

0-RTT TCP Converters

draft-bonaventure-mptcp-converters-01

IETF99, July 2017

O. Bonaventure

M. Boucadair

B. Peirens

Motivation

- Instead of tuning MPTCP to cope middleboxes, can we design middleboxes that would **benefit** to MPTCP ?

Finally, the working group will explore whether an MPTCP-aware middlebox would be useful, where at least one end host is MPTCP-enabled. For example, potentially helping MPTCP's incremental deployment by allowing only one end host to be MPTCP-enabled and the middlebox acts as an MPTCP proxy for the other end host, which runs TCP; and potentially helping some mobility scenarios, where the middlebox acts as an anchor between two MPTCP-enabled hosts. The working group will detail what real problems an MPTCP-enabled middlebox might solve, how it would impact the Multipath TCP architecture (RFC6182), what proxy approach might be justified as compared against alternative solutions to the problems, and the likely feasibility of solving the technical and security issues.

Some proposed solutions

Extensions for Network-Assisted MPTCP Deployment Models

draft-boucadair-mptcp-plain-mode-10

Status	IESG evaluation record	IESG writeups	Email expansions	History							
Versions	00	01	02	03	04	05	06	07	08	09	10



MPTCP Working Group
 Internet-Draft
 Intended status: Informational
 Expires: January 6, 2017

B. Peirens
 Proximus
 G. Detal
 S. Barre
 O. Bonaventure
 Tessares
 July 05, 2016

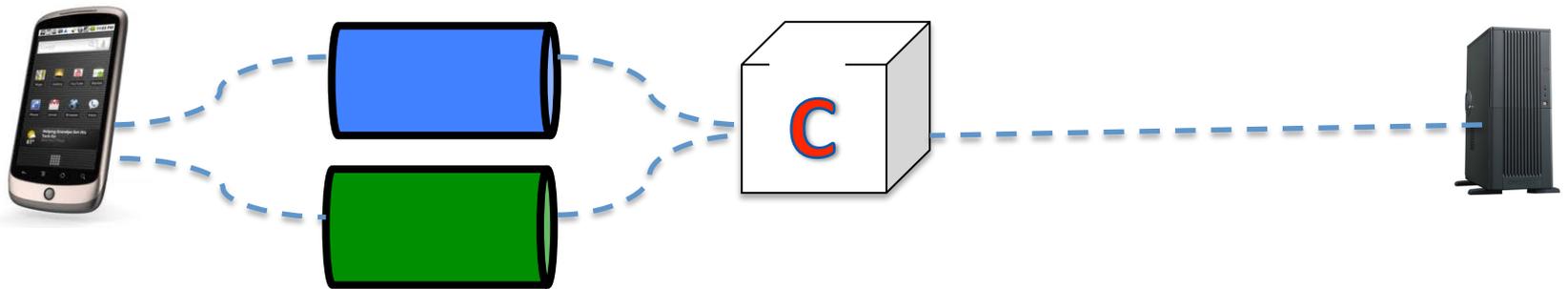
Link bonding with transparent Multipath TCP
draft-peirens-mptcp-transparent-00

Network Working Group
 Request for Comments: 1928
 Category: Standards Track

M. Leech
 Bell-Northern Research Ltd
 M. Ganis
 International Business Machines
 Y. Lee
 NEC Systems Laboratory
 R. Kuris
 Unify Corporation
 D. Koblas
 Independent Consultant
 L. Jones
 Hewlett-Packard Company
 March 1996

Converter

- Motivation
 - Far 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



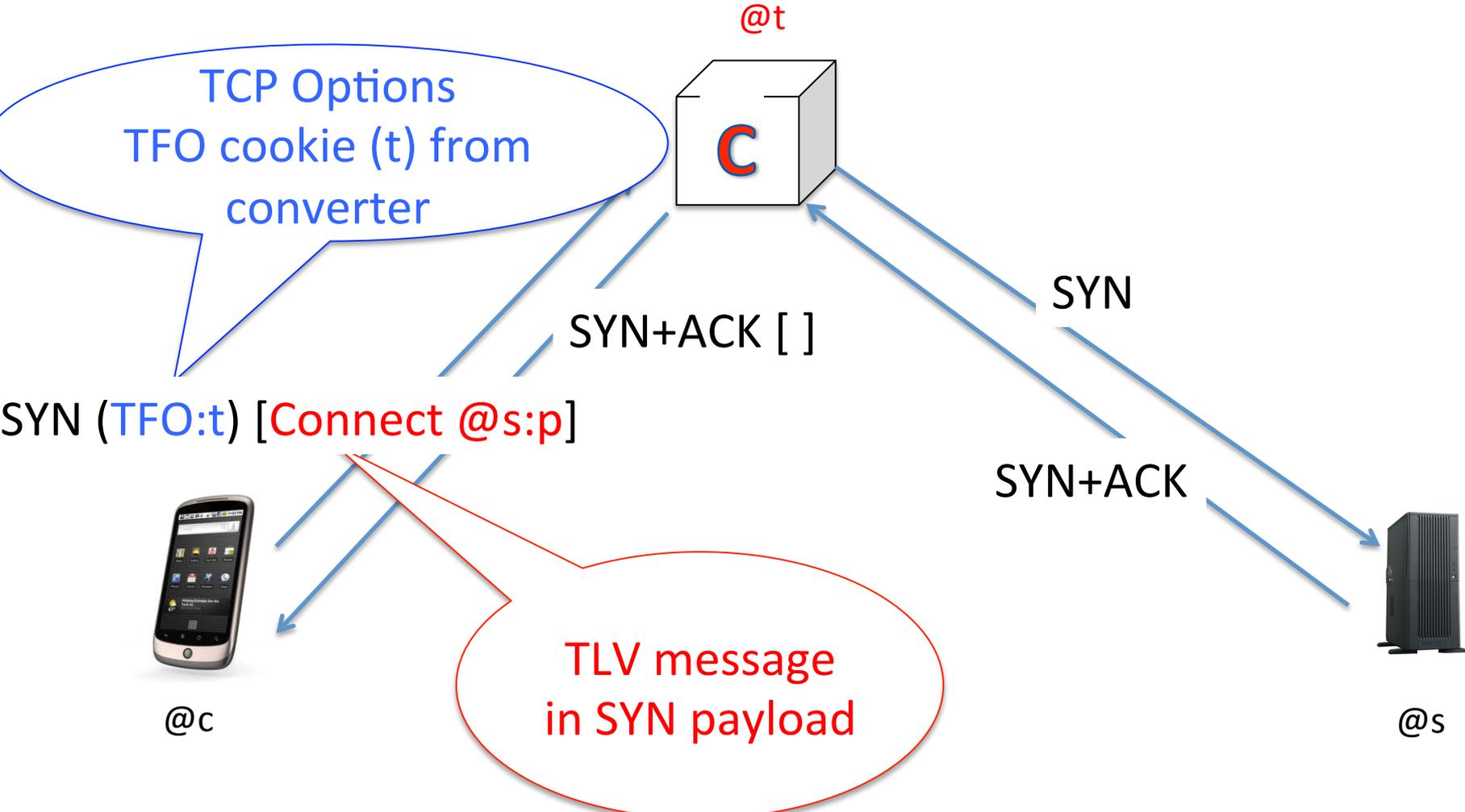
Key points of email discussion

- Using the converter should not significantly increase the connection establishment delay
 - 0-RTT
- Client can decide to use converter based on policies
- Clients should be able to bypass converter when server supports MPTCP
- TFO must be used if data is placed in SYN
 - For client-converter exchanges
 - Clients using converter should still be able to use TFO with servers
- Design should be extensible and future-proof
- Should avoid defining new TCP Options

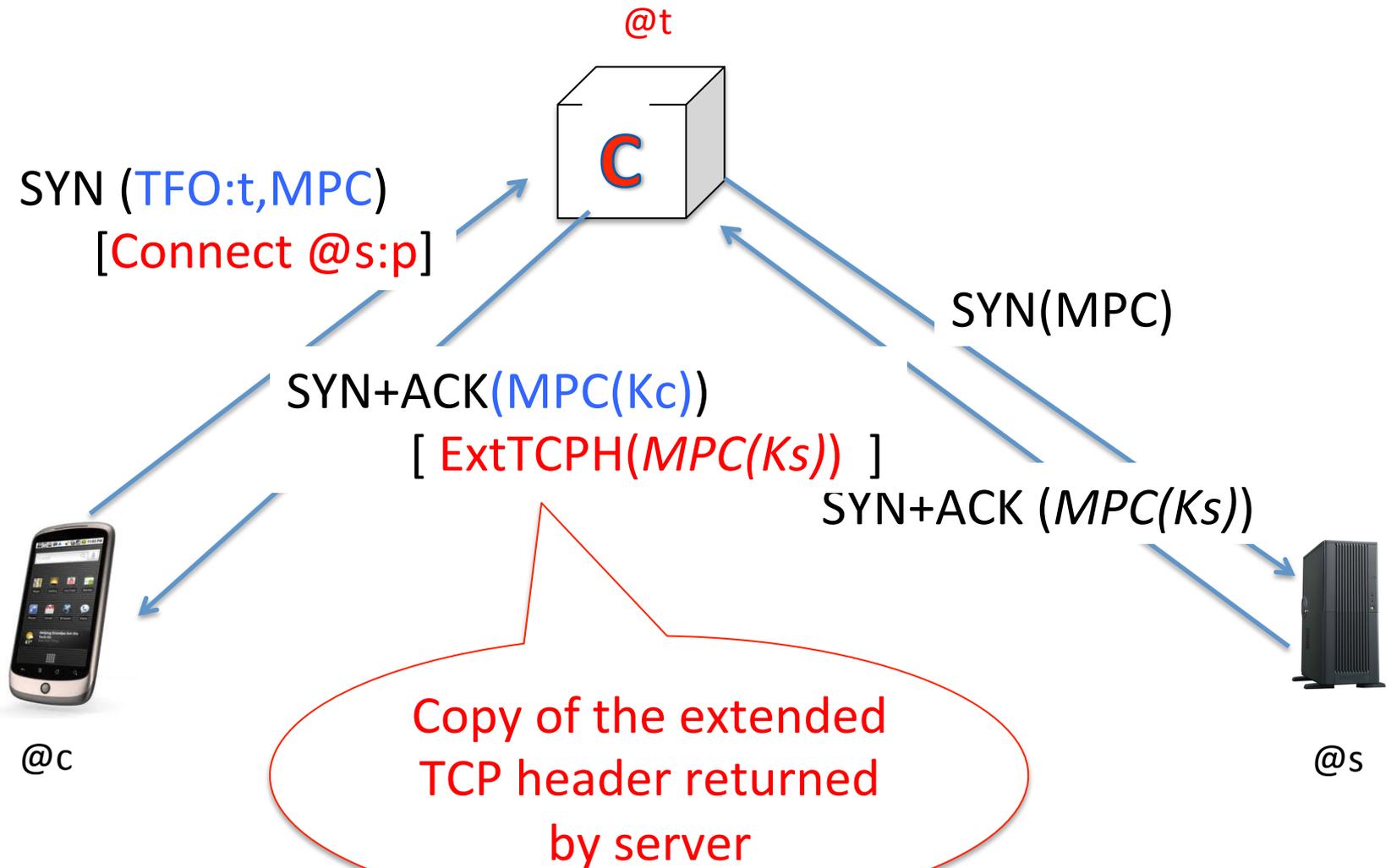
A cleaner design

- Converter Protocol is an application-level protocol listening on a specific TCP port
- Client commands and converter responses are encoded as TLV messages
 - Ensures extensibility
- Converter Protocol leverages TFO
 - Commands and responses can be sent inside SYNs
- Client can learn options supported by server
 - Allows client to bypass converter

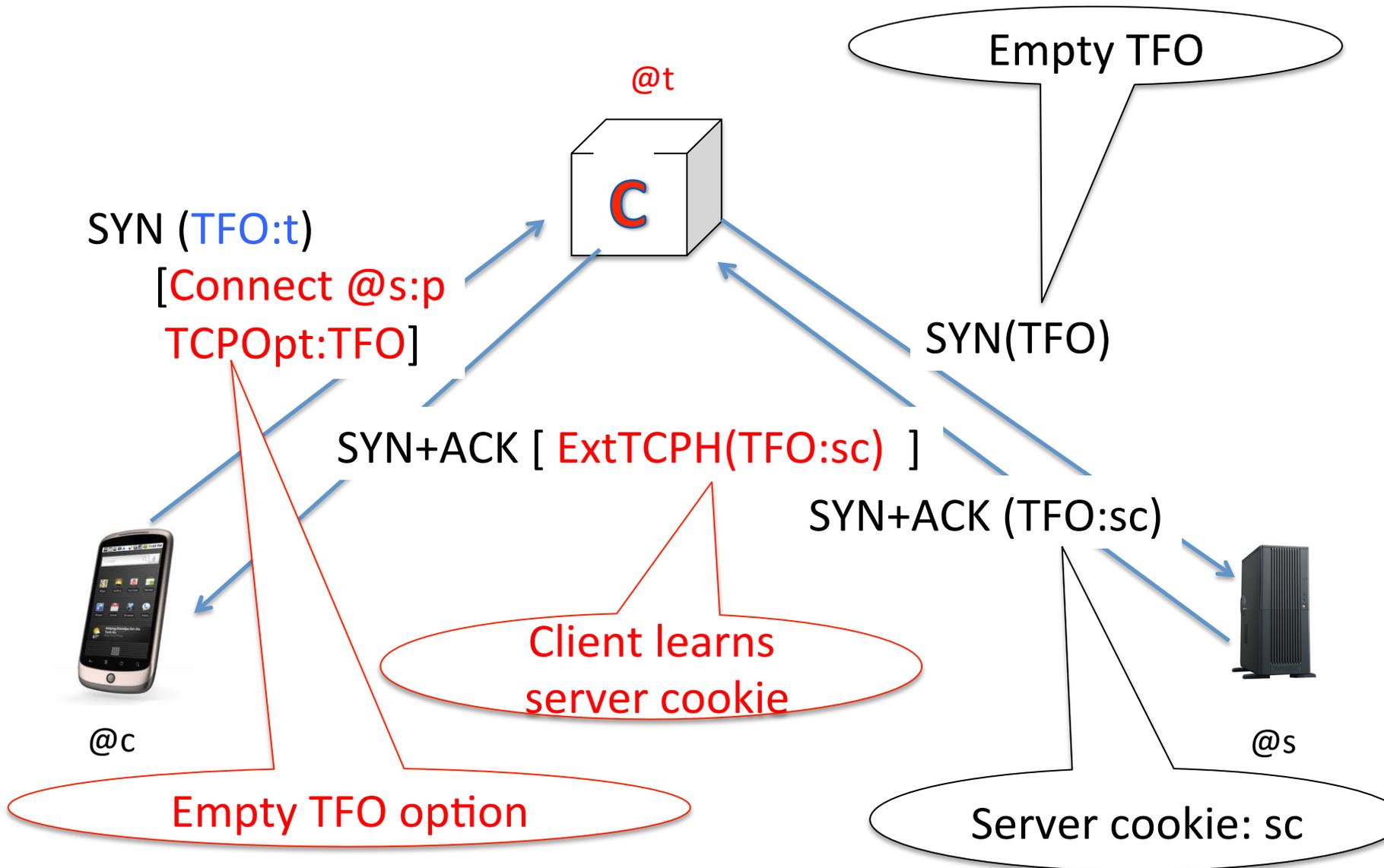
Simplified example



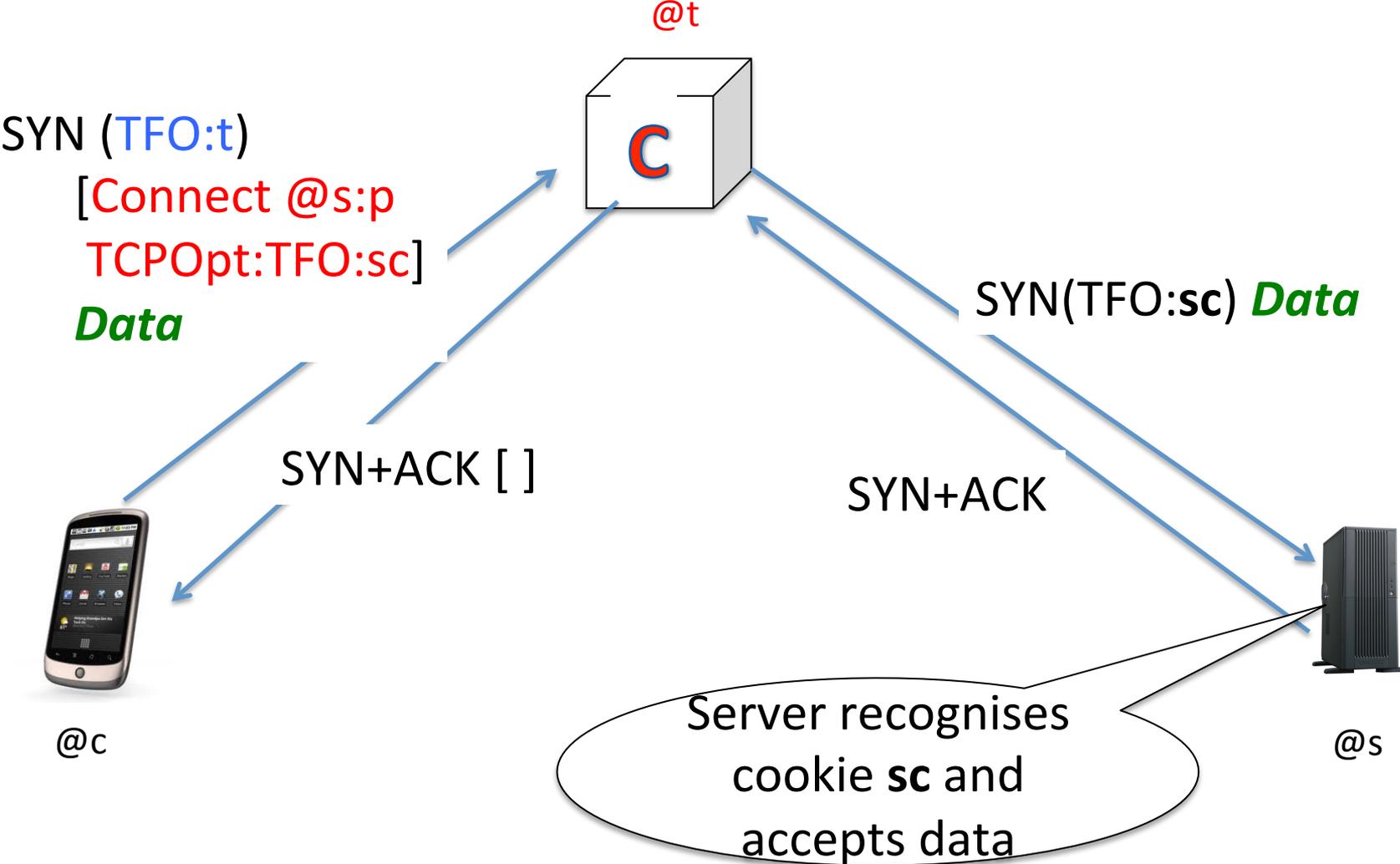
MPTCP connection through converter



TFO connection through converter



TFO connection through converter second connection to server



About extensibility

- Two dimensions were considered in the design
 - Extensibility of the Converter protocol
 - Version number and TLV format ensure that the protocol can be extended for different use cases
 - Evolution of TCP
 - Converter protocol was designed with MPTCP in mind, but other TCP extensions could benefit from it as well
 - Client can detect which TCP options are supported by the server and decide to bypass the converter

Criteria

Main criterias (email from Phil/Yoshi)

- No changes to MPTCP (RFC6824bis)
 - *Converter uses application layer protocol leveraging TFO*
- Proxy is simple to operate and deploy
 - *Converter uses reserved service name/port*
- A session can be initiated from either end
- The set-up time is minimized
 - *0-RTT leveraging TFO*
- minimise the amount of overhead on data
 - *TLV messages appear only in SYN*
- solution needs to work if end to end encryption is in use

Criteria

- Criteria specific to solutions for the single-ended proxy scenario:
 - the proxy is unlikely to be on both default paths
 - *Converter uses reserved service name/port*
 - clarify whether the proxy simply forwards
 - *Converter terminates TCP connections*
 - allows hosts to have traffic that doesn't get proxied
 - *Client can bypass converter (policies or if server supports)*
 - end host and proxy need to authenticate
 - *Not in draft-00 but can be added as new TLV messages*

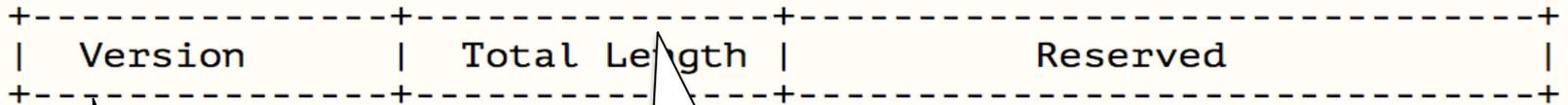
Conclusion

- New design takes into account 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 option
- We request WG adoption for charter item
 - *Finally, the working group will explore whether an MPTCP-aware middlebox would be useful, where at least one end host is MPTCP-enabled.*

Backup slides

Fixed Header

- Sent by client and converter

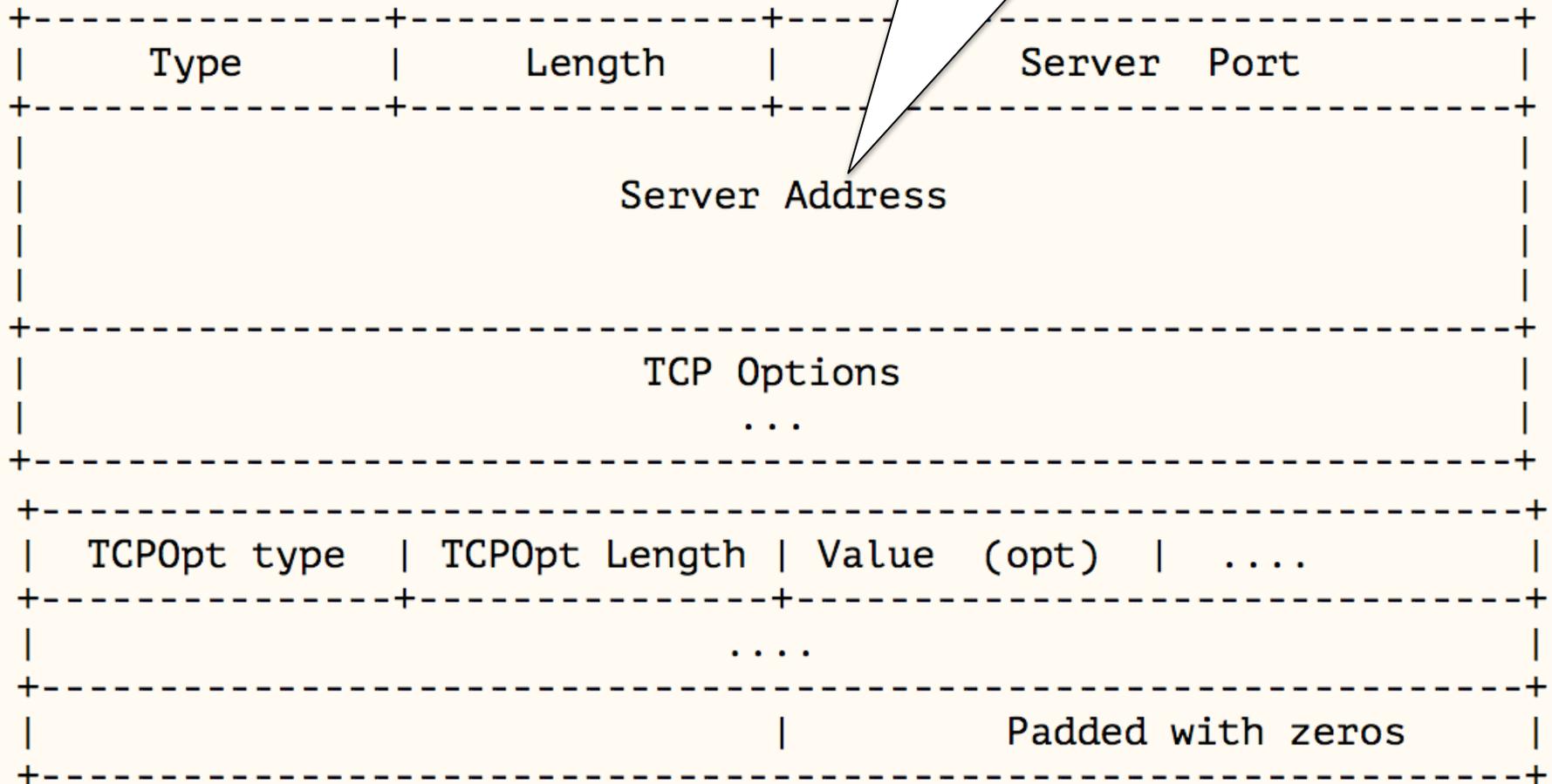


Length of all
TLV messages in
blocks of 32 bits

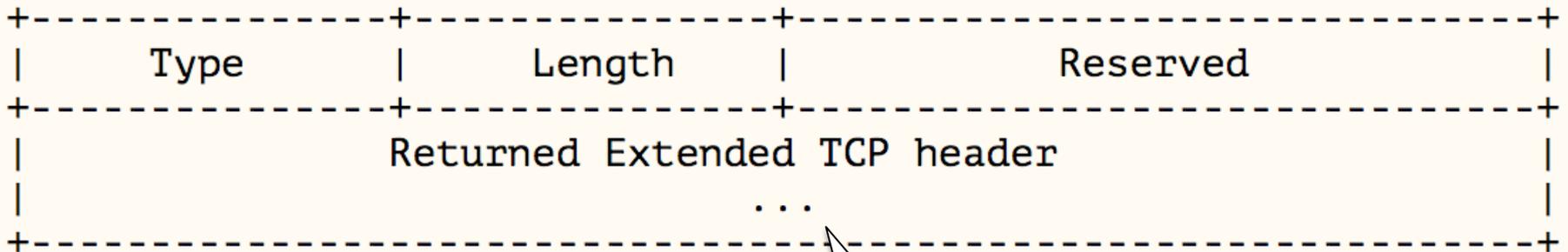
Current version=1

The Connect TLV

IPv6 address

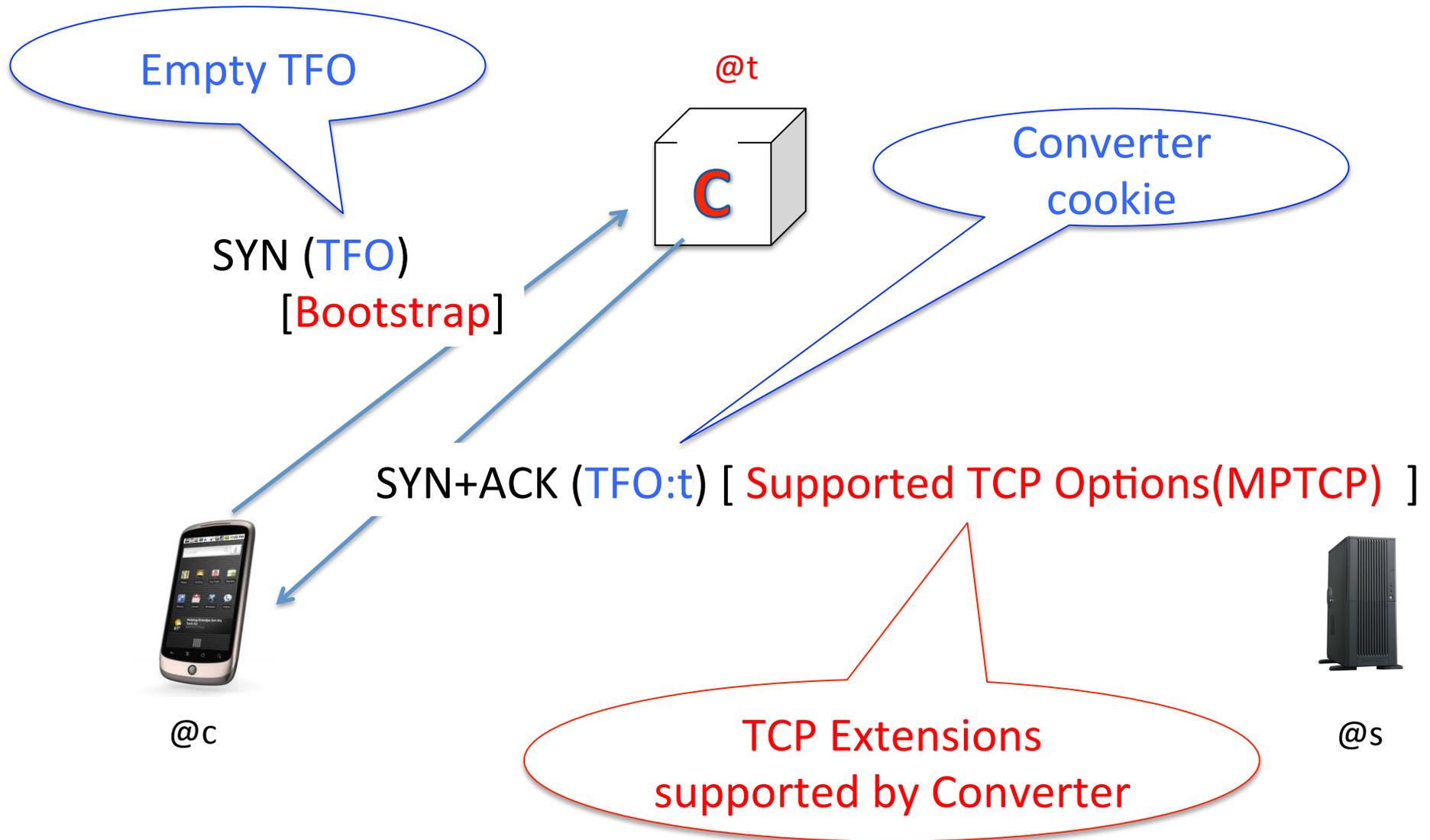


The Extended TCP Header TLV



Copy of the Extended TCP header returned by server in SYN+ACK

Bootstrap procedure



Bootstrap

Type	Length	Zero
------	--------	------

Type	Length	Reserved
Kind #1	Kind #2	...
/		/
/		/
	Kind #n	Zero