0-RTT TCP Convert Protocol

draft-ietf-tcpm-converters-01

IETF101, March 2018

O. Bonaventure, M. Boucadair, B. Peirens, S. Seo, A. Nandugudi
Converter

• Initial Motivation
  – More MPTCP enabled clients than MPTCP enabled servers
  – Clients want to benefit from MPTCP at least on a fraction of the end-to-end path
Objectives of the TCP converter

• Aid the deployment of new TCP Extensions
  – Experience shows that Client OSes deploy new TCP extensions earlier than servers OSes
  – Enterprise or service provider networks can deploy Converters

• Converter proxies Client connections
  – Without requiring additional rtts

• Converter informs Client options on server
  – Enables Client to bypass Converter
Simple example

- Multipath TCP use case
Basic principles

• Converter is explicit TCP proxy between client and server

• Client sends commands in TCP bytestream
  – To achieve 0-rtt, proxy commands are exchanged during handshake leveraging TCP Fast Open
  – Commands/responses are encoded in TLV format

• Converter informs Client of the TCP options supported by server to enable bypass
Reaching the server

Converter TLV
TCP info
Network Layer info

@c
@s
	->	@s
SYN
@t->@s
SYN+ACK
	
@t->@c
SYN+ACK
	
@c->@t
SYN (TFO:t) [Connect @s:p]
	
@s->@t
SYN+ACK

TLV message in SYN payload
Detecting if server supports MPTCP

@c->@t
SYN (TFO:t,MPC)
[Connect @s:p]

@t->@c SYN+ACK (MPC(Kc))
[ExtTCPH(MPC(Ks))]

@t->@s SYN(MPC)

@s->@t
SYN+ACK (MPC(Ks))

Copy of the extended TCP header returned by server
Bootstrap: learning converter cookie

@c -> @t SYN (TFO) [Bootstrap]

@t -> @c SYN+ACK (TFO:t) [Supported TCP Ext. (MPTCP)]

TCP Extensions supported by Converter

Empty TFO

Converter cookie
TFO connection through the converter

- @c -> @t SYN (TFO:t)
  [Connect @s:p
   TCPOpt:TFO]

- @t -> @c SYN+ACK
  [ExtTCPH(TFO:sc)]

- @t -> @s SYN(TFO)

- @s -> @t SYN+ACK (TFO:sc)

Client learns server cookie

Empty TFO option

Server cookie: sc
TFO connection through the converter second connection to server

@c->@t SYN (TFO:t)
[Connect @s:p
TCPOpt:TFO:sc]

Data

@t->@s SYN(TFO:sc) Data

@t

@c->@t SYN+ACK [ ]

@s->@t SYN+ACK

Server recognises cookie sc and accepts data
Changes since WG adoption

• Various editorial changes to clarify and simplify text

• Clarification of how standard TCP extensions should be handled by the Converter
Base TCP Options

• The following options cannot be "converted"
  – Kind=0 (End Of Options List)
  – Kind=1 (No-Operation)
  – Kind=2 (Maximum Segment Size)
Window Scale Option

• Kind=3 (Window Scale)
• Converter can advertise its own window scaling, but no benefit from letting a client propose the WScale that a converter should advertise to a remote server
Timestamp, Selective Ack and Multipath TCP

• The following options can be advertised by a Converter
  – Kind=8 (Timestamp)
  – Kind=4 (SACK permitted)
  – Kind=30 (Multipath TCP)

• Kind=5 (SACK) cannot be advertised since it cannot appear in SYN
TCP Fast Open

• Kind=34

• Can be advertised by Converter, requires special support as shown earlier
TCP User Timeout

• Deployment of the TCP option (Kind=28) unclear
  – feedback requested from working group on the benefits of supporting this extension
TCP Authentication Option

• Main objective of this extension seems incompatible in principle with a TCP proxy

• The TCP-AO-NAT extension might be supported, but feedback from WG is requested on the benefits of supporting it
### Experimental TCP extensions

- Not considered in this draft, we suggest that separate drafts discuss the support of these TCP extensions

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0348</td>
<td>HOST_ID</td>
<td>[RFC7974]</td>
</tr>
<tr>
<td>0x0CA0</td>
<td>TCP Capability Option</td>
<td>[draft-boucdair-tcpm-capability-option]</td>
</tr>
<tr>
<td>0x0ED0</td>
<td>Extended Data Offset</td>
<td>[draft-ietf-tcpm-tcp-edo]</td>
</tr>
<tr>
<td>0x454E</td>
<td>TCP-ENO</td>
<td>[draft-ietf-tcpinc-tcpeno]</td>
</tr>
<tr>
<td>0x5323</td>
<td>Service Number</td>
<td>[draft-touch-tcpm-sno]</td>
</tr>
<tr>
<td>0x75CFFEE</td>
<td>Timestamp Interval</td>
<td>[draft-trammell-tcpm-timestamp-interval]</td>
</tr>
<tr>
<td>0xACCE</td>
<td>AccECN Experimental Option</td>
<td>[draft-kuehlewind-tcpm-accurate-ecn]</td>
</tr>
<tr>
<td>0x2D4C3D9</td>
<td>Shared Memory communications over RMDA protocol</td>
<td>[RFC7609]</td>
</tr>
<tr>
<td>0xF989</td>
<td>Fast Open (current and new implementations SHOULD use option 34)</td>
<td>[RFC7413]</td>
</tr>
<tr>
<td>0xF990</td>
<td>Low Latency</td>
<td>[draft-wang-tcpm-low-latency-opt]</td>
</tr>
</tbody>
</table>
Conclusion

• Initial proposal was focussed on the support of Multipath TCP for which there is a clear demand but other TCP extensions could benefit from such a facility
• draft takes into account major comments raised during email discussions
  – Application level protocol
    • Service name/port to be reserved by IANA
  – Provides 0-RTT using TFO
  – Client can bypass converter if server supports extension
• Next steps
  – Improved support for other TCP extensions
  – Feedback from implementors and interoperability tests