Connection Signaling

Mark Handley
UCL

The Problems

- Mobility
- Multihoming
- Firewalls
- NATs
- Address Spoofing
- DoS Attacks

Departures from End-to-End

- Mobility
  - Need to find host, need to re-bind connections.
- Multihoming
  - Need to bind connection to more than one path without affecting global routing.
- Firewalls

  Middle cares about connections.
- NATs
  - Middle cares about connections, rewrites addresses
- Address Spoofing
  - Prevention involves the middle, detection involves the middle.
- DoS Attacks
  - End can’t defend itself - needs to involve the middle.

Connections

- Perhaps the traditional self-contained TCP model of a session connecting a pair of IP addresses and ports needs revision?

WARNING!

LESS-THAN HALF BAKED IDEAS COMING UP.

IGNORES RELATED WORK.

MAY TREAD ON OTHER PEOPLE'S TURF.

CONTENTS MAY BE HOT.

Philosophy and Assumptions.

- IP Addresses are primarily *addresses*.
  - Identify a location in the network.
  - Should be possible to aggregate routes .
- Transport protocols should be capable of supporting *address and port rebinding*.
  - Before/during connection establishment.
  - In mid connection.

  Plenty of work on this - definitely feasible for TCP, SCTP, DCCP. Feasible for UDP *flows*.

Strawman:

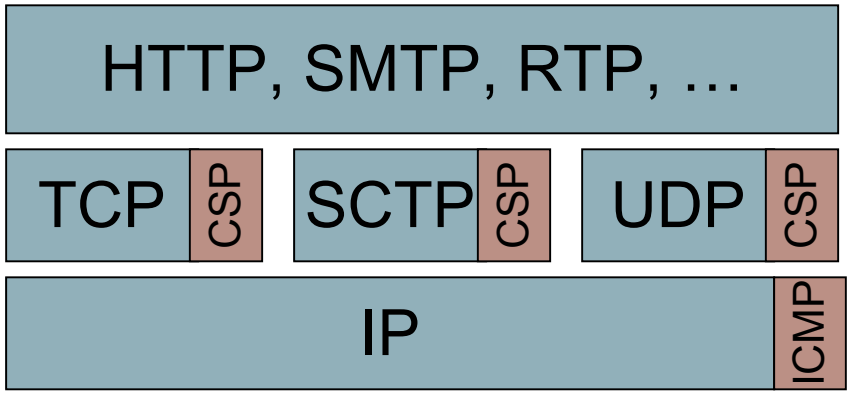
Connection Signaling Protocol (CSP)

- Assume that we use a general purpose Connection Signaling Protocol to signal every transport connection.
  - Intent is *not* to build virtual circuits.
  - Provide a clean place in the architecture to:
    - Signal the application's intent to middleboxes.
    - Signal the middleboxes intent to end hosts.
    - Locate mobile end-systems and signal mobility to everyone.
    - Signal alternative path information to end-systems.
    - Handshake between end-systems before trusting them.
    - Signal middleboxes to deny service.

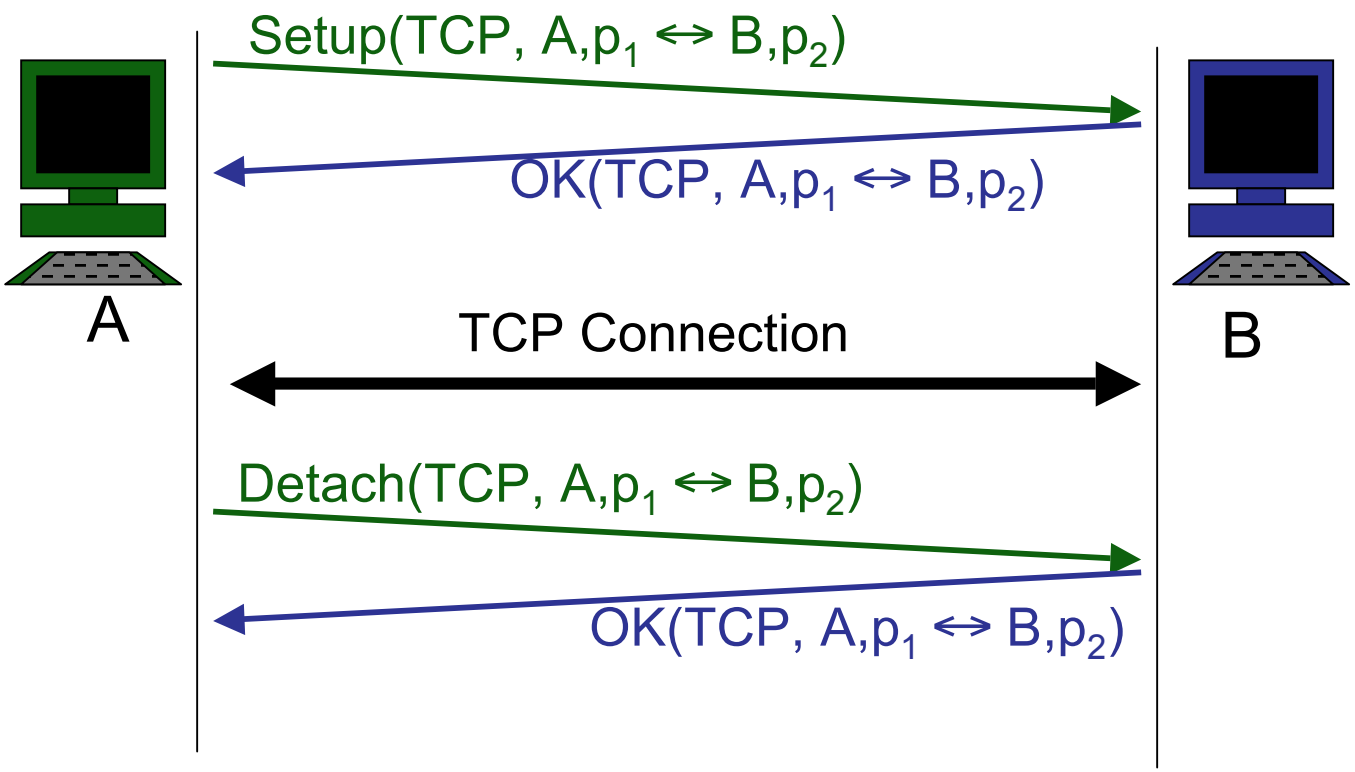
Stack

CSP is not strictly layered under or over transport protocols.

- More like alongside.
- Akin to how ICMP is to IP.

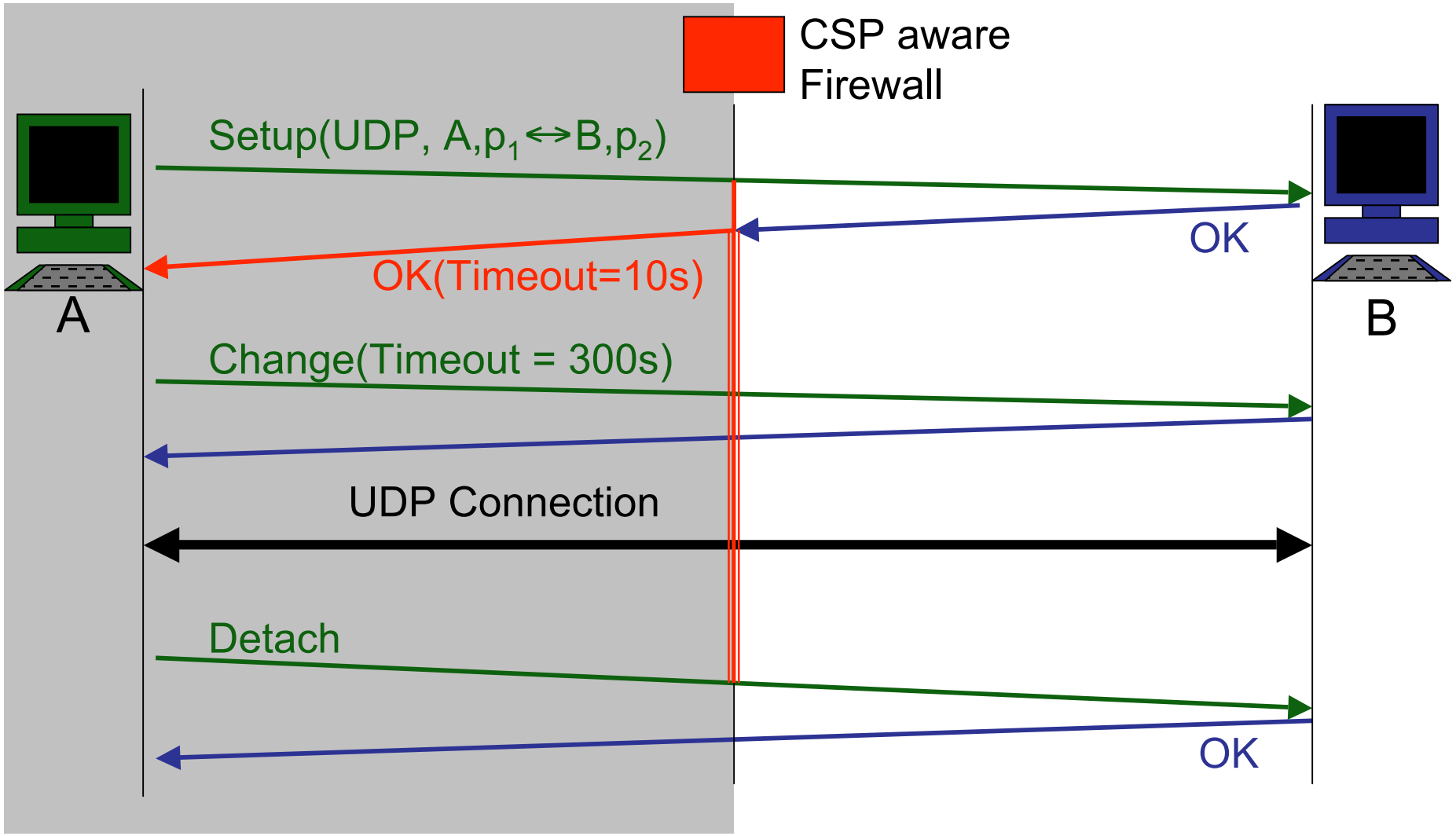


Simple Connection

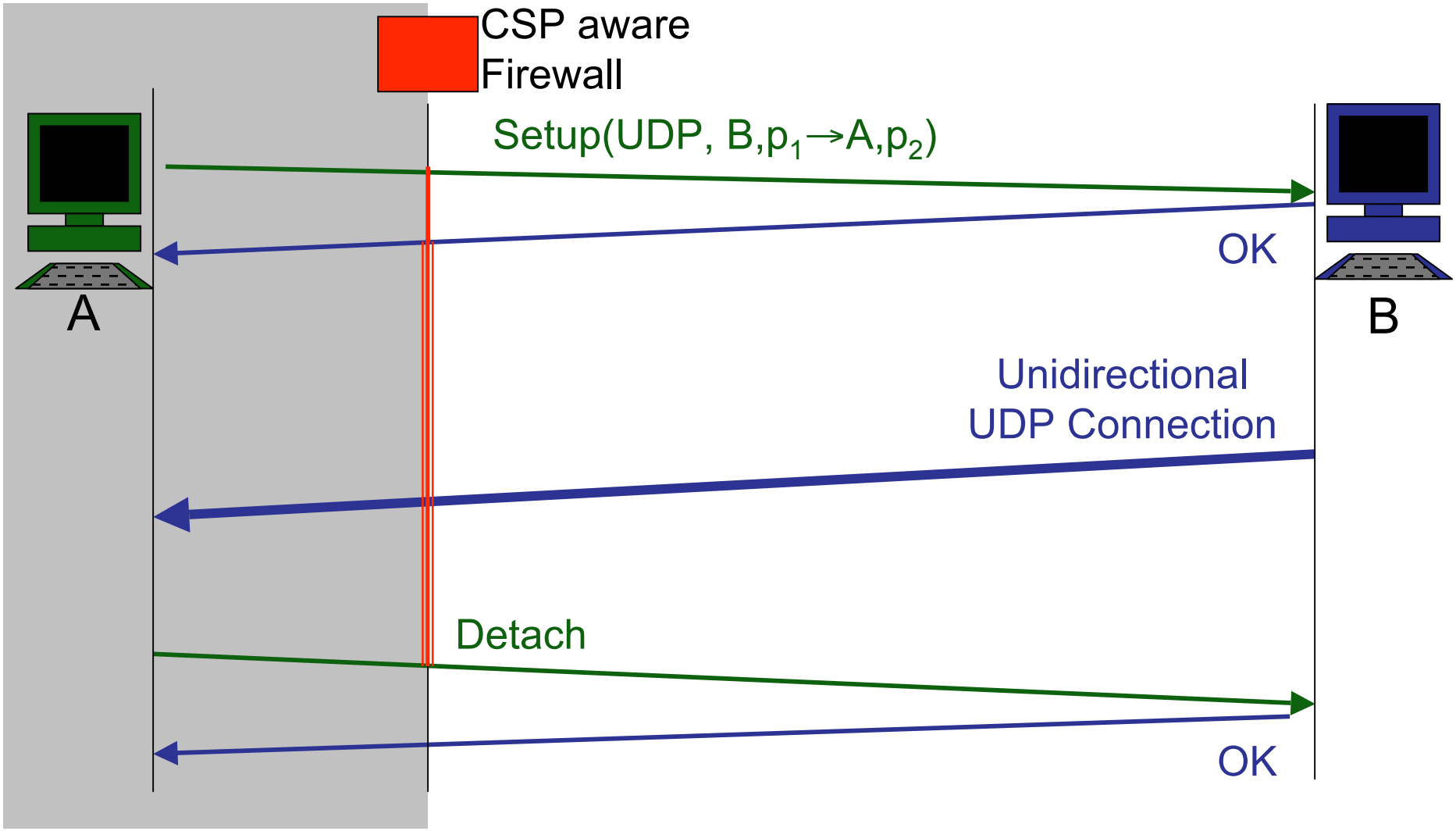


- May be able to piggyback first data packet on signaling.
  - Will ignore optimizations for now.

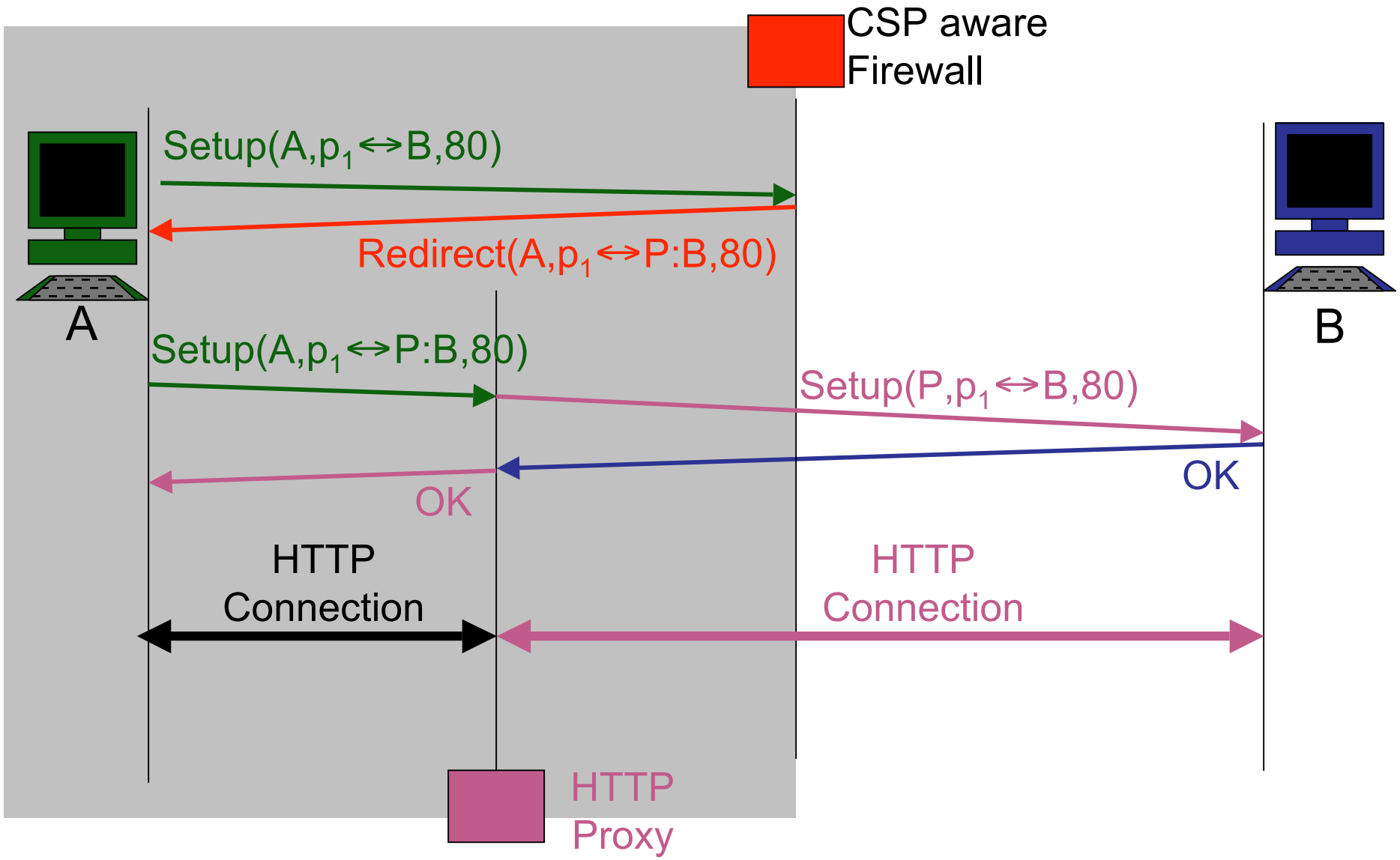
Simple Firewalled UDP Connection



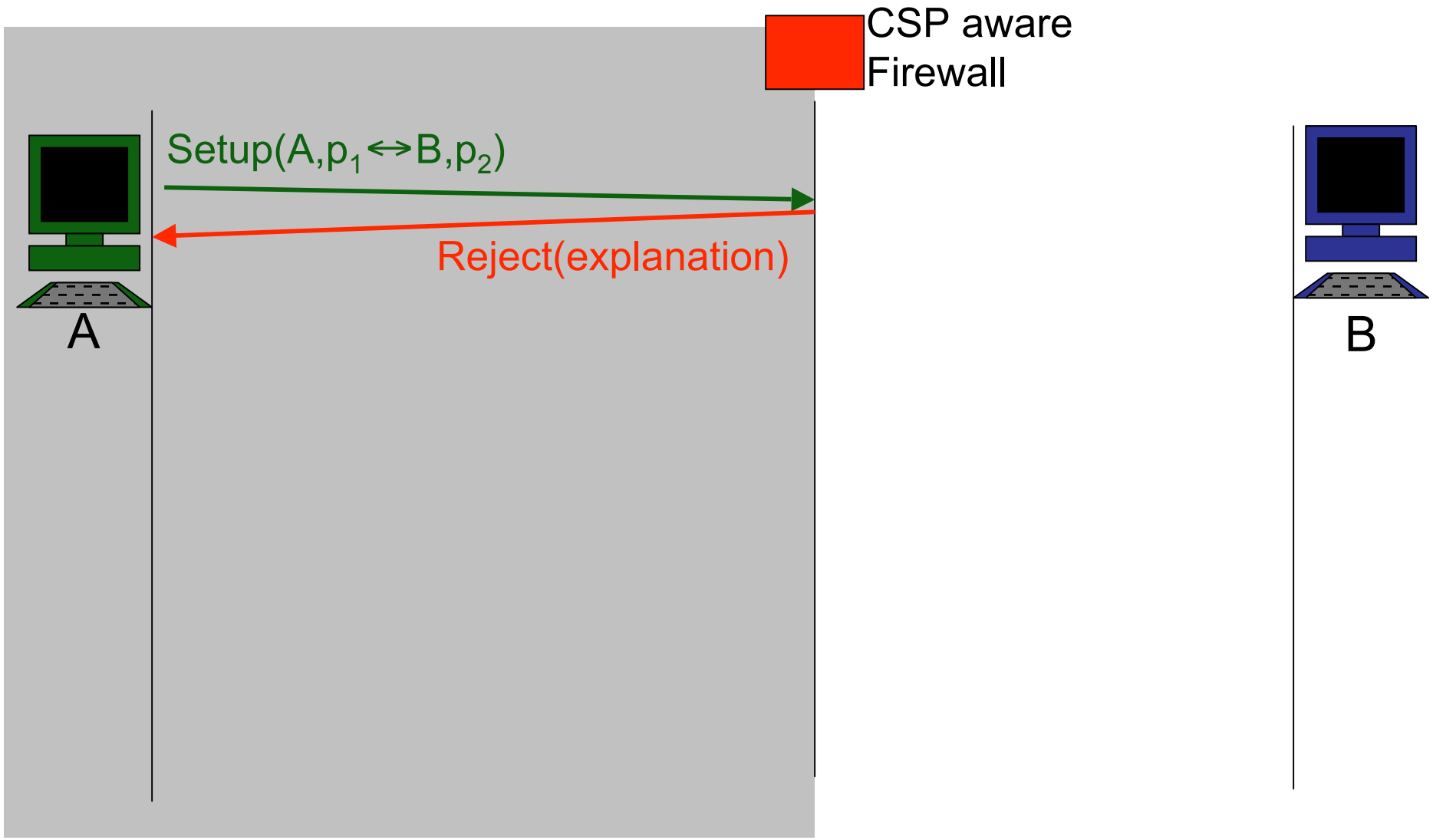
Firewalled Incoming UDP Connection



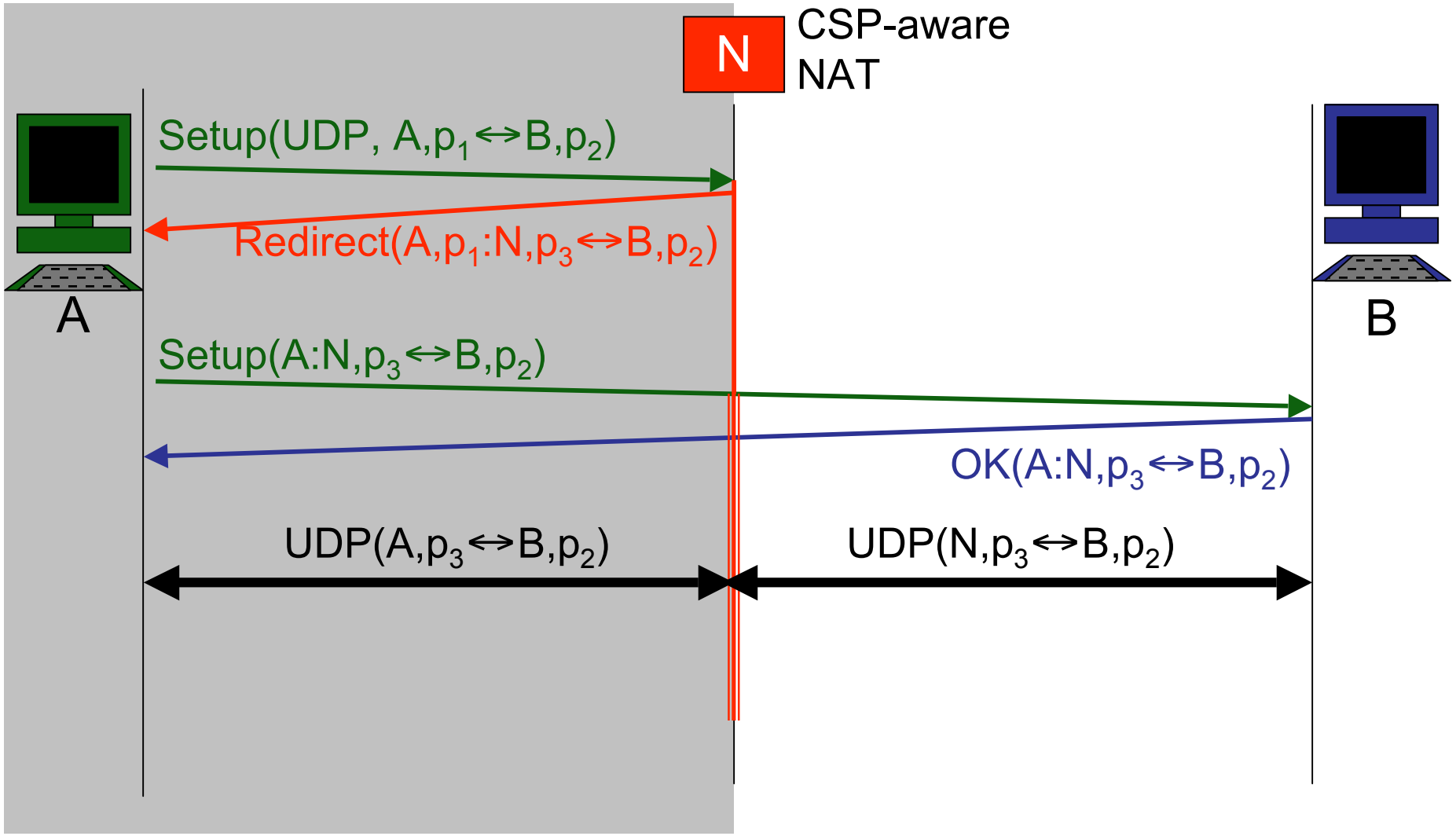
Firewall redirect to offpath proxy



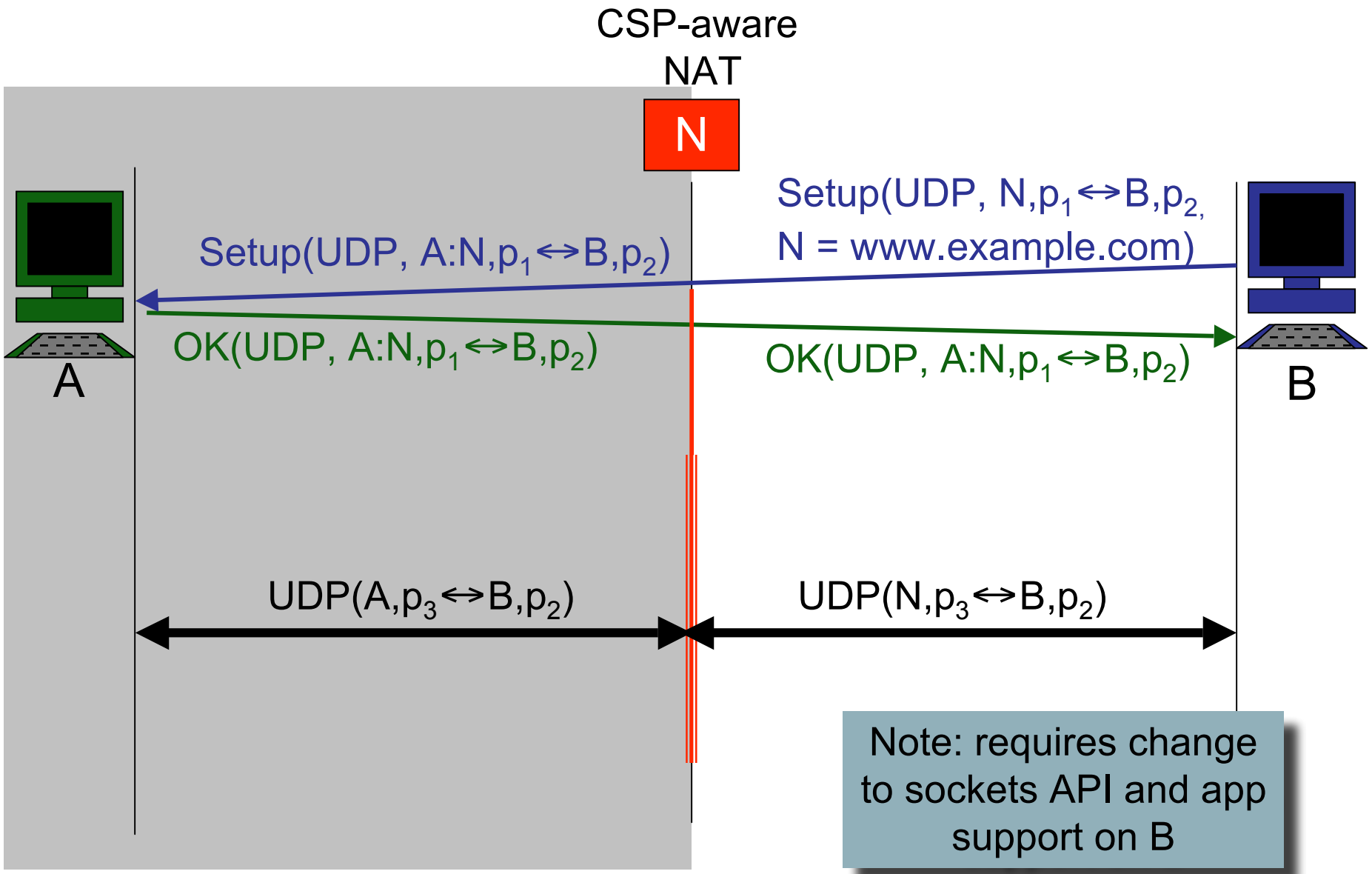
Firewall rejection



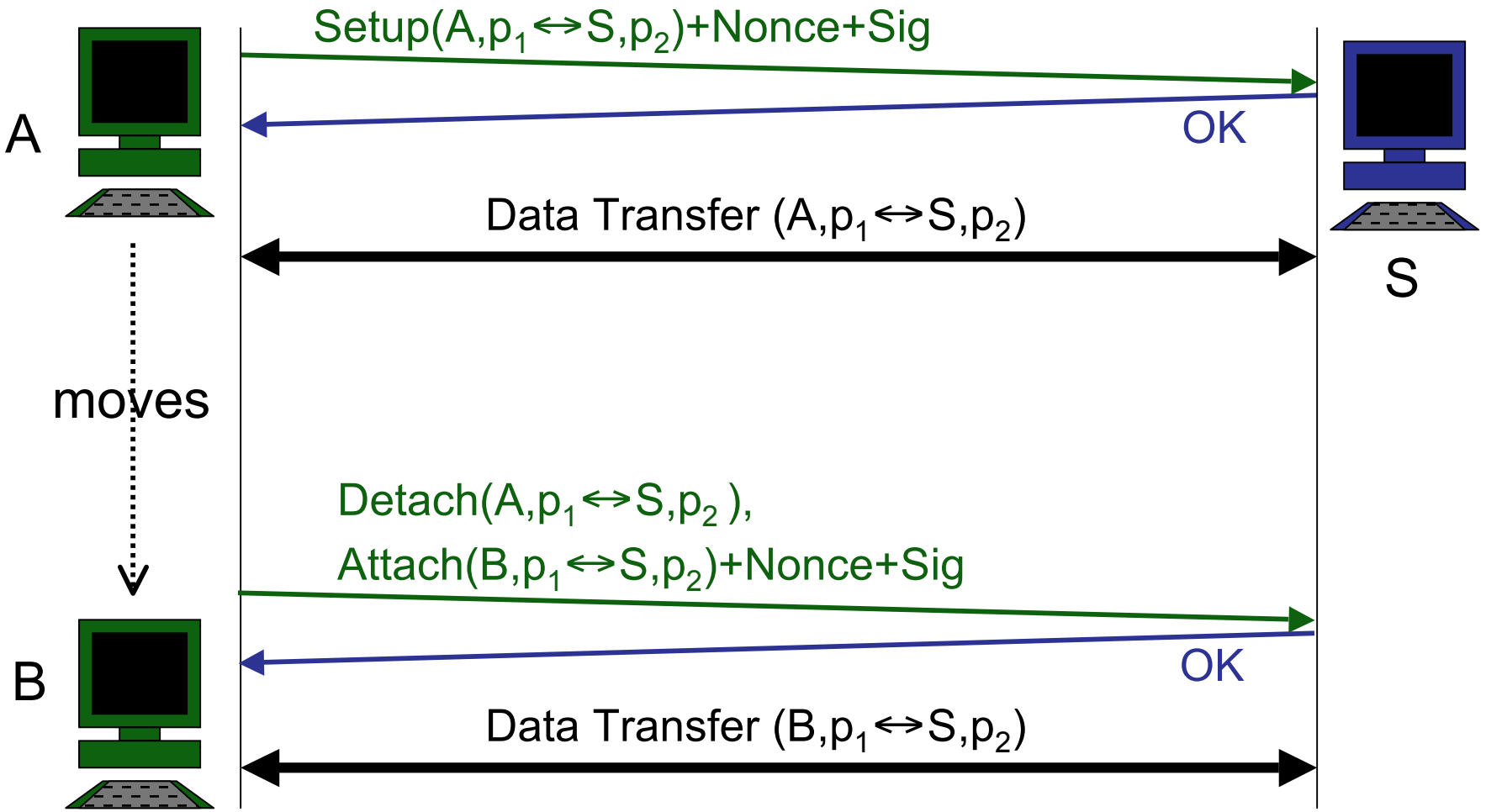
NAT Traversal (1)



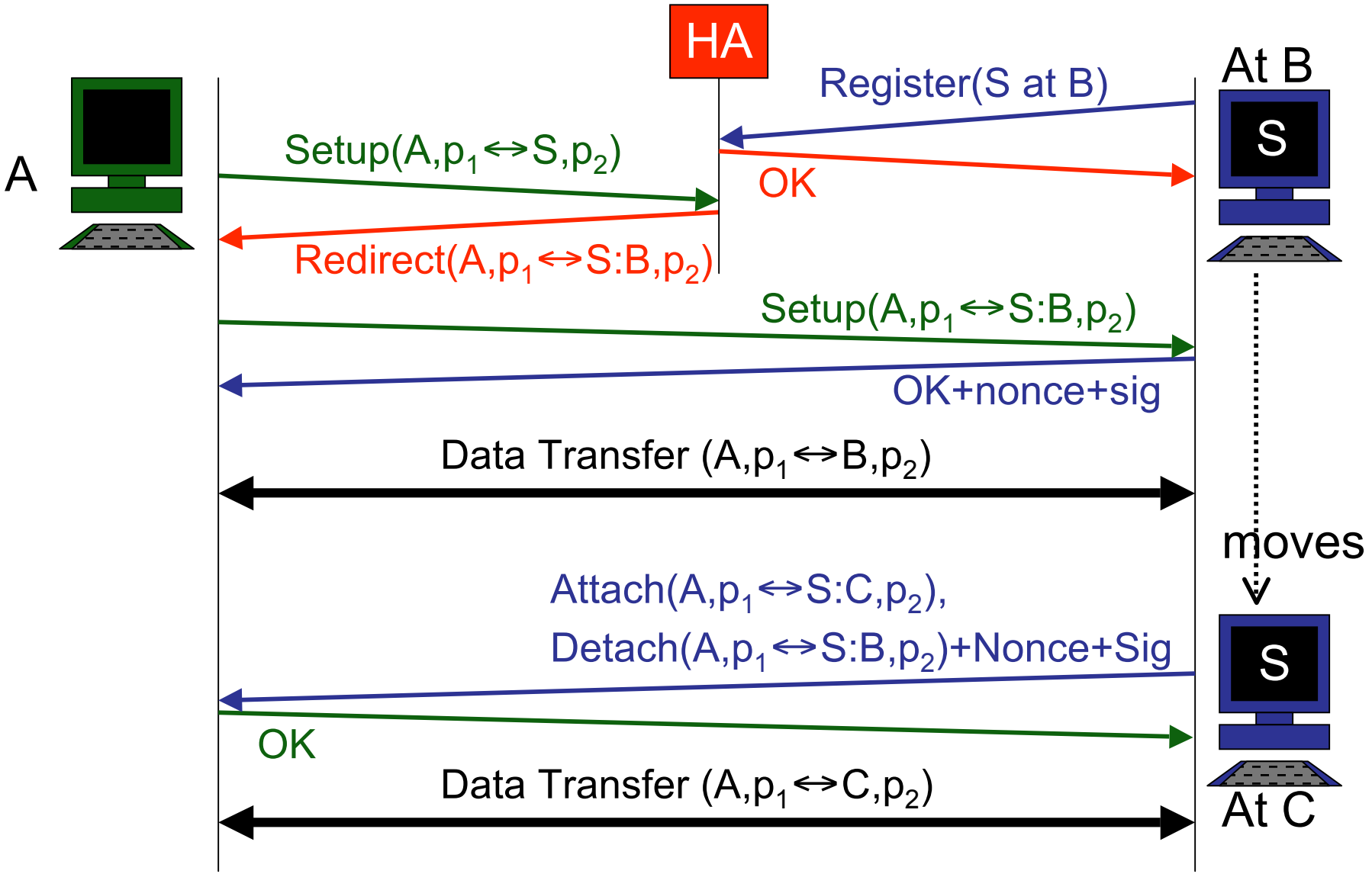
NAT Traversal (2)



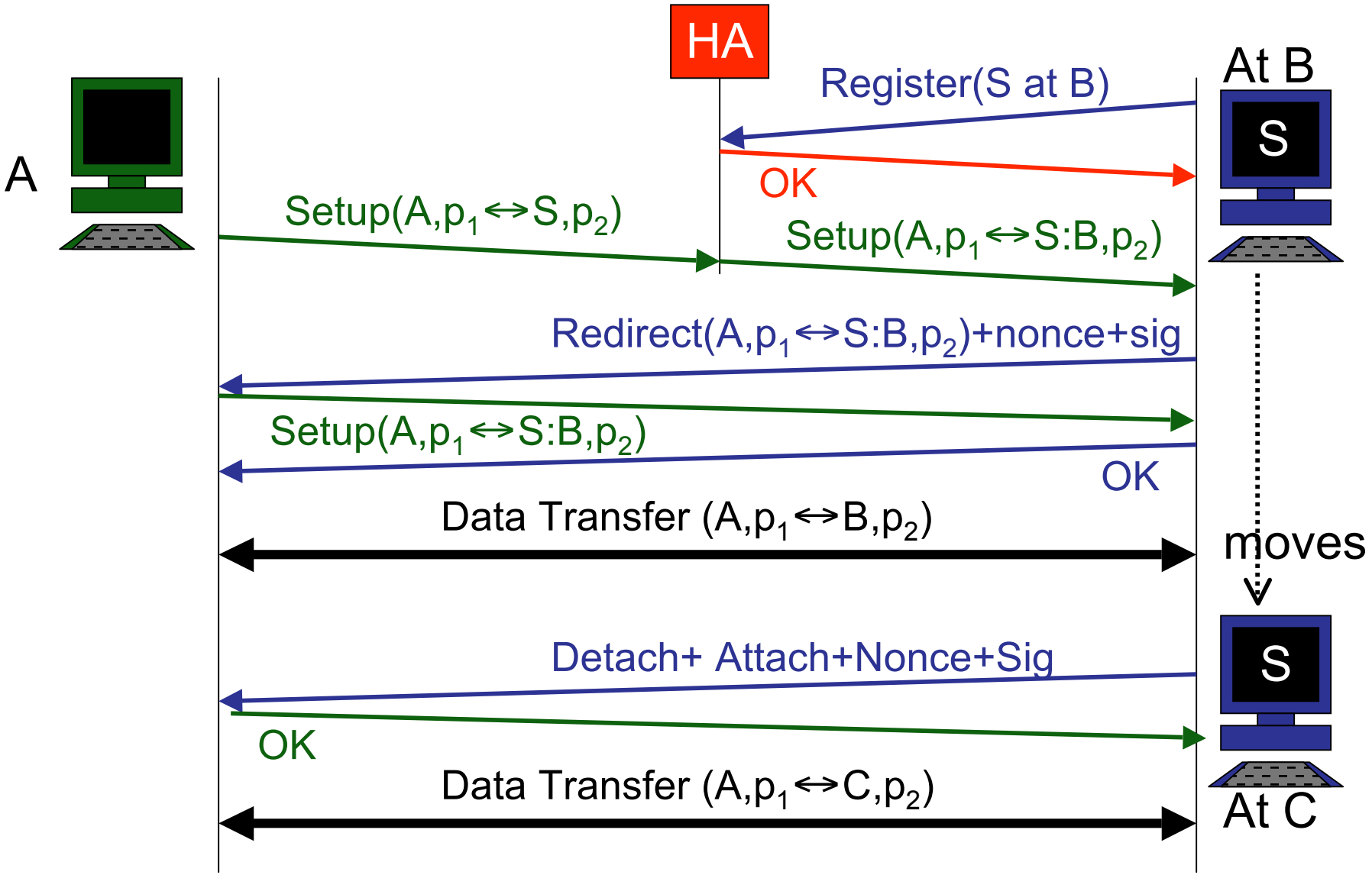
Mobile Client



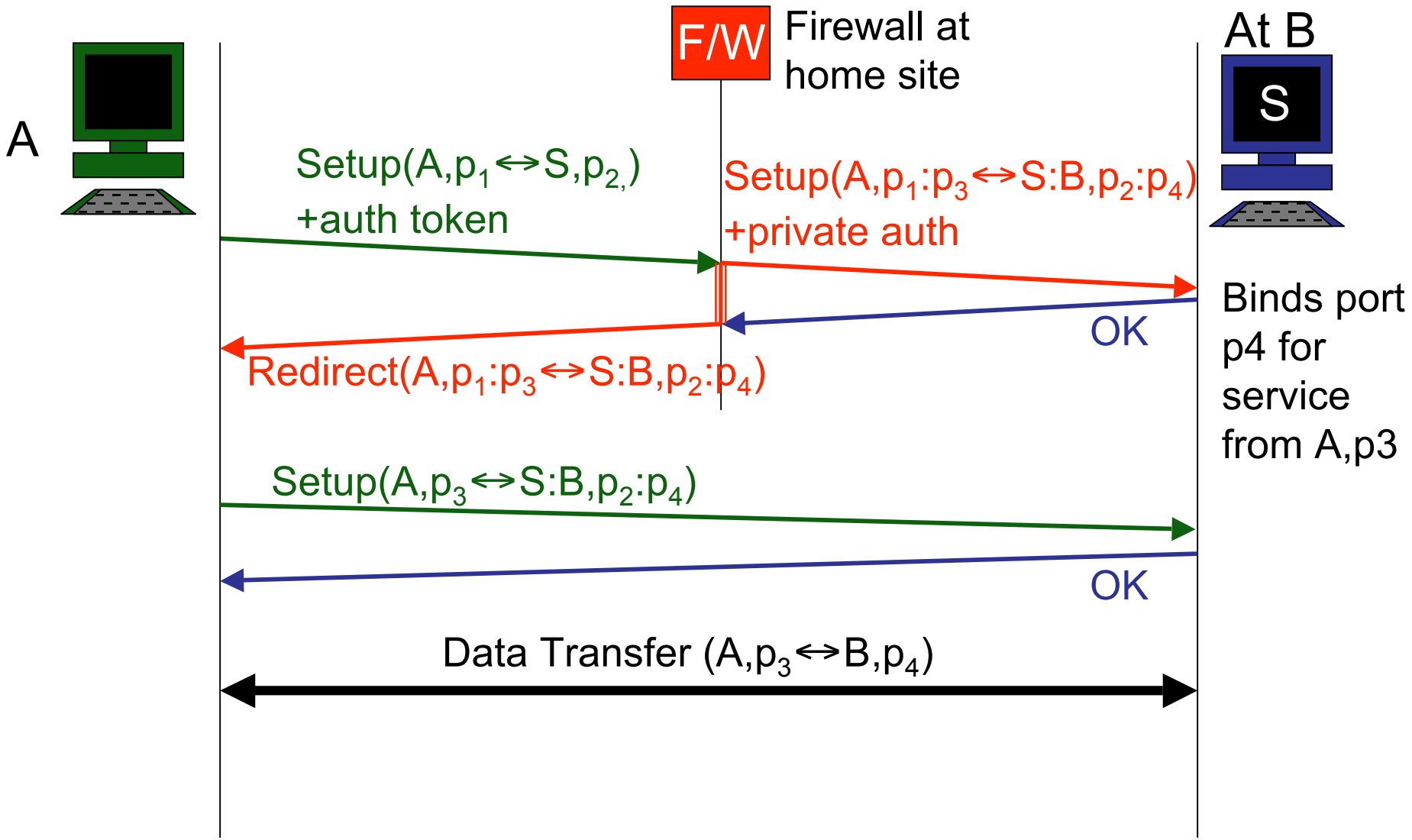
Mobile Server



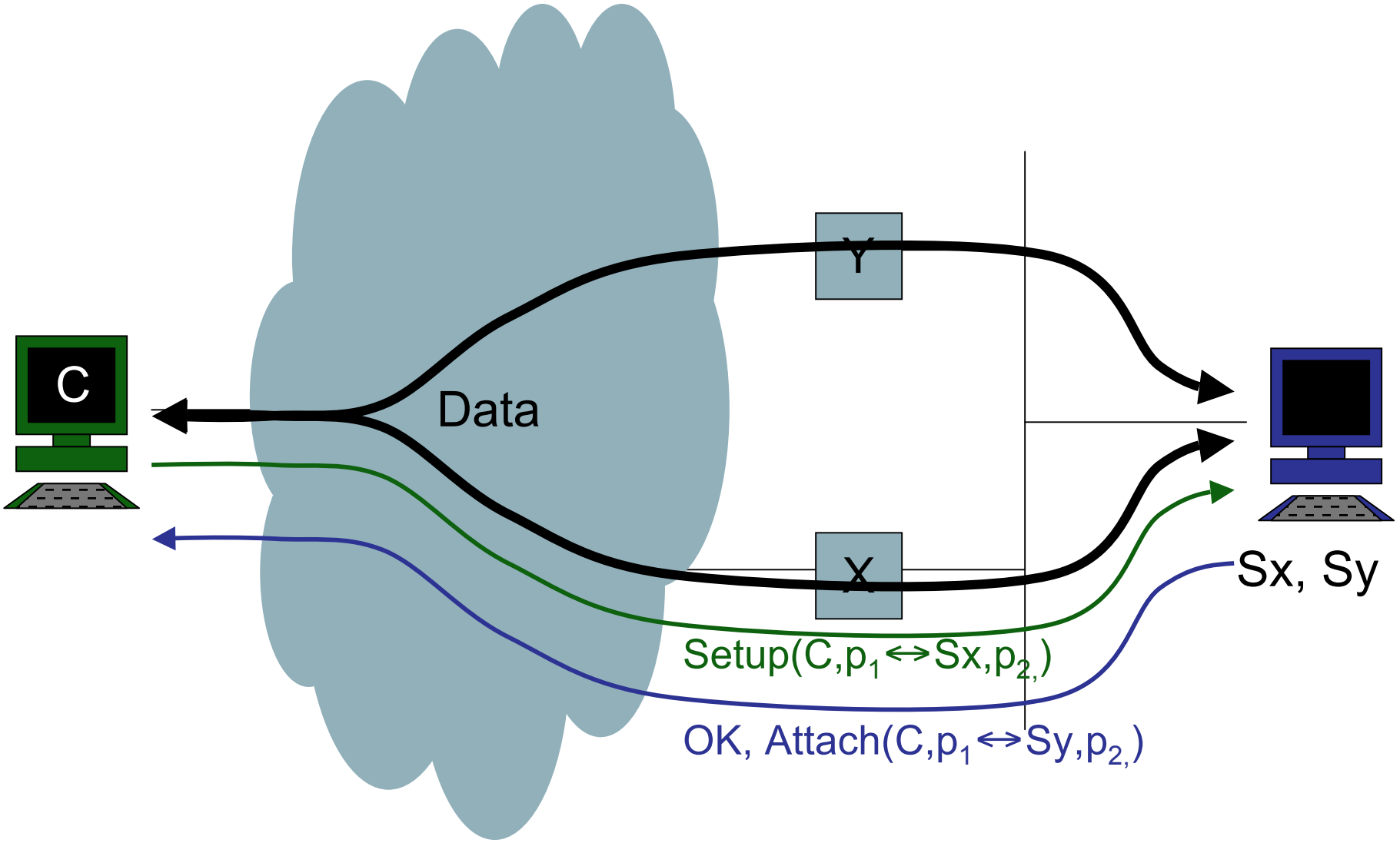
Hidden Mobile Server



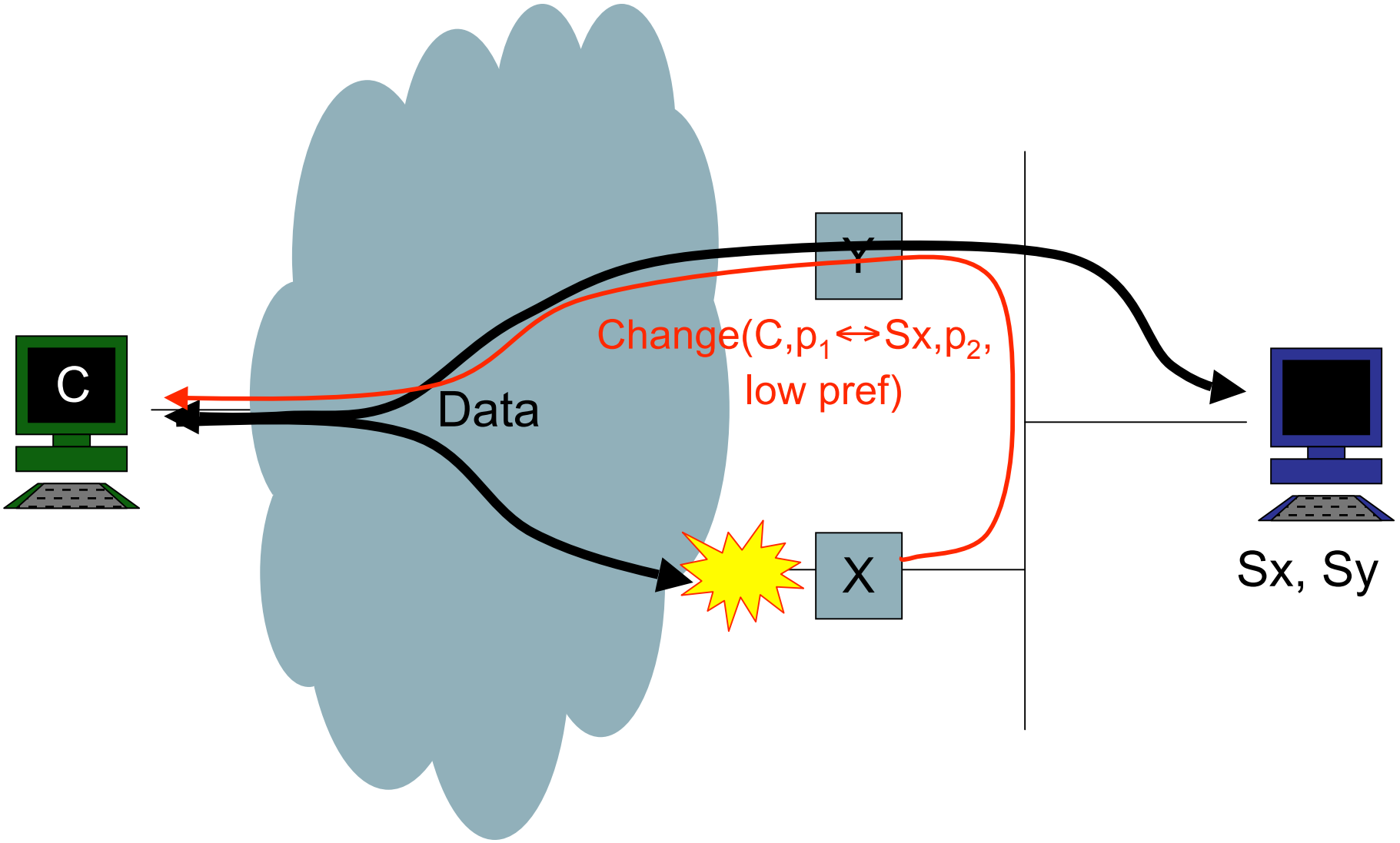
Offpath Firewall for Mobile Host



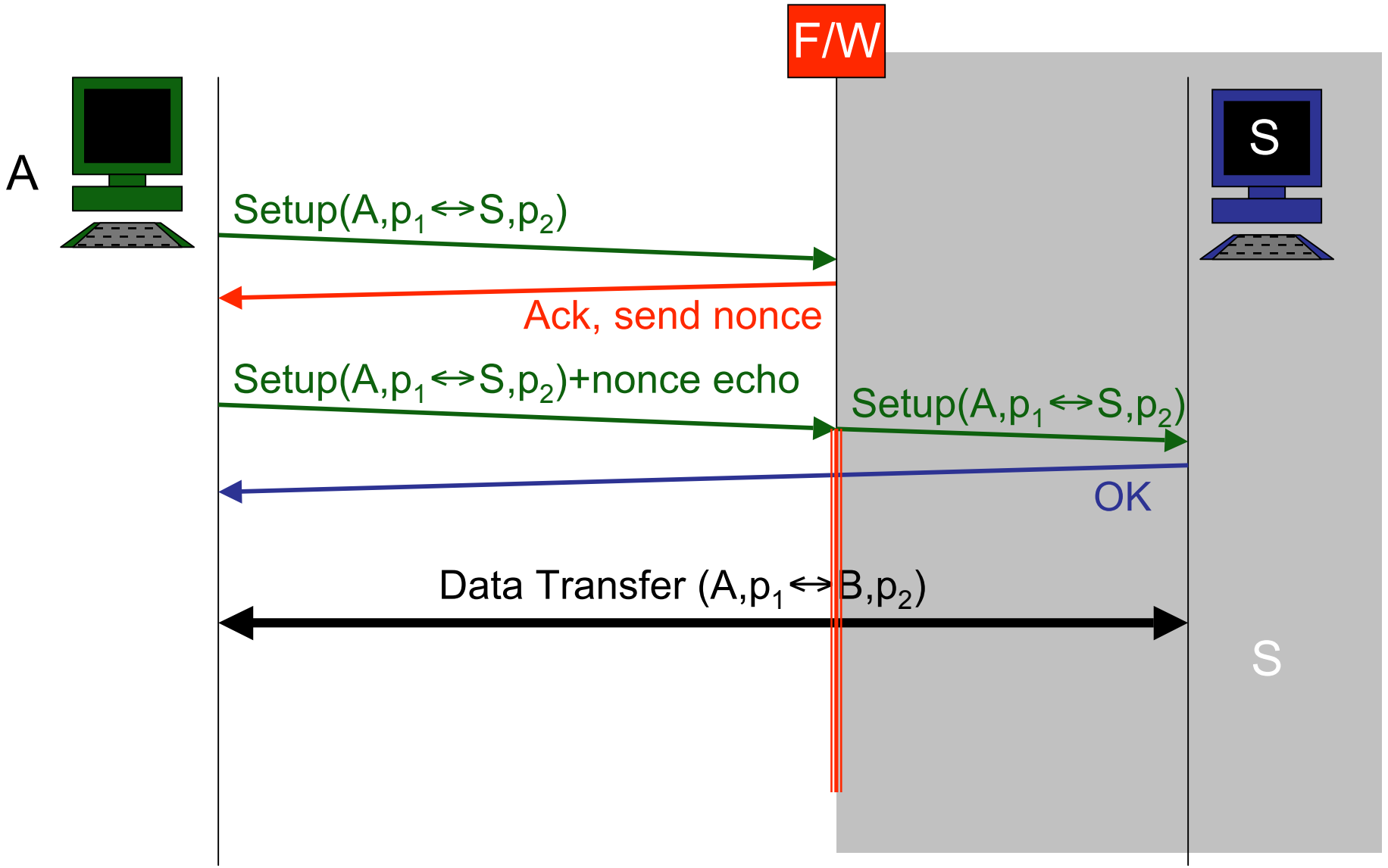
Simple Multihoming



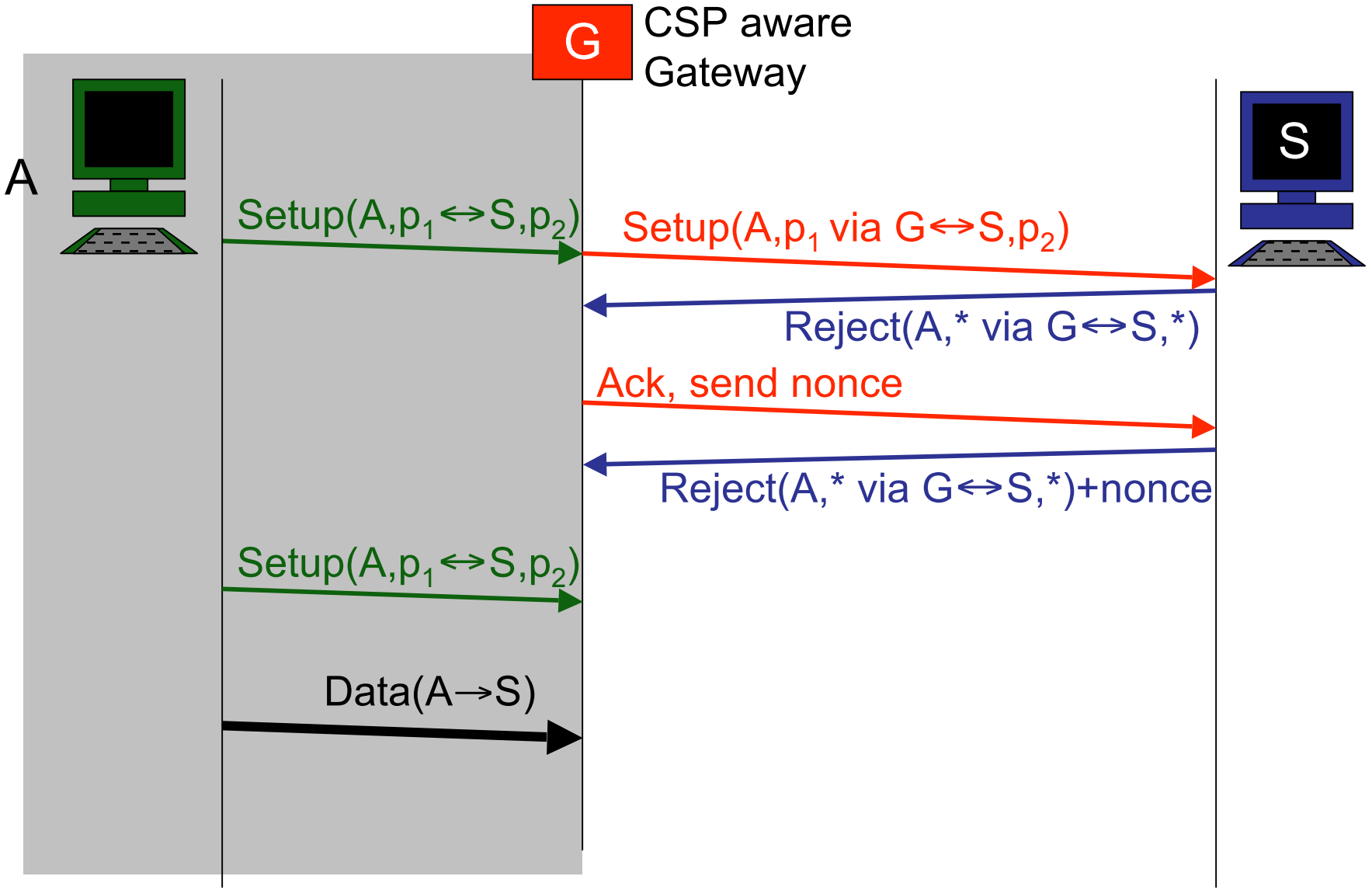
Simple Multihoming



Spoofing


F/W
A
S
Setup(A,p_1↔S,p_2)
Ack, send nonce
Setup(A,p_1↔S,p_2)+nonce echo
Setup(A,p_1↔S,p_2)
OK
Data Transfer (A,p_1↔B,p_2)
S

DoS Prevention



Connection Signaling: Summary

Assertion:

- Many of the architectural problems we currently face can be solved using connection signaling.
- Lots of questions.
  - Efficiency, simplicity vs flexibility
  - Backward compatibility, existing NATs, related work.
  - Which problems to focus on, which to ignore?
- Real danger of second system syndrome.
  - Unless it's simple, no chance of success.