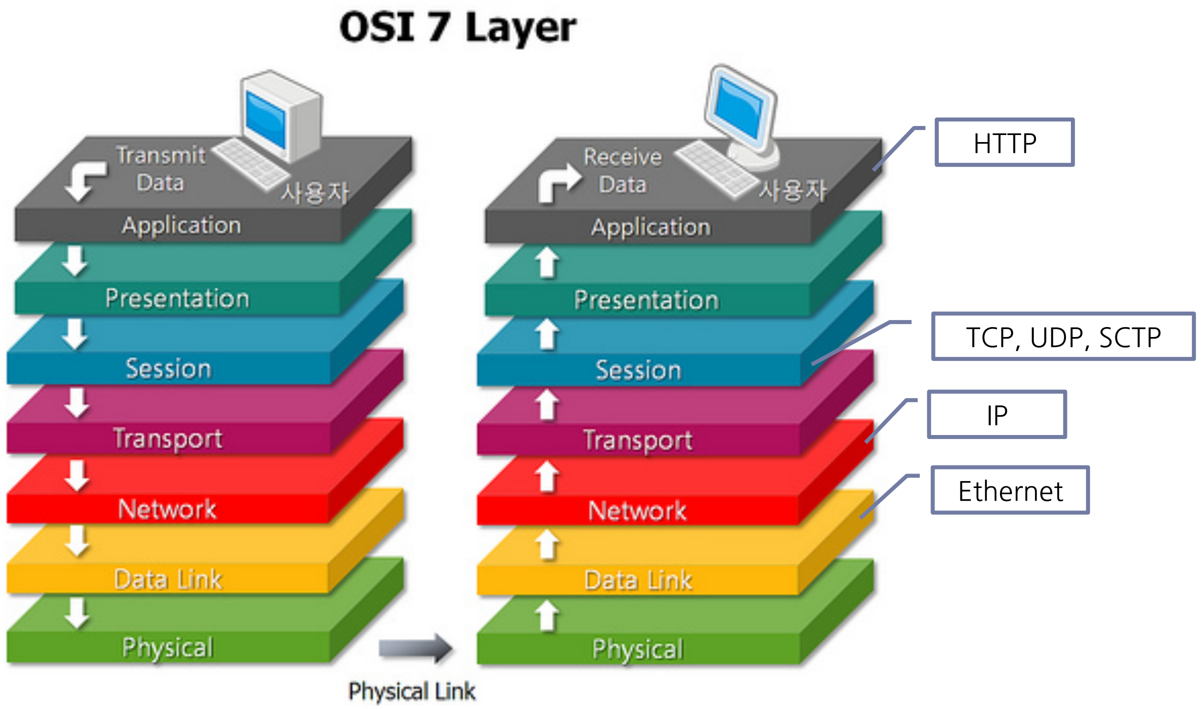
SCTP
(Stream Control Transmission Protocol)

조영일

목차

소개

- IP 기반의 transport layer protocol
- Stack

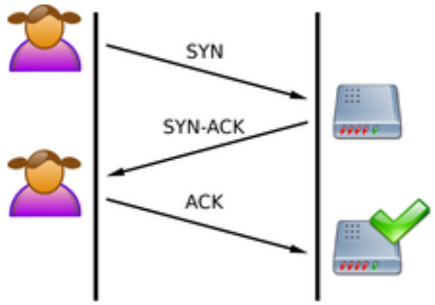


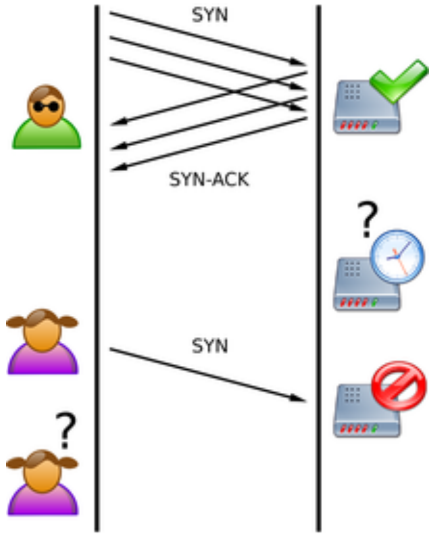
TCP's problems

- head-of-line blocking
  - 많은 application들이 reliable하기를 바라지만 sequence를 보장할 필요까지는 없는데 TCP는 sequence를 중요하게 여김
  - 일부 데이터가 누락되면 재전송이 완료될 때까지 후속 데이터가 대기해야 함
- unnecessary stream-oriented nature
  - 스트림 중에서 메시지를 분해하기 위해 마크를 달아야 함
- complicated setup with highly available transfer using multiple network paths
  - NIC도 여러 개, IP도 여러 개, 연결된 네트워도 여러 개일 때 TCP는 활용하기 복잡함
- vulnerable to DoS attack
  - SYN flooding

TCP's problems

- SYN Flooding Attack



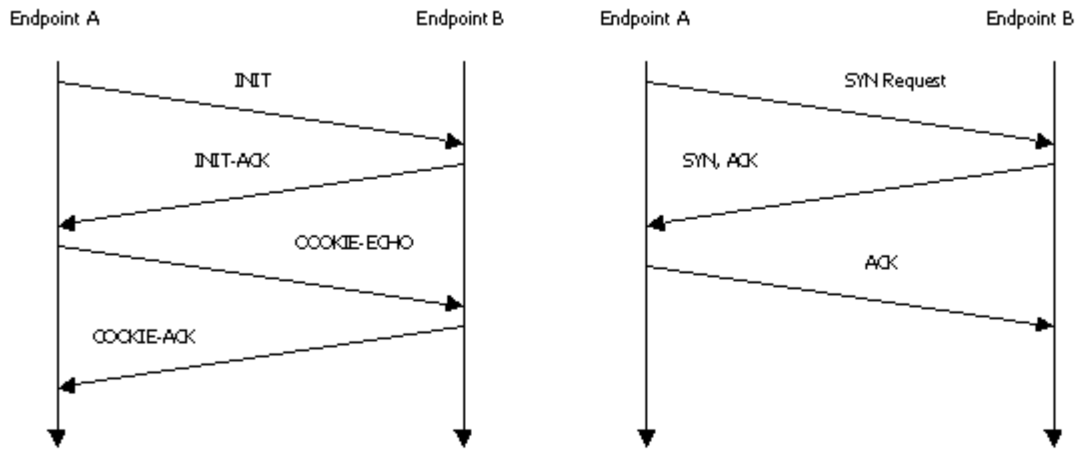


특징

- message-oriented like UDP
- reliable, in-sequence, congestion control like TCP
  - optional message ordering
- session-oriented
- multi-homing
- multi-streaming
  - TCP는 connection 개념만 존재
  - SCTP는 association 안에 여러 stream이 존재
- selective ACK
  - 이미 TCP에도 SACK에 대한 지원이 되고 있음

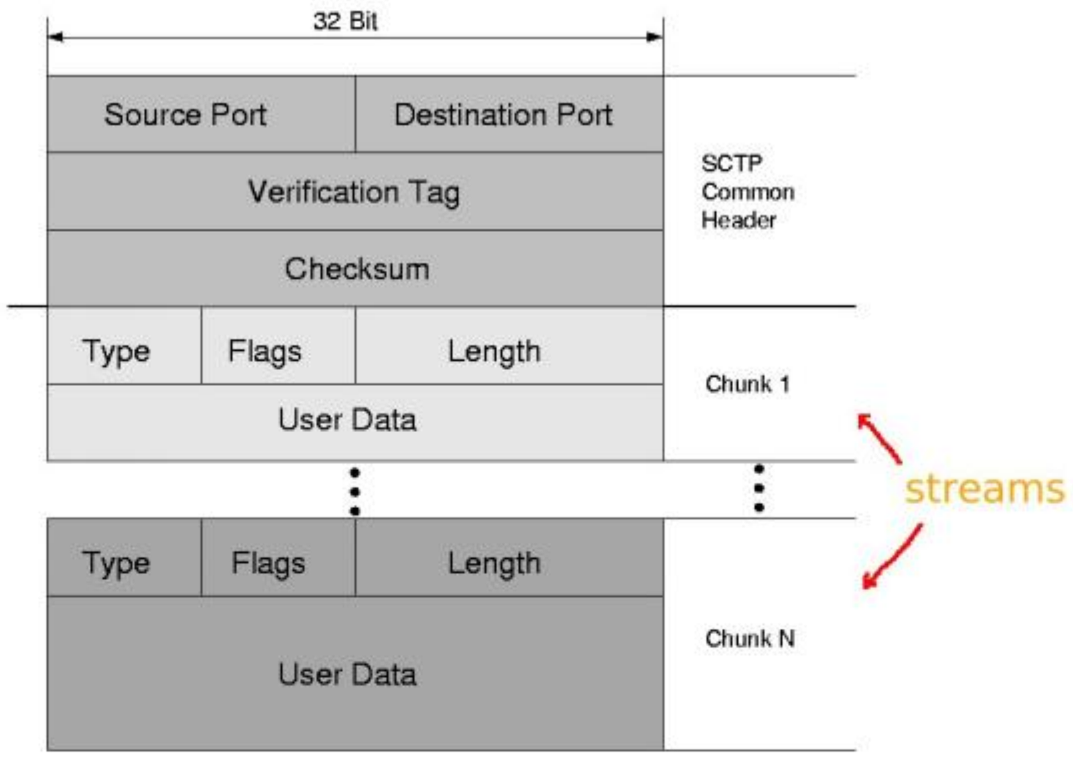
장점

- gets more throughput over TCP
  - by concurrent streams
- eliminates TCP's head-of-line blocking problem
  - stream이 분리되어 있으므로
- security against SYN-flooding attack



- strong resilience
  - when some routes are down

Message Structure



Message Structure

ID	Chunk Type
0	payload data
1	initiation
2	initiation acknowledgement
3	selective acknowledgement
4	heartbeat request
5	heartbeat acknowledgement
6	abort
7	shutdown
8	shutdown acknowledgement
9	operation error
10	state cookie
11	cookie acknowledgement
12	reserved for explicit congestion notification echo
13	reserved for congestion window reduced
14	shutdown complete
	…

Socket API

- com.sun.nio.sctp
  - .MessageInfo
  - .SctpChannel
  - .SctpServerChannel

예제 코드

- Daytime Server

```
public class DaytimeServer {
    static int SERVER_PORT = 3456;
    static int US_STREAM = 0;
    static int FR_STREAM = 1;

    static SimpleDateFormat USformatter = new SimpleDateFormat("h:mm:ss a EEE d MMM yy, zzzz",
                                                                  Locale.US);
    static SimpleDateFormat FRformatter = new SimpleDateFormat("h:mm:ss a EEE d MMM yy, zzzz",
                                                                  Locale.FRENCH);

    public static void main(String[] args) throws IOException {
        SctpServerChannel ssc = SctpServerChannel.open();
        InetSocketAddress serverAddr = new InetSocketAddress(SERVER_PORT);
        ssc.bind(serverAddr);

        ByteBuffer buf = ByteBuffer.allocateDirect(60);
        CharBuffer cbuf = CharBuffer.allocate(60);
        Charset charset = Charset.forName("ISO-8859-1");
        CharsetEncoder encoder = charset.newEncoder();
```

```
while (true) {
    SctpChannel sc = ssc.accept();

    /* get the current date */
    Date today = new Date();
    cbuf.put(USformatter.format(today)).flip();
    encoder.encode(cbuf, buf, true);
    buf.flip();

    /* send the message on the US stream */
    MessageInfo messageInfo = MessageInfo.createOutgoing(null, US_STREAM);
    sc.send(buf, messageInfo);

    /* update the buffer with French format */
    cbuf.clear();
    cbuf.put(FRformatter.format(today)).flip();
    buf.clear();
    encoder.encode(cbuf, buf, true);
    buf.flip();

    /* send the message on the French stream */
    messageInfo.streamNumber(FR_STREAM);
    sc.send(buf, messageInfo);

    cbuf.clear();
    buf.clear();

    sc.close();
}
```

예제 코드

- Daytime Client

```
public class DaytimeClient {
    static int SERVER_PORT = 3456;
    static int US_STREAM = 0;
    static int FR_STREAM = 1;

    public static void main(String[] args) throws IOException {
        InetSocketAddress serverAddr = new InetSocketAddress("localhost", SERVER_PORT);
        ByteBuffer buf = ByteBuffer.allocateDirect(60);
        Charset charset = Charset.forName("ISO-8859-1");
        CharsetDecoder decoder = charset.newDecoder();

        SctpChannel sc = SctpChannel.open(serverAddr, 0, 0);

        /* handler to keep track of association setup and termination */
        AssociationHandler assocHandler = new AssociationHandler();

         /* expect two messages and two notifications */
        MessageInfo messageInfo = null;
        do {
            messageInfo = sc.receive(buf, System.out, assocHandler);
            buf.flip();
            if (buf.remaining() > 0 && messageInfo.streamNumber() == US_STREAM) {
                System.out.println("(US) " + decoder.decode(buf).toString());
            } else if (buf.remaining() > 0 && messageInfo.streamNumber() == FR_STREAM) {
                System.out.println("(FR) " +  decoder.decode(buf).toString());
            }
            buf.clear();
        } while (messageInfo != null);

        sc.close();
    }

    static class AssociationHandler extends AbstractNotificationHandler
    {
        public HandlerResult handleNotification(AssociationChangeNotification not,
                                                PrintStream stream) {
            if (not.event().equals(COMM_UP)) {
                int outbound = not.association().maxOutboundStreams();
                int inbound = not.association().maxInboundStreams();
                stream.printf("New association setup with %d outbound streams" +
                              ", and %d inbound streams.\n", outbound, inbound);
            }

            return HandlerResult.CONTINUE;
        }

        public HandlerResult handleNotification(ShutdownNotification not, PrintStream stream) {
            stream.printf("The association has been shutdown.\n");
            return HandlerResult.RETURN;
        }
    }
}
```