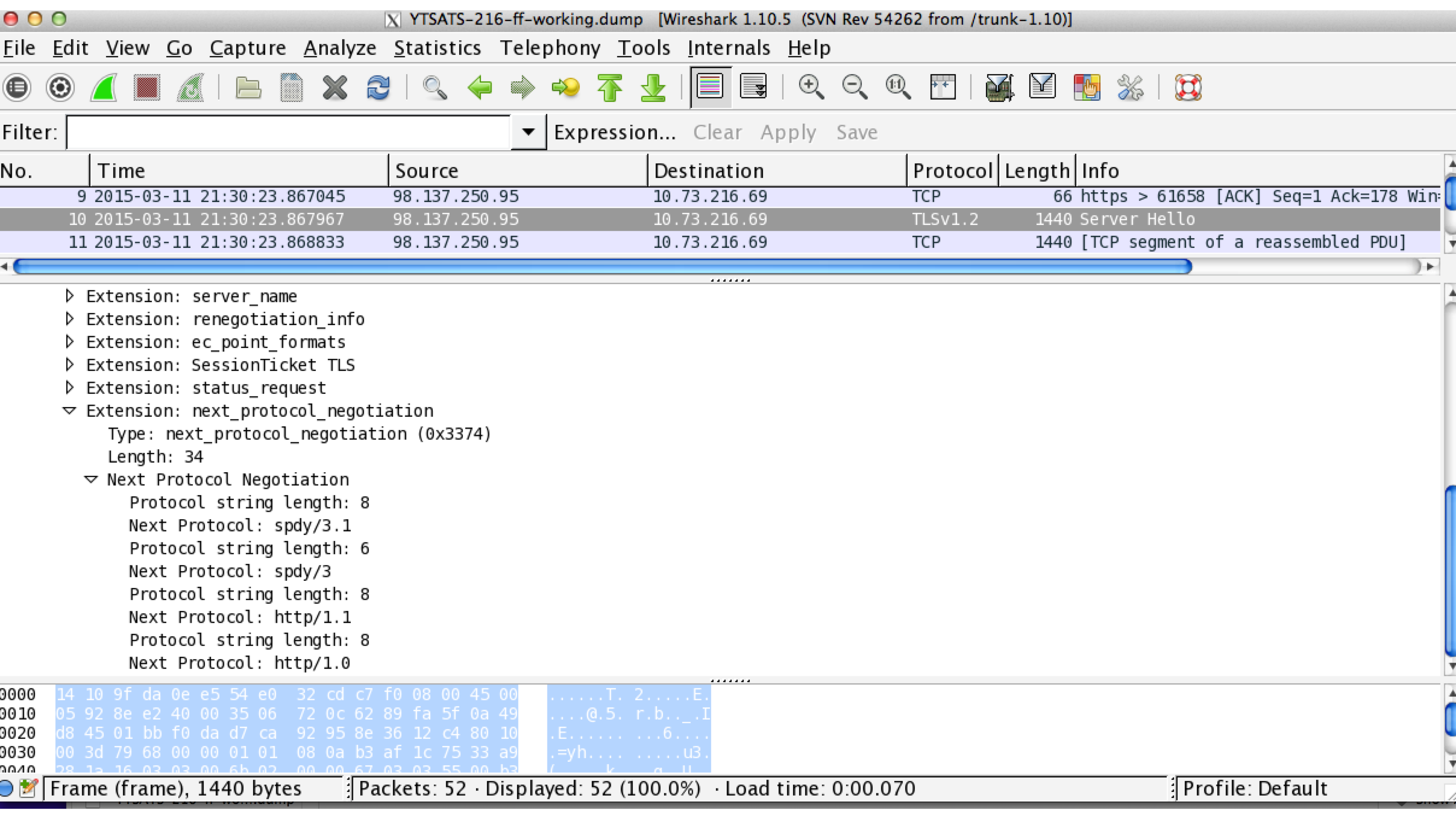
NPN/ALPN Customization

Sudheer Vinukonda

<sudheerv@yahoo-inc.com>

What is NPN/ALPN

- NPN (Next Protocol Negotiation) and ALPN (Application-Layer Protocol Negotiation) are Transport Layer Security (TLS) extensions.
- Allow the application layers to negotiate which protocol should be used over the TLS connection by avoiding additional round trips
- Independent of the application layer protocols.
- NPN used to negotiate SPDY
- ALPN used to select HTTP/2

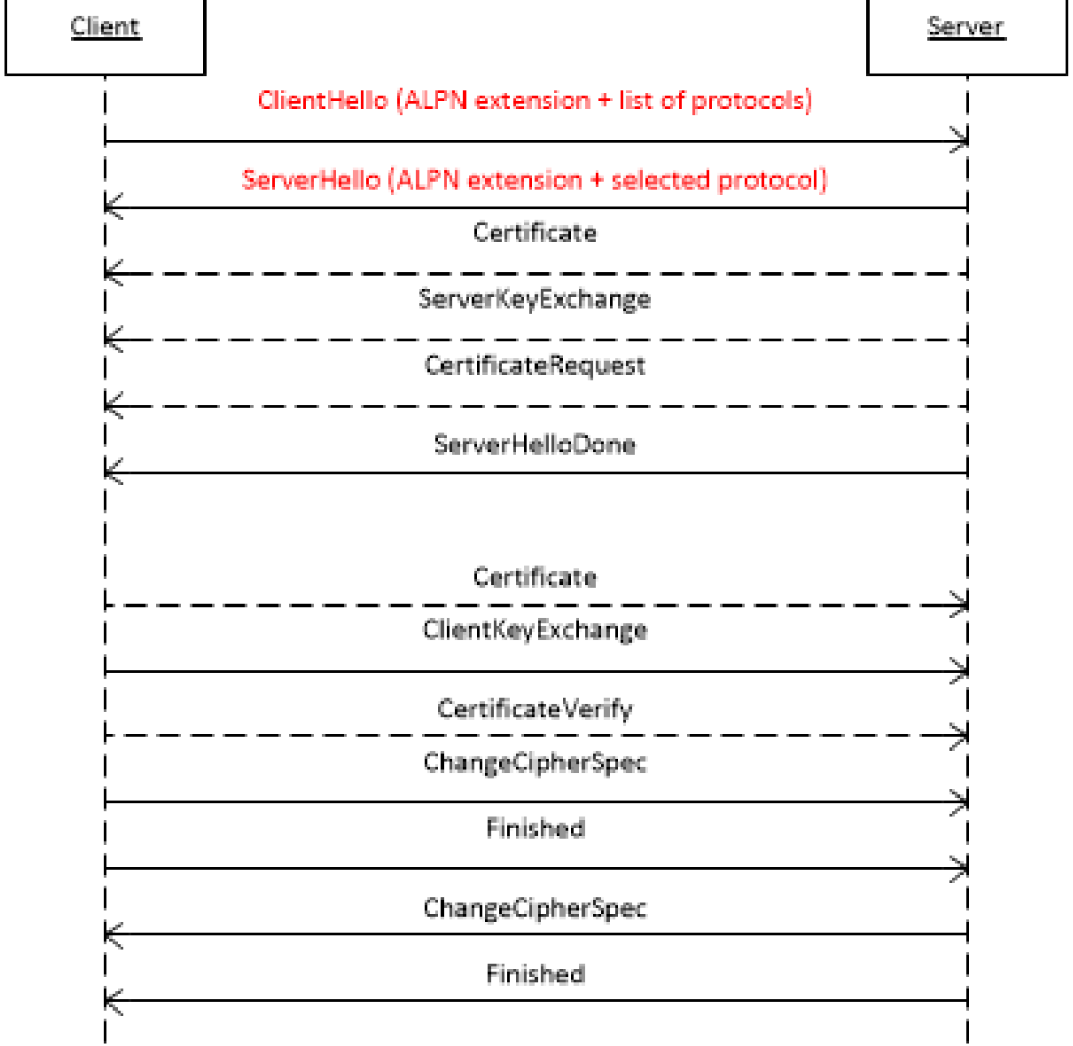

YTSATS-216-ff-working.dump [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time                        Source          Destination    Protocol  Length Info
   9 2015-03-11 21:30:23.867045  98.137.250.95   10.73.216.69   TCP           66 https > 61658 [ACK] Seq=1 Ack=178 Win
  10 2015-03-11 21:30:23.867967  98.137.250.95   10.73.216.69   TLSv1.2     1440 Server Hello
  11 2015-03-11 21:30:23.868833  98.137.250.95   10.73.216.69   TCP         1440 [TCP segment of a reassembled PDU]
```

```
  ▷ Extension: server_name
  ▷ Extension: renegotiation_info
  ▷ Extension: ec_point_formats
  ▷ Extension: SessionTicket TLS
  ▷ Extension: status_request
  ▽ Extension: next_protocol_negotiation
      Type: next_protocol_negotiation (0x3374)
      Length: 34
    ▽ Next Protocol Negotiation
        Protocol string length: 8
        Next Protocol: spdy/3.1
        Protocol string length: 6
        Next Protocol: spdy/3
        Protocol string length: 8
        Next Protocol: http/1.1
        Protocol string length: 8
        Next Protocol: http/1.0
```

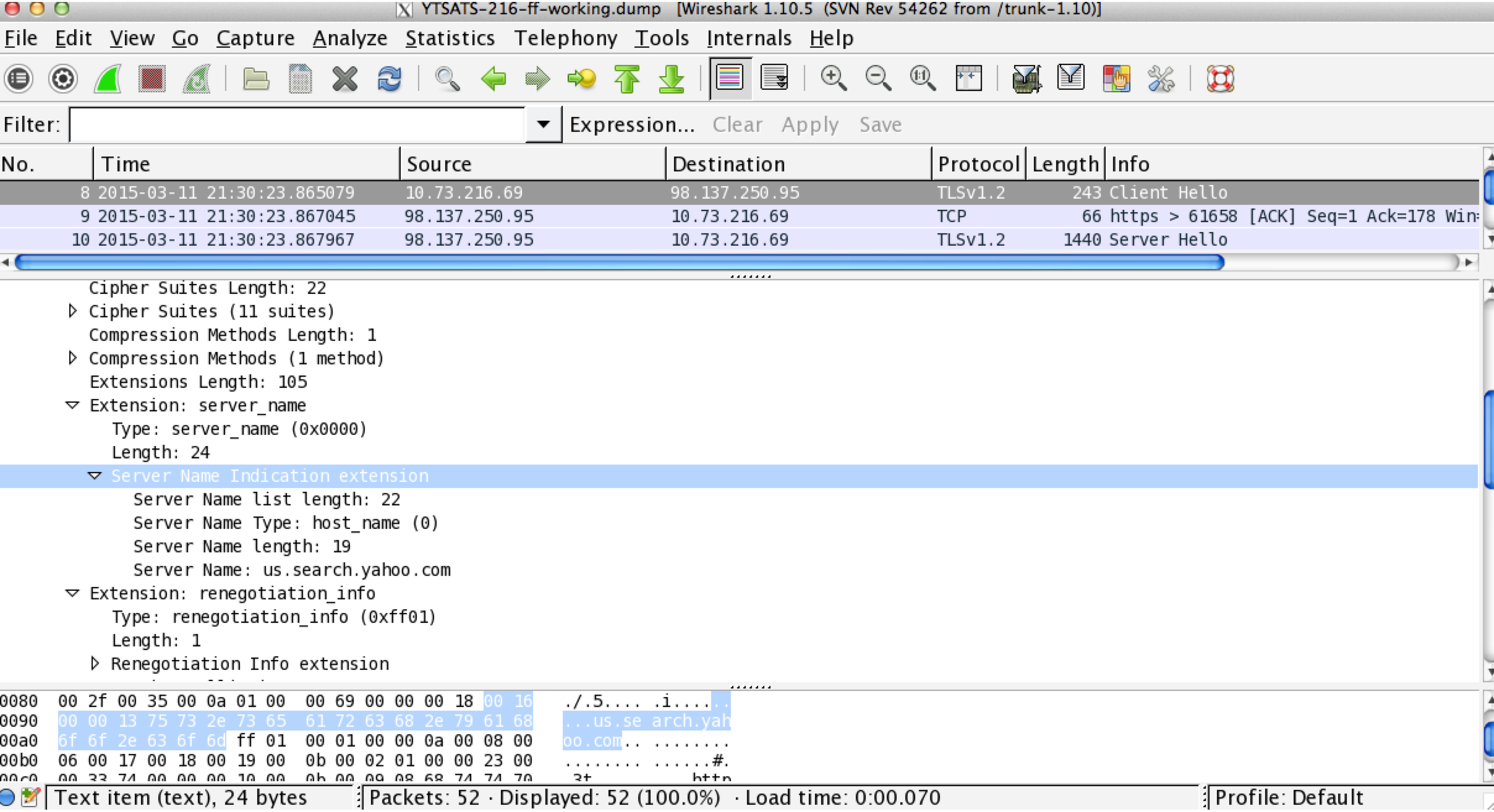
```
0000  14 10 9f da 0e e5 54 e0  32 cd c7 f0 08 00 45 00   ......T. 2.....E.
0010  05 92 8e e2 40 00 35 06  72 0c 62 89 fa 5f 0a 49   ....@.5. r.b.._.I
0020  d8 45 01 bb f0 da d7 ca  92 95 8e 36 12 c4 80 10   .E...... ...6....
0030  00 3d 79 68 00 00 01 01  08 0a b3 af 1c 75 33 a9   .=yh.... .....u3.
0040  28 1a 16 03 03 00 6b 02  00 00 67 03 03 55 00 b3   (.....k. ..g..U..
```

Frame (frame), 1440 bytes | Packets: 52 · Displayed: 52 (100.0%) · Load time: 0:00.070 | Profile: Default


Client
Server
ClientHello (ALPN extension + list of protocols)
ServerHello (ALPN extension + selected protocol)
Certificate
ServerKeyExchange
CertificateRequest
ServerHelloDone
Certificate
ClientKeyExchange
CertificateVerify
ChangeCipherSpec
Finished
ChangeCipherSpec
Finished

Issues in current implementation

- Fixed NPN list advertised per TLS port
- ALPN selects the first server-offered protocol from the advertised list
- Hard to introduce new protocols
- Need to be able to customize NPN list for different domains
- SNI extension from Client-Hello



```
No.  Time                        Source          Destination     Protocol  Length Info
   8 2015-03-11 21:30:23.865079  10.73.216.69    98.137.250.95   TLSv1.2      243 Client Hello
   9 2015-03-11 21:30:23.867045  98.137.250.95   10.73.216.69    TCP           66 https > 61658 [ACK] Seq=1 Ack=178 Win=
  10 2015-03-11 21:30:23.867967  98.137.250.95   10.73.216.69    TLSv1.2     1440 Server Hello
```

```
        Cipher Suites Length: 22
      ▷ Cipher Suites (11 suites)
        Compression Methods Length: 1
      ▷ Compression Methods (1 method)
        Extensions Length: 105
      ▽ Extension: server_name
          Type: server_name (0x0000)
          Length: 24
        ▽ Server Name Indication extension
            Server Name list length: 22
            Server Name Type: host_name (0)
            Server Name length: 19
            Server Name: us.search.yahoo.com
      ▽ Extension: renegotiation_info
          Type: renegotiation_info (0xff01)
          Length: 1
        ▷ Renegotiation Info extension
```

```
0080  00 2f 00 35 00 0a 01 00  00 69 00 00 00 18 00 16   ./.5.... .i......
0090  00 00 13 75 73 2e 73 65  61 72 63 68 2e 79 61 68   ...us.se arch.yah
00a0  6f 6f 2e 63 6f 6d ff 01  00 01 00 00 0a 00 08 00   oo.com.. ........
00b0  06 00 17 00 18 00 19 00  0b 00 02 01 00 00 23 00   ........ ......#.
00c0  00 33 74 00 00 00 10 00  0b 00 09 08 68 74 74 70   .3t..... ....http
```

Text item (text), 24 bytes | Packets: 52 · Displayed: 52 (100.0%) · Load time: 0:00.070 | Profile: Default

Customize the list..

- The knowledge of what protocols are available/registered is in the Acceptor objects created during initialization
- Plugins do not have access to the Acceptor object associated with an incoming TLS connection
- SSLNetVConnection has a npnSet that is fixed per TLS port based on the protocols/endpoints available on that port
- Proposal is to add a pointer to the Acceptor object (base class SessionAccept) in the SSLNetVConnection (netVC)
- Initialize the SesisonAccept pointer in the netVC during Accept

Plugin design proposal

- Plugin allows configuring a custom NPN list based on SNI
- During init, plugin calls a TS API to validate the configured NPN list against each Acceptor object and return the allowed Acceptor objects
- Plugin then maintains a mapping of {SNI, Acceptor} to configured custom list
- When a TLS connection is made and the SNI hook is invoked, the plugin would use the SNI + netVC’s acceptor object to locate the custom list