Multipath TCPの紹介と最近の動向

Yoshifumi Nishida
GE Global Research

GE imagination at work

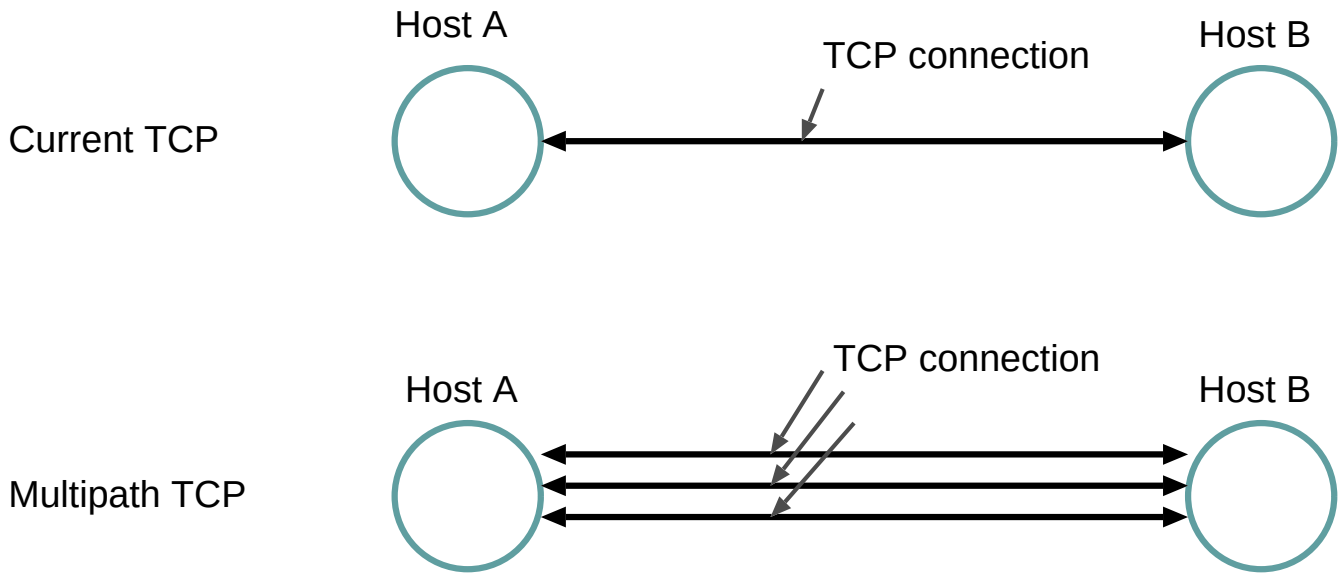
Who Am I?

- Name: Yoshifumi Nishida
- Current Job
  - Senior Researcher at GE Global Research in San Ramon, CA
- IETF Activities
  - Co-chair of TCPM Working Group
  - Co-chair of Multipath TCP Working Group
  - Transport Area Directorate

What Is Multipath TCP?

What is Multipath TCP (MPTCP)?

- An extension to TCP
  - Not a new protocol
- Allow single TCP session to use multiple addresses
  - Utilize multiple TCP connections, but expose only one TCP connection to upper layer



Benefit for MPTCP

- Basic advantages
  - Increase throughput
    - Utilize multiple paths simultaneously
  - Increase resiliency
    - Failover to other paths when one path becomes unavailable
  - Dynamic address configuration
    - Add or delete IP addresses without terminating connection
- Additional possibilities
  - Mobility support
    - Don’t need to use Mobile IP
  - Stimulate IPv6 transition
    - Legacy IPv4 applications will start using IPv6 without any modification

Why Not Multiple TCP Connections?

- Applications can use multiple TCP connections!
- But,
  - Need to rewrite existing applications
  - It can be too aggressive than normal TCP
    - Especially when all paths share the same bottleneck
  - Sophisticated data transmission will be difficult
    - Retransmit data to other paths will be tricky
    - Applications need to decide how much data to be sent on each path

Why Not SCTP?

- SCTP already supports using multiple addresses!
- But,
  - Middlebox traversal can be problematic (especially NAT)
    - Some middleboxes don’t understand SCTP traffic
  - Need to rewrite existing applications to use SCTP
    - SCTP uses different APIs
  - Not easy to fallback to TCP
    - It can be cumbersome when peer doesn’t support SCTP
  - Offload engine is not prevailed very much

Isn’t It Too Aggressive?

- MPTCP utilizes multiple TCP connections!
- But,
  - MPTCP employs new congestion control logic
    - Coupled Congestion Control
      - Adjust transfer rate of single flow from total transfer rate
  - Design criteria for coupled congestion control
    - Should coexist gracefully with existing legacy TCP flows
    - Should not be neither too aggressively nor too timidly
  - You can also specify a path to be used as "Backup"

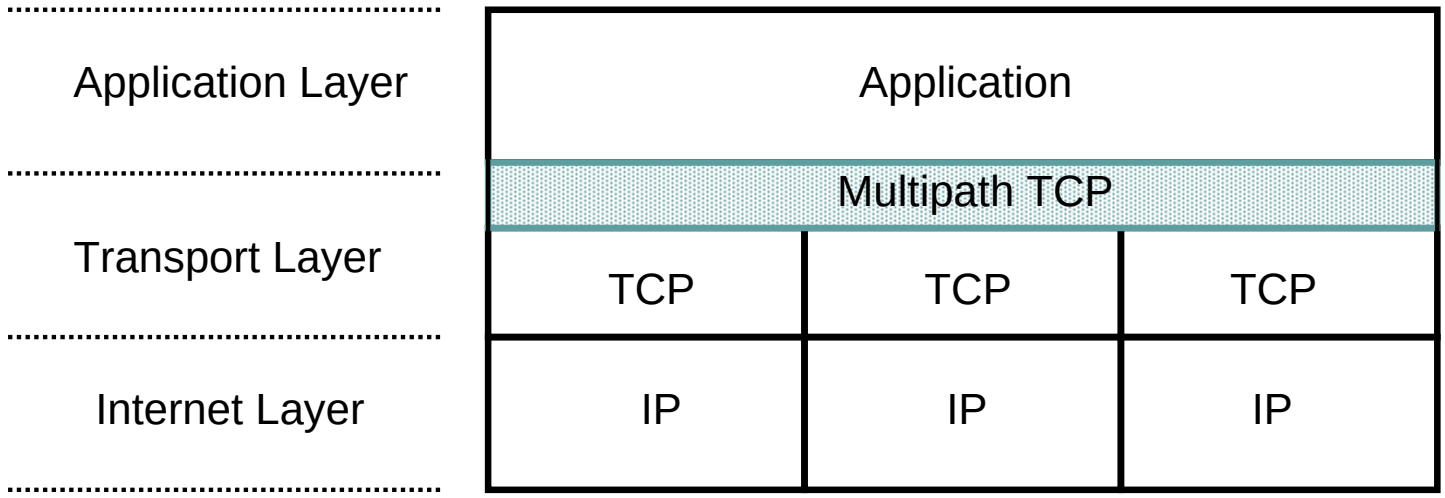
Do I Need to Modify My Application?

- Applications will not be required to update for MPTCP
  - MPTCP can work with current socket API for TCP
  - If your kernel support MPTCP, TCP applications can start using MPTCP
- For advanced features, special APIs for MPTCP will be needed

Multipath TCP Architecture

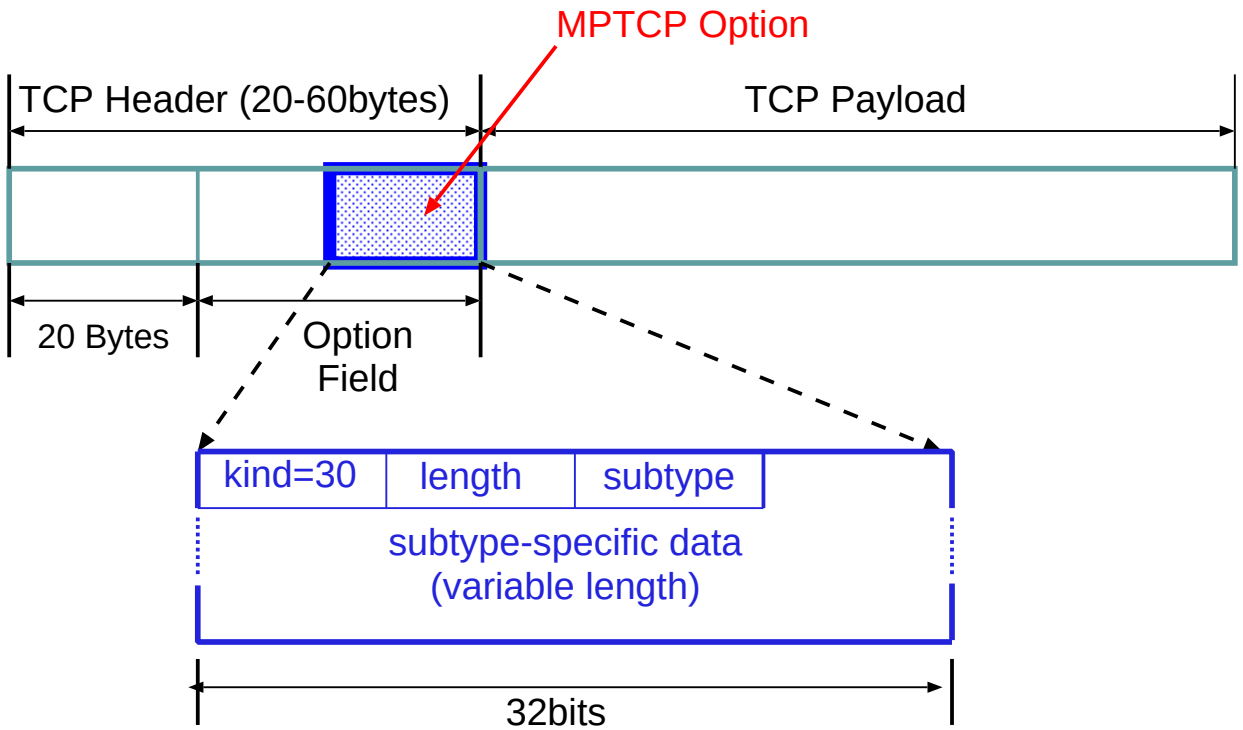
Layer Architecture

- Multipath TCP operates at the transport layer
  - Transparent to both higher and lower layers
- MPTCP layer is upper layer on TCP
  - It controls multiple TCP sessions as subflows



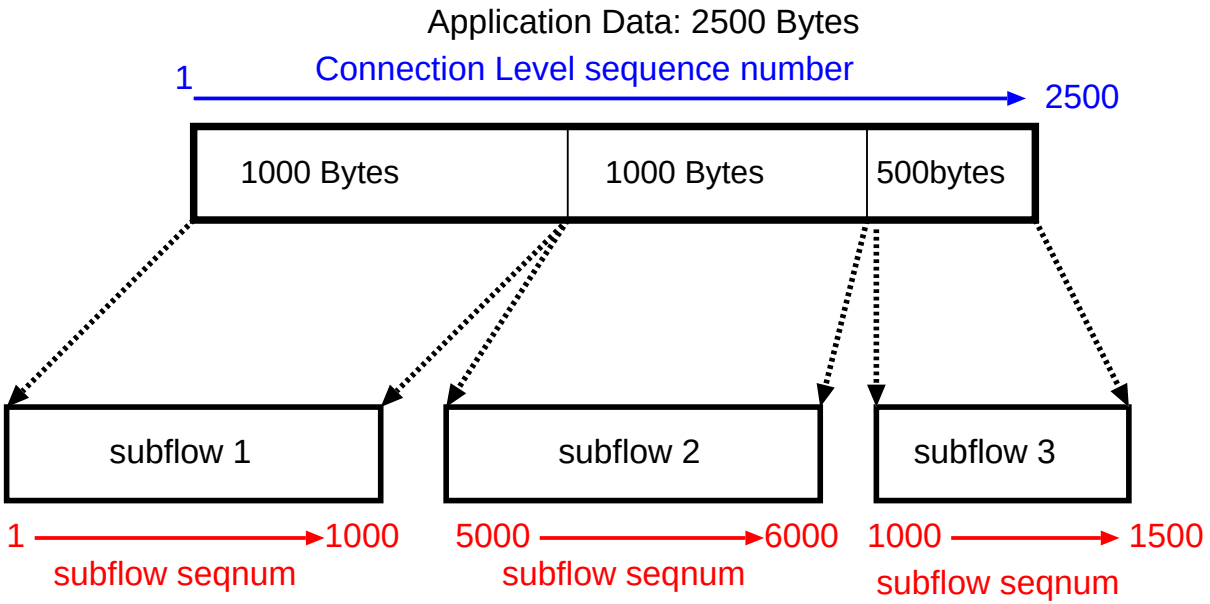
Signalling

- All control Information for MPTCP is sent in TCP options
- Option Kind: 30
  - subtype field is used to identify the type of suboptions
  - 7 types are currently defined
    - E.g. MP_CAPABLE, DSS, MP_JOIN, ADD_ADDR



Sequence Numbering

- Use two layers of sequence spaces
  - connection level sequence number
  - subflow sequence number (TCP's sequence number)
- Sender sends mapping information in TCP options
  - Receiver assembles data from multiple flows by mapping info

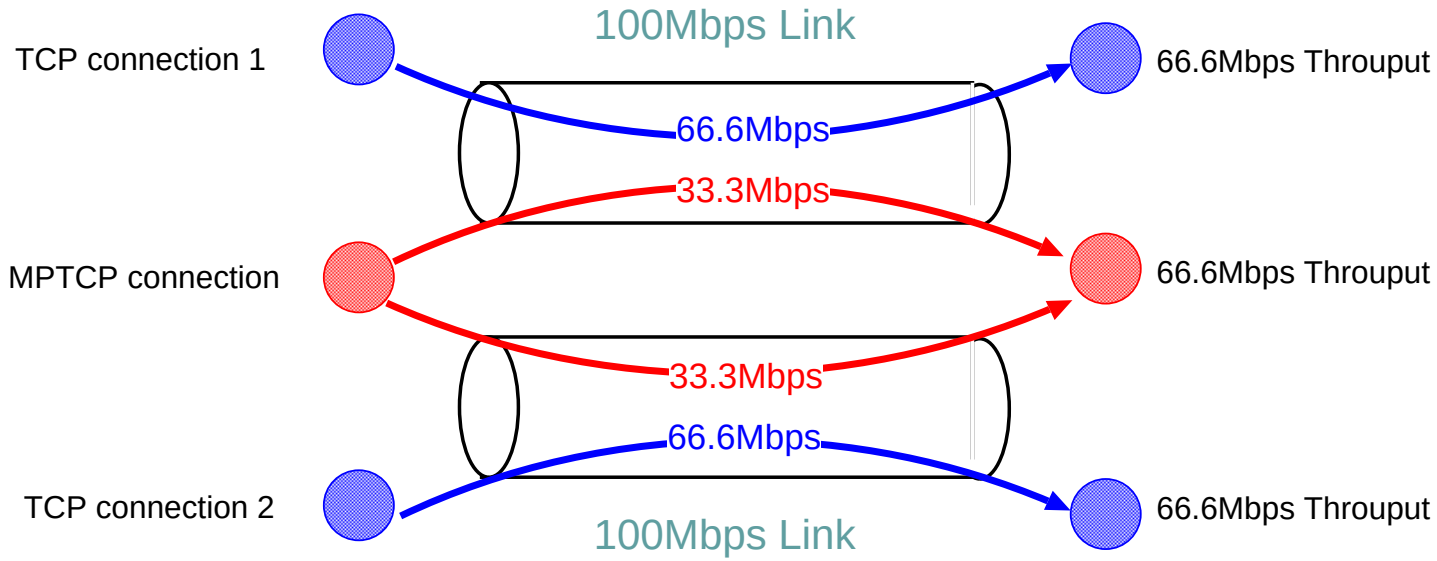


MPTCP Mapping

Connection Level Seq Num	subflow ID	subflow Seq num
1-1000	1	1-1000
1001-2000	2	5000-6000
2001-2500	3	1000-1500

Congestion Control (1)

- A simple sample target scenario
  - Network resources behave like a single pooled resource
  - MPTCP uses two links modestly, but efficiently
    - Compete normal TCP modestly
    - Outperform single path TCP



Congestion Control (2)

- Coupled Congestion Control
  - Affect only increase phase of the congestion avoidance state
    - Use Linked Increase Algorithm
  - Slow-Start, Fast Retransmission, Fast Recovery algorithm are not changed
- Linked Increase Algorithm
  - For each ACK received on subflow i, increase cwnd_i by

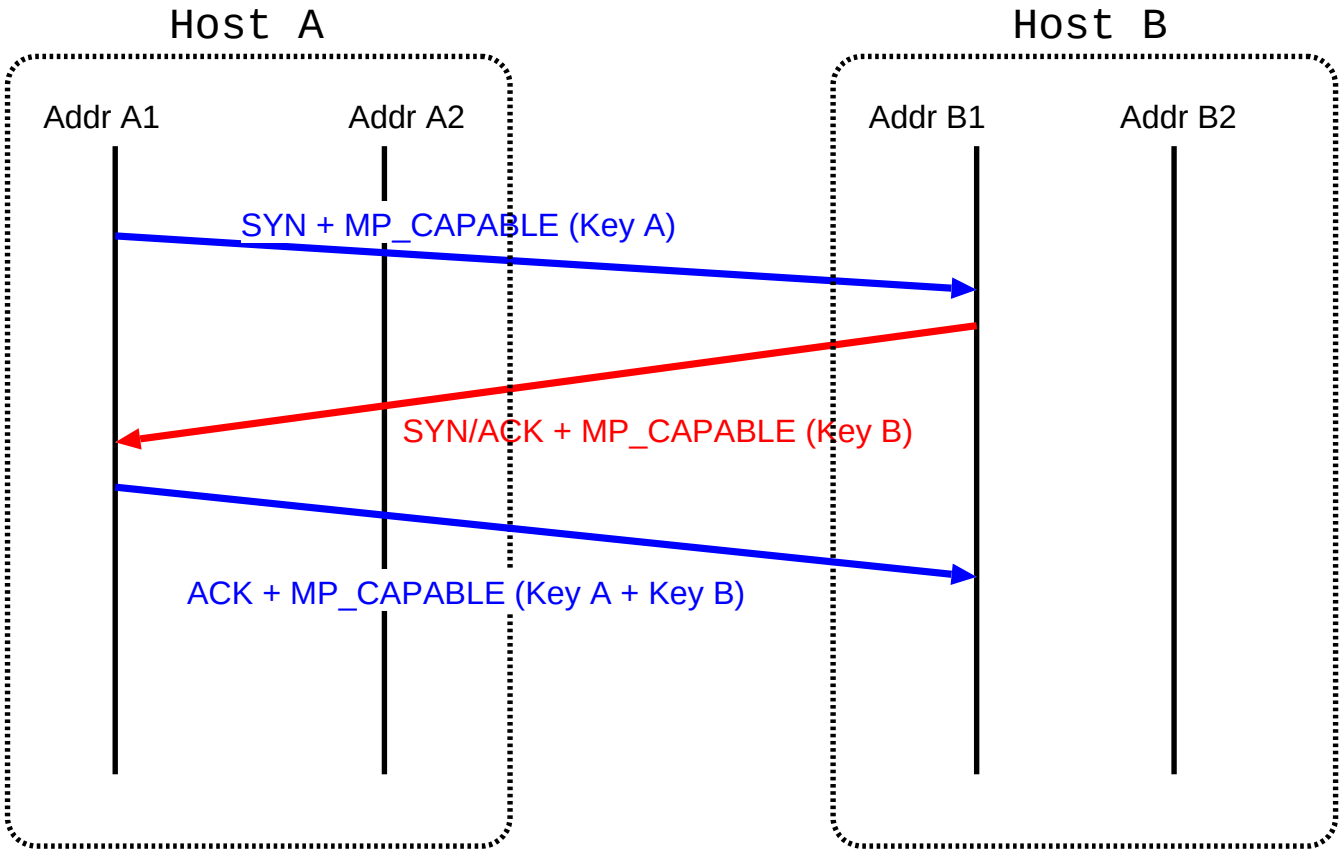
$$min(\frac{(\alpha \times bytes\_acked \times mss_i)}{total\_cwnd}, \frac{bytes\_acked \times mss_i}{cwnd_i})$$

$$\alpha = total\_cwnd \times \frac{\max_i(\frac{cwnd_i \times mss_i^2}{rtt^2})}{\sum_i(\frac{cwnd_i \times mss_i}{rtt_i})^2}$$

    - 'alpha' needs to be computed in case of packet lost or once per RTT

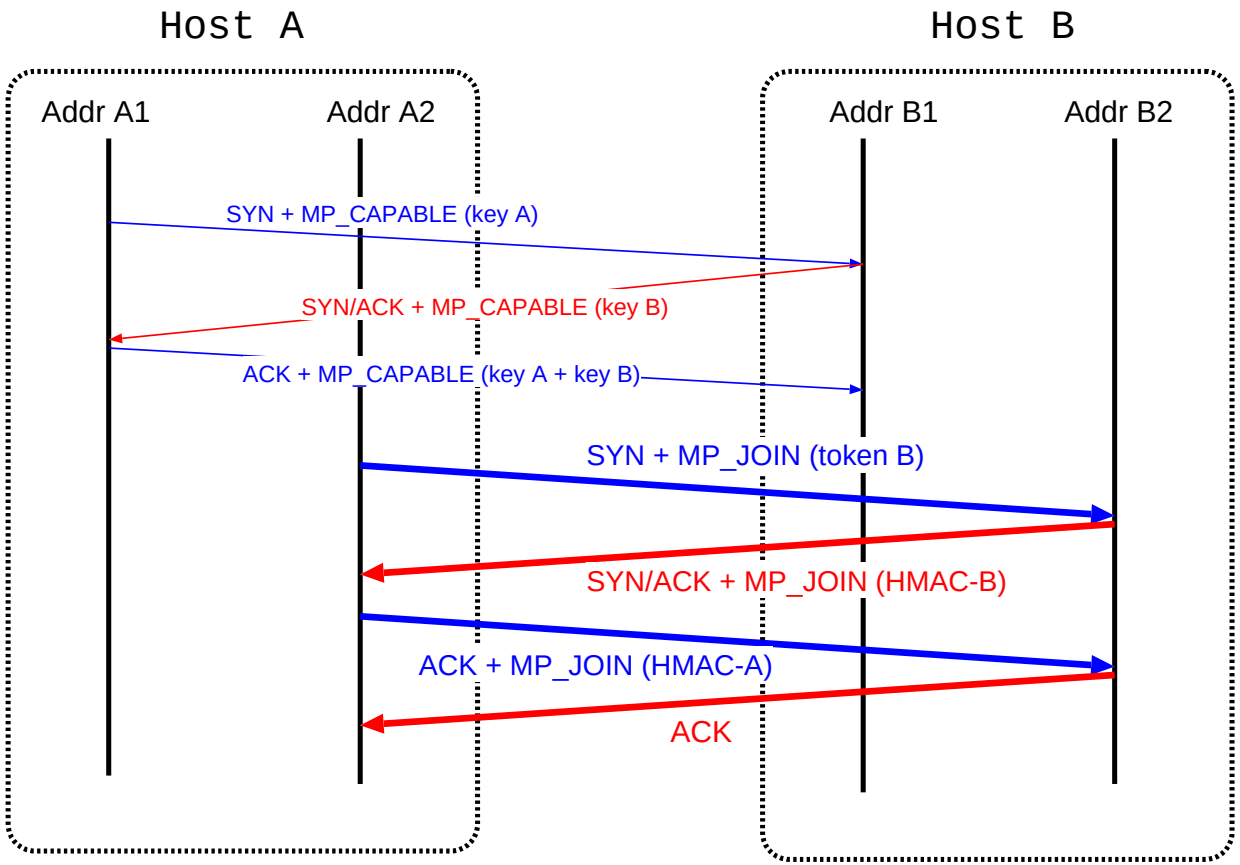
Protocol Example (1)

- Connection Setup
  - Exchange MP_CAPABLE option in SYN exchange
    - Notify it has multipath capability
    - Send a token to enhance security



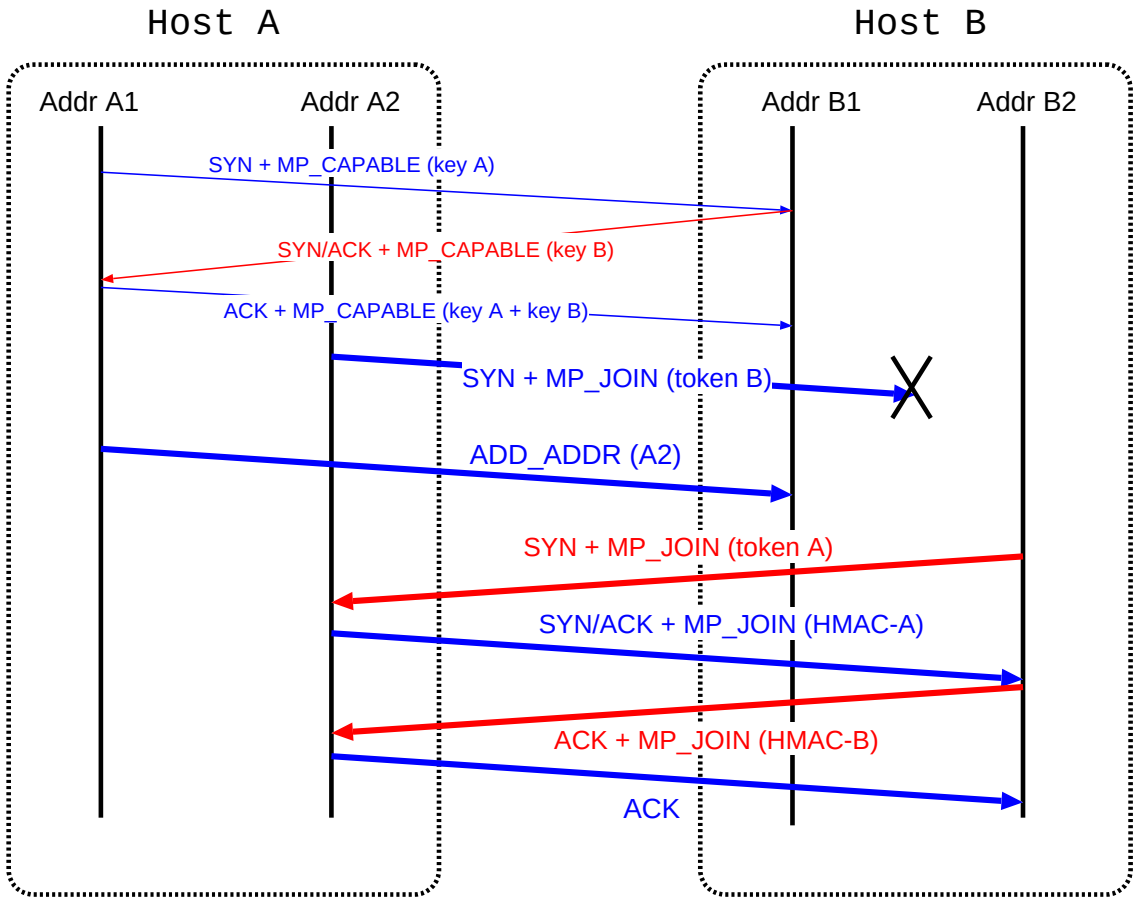
Protocol Example (2)

- Starting New Subflow
  - Send JOIN option in SYN packet from new address
    - Attach peer's token to identify multipath TCP session



Protocol Example (3)

- Address Knowledge Exchange
  - Notify additional address info to the peer
    - Useful for NATed host



Multipath TCP WG Status

Multipath TCP WG

- Established in November 2008
- Current Status
  - Finished all initial milestones by March 2013
    - RFC6181 (Threat Analysis) .. Informational
    - RFC6182 (Architectural Guideline) .. Informational
    - RFC6356 (Congestion Control) .. Experimental
    - RFC6824 (Protocol Spec) .. Experimental
    - RFC6897 (API Consideration) .. Informational

- Next Step
  - Proceed MPTCP protocol spec to Proposed Standard
  - Publish supplemental documents
    - Implementation advice
    - Use cases and operational experiences
    - Middlebox behavior

Discussions at 88th Meeting (1)

- Two sessions
  - Monday (17:40-19:40) and Wednesday (15:50-16:50)

- How to advance protocol spec?
  - IESG requests strong security mechanism for PS drafts
  - Current consensus: two-pronged approach
    - Prong 1: Minor updates to address some potential risks
      - Provide the same security level as SCTP Dynamic Address Configuration
      - RFC5061 is PS. So, this is good enough to be PS
    - Prong 2: Major updates for more advanced security
      - TCPCrypt can be a good candidate as base technology
      - But, we will need more investigation

Discussions at 88th Meeting (2)

- Q&A session for MPTCP activities in Apple Inc.
  - Invite Stuart Cheshire as a speaker

- Some comments from Stuart
  - MPTCP is currently used only for Siri
    - Migrate between interfaces (3G/LTE, Wifi) based on performance
  - If you want to use MPTCP for your appl, use bugreport system
  - MPTCP traffic seems to go through most of the Internet
    - Most middleboexs don’t affect MPTCP
  - Mobile IP was also considered, but we chose MPTCP
    - More host-level solution and requires home-agents, etc
  - Cannot comment on future plans

Additional Information

MPTCP Implementations

- Linux (plus Android)
  - http://www.multipath-tcp.org/
- FreeBSD
  - http://caia.swin.edu.au/urp/newtcp/mptcp/
- Citrix
  - Netscaler release 10.1
- Apple Inc.
  - Used for Siri
- Others
  - Multipath Networks
    - MPTCP supported router

MPTCP Documentations

- RFCs
  - RFC6181 (Threat Analysis)
  - RFC6182 (Architectural Guideline)
  - RFC6356 (Congestion Control)
  - RFC6824 (Protocol Spec)
  - RFC6897 (API Consideration)

- Technical Background
  - ‘The Resource Pooling Principle’, Damon Wischik, Mark Handley and Marcelo Bagnulo Braun. ACM/SIGCOMM CCR.
  - ‘Practical Congestion Control for Multipath Transport Protocols’, Costin Raiciu, Damon Wischik, Mark Handley, UCL Tech Report
  - ’Control of multipath TCP and optimization of multipath routing in the Internet’, Damon Wischik, Mark Handley and Costin Raiciu, Proc. NetCOOP 2009.