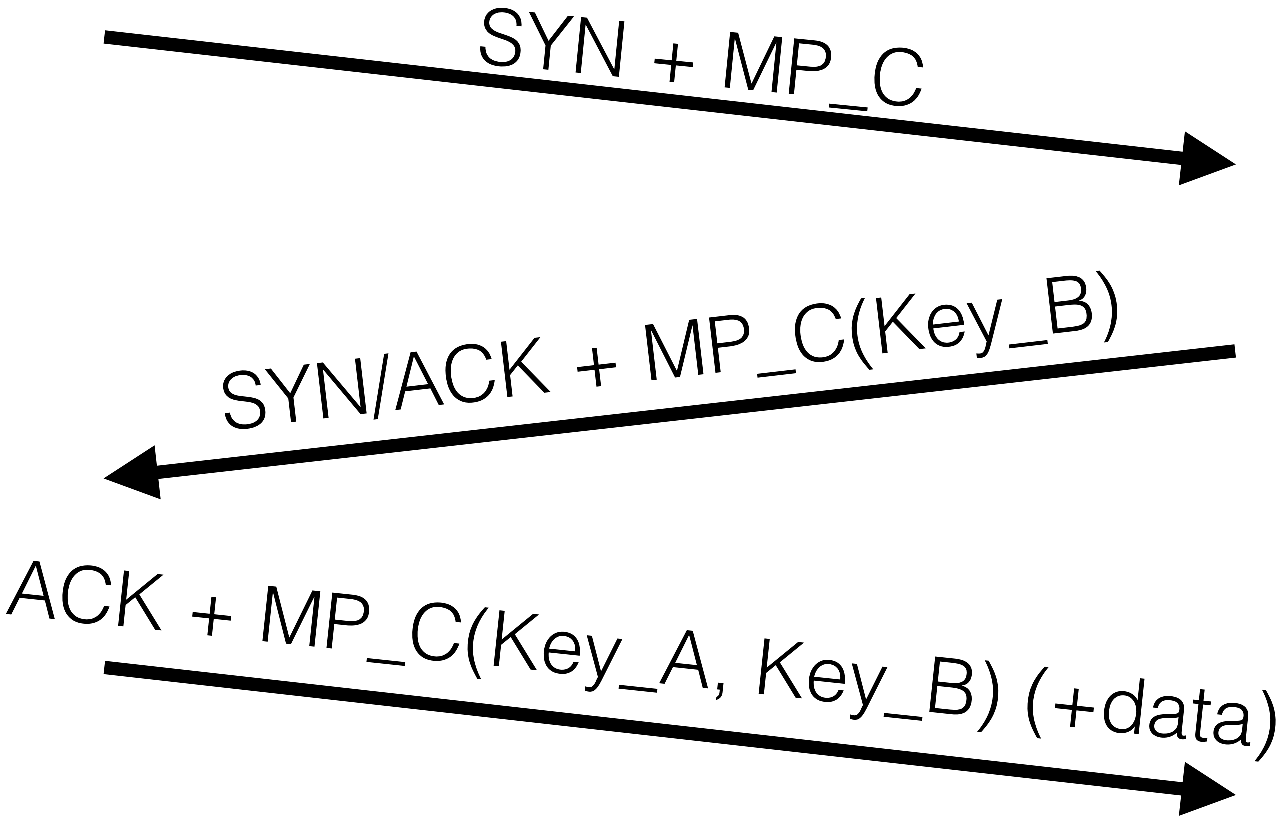
Updating the MPTCP Handshake

Christoph Paasch <cpaasch@apple.com>

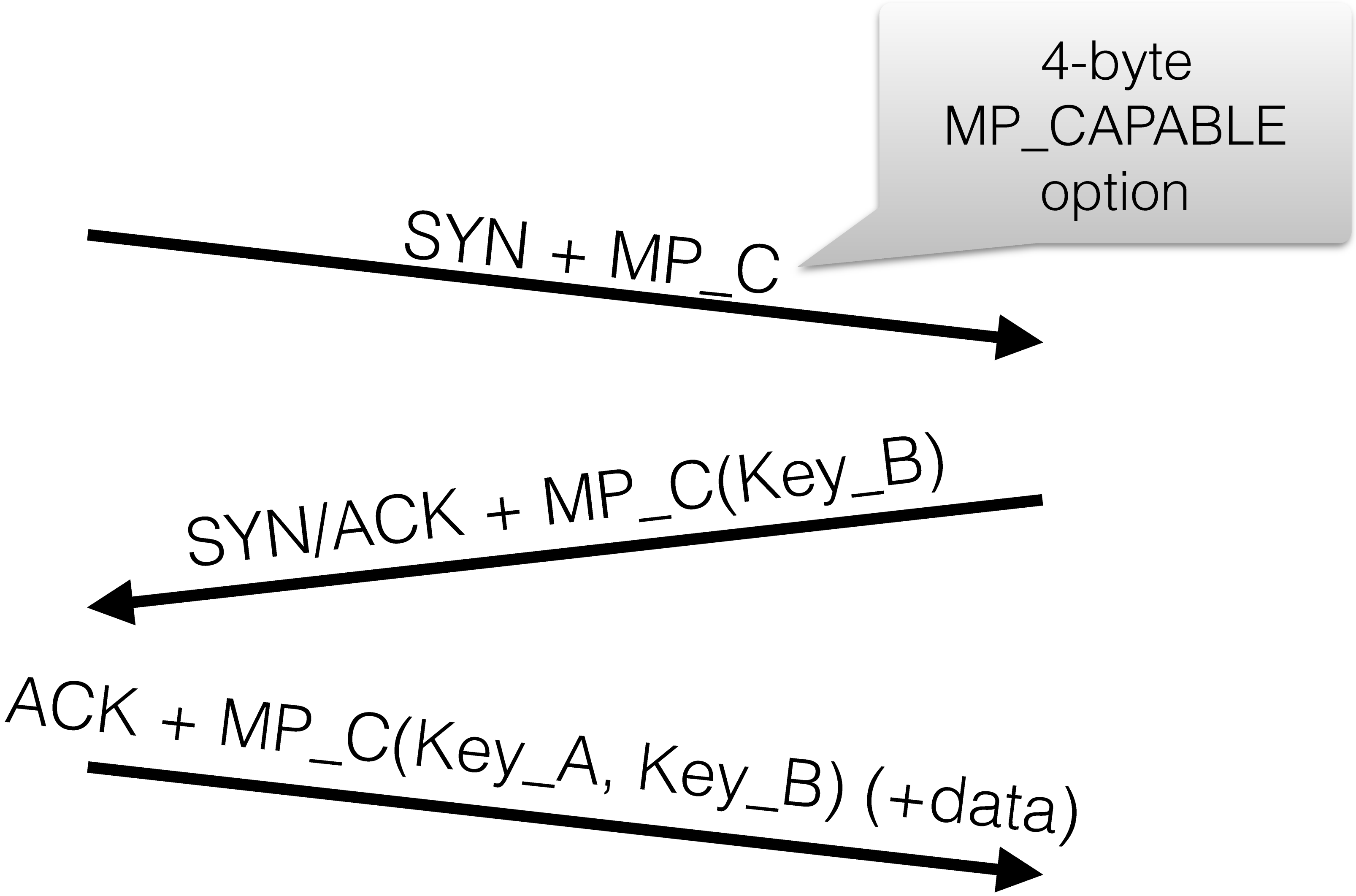
RFC6824bis - v05

Updated handshake, to allow SYN-COOKIE support:



RFC6824bis - v05

Updated handshake, to allow SYN-COOKIE support:



Evolving the handshake

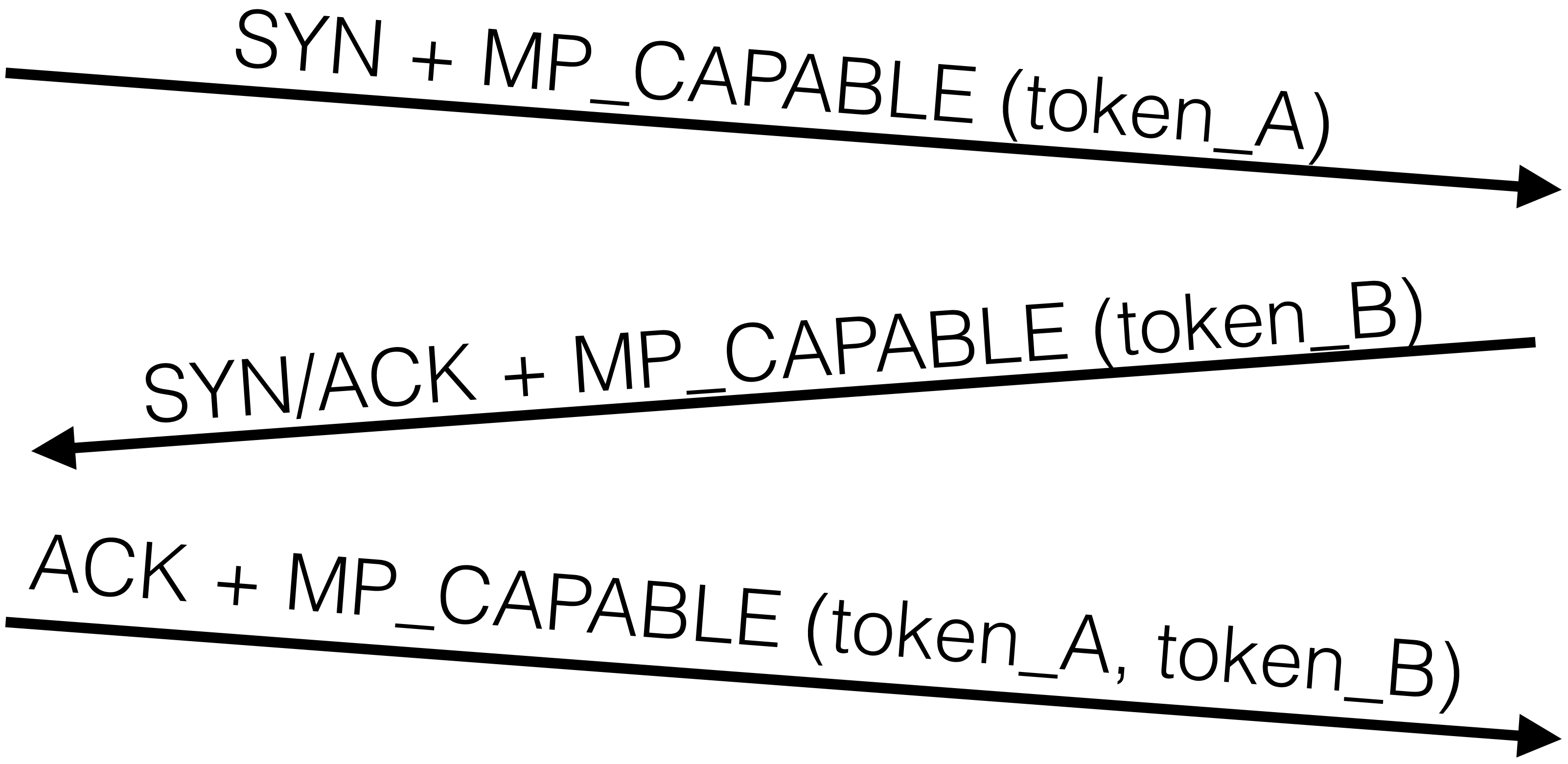
- 4-byte MP_CAPABLE -> Plenty of space to use :)
- From previous meeting:

  Separate the token from the key

  - Benefits load balancers
  - Benefits token-generation (easy to guarantee uniqueness)

Separate Token from Key

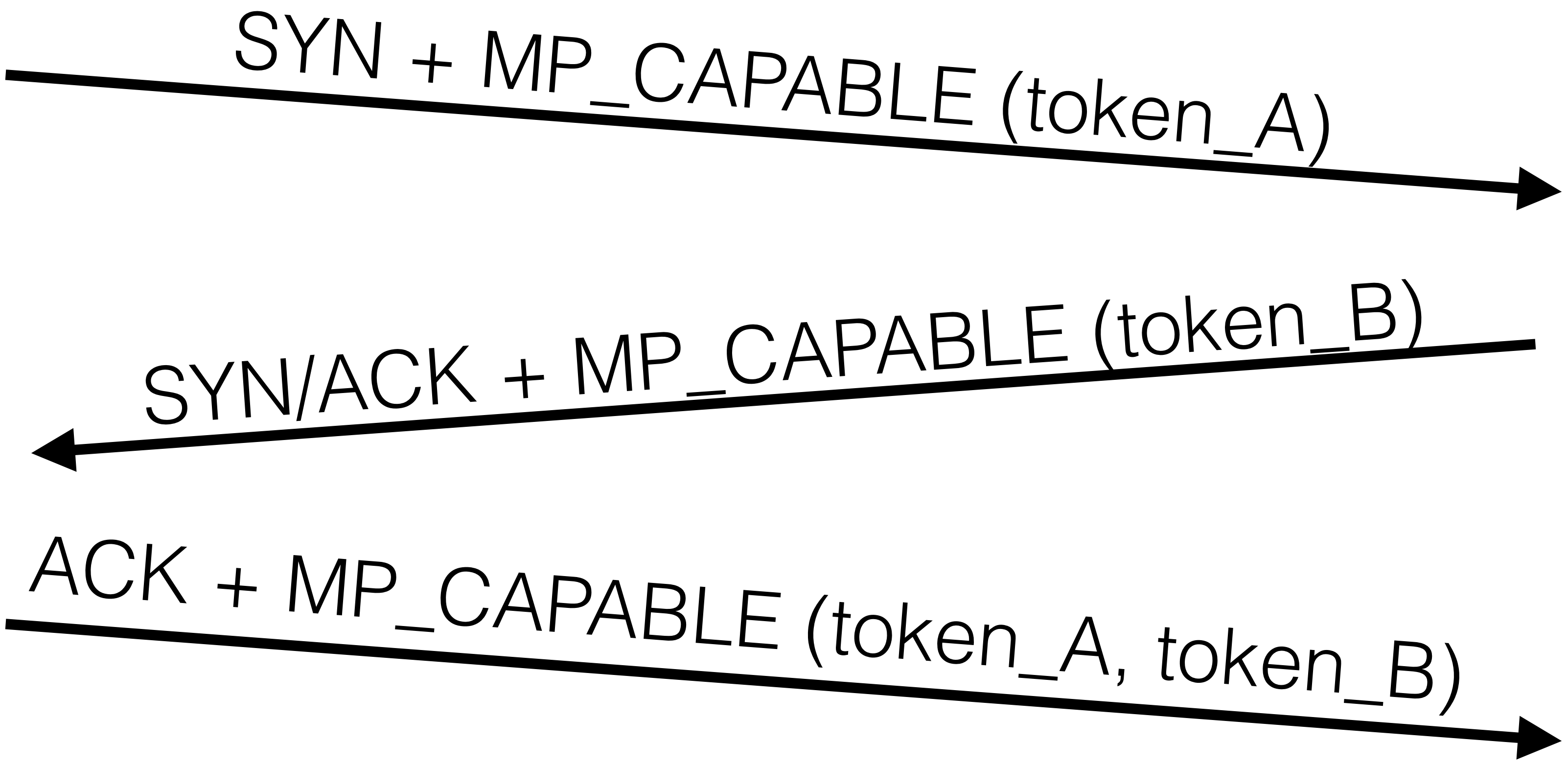
1. Specify token explicitly in 3-way handshake:



SYN + MP_CAPABLE (token_A)
SYN/ACK + MP_CAPABLE (token_B)
ACK + MP_CAPABLE (token_A, token_B)

Separate Token from Key

1. Specify token explicitly in 3-way handshake:



SYN + MP_CAPABLE (token_A)

SYN/ACK + MP_CAPABLE (token_B)

ACK + MP_CAPABLE (token_A, token_B)

- No more space for the key
- Key could come from out-of-band (e.g., TLS)
- Still need a “fallback” for non-TLS traffic

Separate Token from Key

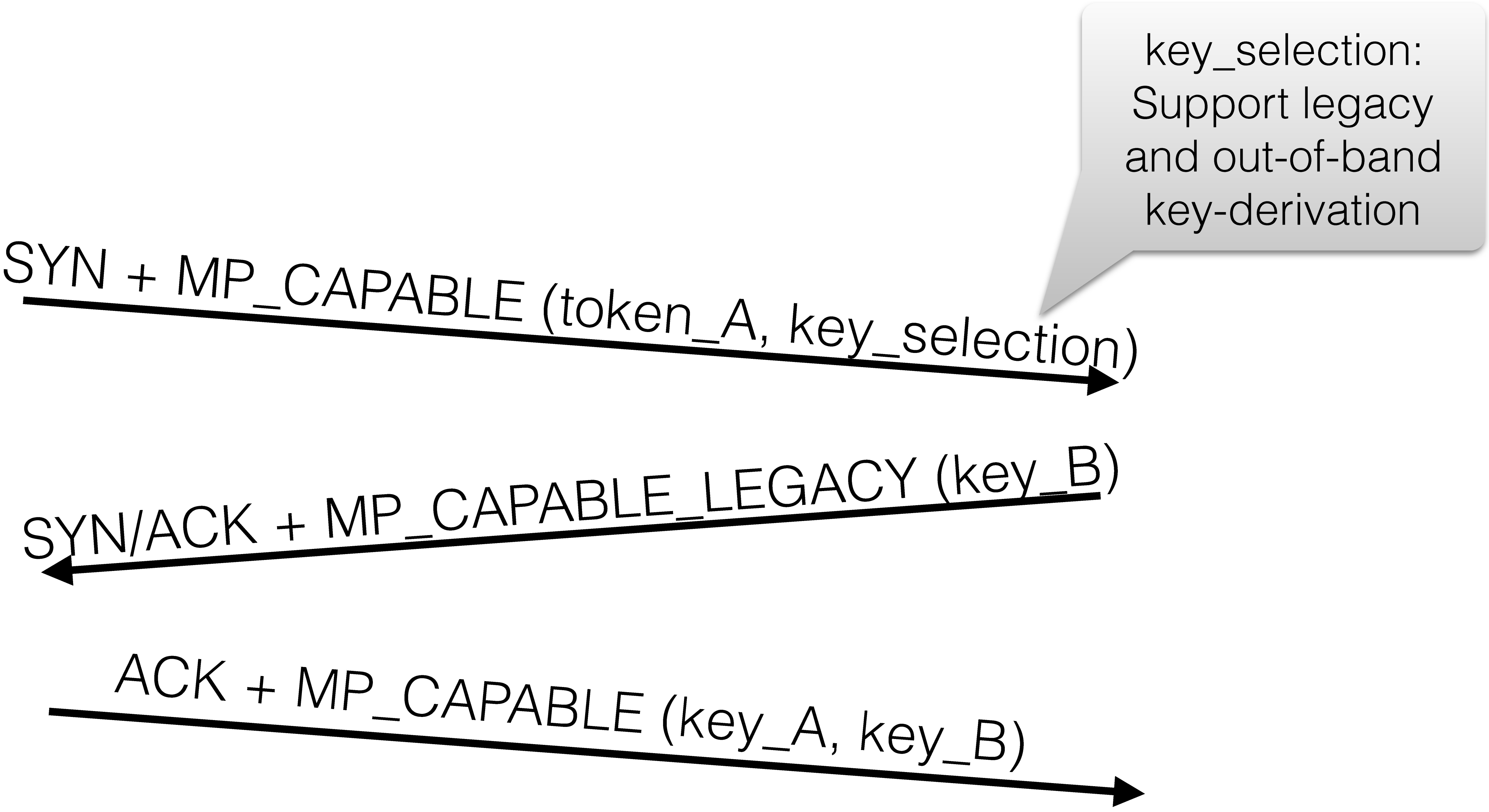
2. Key-selection during handshake

    - Client announces its supported key-selection methods

    - Server decides which one to choose

    - allows “legacy”-fallback

Legacy-fallback



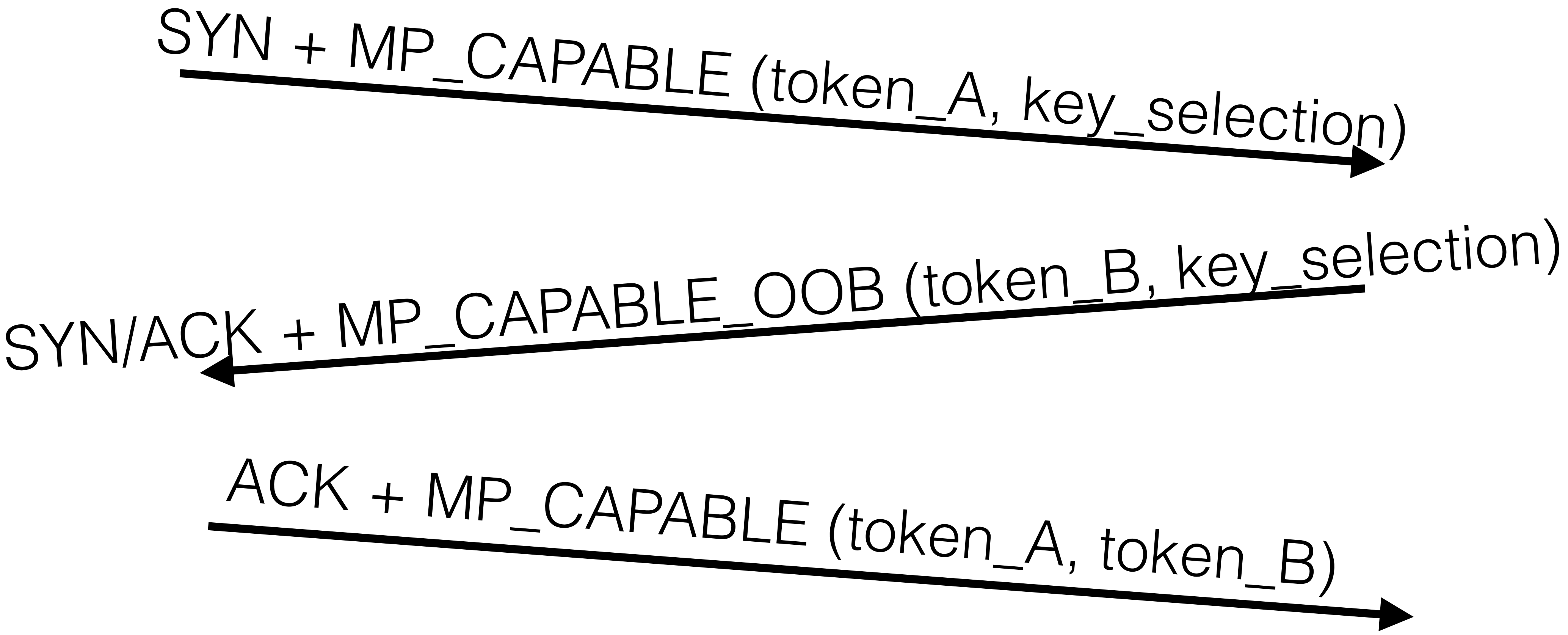
key_selection:
Support legacy
and out-of-band
key-derivation

SYN + MP_CAPABLE (token_A, key_selection)

SYN/ACK + MP_CAPABLE_LEGACY (key_B)

ACK + MP_CAPABLE (key_A, key_B)

Out-of-band key



SYN + MP_CAPABLE (token_A, key_selection)

SYN/ACK + MP_CAPABLE_OOB (token_B, key_selection)

ACK + MP_CAPABLE (token_A, token_B)