0-RTT TCP Convert Protocol

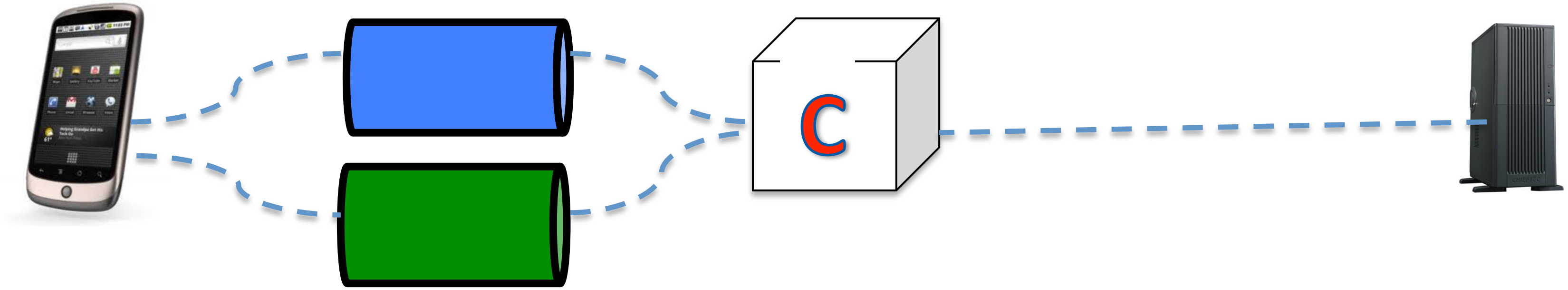
draft-ietf-tcpm-converters-01

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

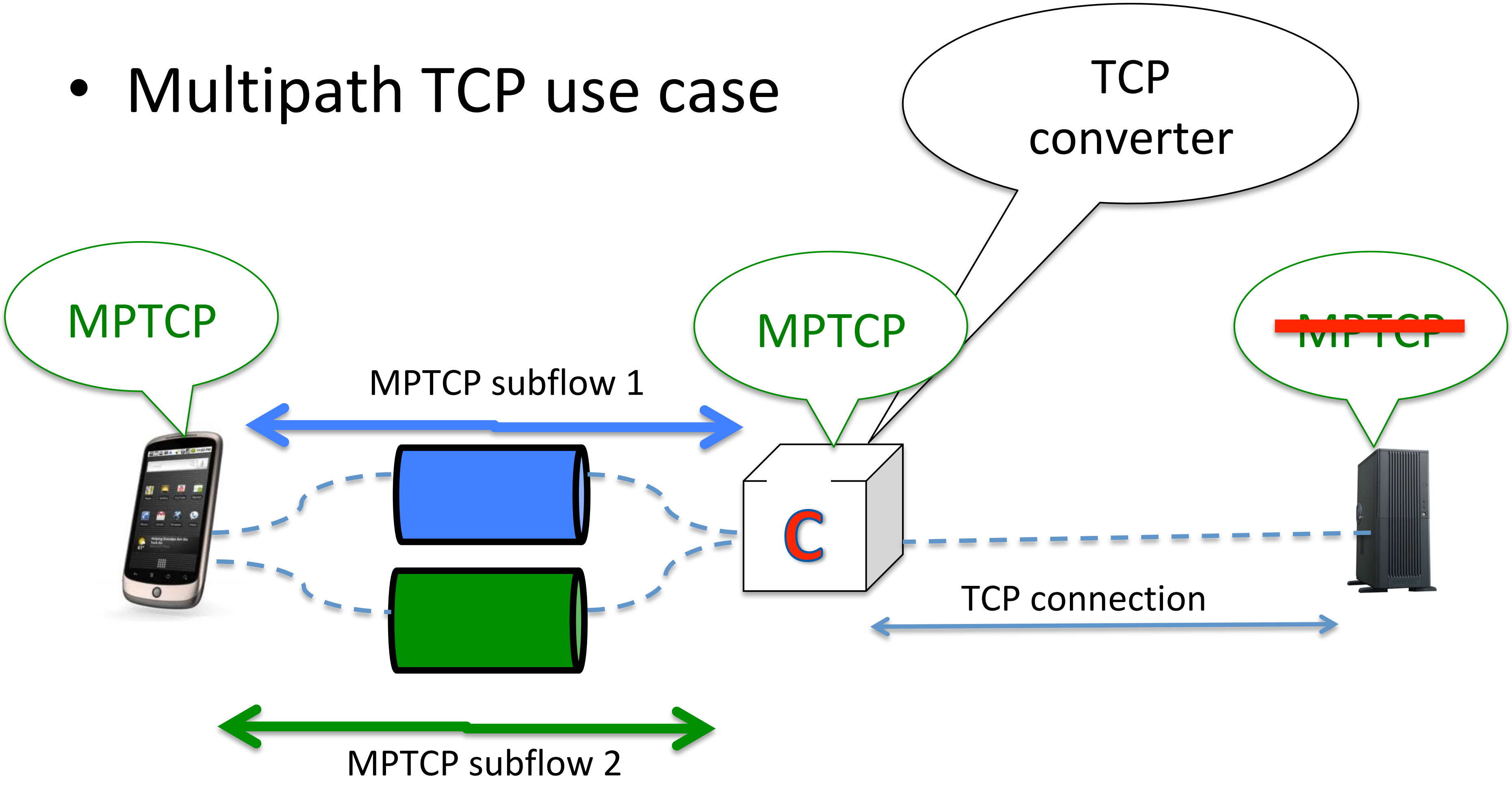


Objectives of the TCP converter

- 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

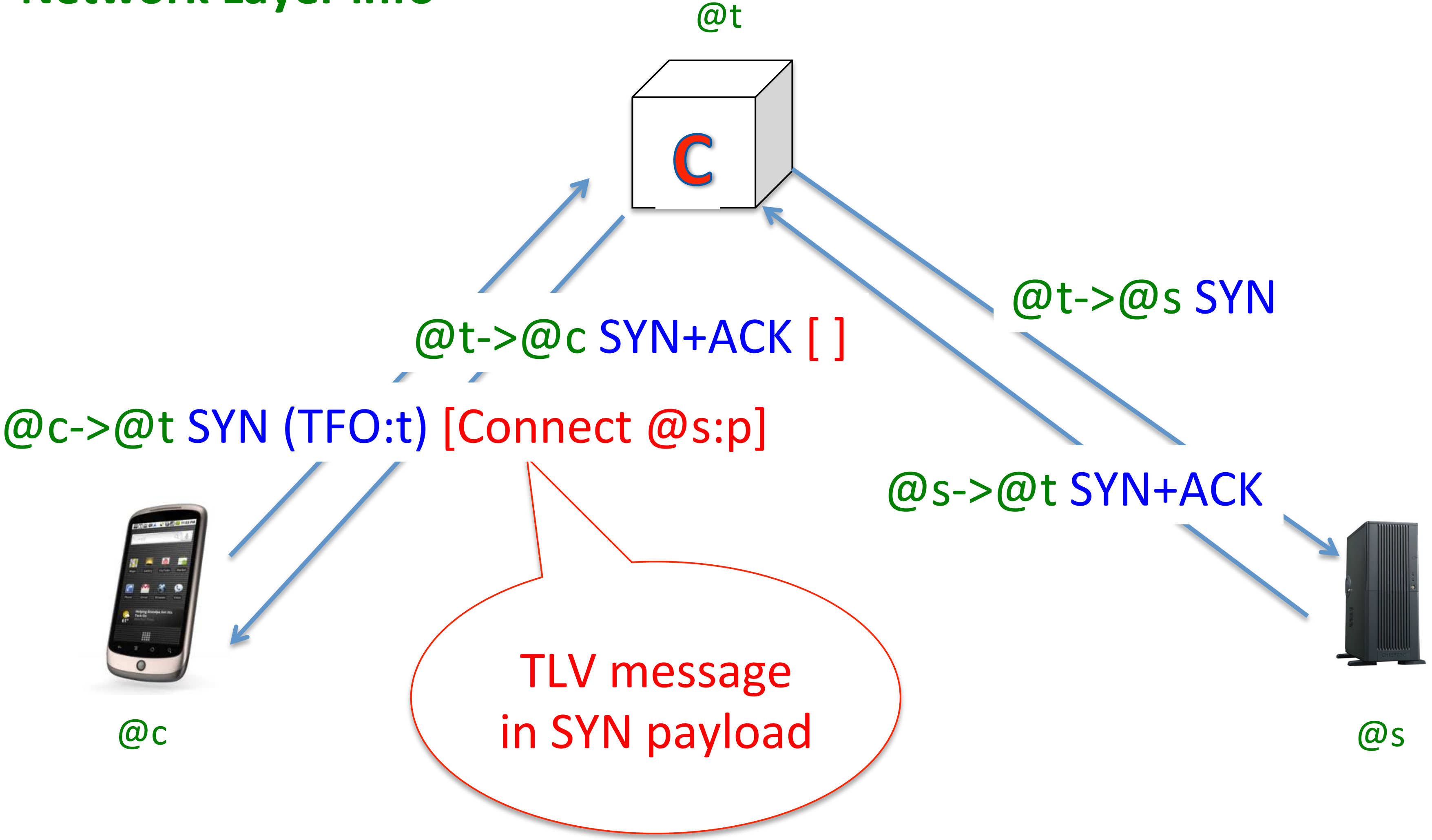
- Converter is explicit TCP proxy between client and server
- Client sends commands in TCP bytestream
  - To achieve 0-rtt, proxy commands are exchanged during handshake leveraging TCP Fast Open
  - Commands/responses are encoded in TLV format
- Converter informs Client of the TCP options supported by server to enable bypass

Reaching the server

Converter TLV

TCP info

Network Layer info



@t

C

@t->@c SYN+ACK [ ]

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

@t->@s SYN

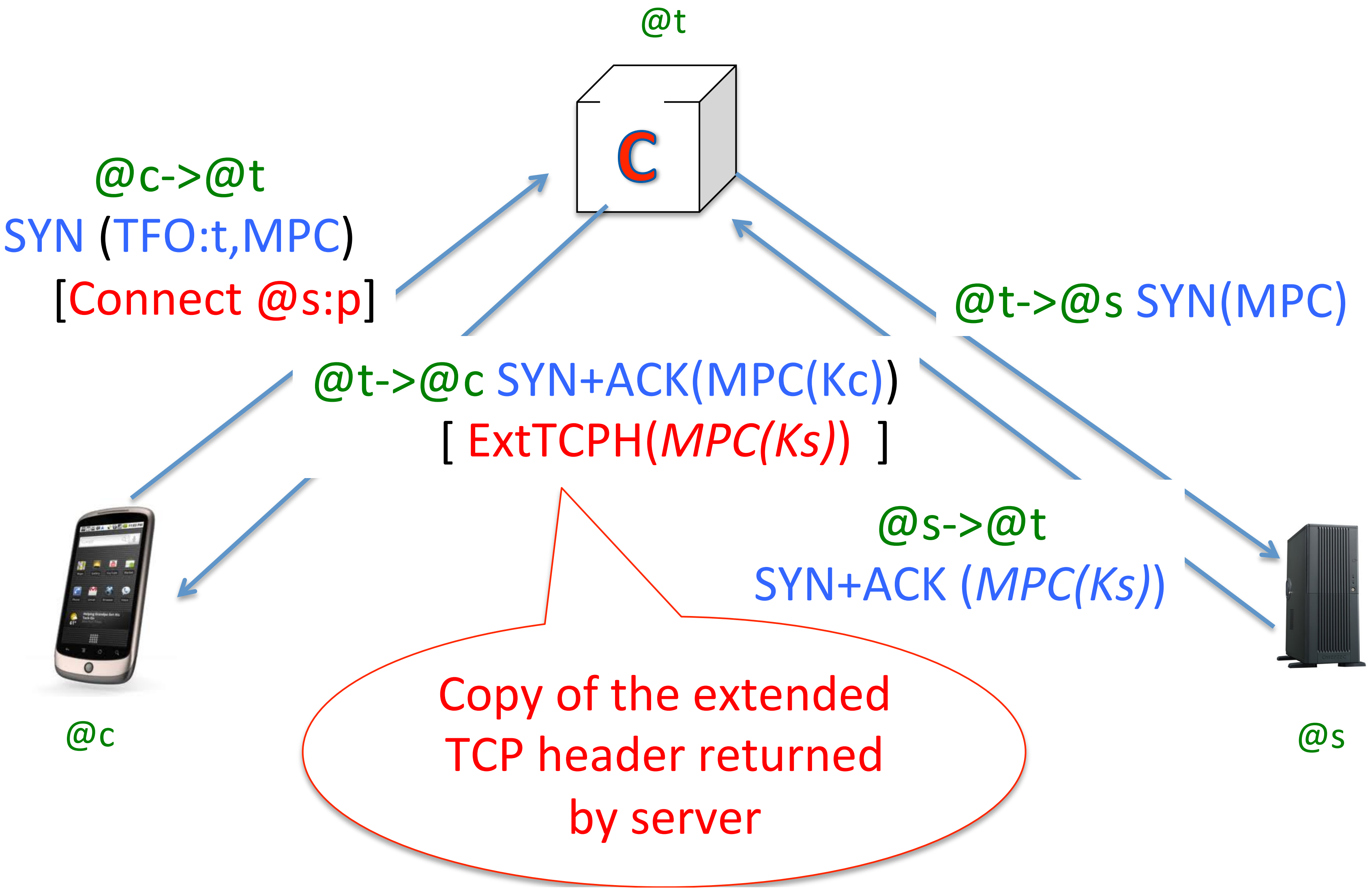
@s->@t SYN+ACK

@c

TLV message in SYN payload

@s

Detecting if server supports MPTCP



@t

C

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

@t->@s SYN(MPC)

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

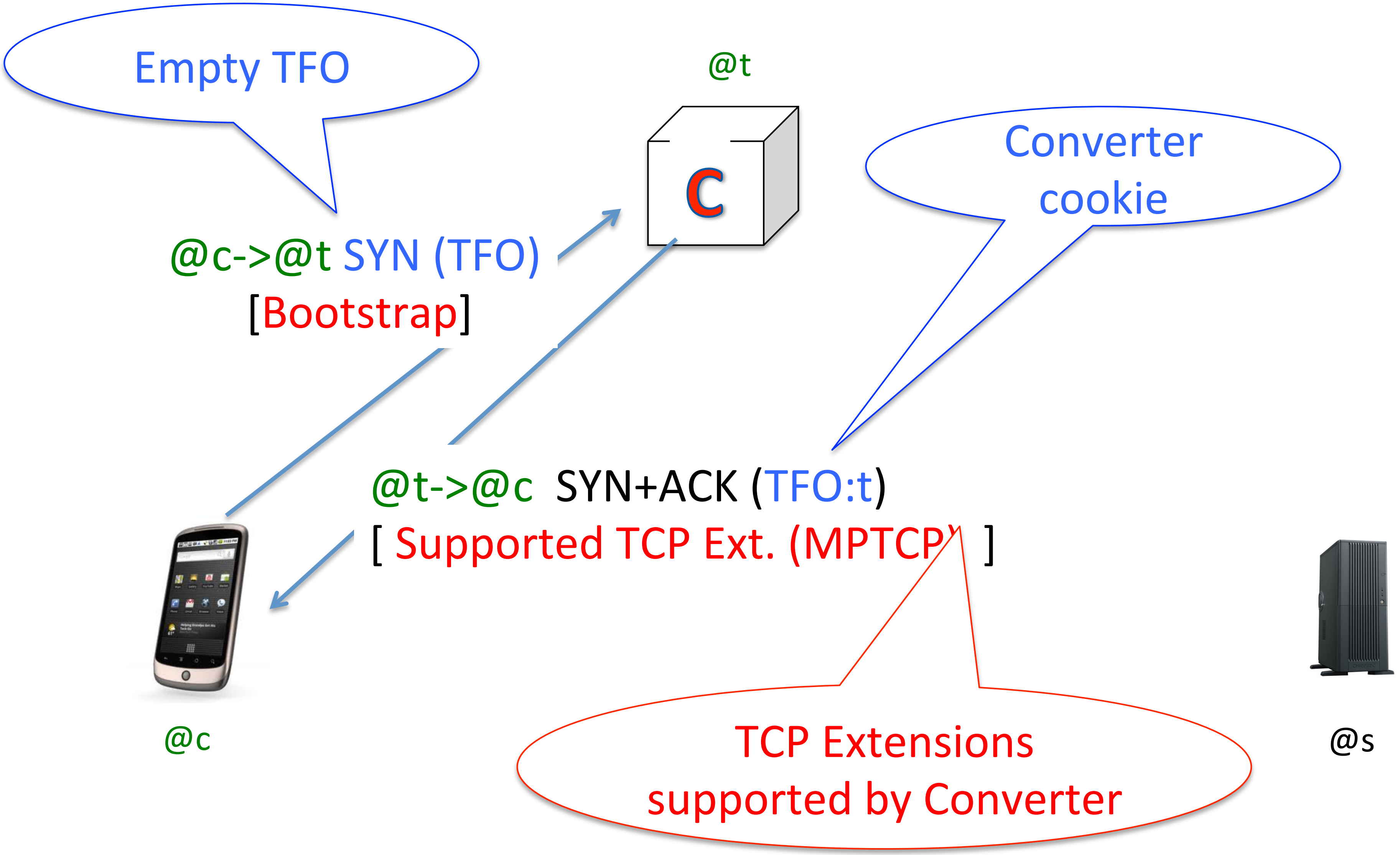
@s->@t
SYN+ACK (*MPC(Ks)*)

@c

@s

Copy of the extended TCP header returned by server

Bootstrap : learning converter cookie



Empty TFO

@t

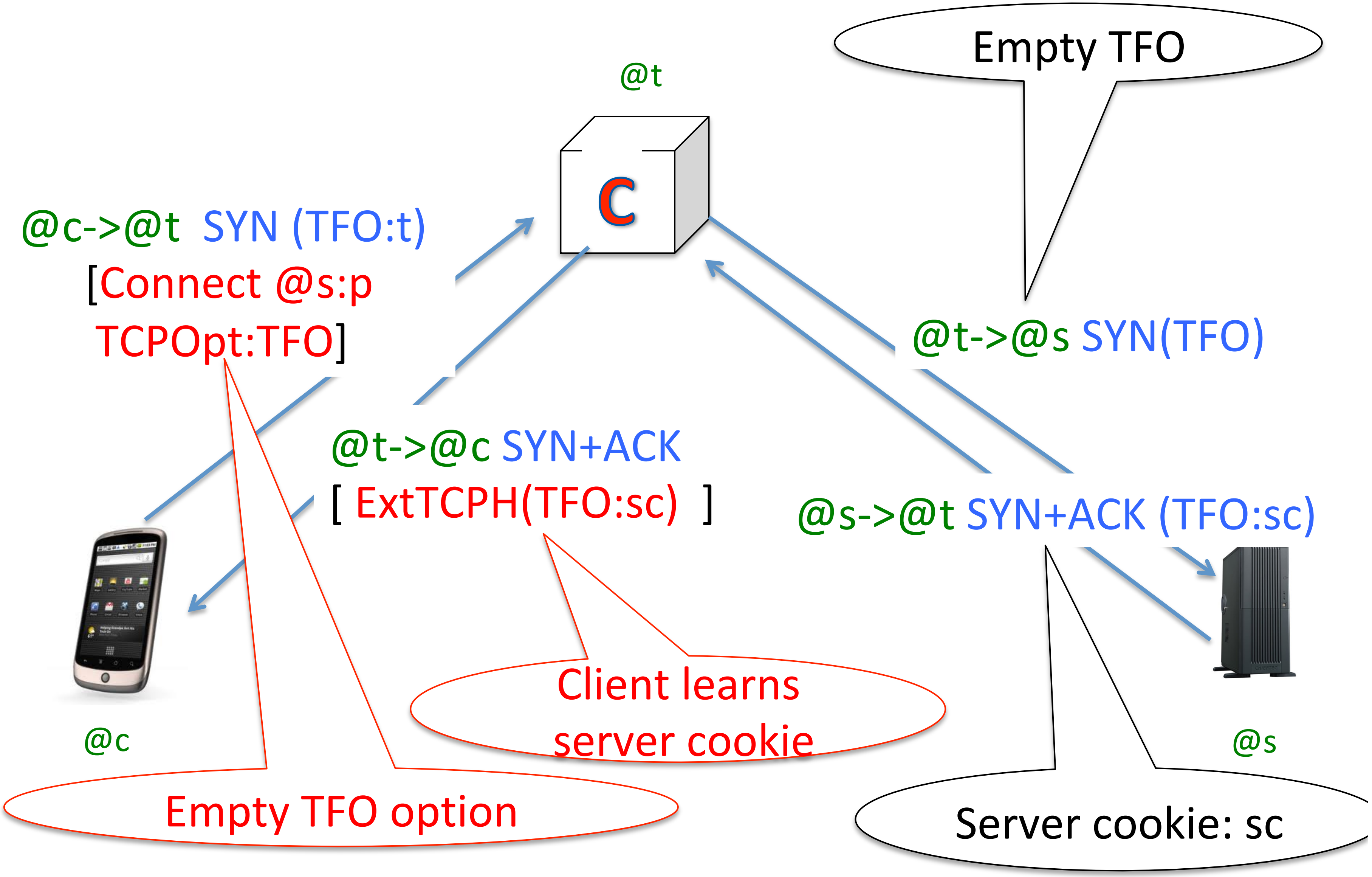
Converter cookie

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

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

TCP Extensions supported by Converter

TFO connection through the converter



@t

Empty TFO

C

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

@t->@s SYN(TFO)

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

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

Client learns
server cookie

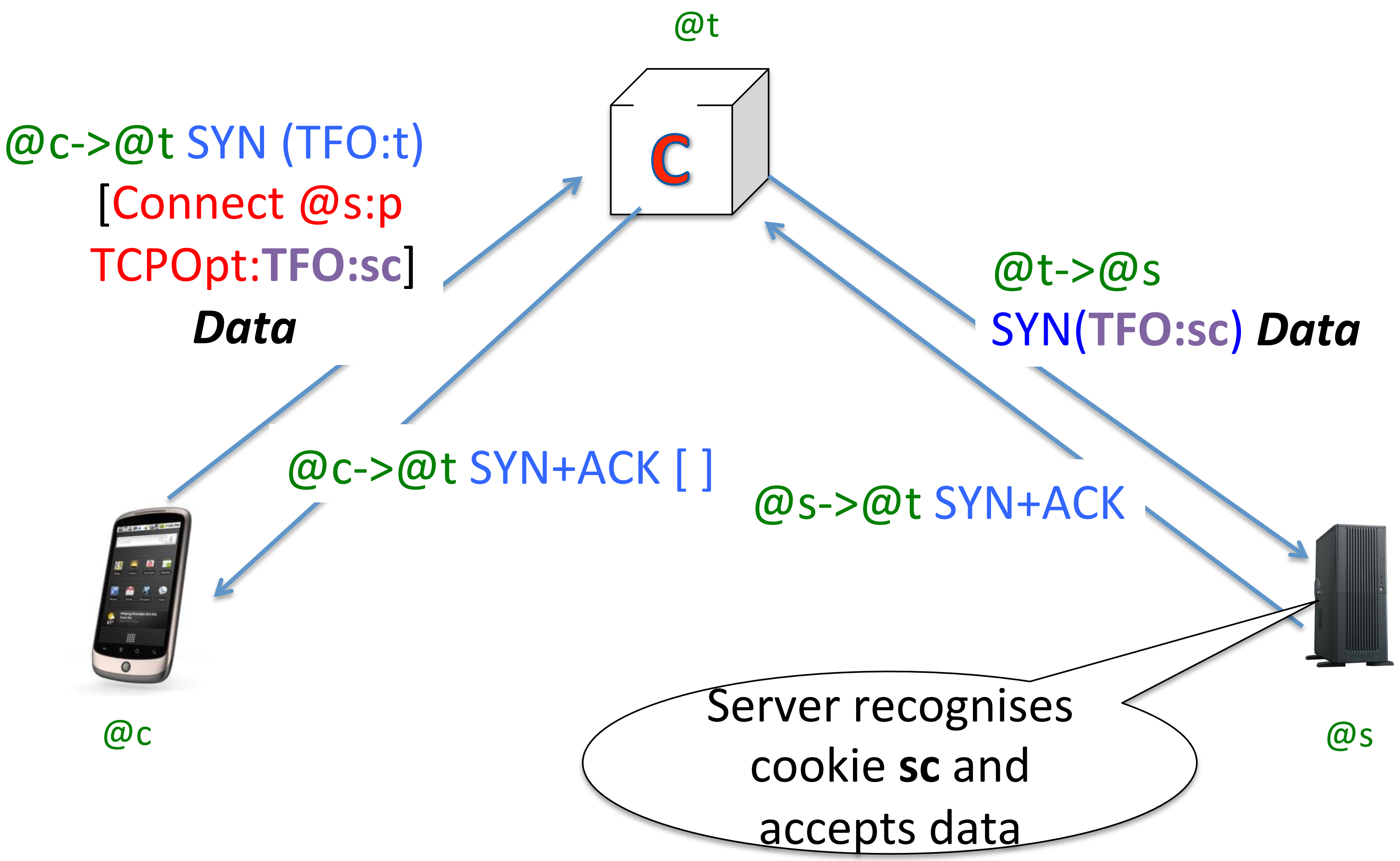
@c

@s

Empty TFO option

Server cookie: sc

TFO connection through the converter second connection to server



@t

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

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

@c->@t SYN+ACK [ ]

@s->@t SYN+ACK

Server recognises cookie **sc** and accepts data

@c

@s

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

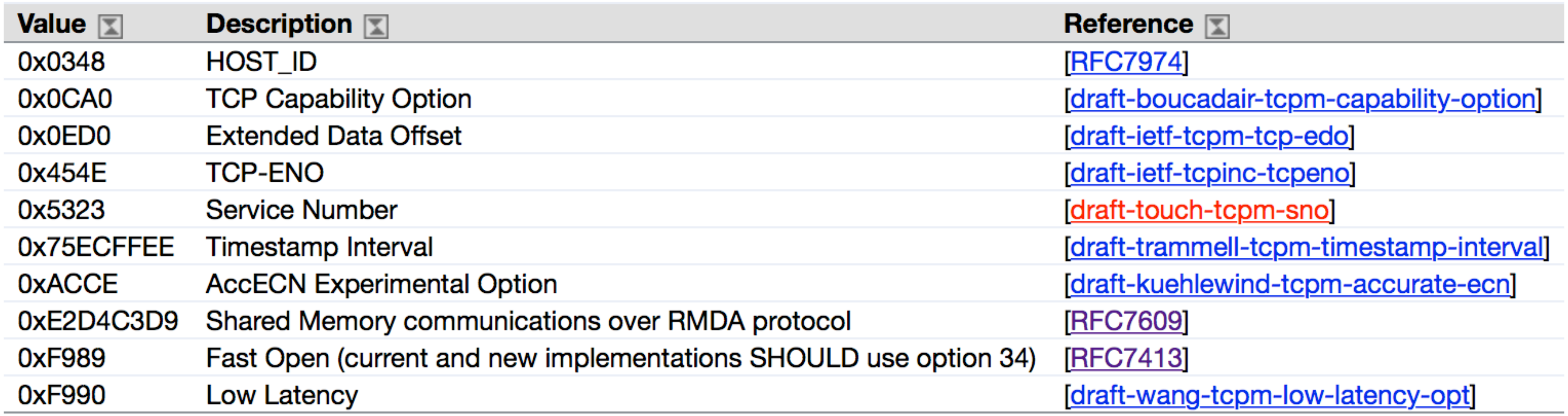
TCP Authentication Option

- 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



Value	Description	Reference
0x0348	HOST_ID	[RFC7974]
0x0CA0	TCP Capability Option	[draft-boucadair-tcpm-capability-option]
0x0ED0	Extended Data Offset	[draft-ietf-tcpm-tcp-edo]
0x454E	TCP-ENO	[draft-ietf-tcpinc-tcpeno]
0x5323	Service Number	[draft-touch-tcpm-sno]
0x75ECFFEE	Timestamp Interval	[draft-trammell-tcpm-timestamp-interval]
0xACCE	AccECN Experimental Option	[draft-kuehlewind-tcpm-accurate-ecn]
0xE2D4C3D9	Shared Memory communications over RMDA protocol	[RFC7609]
0xF989	Fast Open (current and new implementations SHOULD use option 34)	[RFC7413]
0xF990	Low Latency	[draft-wang-tcpm-low-latency-opt]

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