Inner Space

Bob Briscoe

Nov 2014

draft-briscoe-tcpm-inner-space-01

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

problem (Inner Space addresses all these)

- no arbitrary limit to TCP option space on all segments
  - SYN, SYN/ACK, non-SYN
- middlebox traversal
  - not just detect-and-die
  - traverse resegmentation, option-stripping, DPI Web filters etc.
  - for itself and for all TCP options it supports
- legacy server fall-back with no added latency

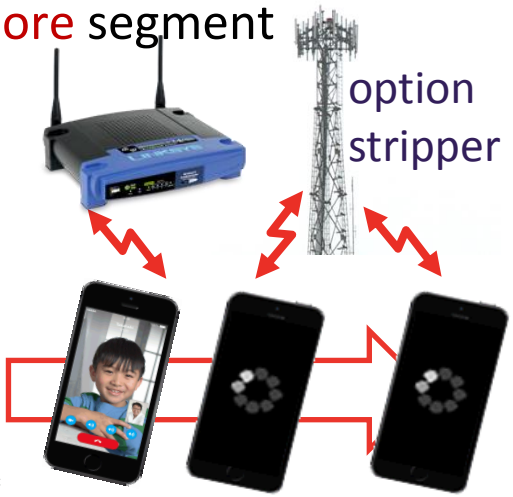
- make TCP options easy
  - they will just work
  - from the SYN onwards

middleboxes: detect-and-die?

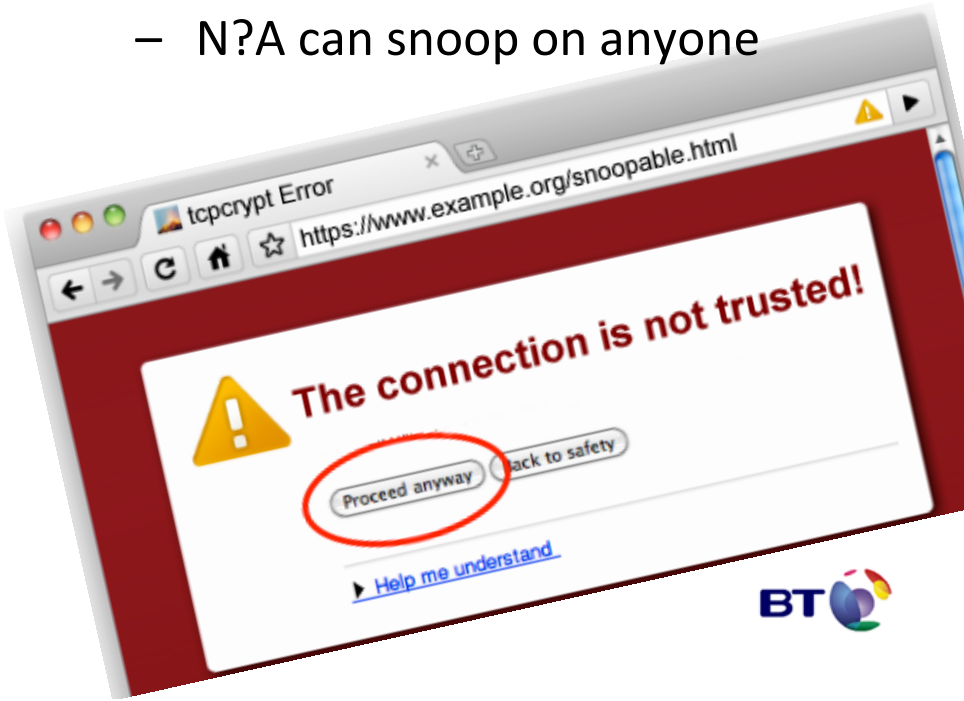
to port	% paths stripped
80 (HTTP)	14%
443 (HTTPS)	6%
34343 (unassigned)	4%

[Honda11]

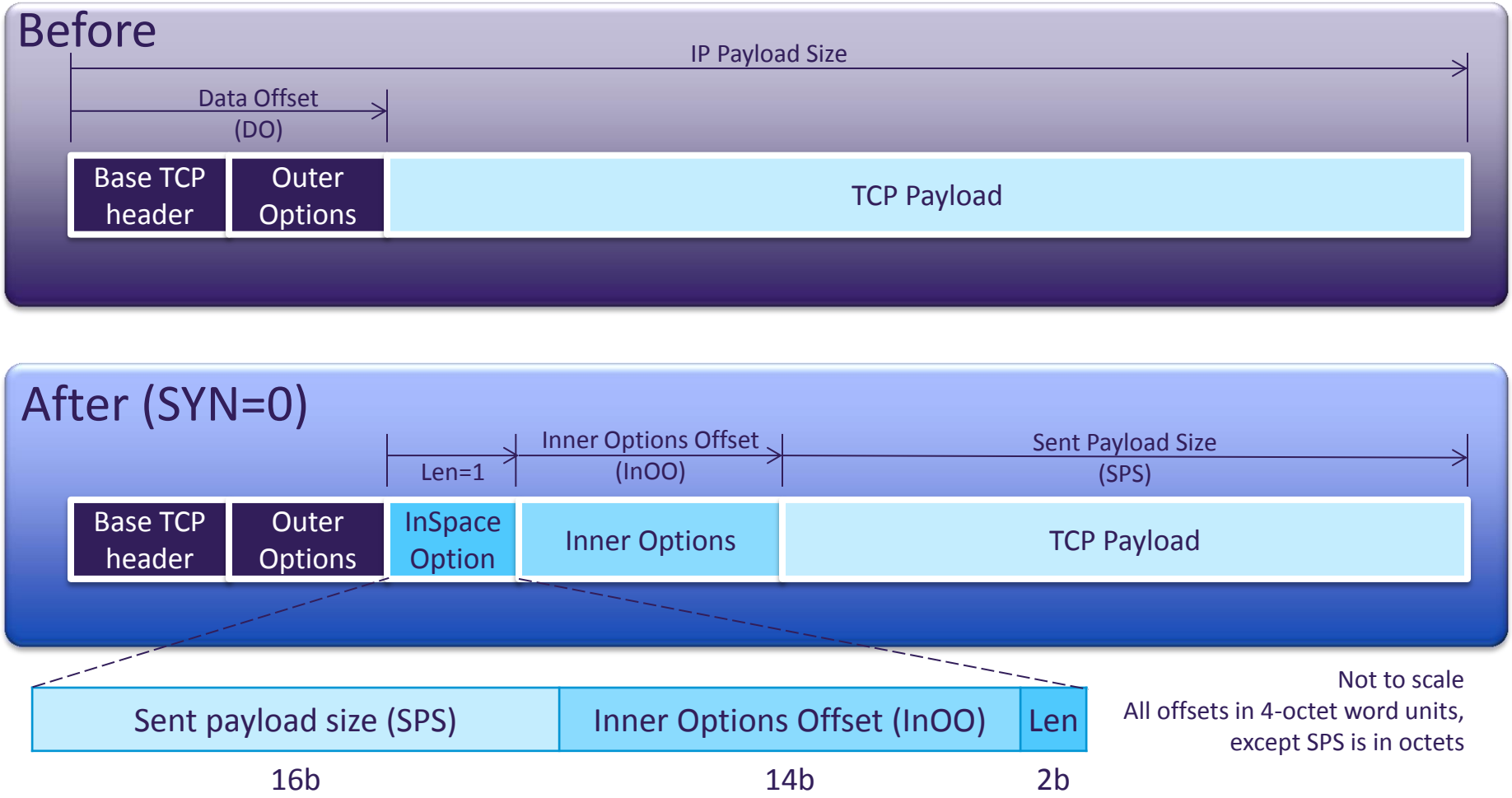
- EDO
  - if stripped from SYN or SYN/ACK: disable EDO
  - required on all segments, even if space not needed
  - if stripped mid-connection, ignore segment



- tcpcrypt + EDO
  - tcpcrypt will disable itself on ~10% of paths
  - downgrade as the norm
  - N?A can snoop on anyone

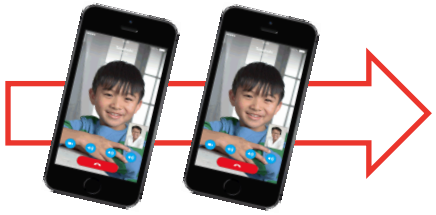


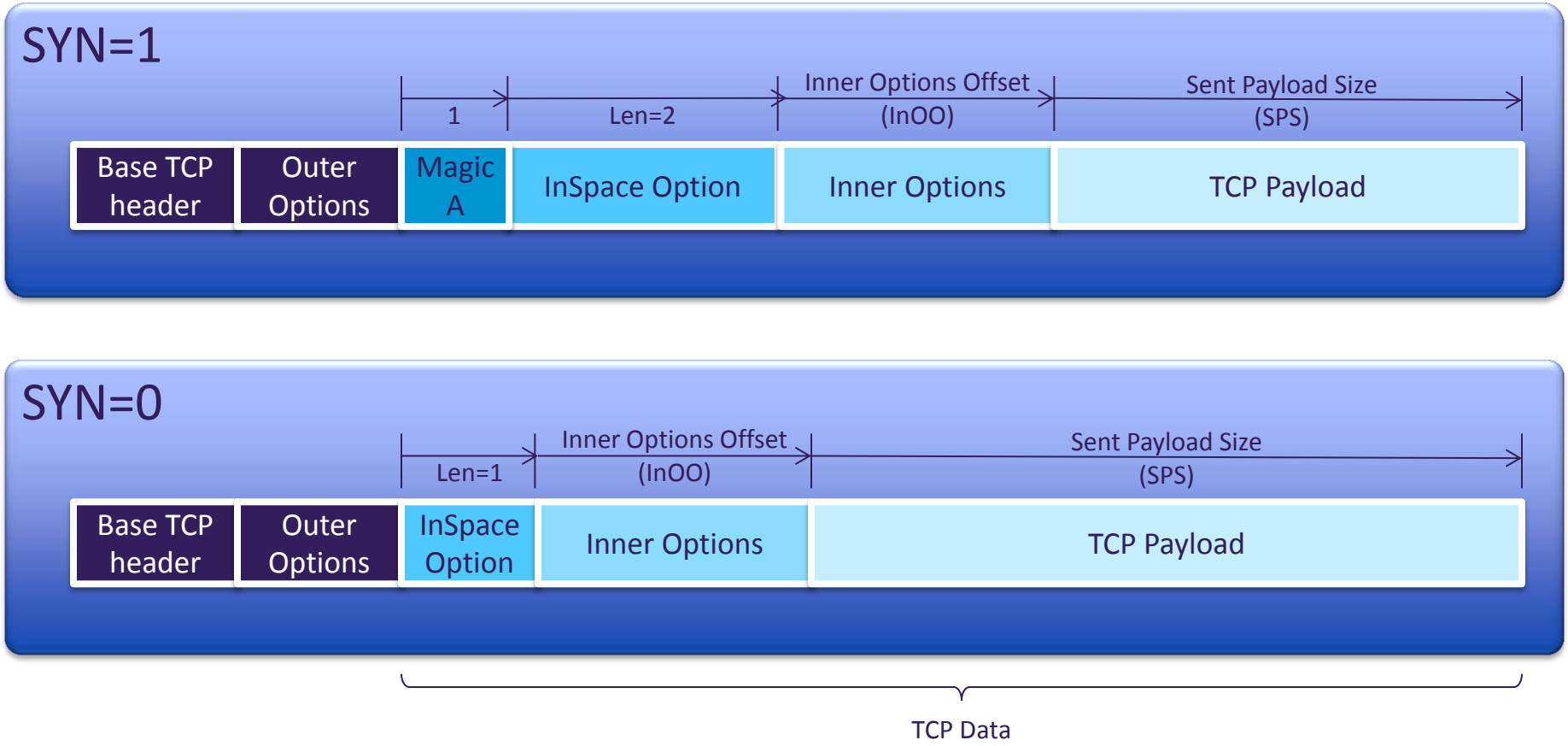
Inner Space – TCP segment structure (SYN=0)



- InSpace solely contains frame size info

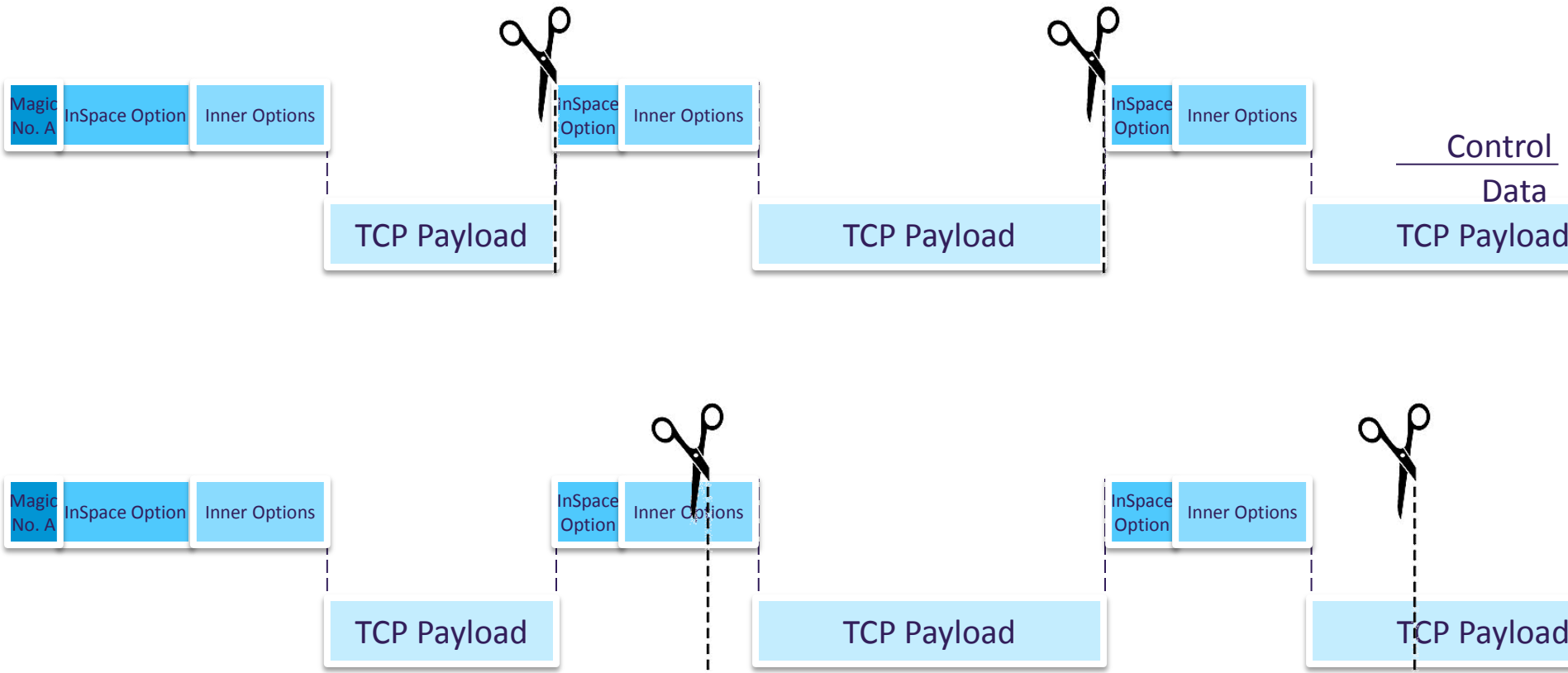
Inner Space – TCP segment structure

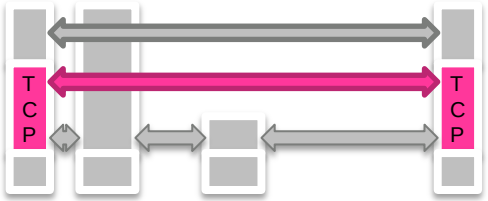




- presence of Inspace flagged by magic no. at start of each stream
- avoided an Outer TCP Option as the flag, which could be stripped
- inherently safe to flag within the payload – shares fate with options

Inner Space – TCP byte-stream



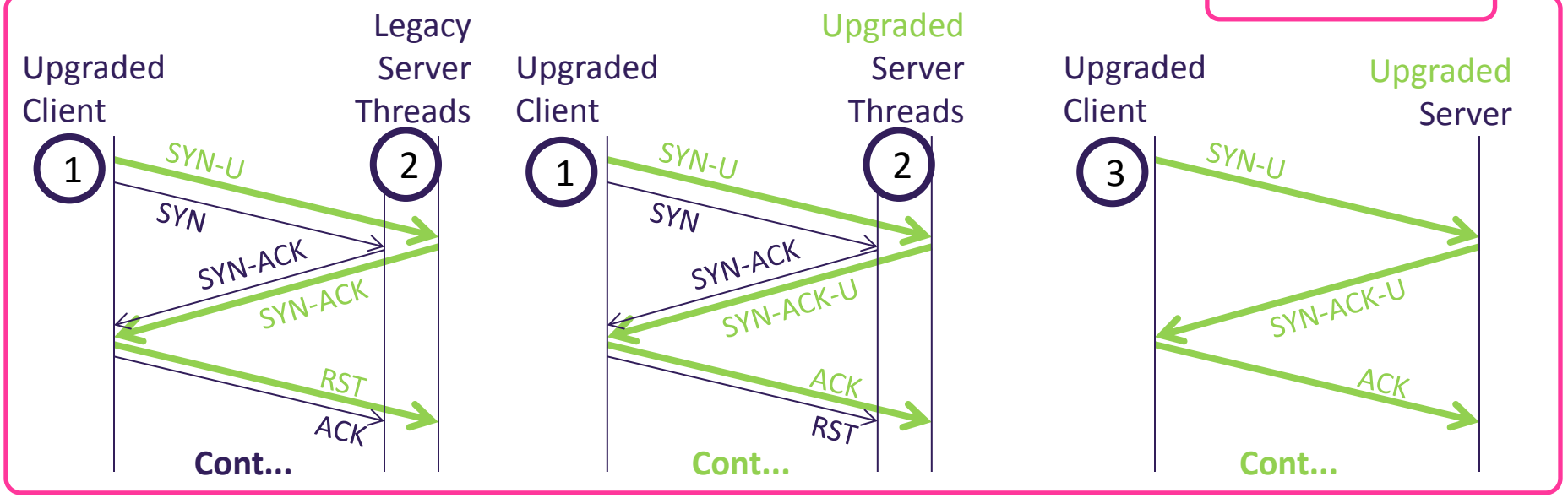


- robust to resegmentation
- Inner Options not prone to stripping
- reliable ordered delivery of Inner Options

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

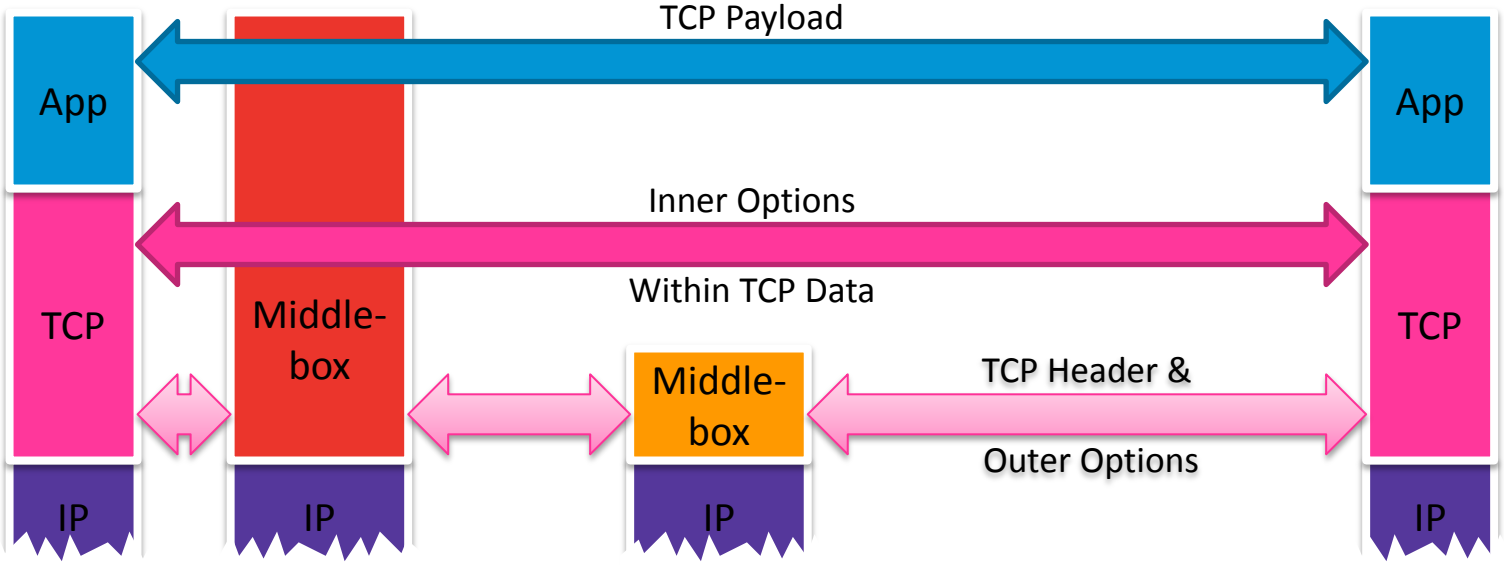
-U = upgraded,
i.e. magic no.
at start of TCP Data



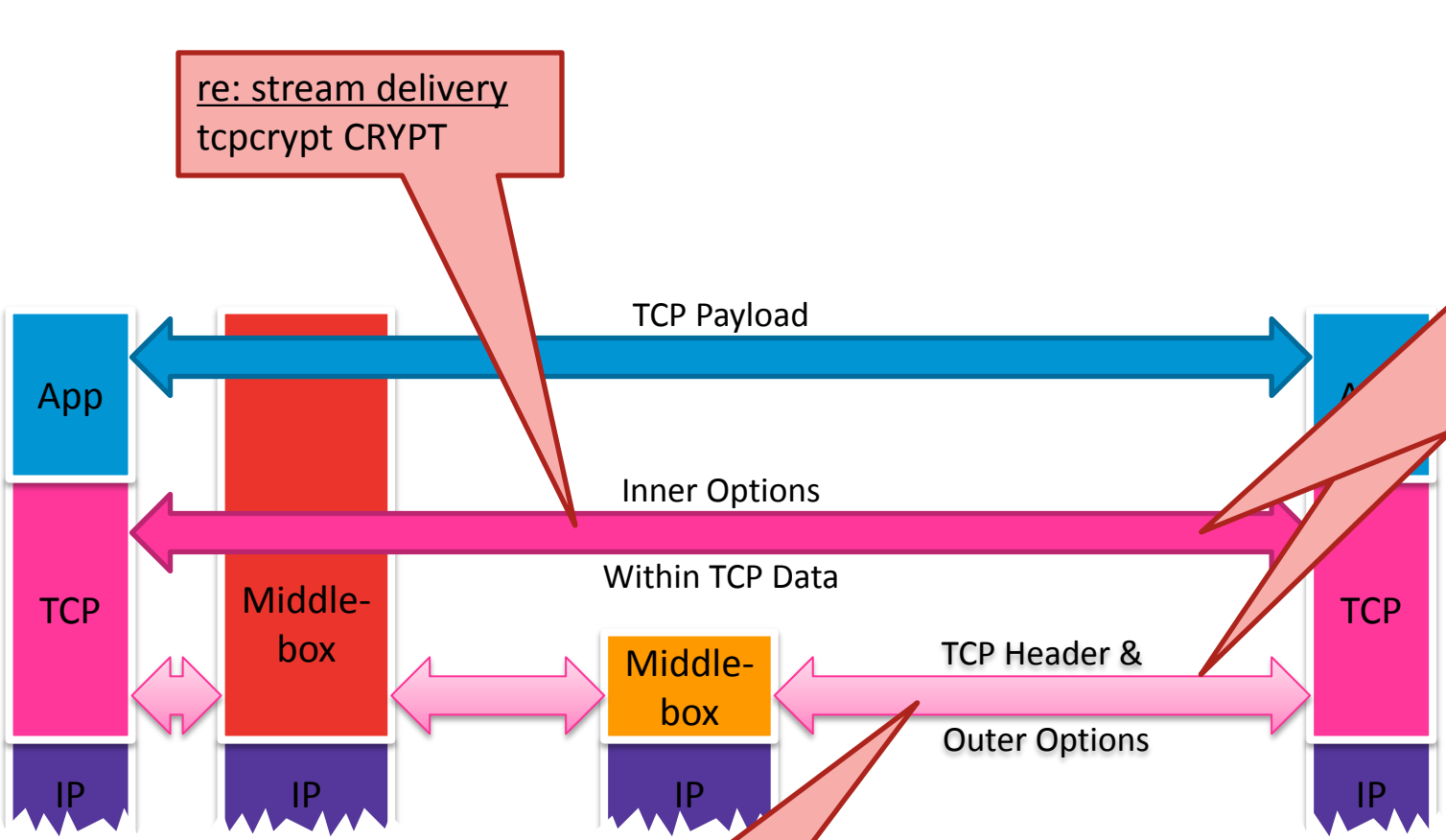
3. Can use single SYN-U handshake
   - when server is in cached white-list
   - once deployment is widespread (no need for white-list)

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

Inner Space – encapsulation model



Inner Space – applicability & compatibility*



* Many of the above schemes involve multiple different types of TCP option, see draft.

middlebox domination strategy

long term aim

- authenticate options

- if turned on option authentication today
  - ~10% 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

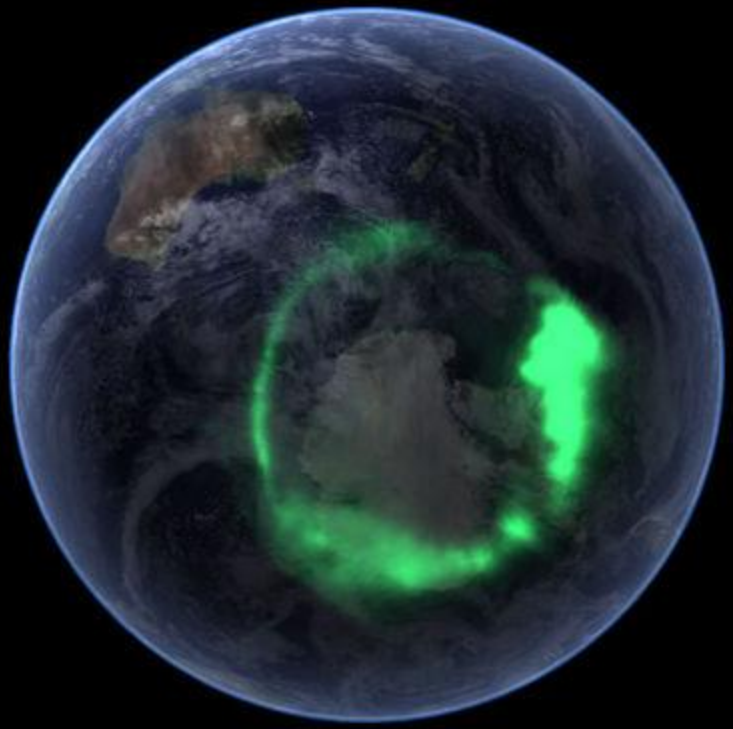
if you want to shoot them,
why shoot yourself in the foot
when you can make them shoot themselves in the foot?

summary

- make TCP options easy
  - they will just work
  - from the SYN onwards

next steps

- review with a view to adoption
  - focus: mandatory vs. optional elements
- path testing
  - data in SYN, is DPI bypass necessary? viable?
- implementation
  - compatibility testing
- IAB workshop on stack evolution in a middlebox Internet
  - principles



Inner Space

Q&A

Spare slides

menu

- Problem (= summary of benefits)
- Inner Space protocol
- Applicability / compatibility
- middlebox domination strategy
- Next steps

Spare slides

- Benefits & drawbacks
- Tricky bits
- Extensions

☹ drawbacks - overheads

		Example
• Dual Handshake		
– Latency	(Upgraded Server)	Zero
	(Legacy Server)	Worst of 2
– Connection Rate	P*D	8%
– Connection State	P*D/R	2.7%
– Network Traffic	2*H*P*D/J counting in bytes	0.03%
	2*P*D/K counting in packets	0.2%
– Processing	{pending implementation}	?
• Option on every non-empty segment		
– Network Traffic	P*Q*4/F	0.04%
– Processing	{pending implementation}	?

Example

P : [0-100%] proportion of connections that use extra option space	80%
D : [0-100%] proportion of these that use dual handshake	10%
R : [round trips] ave. hold time of connection state	3
H : 88B for IPv4 or 108B for IPv6 (see draft for assumptions)	
J : ave bytes per connection (in both directions)	50KiB
K : ave packets per connection (in both directions)	70 packets
Q : ave prop'n of InSpace connections that use it after handshake	10%
F : [B] ave frame size	750B

BT

☹ drawbacks - non-deterministic

- the magic number approach traverses option stripping middleboxes, but...
- probability that an Upgraded SYN or SYN/ACK is mistaken for an Ordinary Segment: Zero
- probability that an Ordinary SYN or SYN/ACK with zero payload is mistaken for an Upgraded Segment: Zero
- probability that payload data in an Ordinary SYN or SYN/ACK is mistaken for an Upgraded Segment: $<< 2^{-66}$
  (roughly 1 connection collision globally every 40 years)

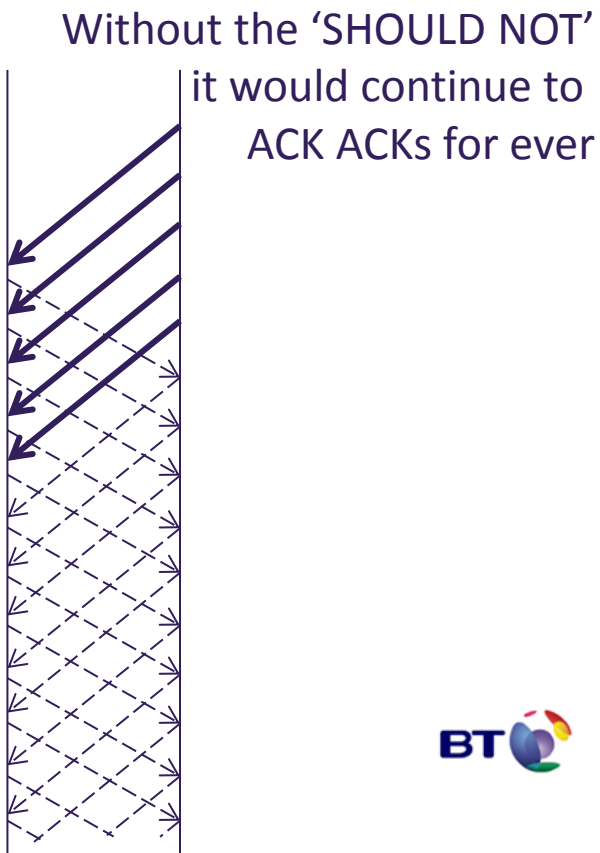
disabling Inner Space temporarily

- set Sent Payload Size (SPS) to special max value 0xFFFF
  - sent segment was not 0xFFFF octets, but behave as if it was
  - values above 0xFFE8 (= 2^{16} – 25) are usable but not believable
- regularly repeat just the 4B InSpace option
  - every 0xFFFF octets (=44.7 * 1466B typical full-sized segments)

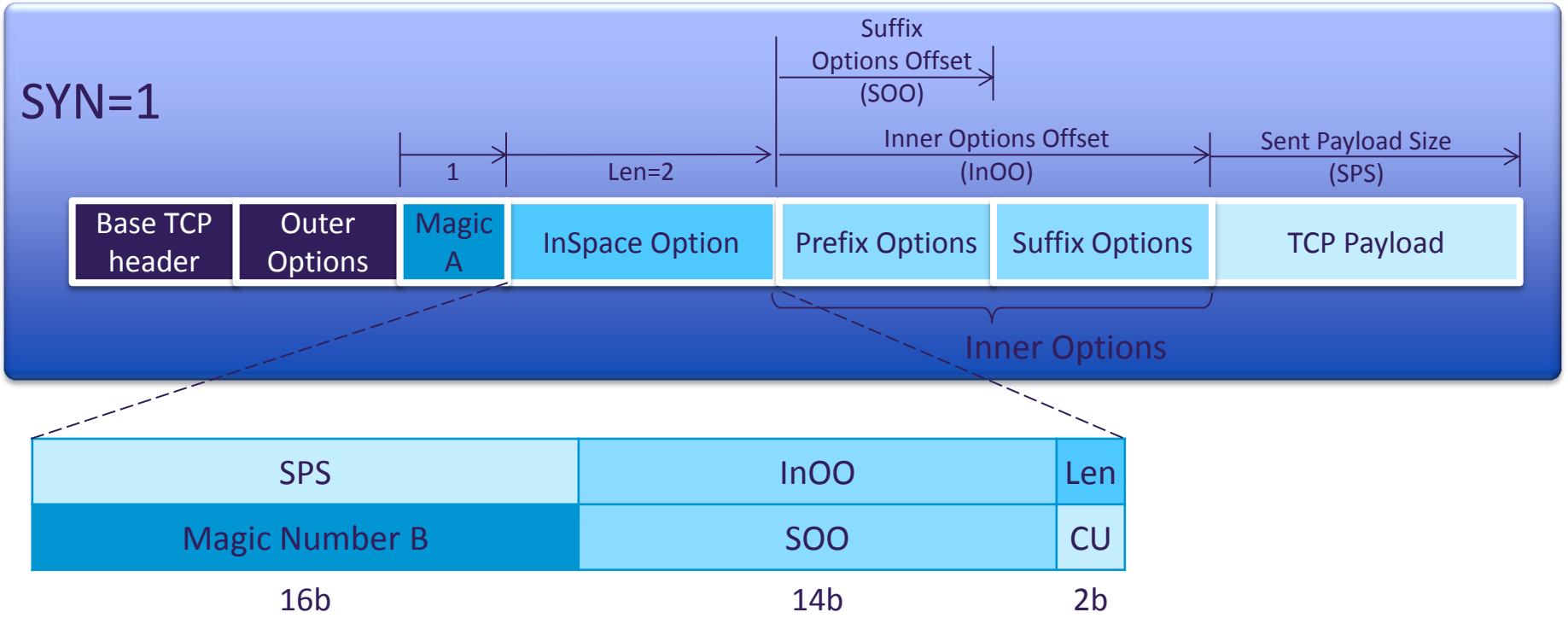
tricky bits - zero payload segments

- zero payload segments
  - MAY include an Inner Option
  - SHOULD NOT repeat the same Inner Options until more payload

- other tricky bits → spare slides or draft
  - option processing order
  - options that alter byte-stream
    - e.g. encrypt or compress
  - the EchoCookie for SYN floods
  - retransmissions during handshake



tricky bits – option processing order



- only on the first segment of each half-connection
  - on later segments, Outer Options have to be processed before Inner
  - reason: can't find Inner Options if still waiting to fill a sequence gap

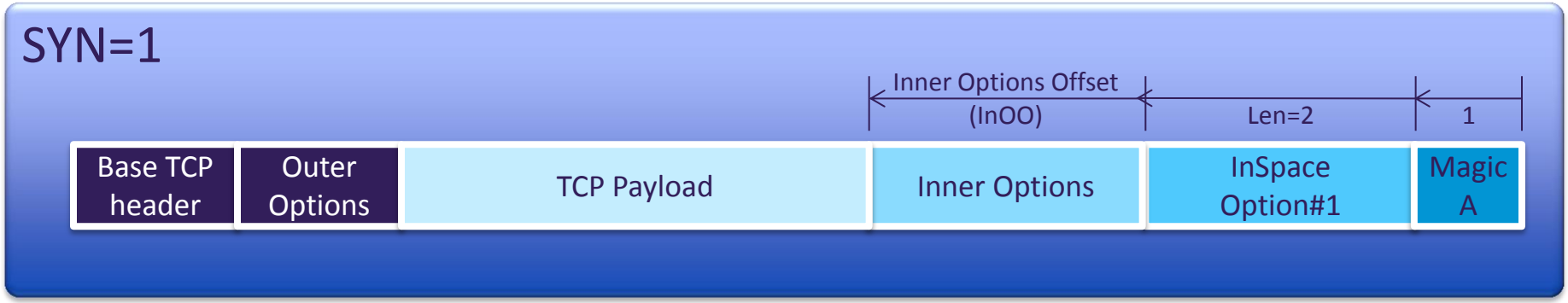
tricky bits – processing order: one level of recursion

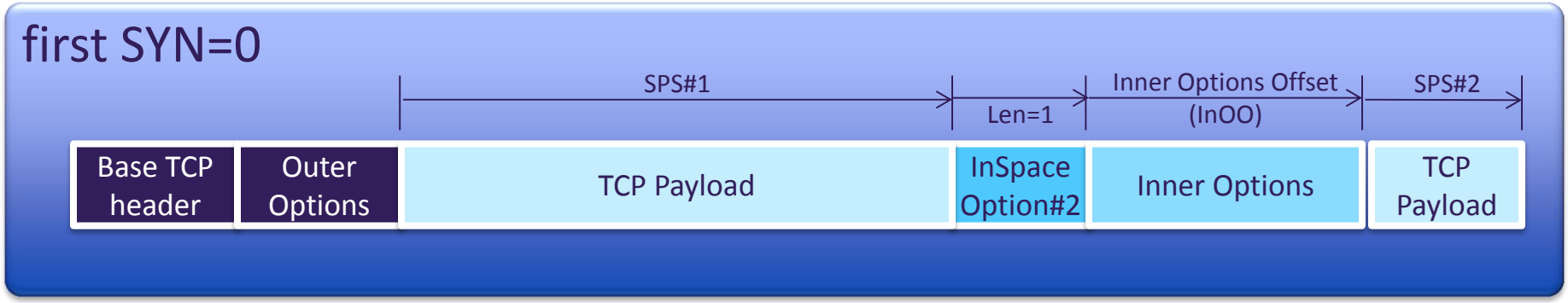
- If TCP alters the TCP Data (e.g. decrypt, decompress in the receiving case, for example)
  - SYN=1: if it hasn't previously found MagicA, it looks again

  - SYN=0: There might be a rekey command in an encrypted Inner Option. So the TCP receiver decrypts up to the end of each set of Inner Options, processes those options, then continues decrypting (which might be with a new key).

extension – DPI traversal

- conjecture: DPI often parses payload & stops when it finds what it needs
- solution?: locate MagicA at the end of the segment
  - server searches for MagicA at end if not at start





- can't work from the end of every segment, only the first
  - then use the spare first SPS (SPS#1) for the second segment

spare slides - to write, see draft

- handshake retransmissions
- explicit dual handshake
  - corner cases of dual handshake
  - deferred data in SYN

Extensions – summary of dependencies

- mandatory if implement Inner Space

EchoCookie TCP option

- extensions: optional while Inner Space is Experimental

- ModeSwitch TCP Option (scope wider than Inner Space)

- Explicit Dual Handshake (2 Outer TCP Options)

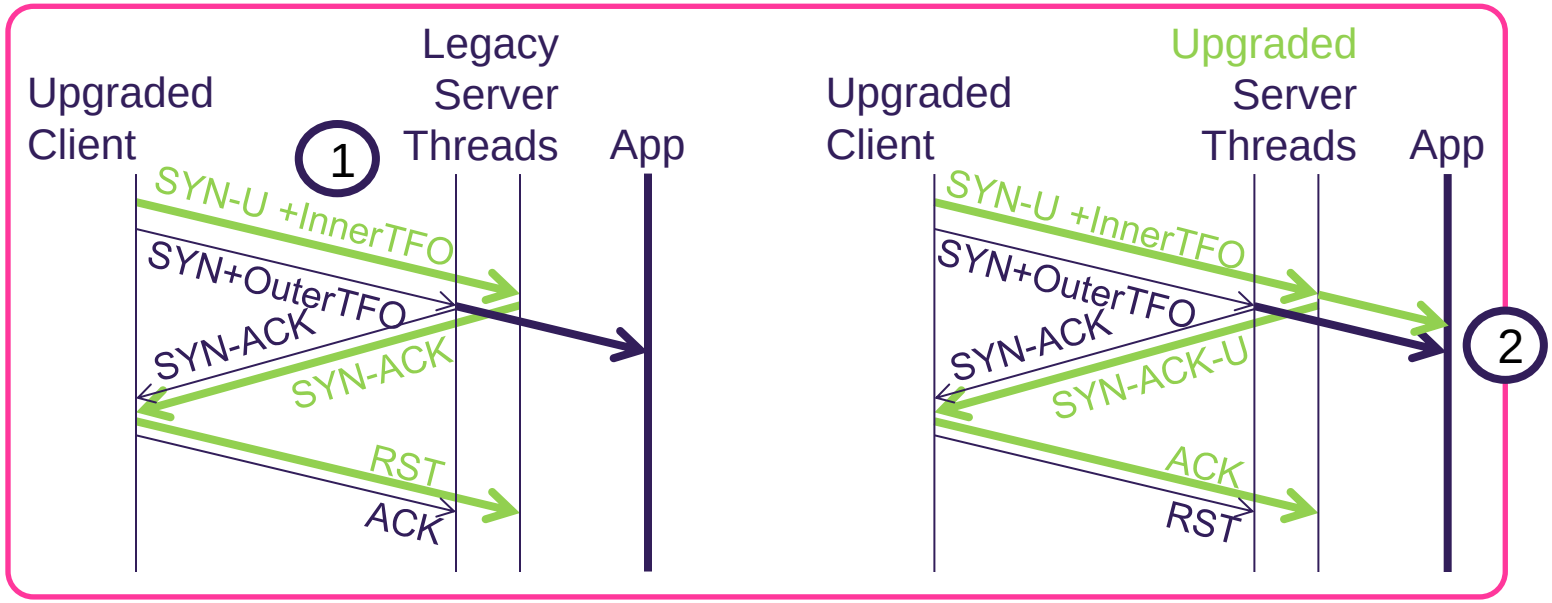
- Jumbo InSpace Option

- Inner Space segment structure for DPI traversal

see spare slides or draft

Inner Space & TCP Fast Open (TFO)

1. If Upgraded Client uses TFO
   - MUST place cookie in Inner of SYN-U
   - then Legacy Server will not pass corrupt TCP Data to app before RST



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

2. If dual h/s, Upgraded Server will pass payload to app twice
   - OK, because TFO only applicable if app immune to duplication

Inner Space & tcpcrypt

- tcpcrypt capability negotiation currently adds a round trip
  - **not viable** to add 1RTT delay to every connection to introduce opportunistic encryption

- tcpcrypt currently attempts most of Inner Space
  - in various complicated bespoke ways
  - have proposed how to structure tcpcrypt over Inner Space
  - cuts 1.5 rounds → **makes tcpcrypt viable**
  - cuts out two states – greatly simplifies
  - (?) decouples tcpcrypt from TCP state m/c
  - tcpcrypt can encrypt Inner Options (incl. its own)
    - because that needs reliable ordered delivery

Inner Space & MPTCP

- MPTCP adds Data ACK (in the DSS TCP Option)
  - cumulative ACK of the set of sub-flows cannot be inferred
- Data ACK is a per-segment message
  - cannot use Inner Options
    - would not be interpretable on reception (if out of order)
    - potential deadlock: must not require receive buffer to ACK [1]
- Three ways forward
  1. give up – leave all MPTCP TCP Options as Outer Options
  2. use Inner Space for a low latency MPTCP,
     except DSS and a way to test path for stripping DSS
  3. extend Inner Space to include Outer Options within TCP Data
     without using RWND or sequence space (hard – see next)

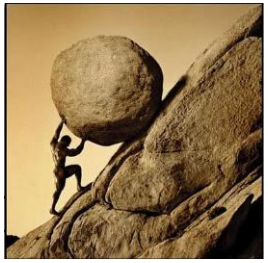
[1] Raiciu et al "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP"

BT

opportunities / further work

- tcpcrypt-v2 decomposition
- probes
  - any Inner Options delivered reliably in order
- relation to Minion, and multi-stream protocols

- Outer Options in Inner for middlebox traversal



- without consuming rwnd (cf. fixed space for Outer Options)
- without consuming sequence space (avoiding middlebox 'correction')
- delivered immediately in received order, not sent order