Minion

Jana Iyengar, *Google*

Stuart Cheshire, *Apple*

Josh Graessley *Apple*

Joint Work

- Bryan Ford (Yale), Fitz Nowlan (Yale), S. Obaid Amin (Yale/F&M), Nabin Tiwari (F&M), Padma Bhooma (Apple)

Ancient History

- Internet application choices in 1983:
  - TCP
    - Version 4 from ISI, February 1979
    - RFC 793, September 1981
  - UDP
    - RFC 768, August 1980

UDP

- [x] Preserves message boundaries
- [ ] Unbounded data size
- [ ] Reliable delivery
- [ ] In-order delivery
- [ ] Flow control
- [ ] Congestion control

TCP

Virtual Serial Port

- ☐ Preserves message boundaries
- ☑ Unbounded data size
- ☑ Reliable delivery
- ☑ In-order delivery
- ☑ Flow control
- ☑ Congestion control

Applications Want Messages

- Email (POP, IMAP, SMTP)
- File Sharing (AFP, SMB)
- Web browsing (HTTP, SPDY)
- Instant messaging (AIM, XMPP)
- etc.

TCP

- Need application-layer framing protocol
  - Length header
  - Byte stuffing and framing
  - Interleaving to avoid Head of Line Blocking
- Enforces in-order delivery
  - Poor for latency-sensitive data
    when there's packet loss or reordering

UDP

- No mandatory in-order delivery
  - Better for latency-sensitive data
- But… no reliability, no flow control, no congestion control
  - All have to be reinvented by application

Present Day

- Internet application choices in 2013: UDP, TCP
- TCP *protocol* has had significant advances
  - Selective ACK
  - Explicit Congestion Notification
  - Multipath
- TCP application service model remains unchanged

And the Others…

- SCTP (Stream Control Transmission Protocol, RFC 4960)
- DCCP (Datagram Congestion Control Protocol, RFC 4340)
- RDP (Reliable Datagram Protocol, RFC 1151)
- SST (Structured Stream Transport, research protocol from MIT)
- POC (Partial Order Connection, RFC 1693)
- BEEP (Blocks Extensible Exchange Protocol, RFC 3080)

What Brought us Here?

- "NATs are evil. We won't care about them. It will all change with IPv6." — Denial
- "Don't design around middleboxes; that will only encourage them!" — Anger
- "Wait, wait… we'll accept middleboxes, but we'll specify how they ought to behave!" — Bargaining
- "Why build a new transport? It won't get deployed." — Depression

Middleboxes are Here to Stay

Acceptance *

- Design of new end-to-end services should not require changes to middleboxes
- Consequence: New end-to-end services must appear as legacy protocols on the wire

* Kübler-Ross model: Five stages of grief

Minion

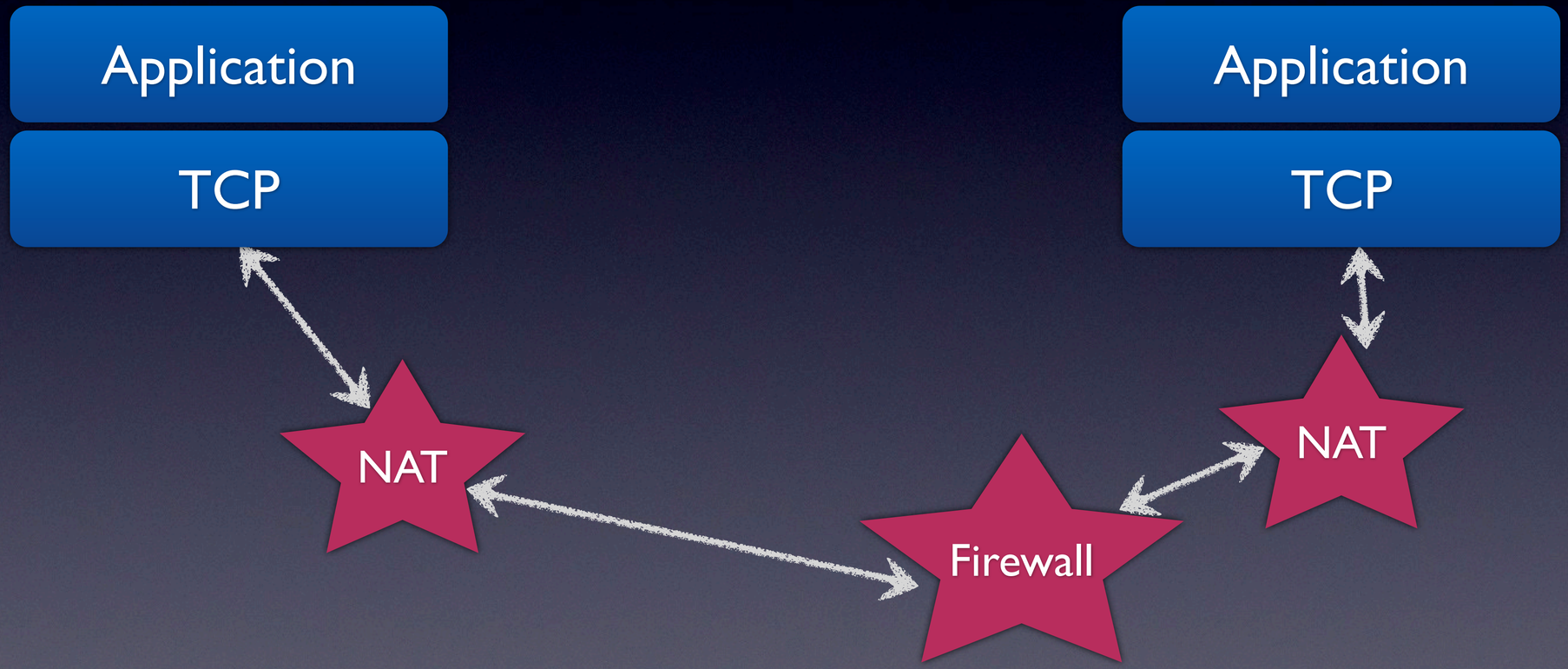
- Provides some SCTP-like features
- ... But looks like TCP on the wire
  - Works through NATs, Firewalls, etc.
- Incremental deployment path
  - Doesn't require kernel support
  - Can optionally benefit from it

Incremental Deployment

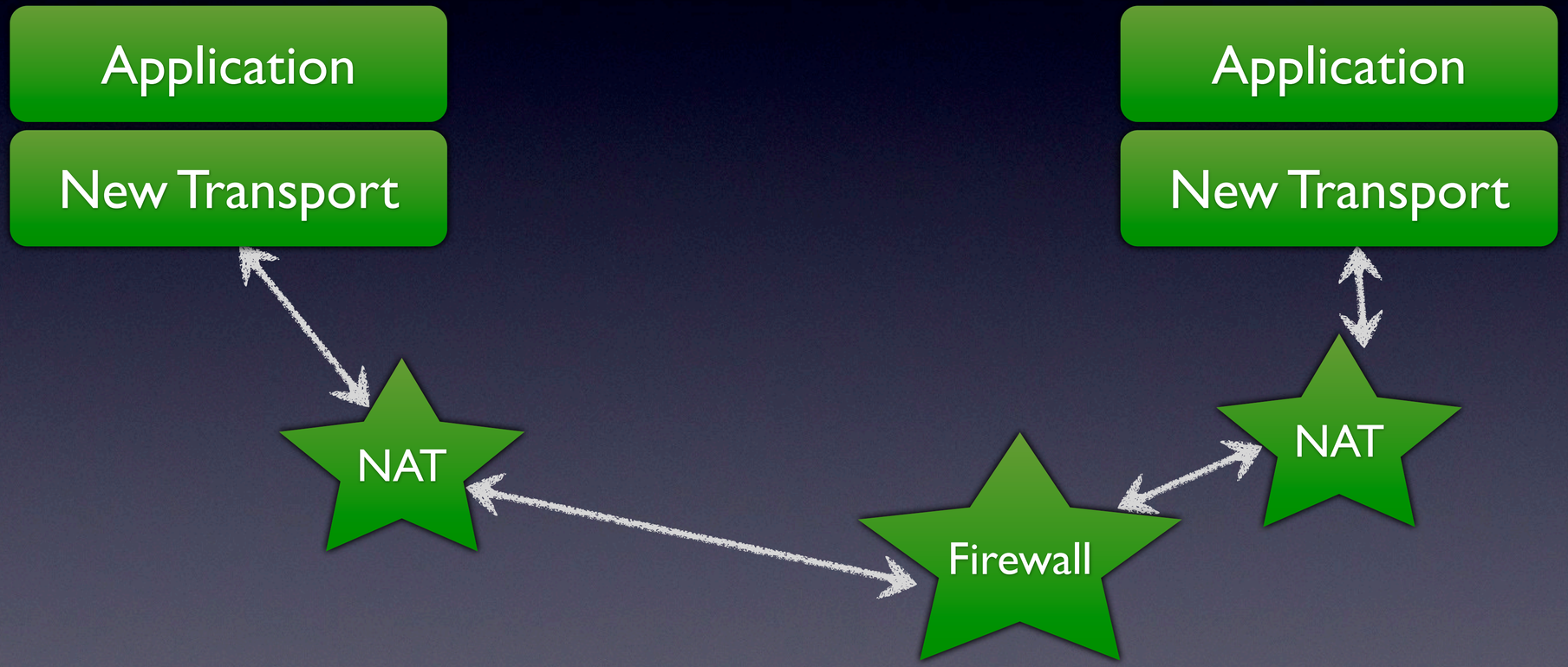
Incremental Deployment

Incremental Deployment

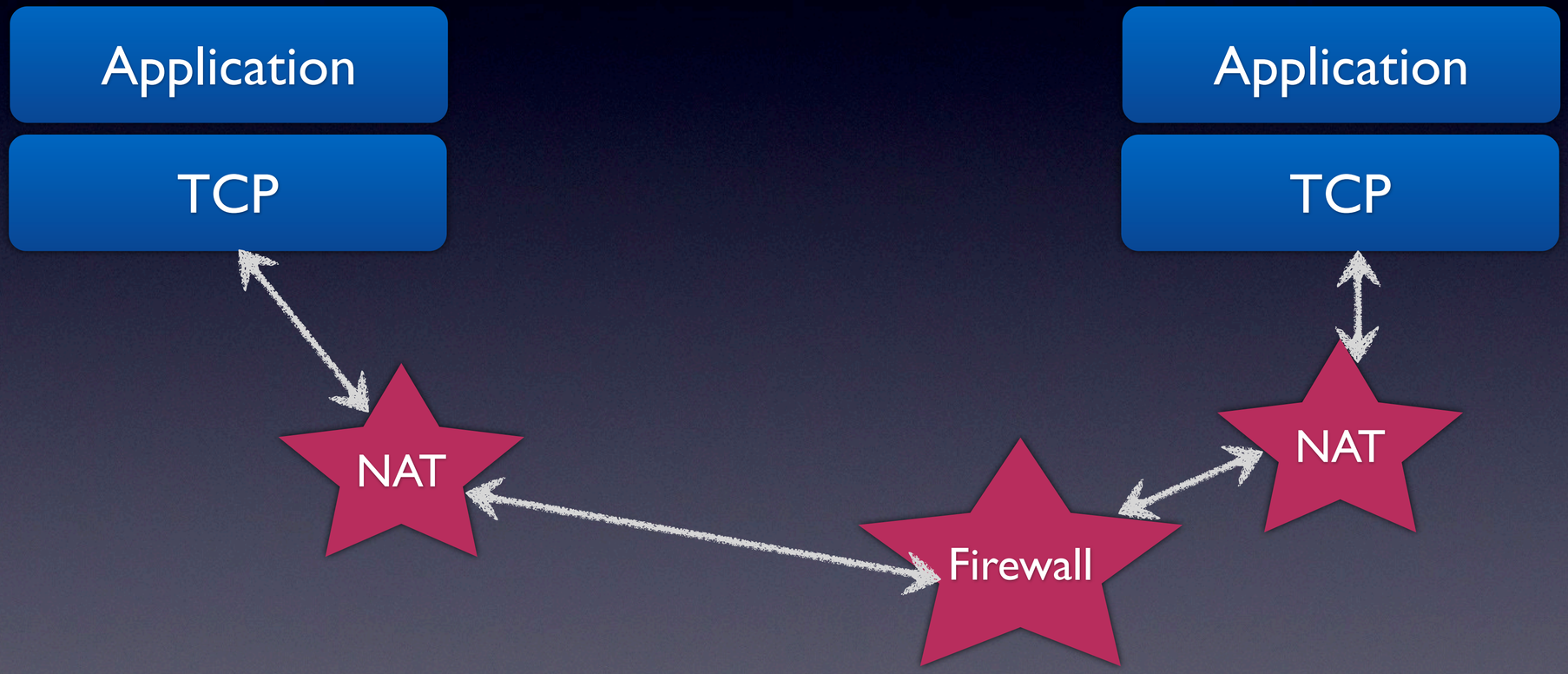
Incremental Deployment



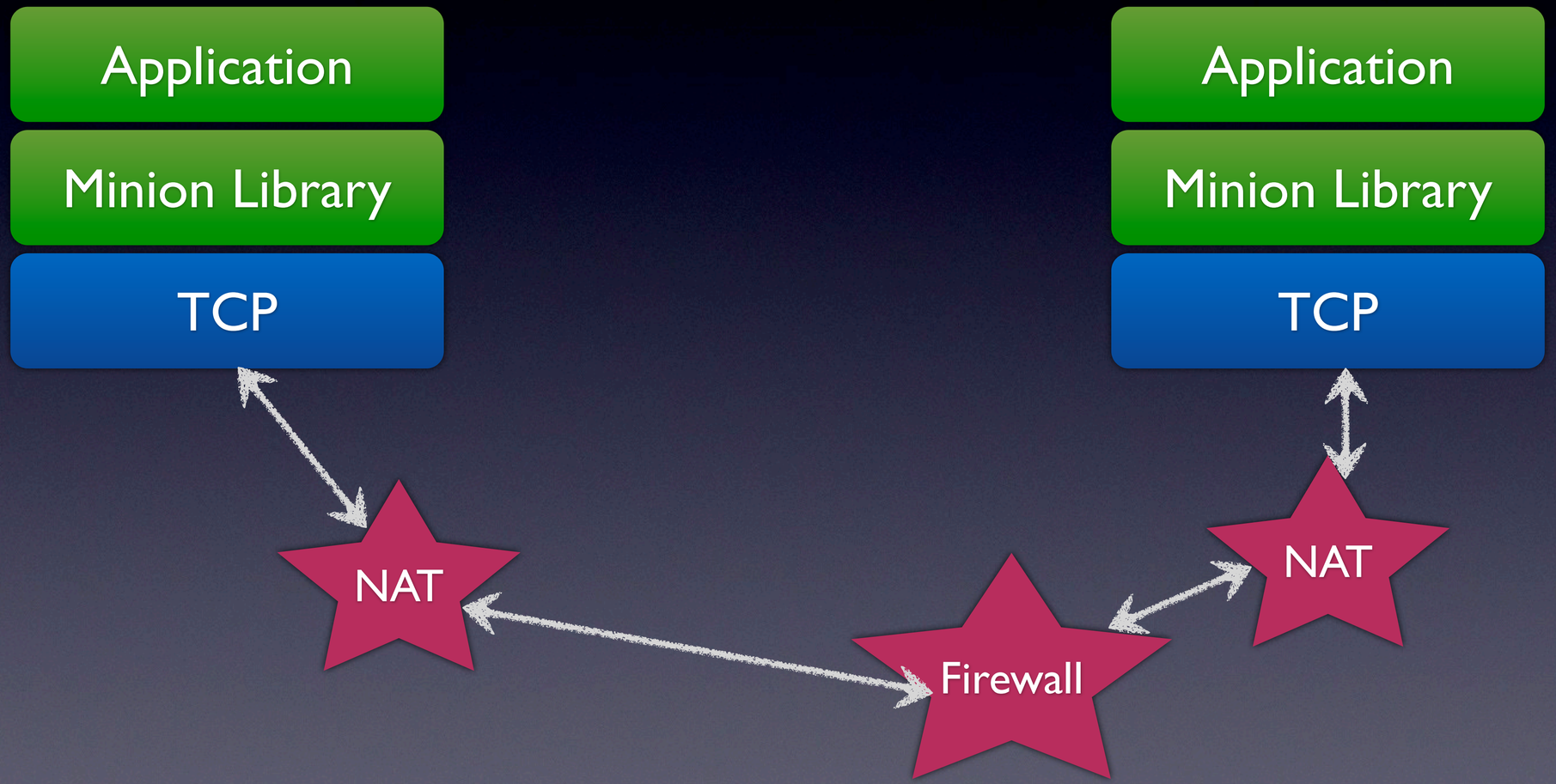
Incremental Deployment



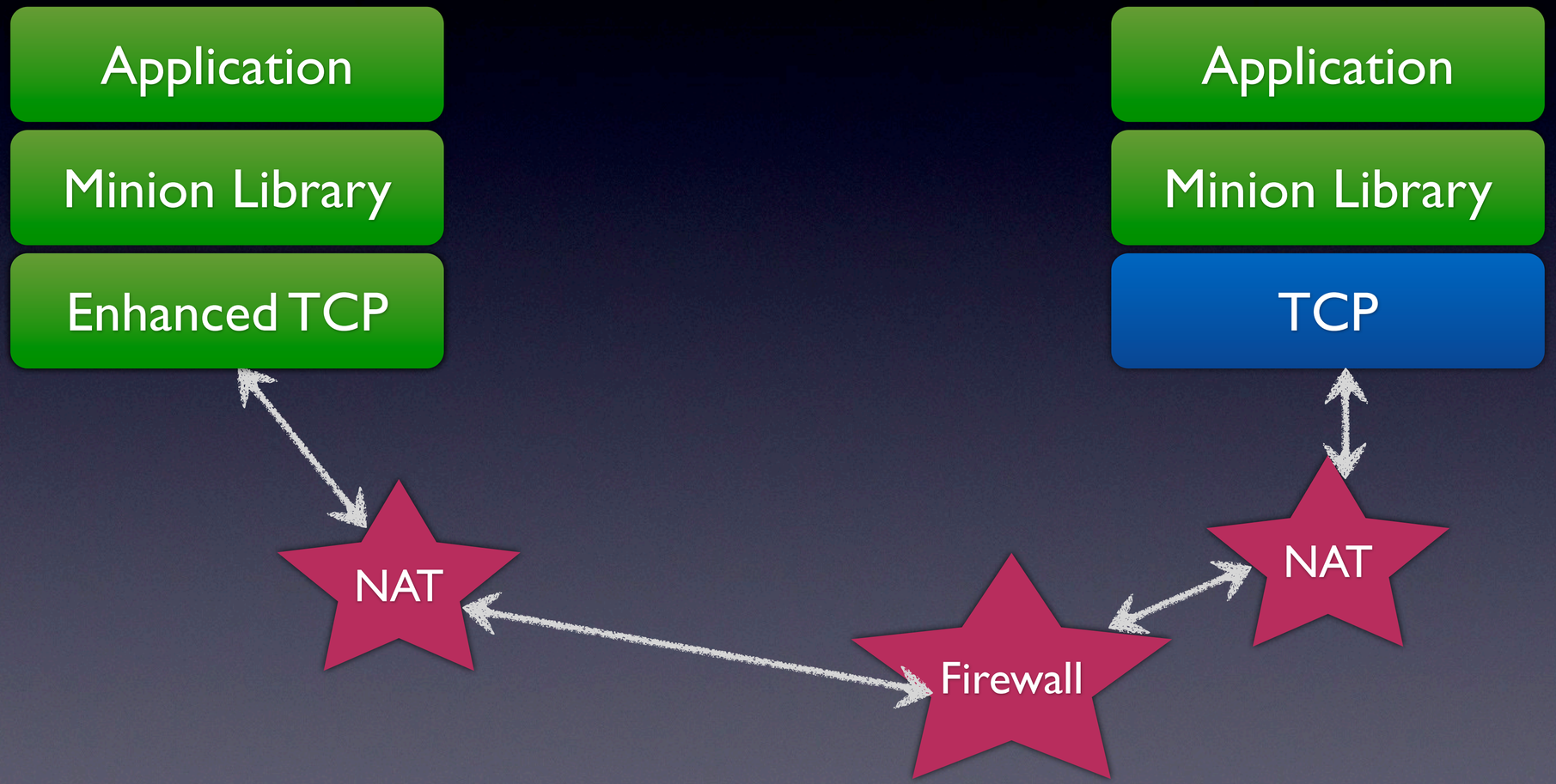
Incremental Deployment



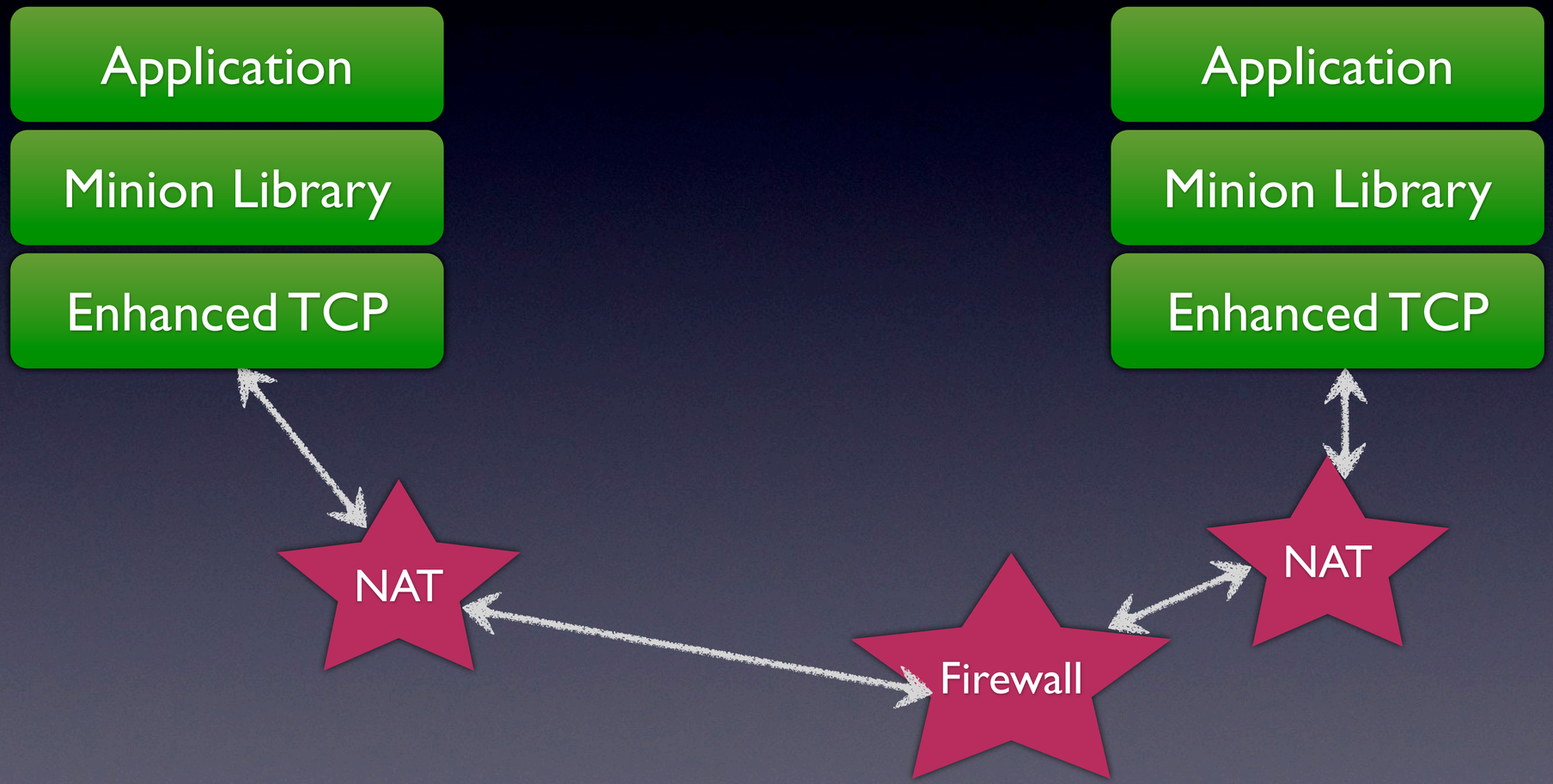
Incremental Deployment



Incremental Deployment



Incremental Deployment

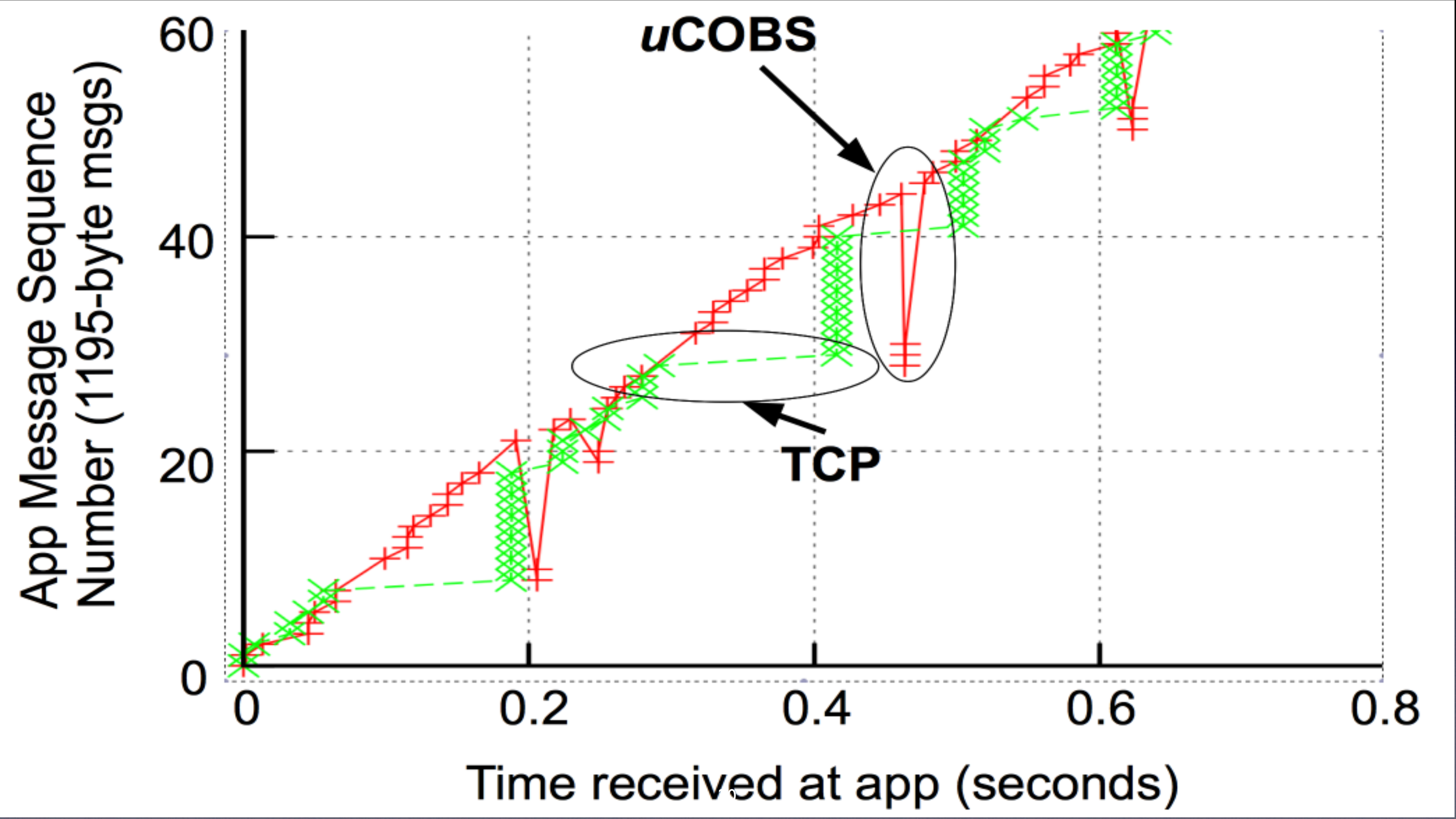


Minion Services

- Reliability, Flow Control, Congestion Control
- Message-oriented, like UDP
  - Unbounded message size
    - Message cancellation
  - Message replies
  - Message dependencies
- DTLS security

Minion: Unordered Messages

- For low-latency data
- TCP socket option enables out-of-order delivery


uCOBS
TCP
App Message Sequence Number (1195-byte msgs)
Time received at app (seconds)
0
20
40
60
0
0.2
0.4
0.6
0.8

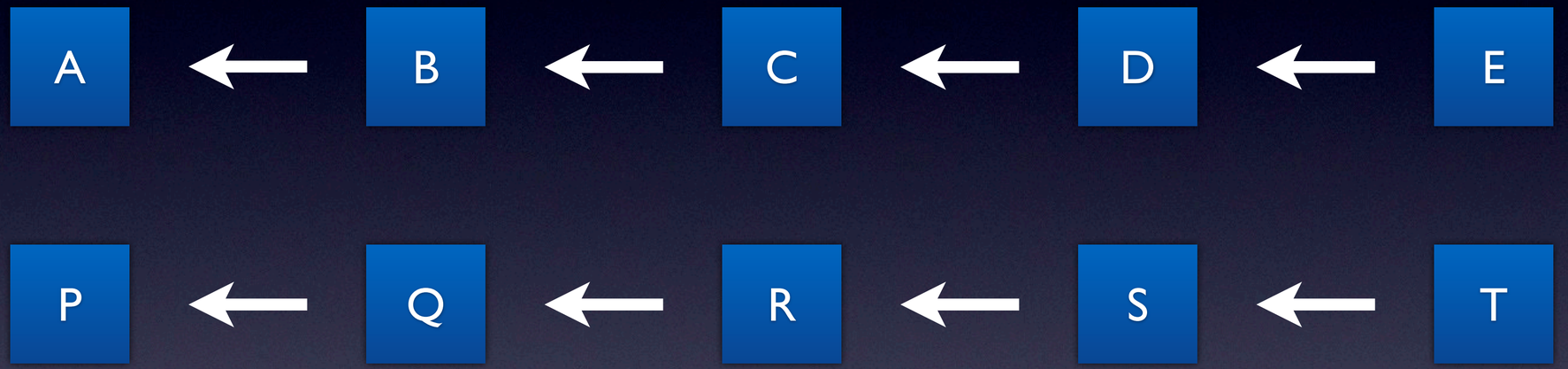
Minion: Multistreaming

- Multiple messages multiplexed on single connection
  - Shared congestion state
  - Shared loss detection and recovery
- Message Interleaving (chunk-level)
- Message Dependencies (partial ordering of messages)
- Multiple Priority Levels (with byte-level pre-emption)

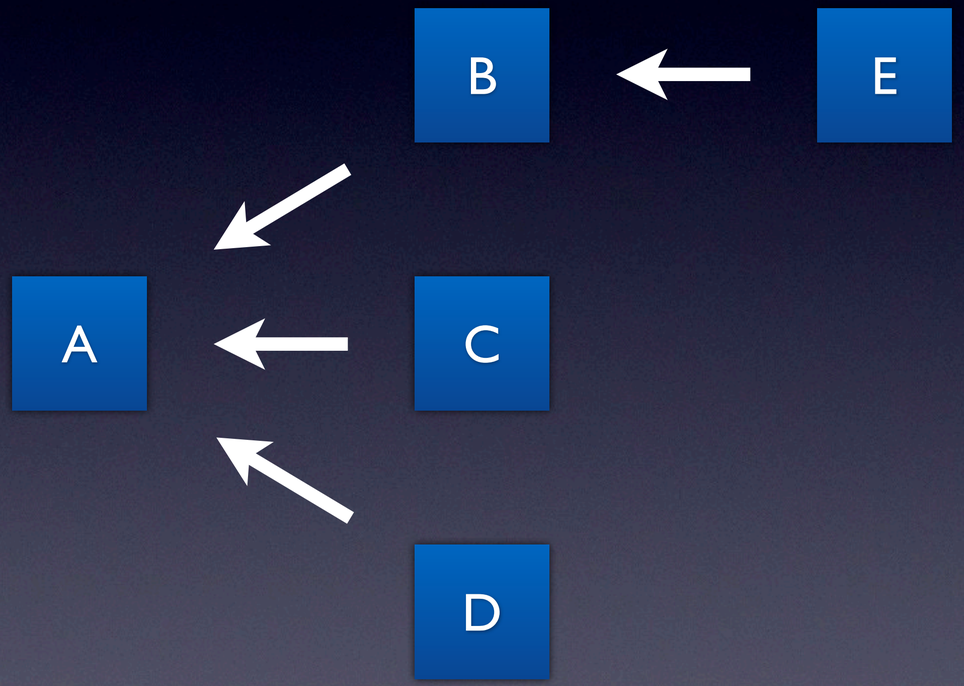
Multistreaming without Streams

- Ordered messages (aka “Streams”)
- Partially ordered messages

Ordered Messages



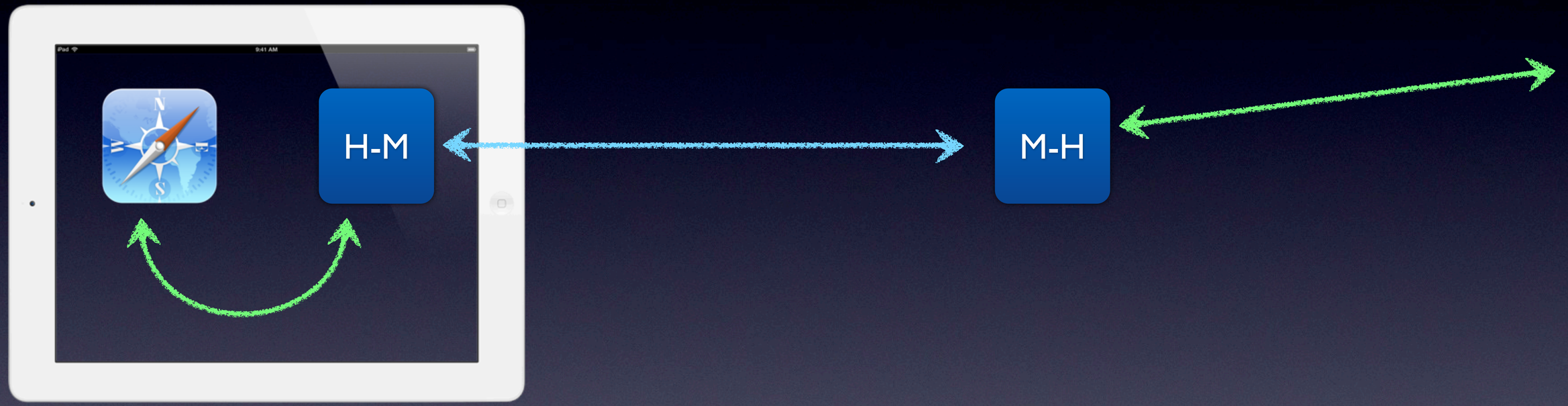
Partially Ordered Messages



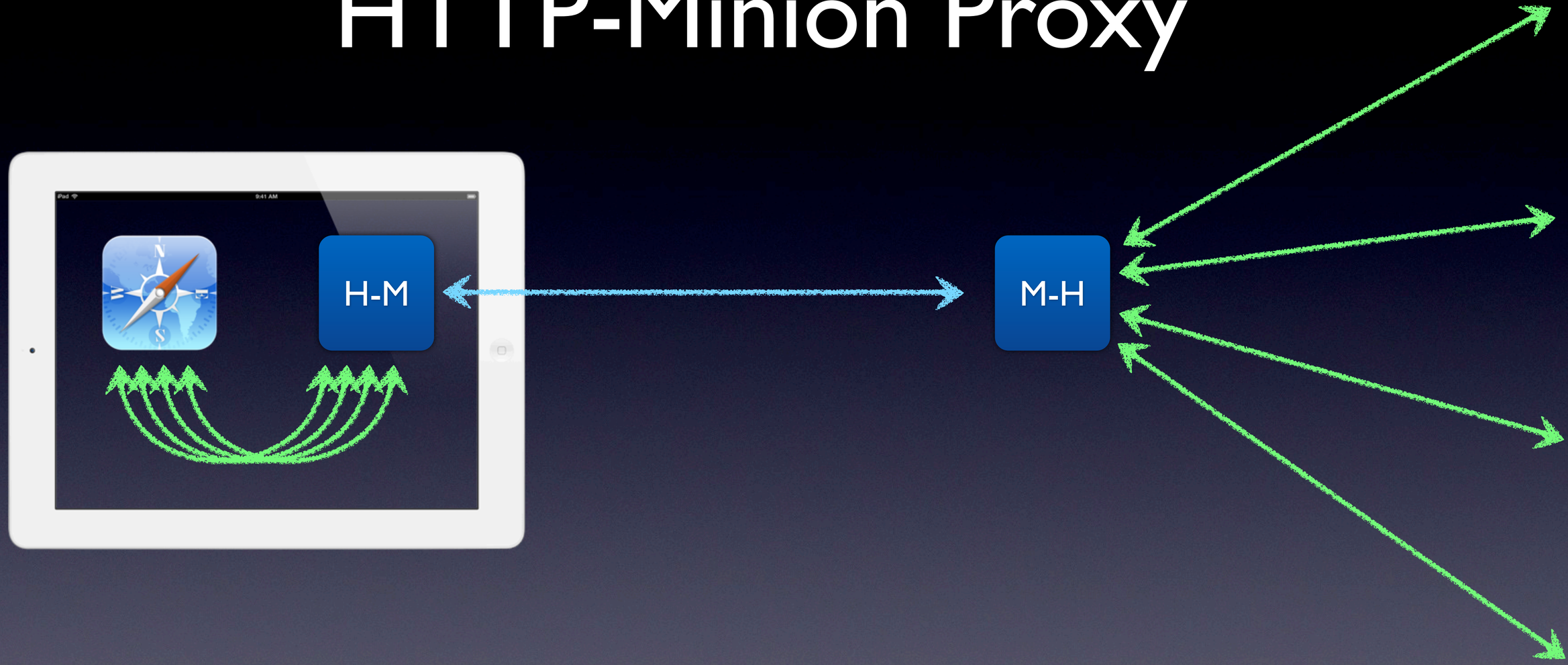
Minion: Request/Reply

- Sending app can designate a reply handler block for message
- Peer can designate that
  message Y is reply to message X
- On reception of the response, message Y gets passed to reply handler block for message X
- Responses for (pipelined) requests can be interleaved, prioritized and sent in any order

HTTP-Minion Proxy



HTTP-Minion Proxy



Q&A

draft-iyengar-minion-concept-01
draft-iyengar-minion-protocol-01