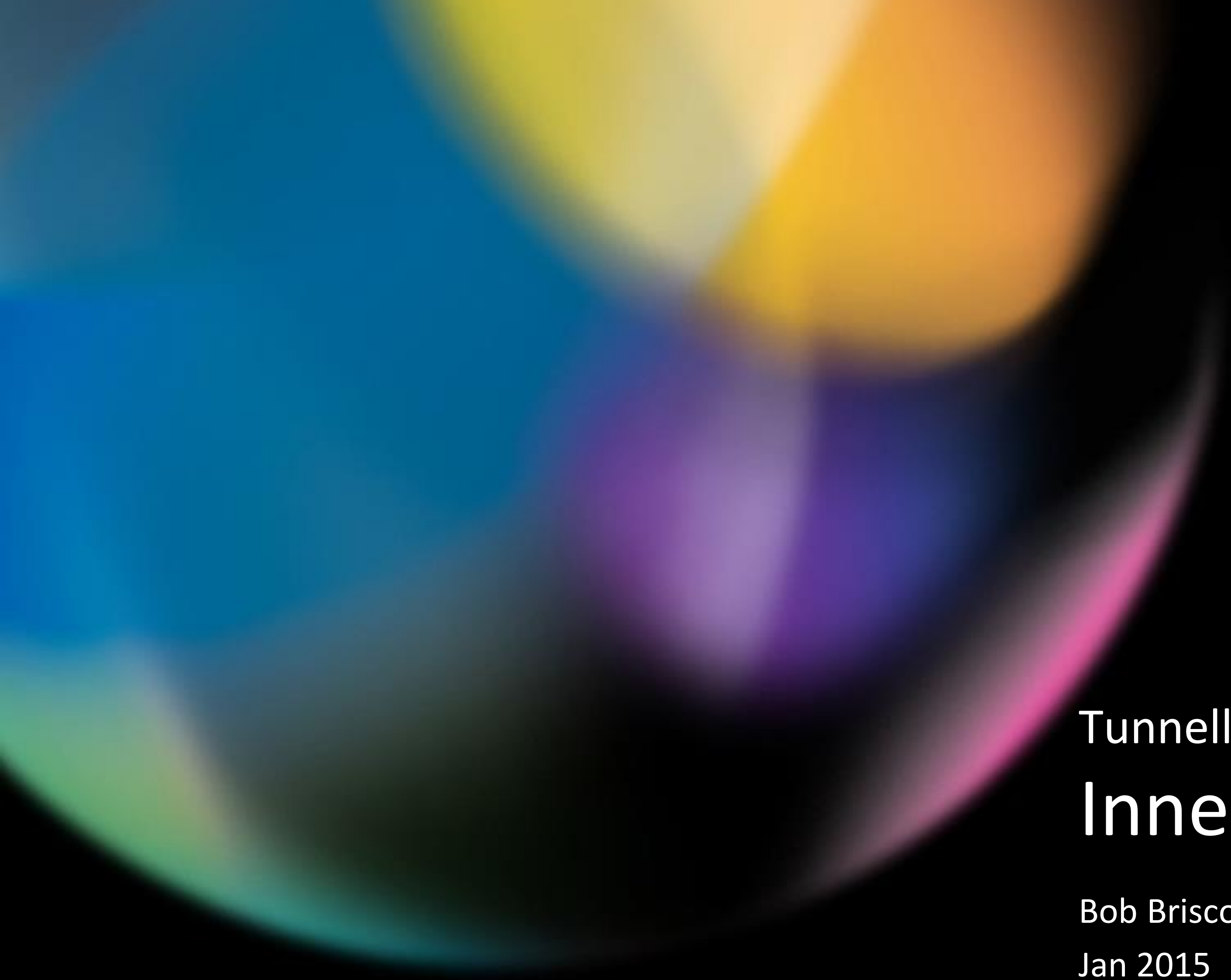


Tunnelling Through Inner Space

Bob Briscoe

Jan 2015

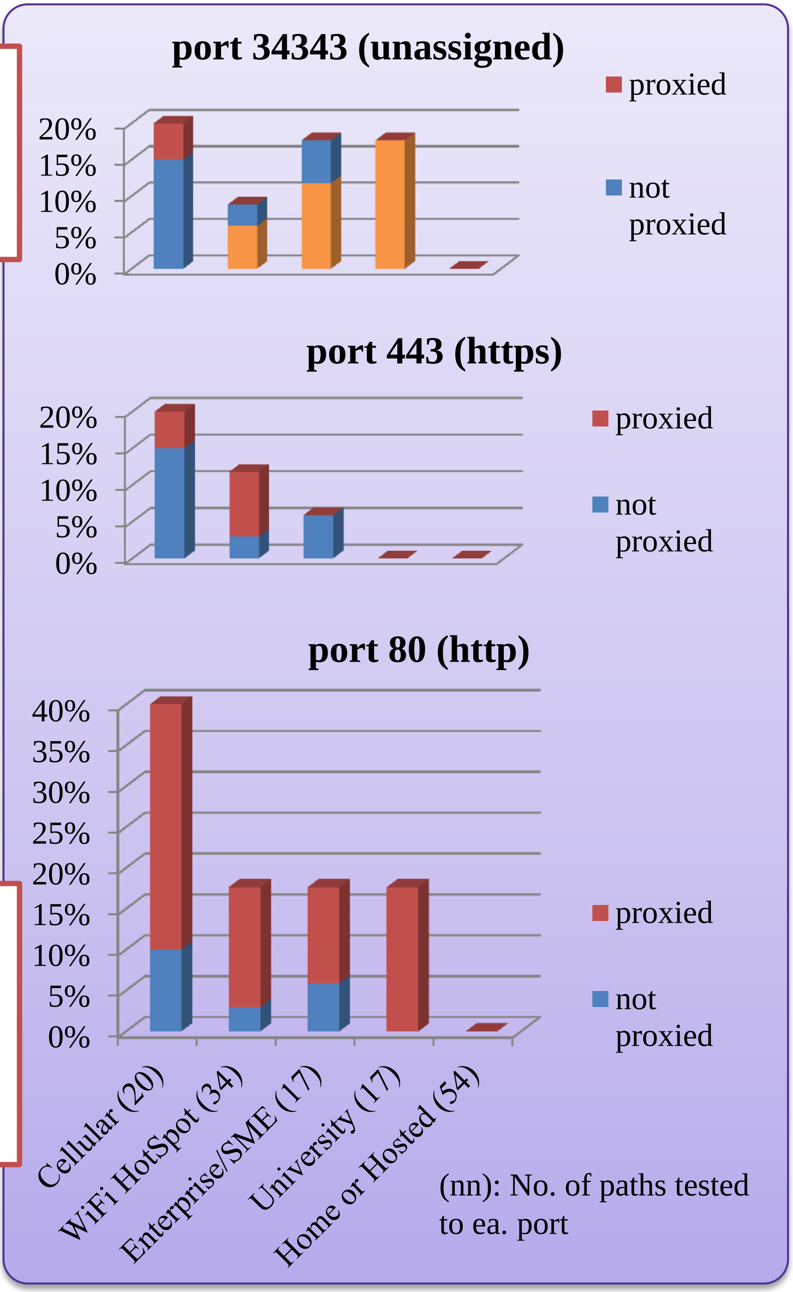
Bob Briscoe's work is part-funded by the European Community
under its Seventh Framework Programme through the
Trilogy 2 (ICT-317756) and the RITE (ICT-317700) projects

the old transport extensibility architecture

Unknown option stripped from TCP SYN

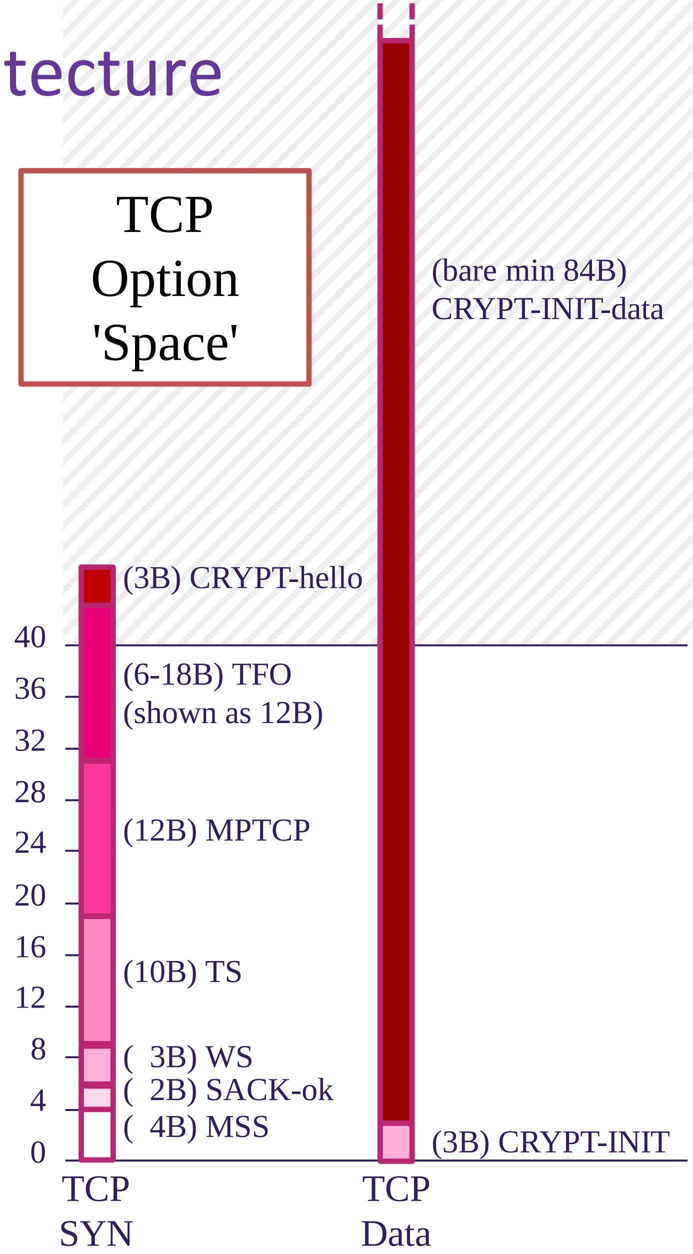
Source: The dataset collected for:

Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A., Handley, M., and H. Tokuda,
"Is it Still Possible to Extend TCP?", Proc. ACM Internet Measurement Conference (IMC'11) 181--192, Nov 2011

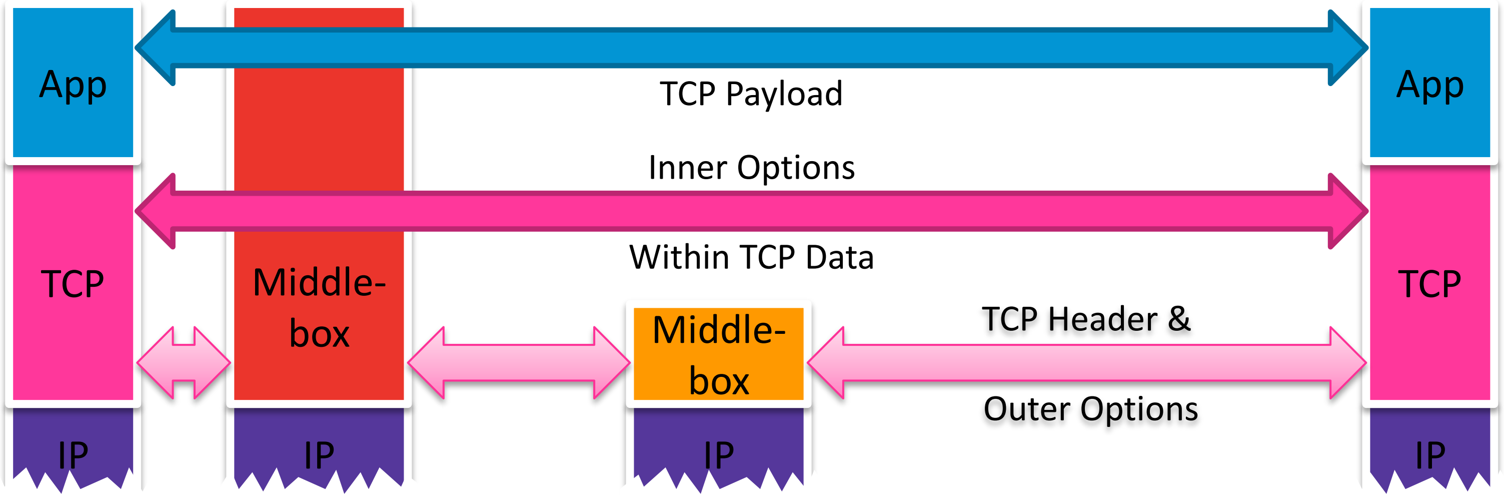


(nn): No. of paths tested to ea. port

TCP Option 'Space'



Approach: Tunnel through Inner Space



Strawman principle: In a middlebox world, it is both more principled and more pragmatic to extend the layer *X* header within layer *X+1* *

- Why should an implementation walk a list of extensions for which it has no code?
- Extension can be coded to know where to look

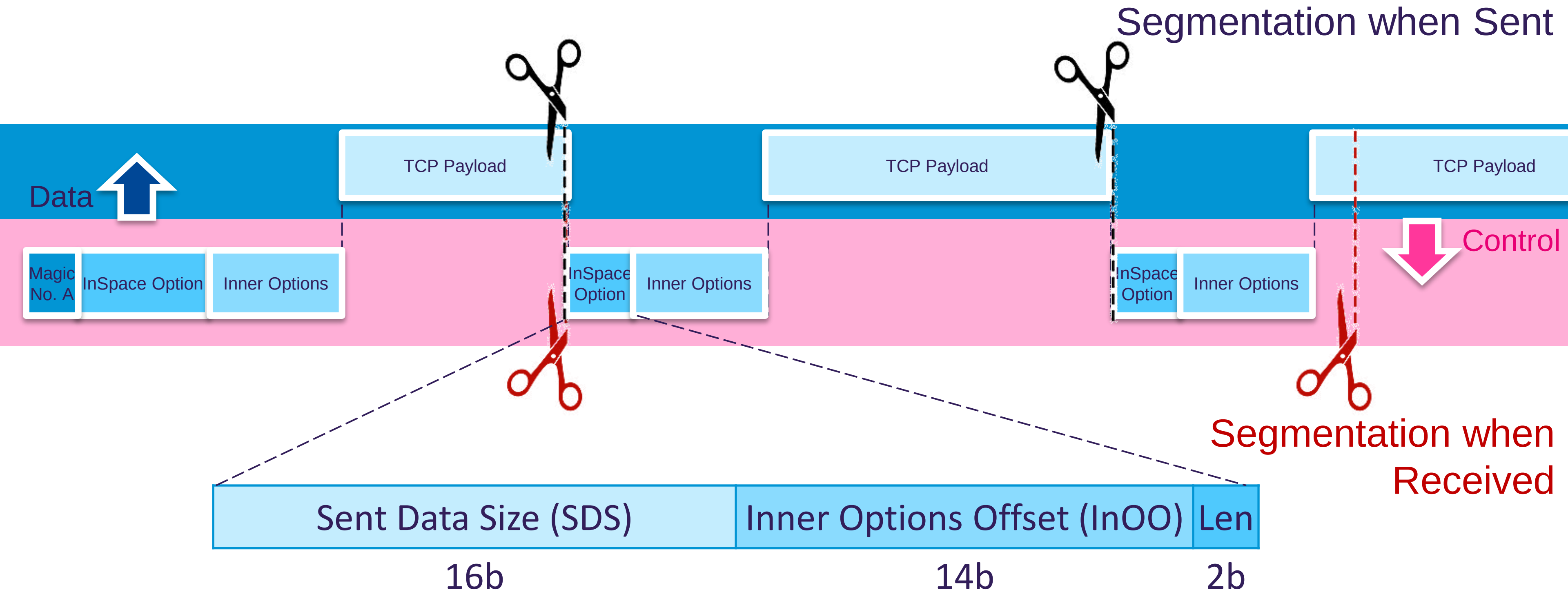
[Rob Hancock re. IPv6 extensions (Trilogy project, Feb 2010)]

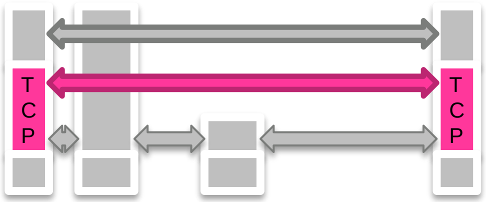
• options:	layer *X*
• extensions:	layer *X+1* *
• end-2-middle:	layer *X*

- How to prevent legacy layer *X* passing corrupt payload to *X+1*?
- Examples (see position paper):
  - (L4) Minion
  - (L4) Inner Space
  - (L3) ConEx
  - (L3) Generic UDP tunnelling (GUT)

* In Internet arithmetic, 4+1 = 7

Inner Space: in the TCP datastream





- robust to resegmentation
- Inner Options not prone to stripping
- in-order delivery of Inner Options
- out-of-order delivery also available

middlebox domination strategy

long term aim

- authenticated control channel

- if turned on option authentication today
  - up to 40% of connections would break
  - the ends break a working service

- middlebox domination strategy
  - Inner Space + option authentication (breaks 0%)
- then, if middleboxes move into the TCP data
  - the middleboxes break a working service

why shoot yourself in the foot

when you can make them shoot themselves in the foot?

Inner Space: Implications & Status

draft-briscoe-tcpm-inner-space-sink-00c
(splitting into sub-drafts - in progress)

draft-briscoe-tcpm-inner-space-01

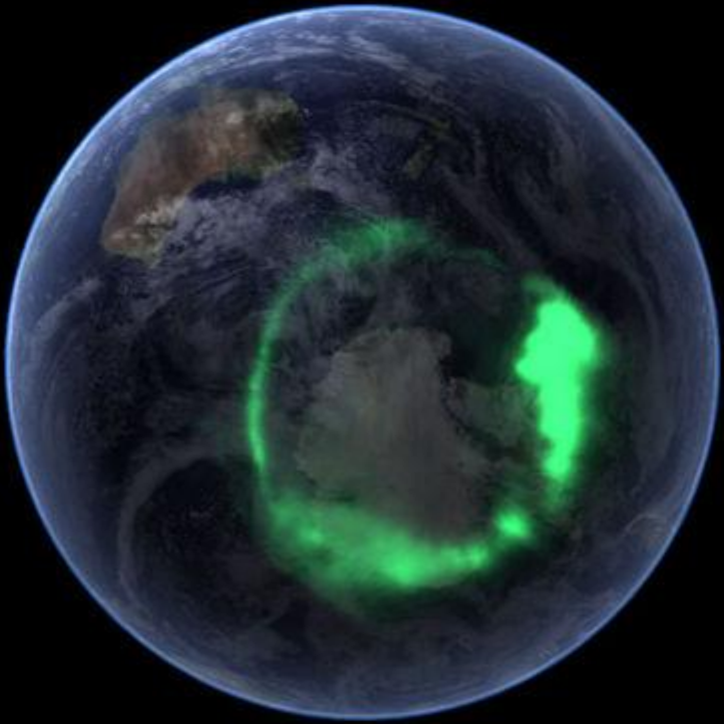
Payload	Control Options		
	in-order	out-of-order	both
in-order	Default	(TCP)	TCPbis
out-of-order		(UDP)	UDPbis
both		(SCTP)	'TCP2'

Assessing whether 'TCP2' could satisfy HTTP2 reqs

- Switchable transport semantics
  - Looks like vanilla TCP on the wire
  - switch inner semantics with TCP options e.g. ordering, encryption, compression
  - think "extensible Minion"

- Example: tcpcrypt decomposition
  - cut from 18 to 9 CRYPT sub-options
  - removed handshake latency
  - can encrypt control options, and MAC pure ACKs

- Progress since Jul'14
  - Default mode: Full spec as individual draft (5 revs, presented in tcpm & tcpinc)
  - TCPbis mode: Full spec available but not submitted <http://bobbriscoe.net/projects/2020comms/tcp/draft-briscoe-tcpm-inner-space-sink-00c.txt>
  - ad hoc team formed (~20 people on mailing list)
  - half-a-dozen doing or planning path traversal testing
  - 2 or 3 planning to implement, including upstreaming



Inner Space

Q&A

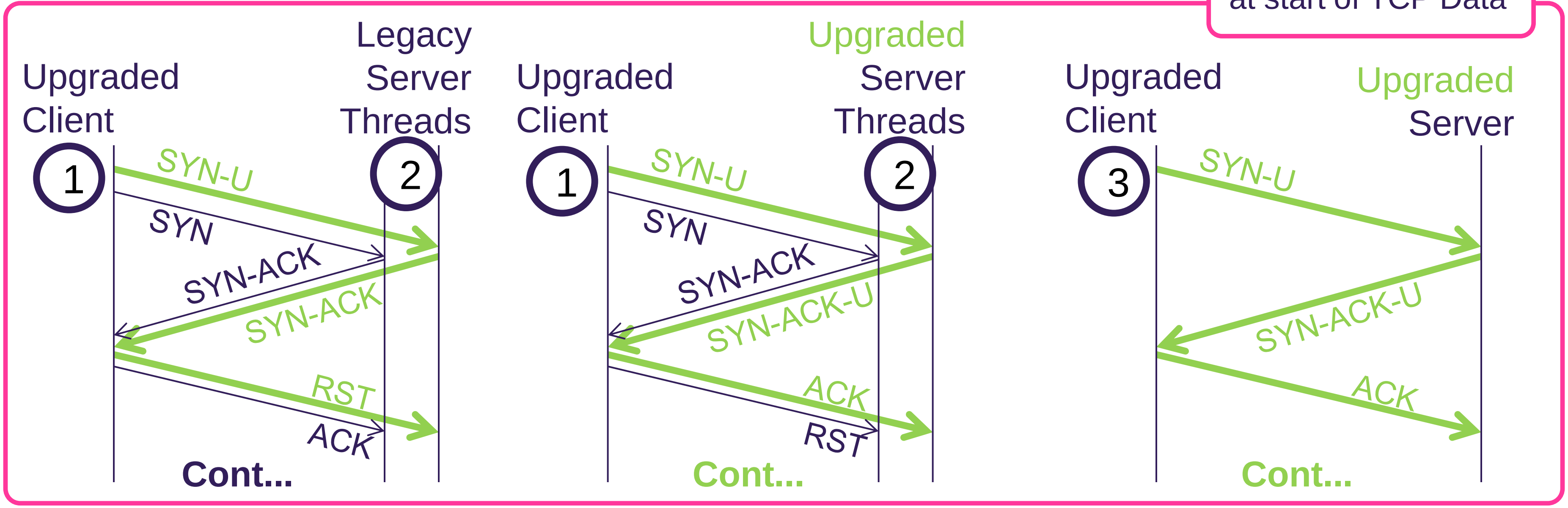
Spare slides

dual handshake... and migration to single

1. different source ports, same dest. port
2. no co-ordination needed between server threads

   can be physically separate replicas

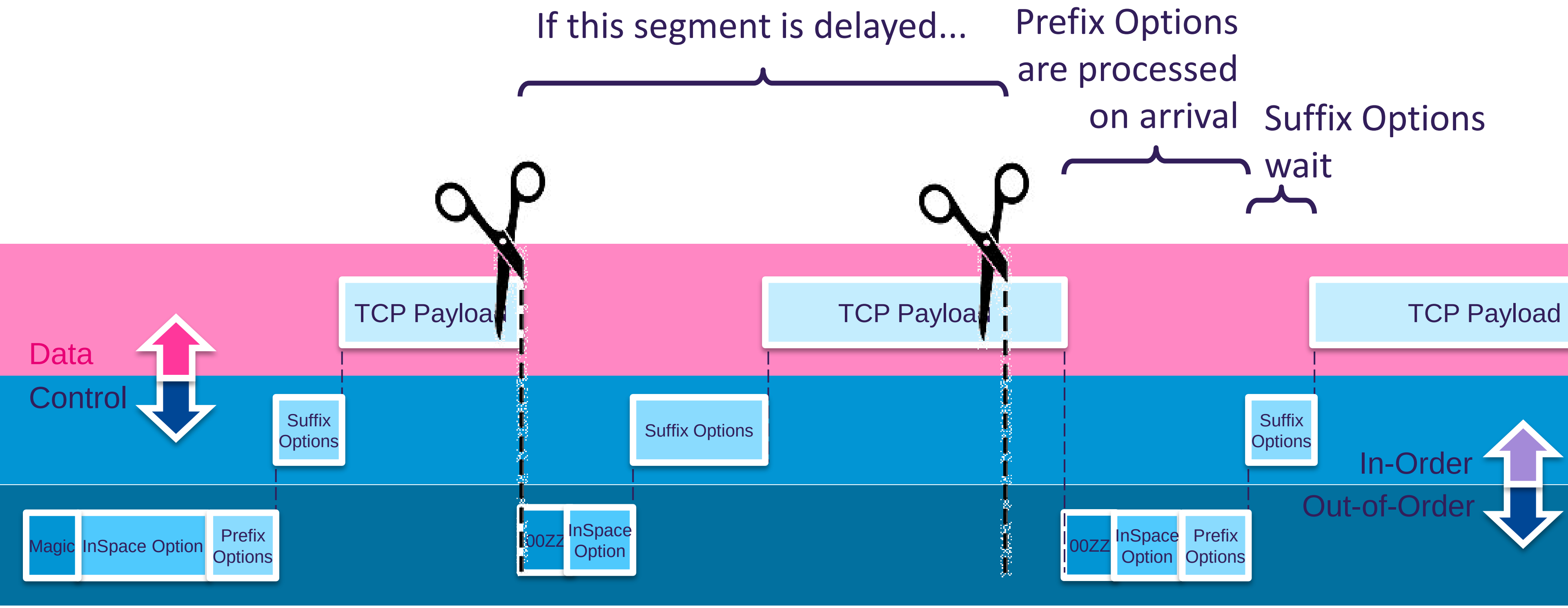
-U = upgraded,
i.e. magic no.
at start of TCP Data



3. Can use single SYN-U handshake
   - when server is in cached white-list
   - once deployment is widespread (no need for white-list)

   Fall-back to SYN if no SYN-ACK-U

TCPbis mode: 2 control channels in the datastream



- Rcvr can reconstruct sent segments - robust to resegmentation
- TCP has always processed Outer Options on arrival (out-of-order)
- Inner Space adds two types of Inner Option to avoid middlebox interference
  - In-order Suffix Options – for stream control
  - Out-of-order Prefix Options
    - essential for a few ACK-related options* to avoid flow-control deadlock

* SACK, MPTCP Data ACK, tcpcrypt MAC of ACK