SIGCOMM'20 Tutorial

Multipath Transport Protocols

- Presenters
  - Olivier Bonaventure, UCLouvain
  - Quentin De Coninck, UCLouvain
- Schedule
  - 10:00-11:00 EDT Multipath TCP, Olivier
  - 11:00-11:30 EDT Hands-on, Quentin
  - 12:00-13:00 EDT Multipath QUIC, Quentin
  - 13:00-13:30 EDT Hands-on, Quentin
- **Please download and install our VM**
  - https://github.com/qdeconinck/sigcomm20_mptp_tutorial
  - And use slack channel for questions

Multipath TCP

The VM prepared by Quentin De Coninck
for the hands-on is available from
https://github.com/qdeconinck/sigcomm20_mptp_tutorial

Olivier Bonaventure
https://inl.info.ucl.ac.be
https://perso.uclouvain.be/olivier.bonaventure

Thanks to Sébastien Barré, Christoph Paasch, Grégory Detal, Mark Handley, Costin Raiciu, Alan Ford, Micchio Honda, Fabien Duchene, Quentin De Coninck, Benjamin Hesmans, Viet-Hoang Tran and many others

SIGCOMM'20 Tutorial

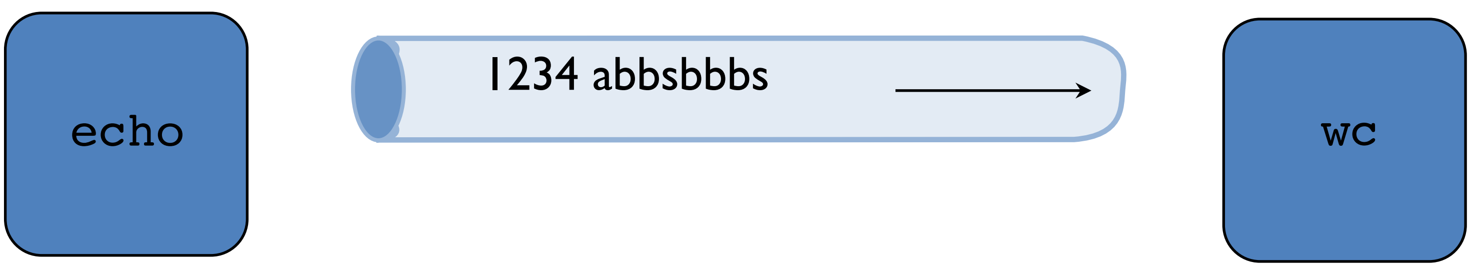
Agenda

- Why Multipath TCP ?
- Barriers to Multipath TCP
- A closer look at the Multipath TCP protocol

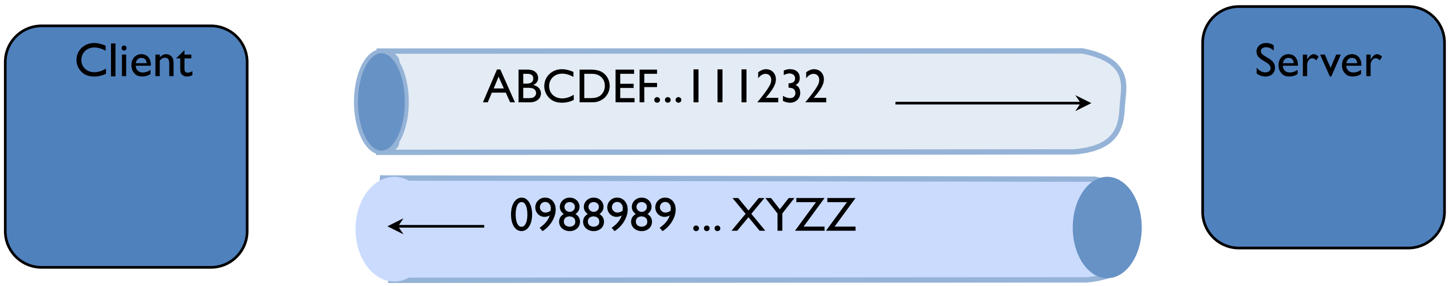
The Unix pipe model

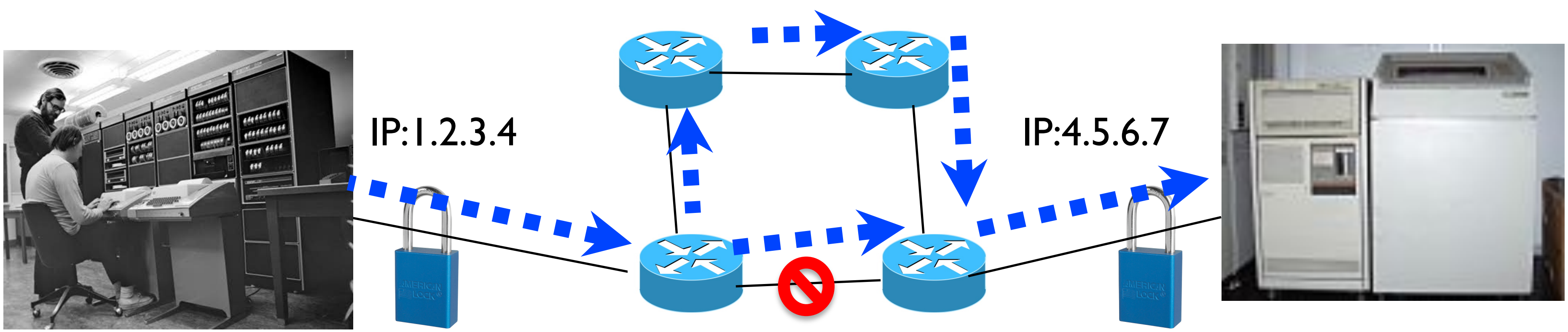
Terminal — bash — 49×7

```
Last login: Tue Nov 13 10:07:47 on ttys006
You have new mail.
mbpobo:~ obo$ echo "1234 abbsbbbs" | wc -c
      14
```



The TCP bytestream model





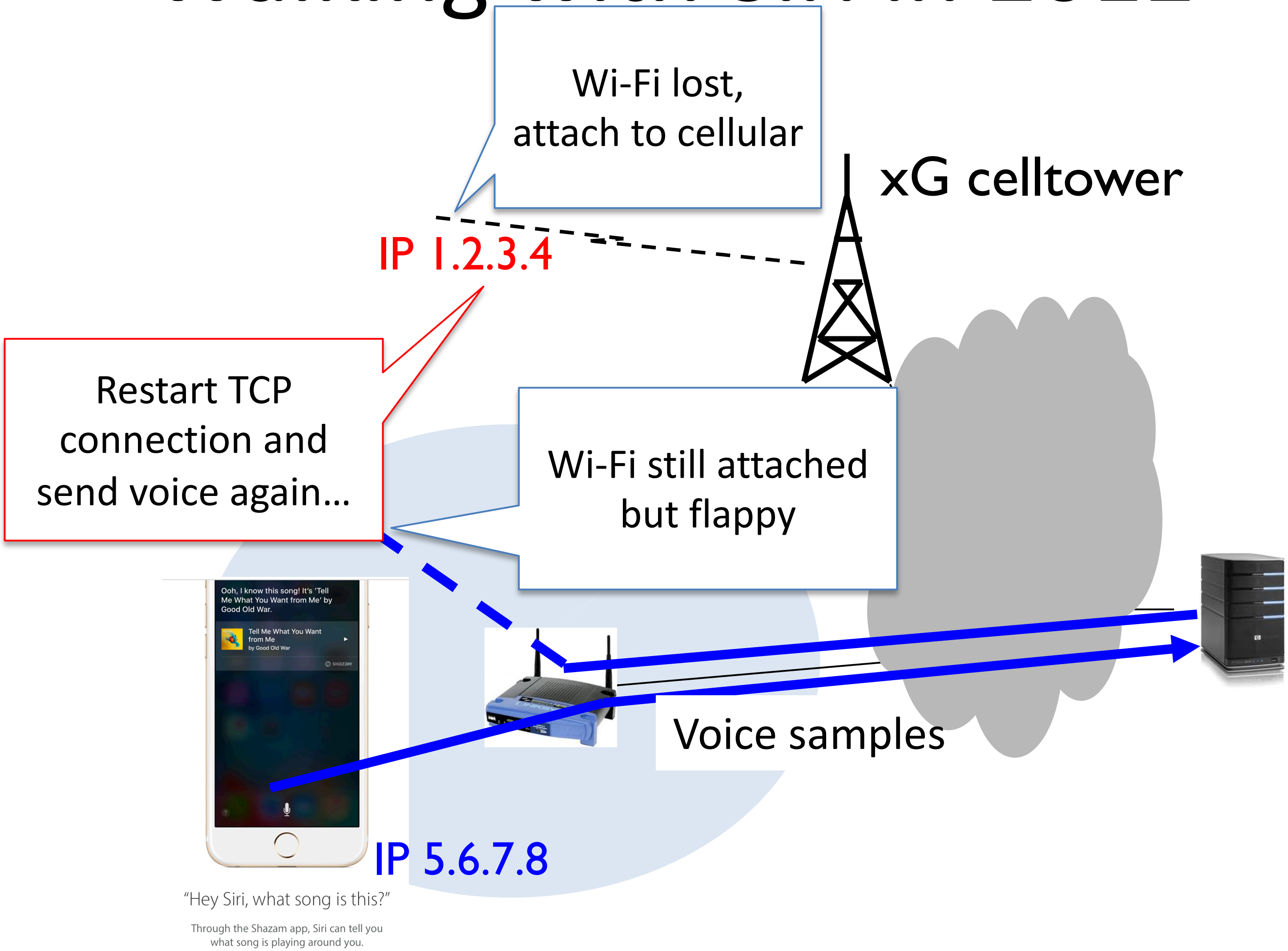
Endhosts have evolved

Mobile devices have multiple wireless interfaces and those using a single interface often support IPv4 and IPv6

Two main types of use cases

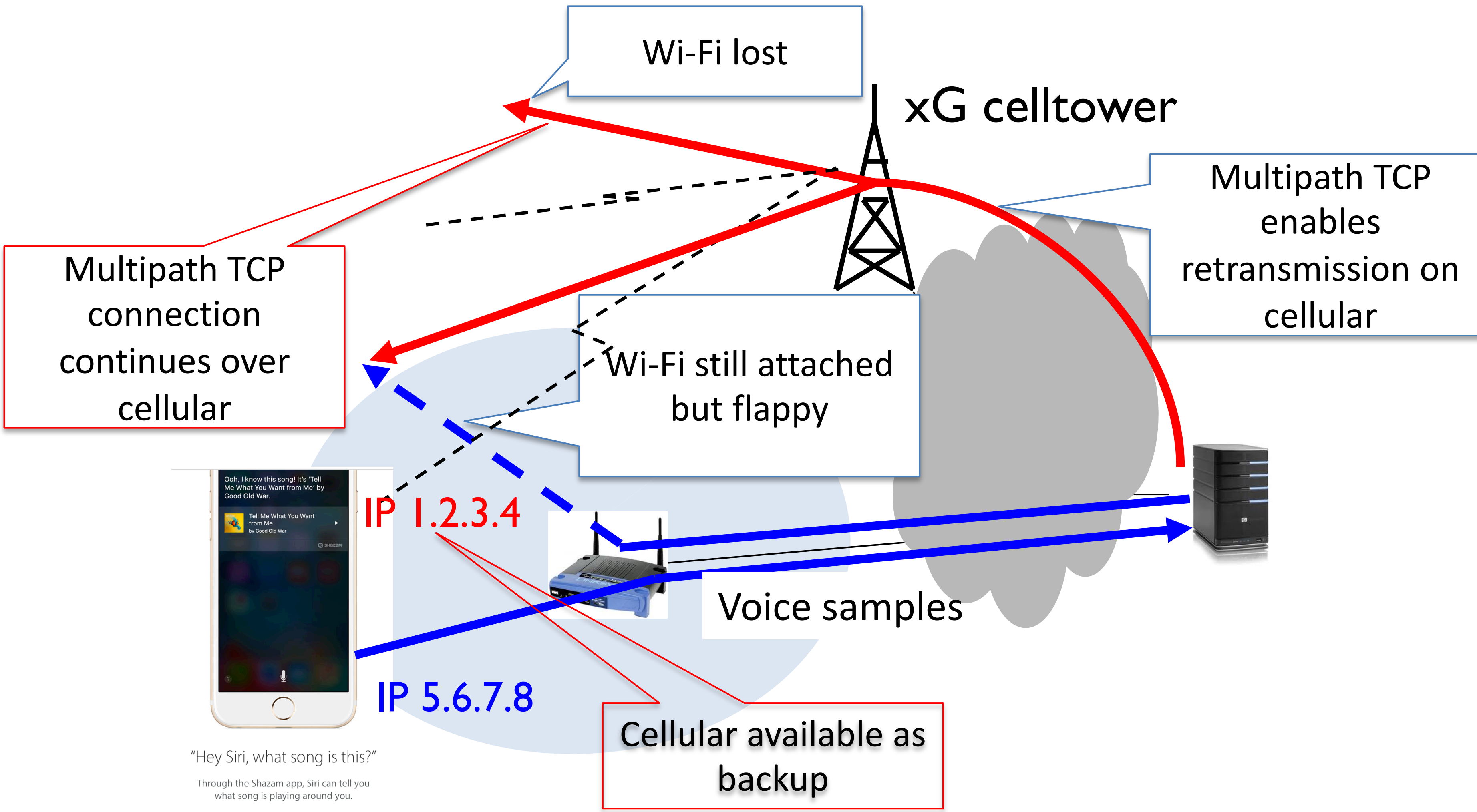
- Seamless handovers on smartphones
  - With Multipath TCP, connections persit on smartphones even when they move from Wi-Fi to cellular and back
- Bandwidth aggregation
  - With Multipath TCP, heterogeneous networks can be combined to obtain higher bandwidth

Walking with Siri in 2012



Siri uses long-lived TLS connections

Walking with Siri and Multipath TCP



Same applies to Apple Music and Apple Maps

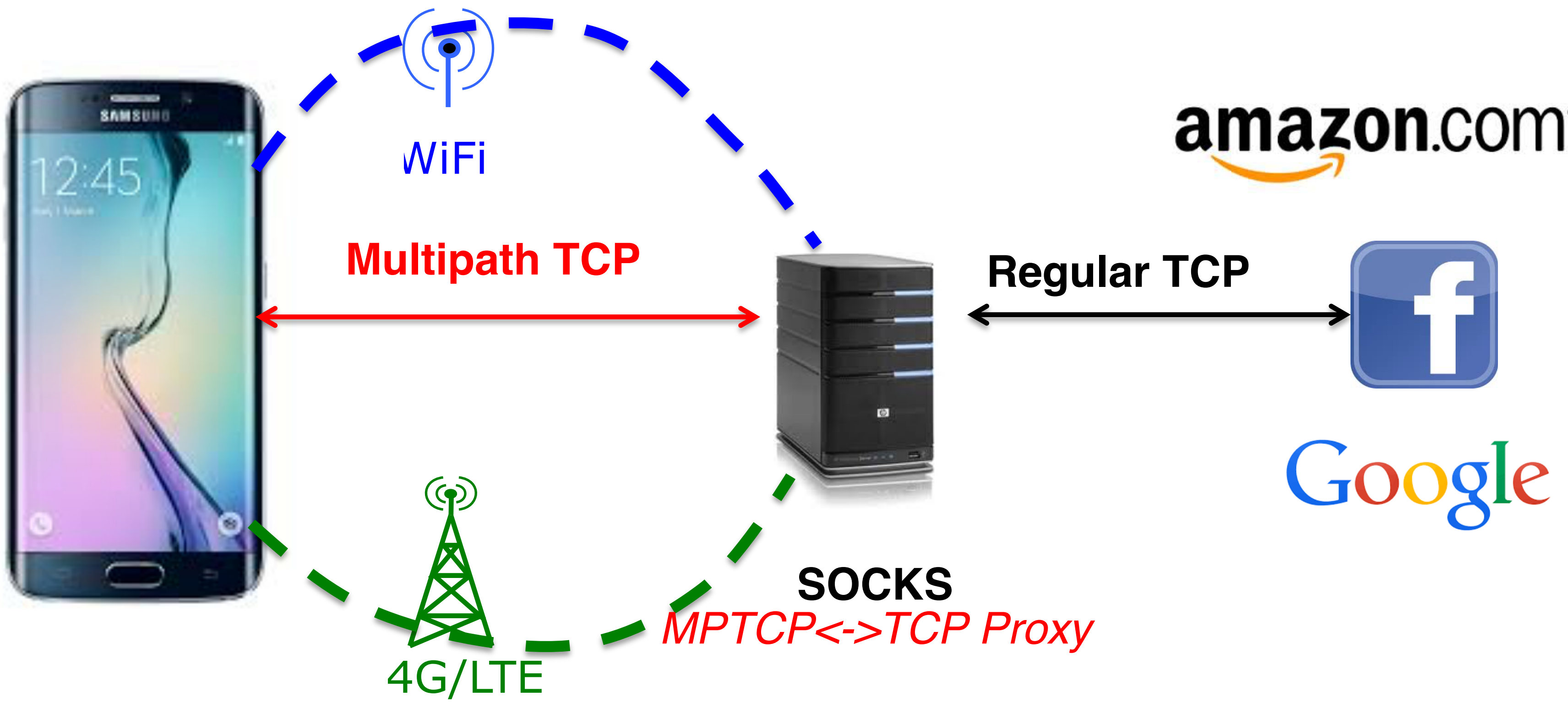
Limitations of this use case today

- Multipath TCP needed on smartphone+server
  - Smartphone support for Multipath TCP
    - Apple iOS since 2013, initially for Siri only and later for third-party applications and also Maps and Music
    - Android, not part of the official release (Linux kernel)
      - Samsung, LG, Huawei use it in specific markets
  - Server support
    - Well tested Linux patch https://www.multipath-tcp.org
    - Ongoing work to bring it to mainline Linux kernel
  - Load balancers

Duchene, F., & Bonaventure, O. (2017, October). Making Multipath TCP friendlier to load balancers and anycast. In 2017 IEEE 25th International Conference on Network Protocols (ICNP) (pp. 1-10). IEEE.

Multipath TCP use cases
High bandwidth on smartphones

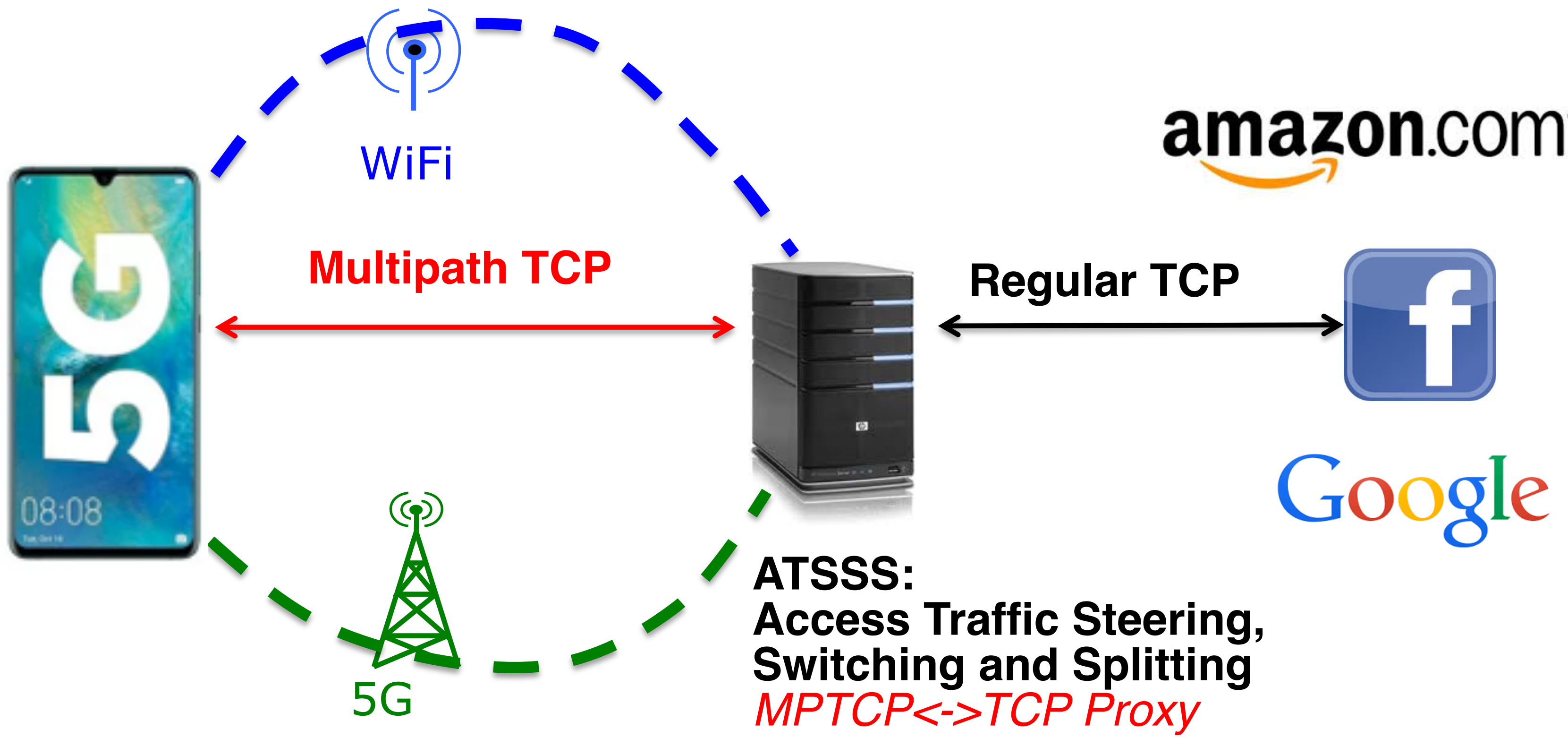
- Koreans want 800+ Mbps on smartphones



Bonaventure, O., and Seo, S.. "Multipath TCP deployments." *IETF Journal* 12.2 (2016): 24-27.

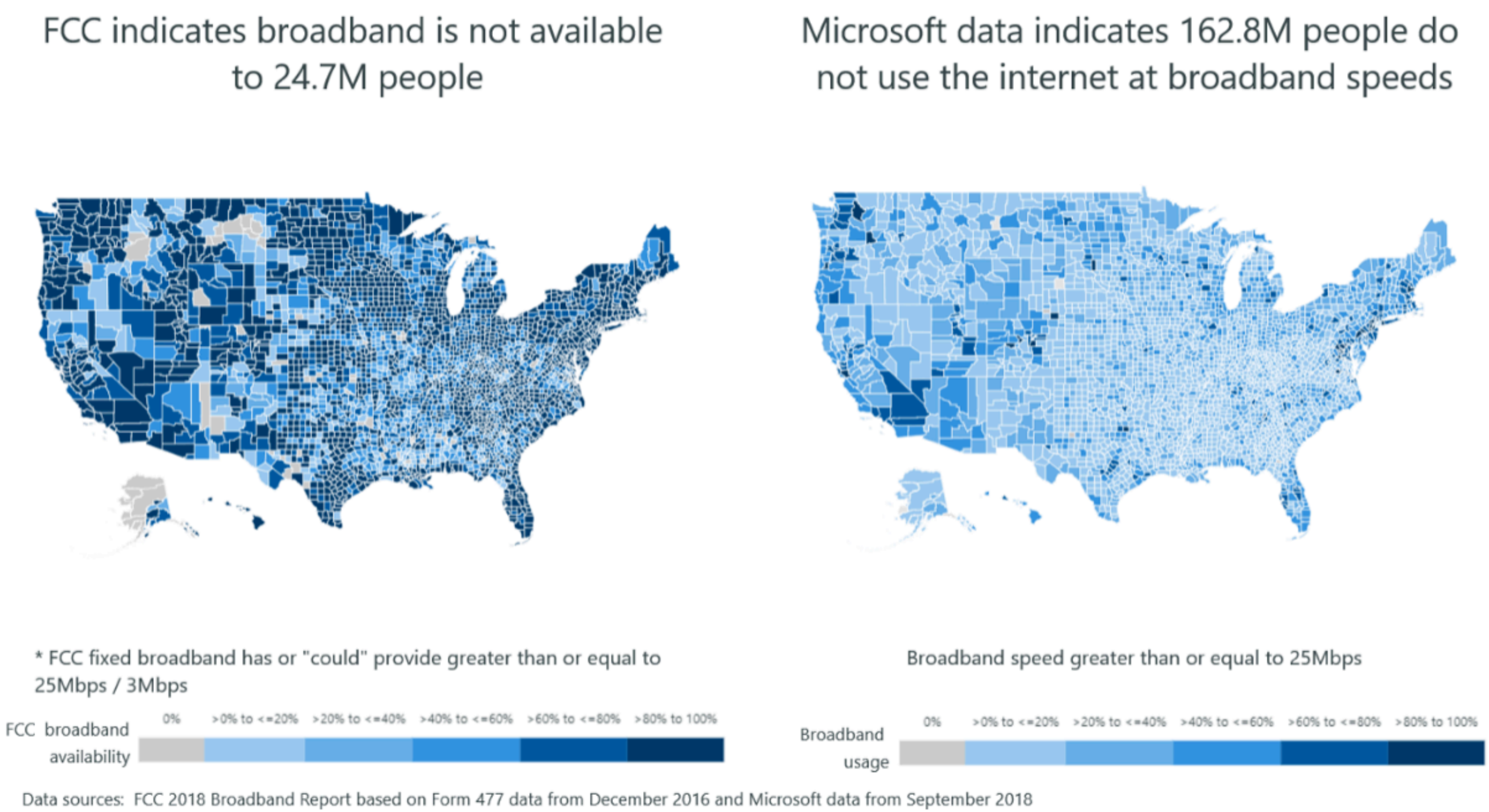
5G/Wi-Fi coexistence

- Same approach as for hybrid access networks



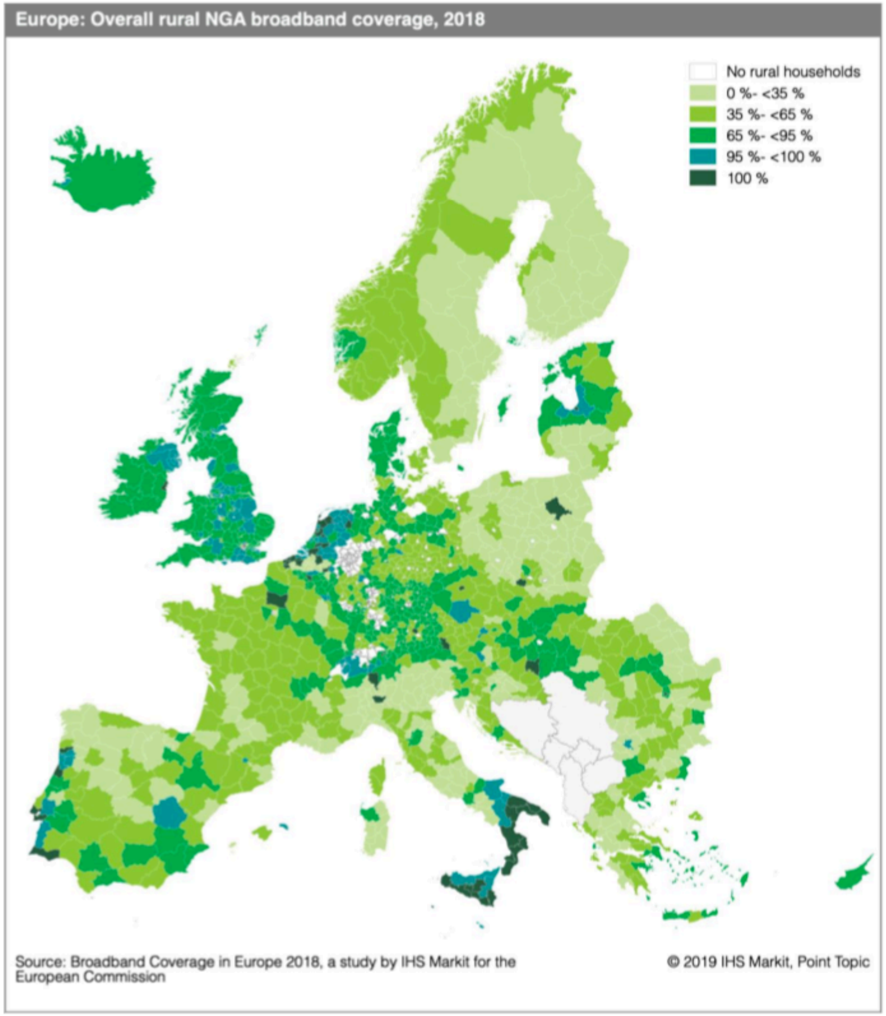
Bonaventure, O., Boucadair, M. et al. ”0-RTT Convert Protocol" *RFC8803, 2020*
https://tools.ietf.org/html/rfc8803

Broadband in the USA



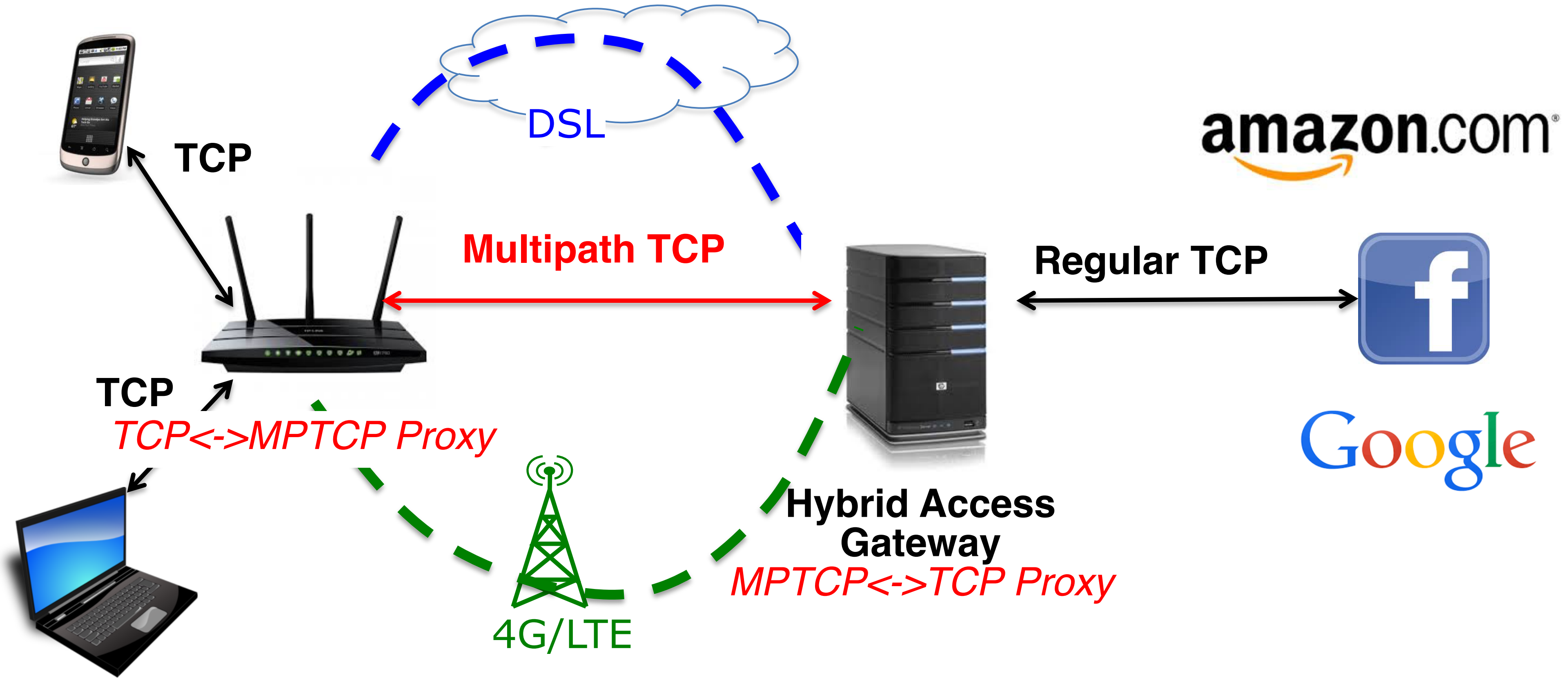
Source: https://blogs.microsoft.com/on-the-issues/2019/04/08/
its-time-for-a-new-approach-for-mapping-broadband-data-to-better-serve-americans/

Fast broadband in rural areas Europe



Source: https://ec.europa.eu/digital-single-market/en/news/study-broadband-coverage-europe-2

Hybrid Access Networks

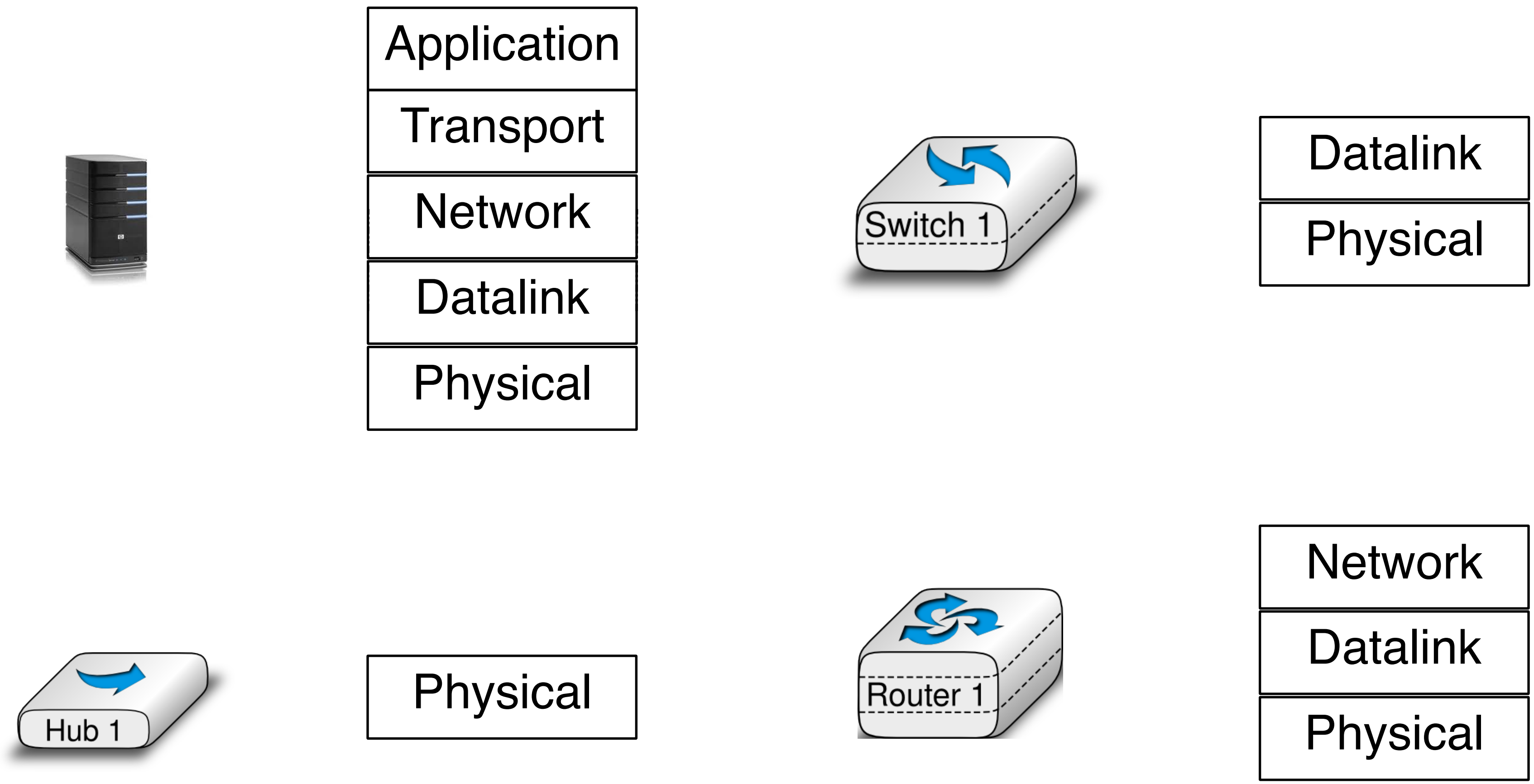


Keukeleire, N., et al. (2020). Increasing Broadband Reach with Hybrid Access Networks. *IEEE Communications Standards Magazine*, *4*(1), 43-49.

Agenda

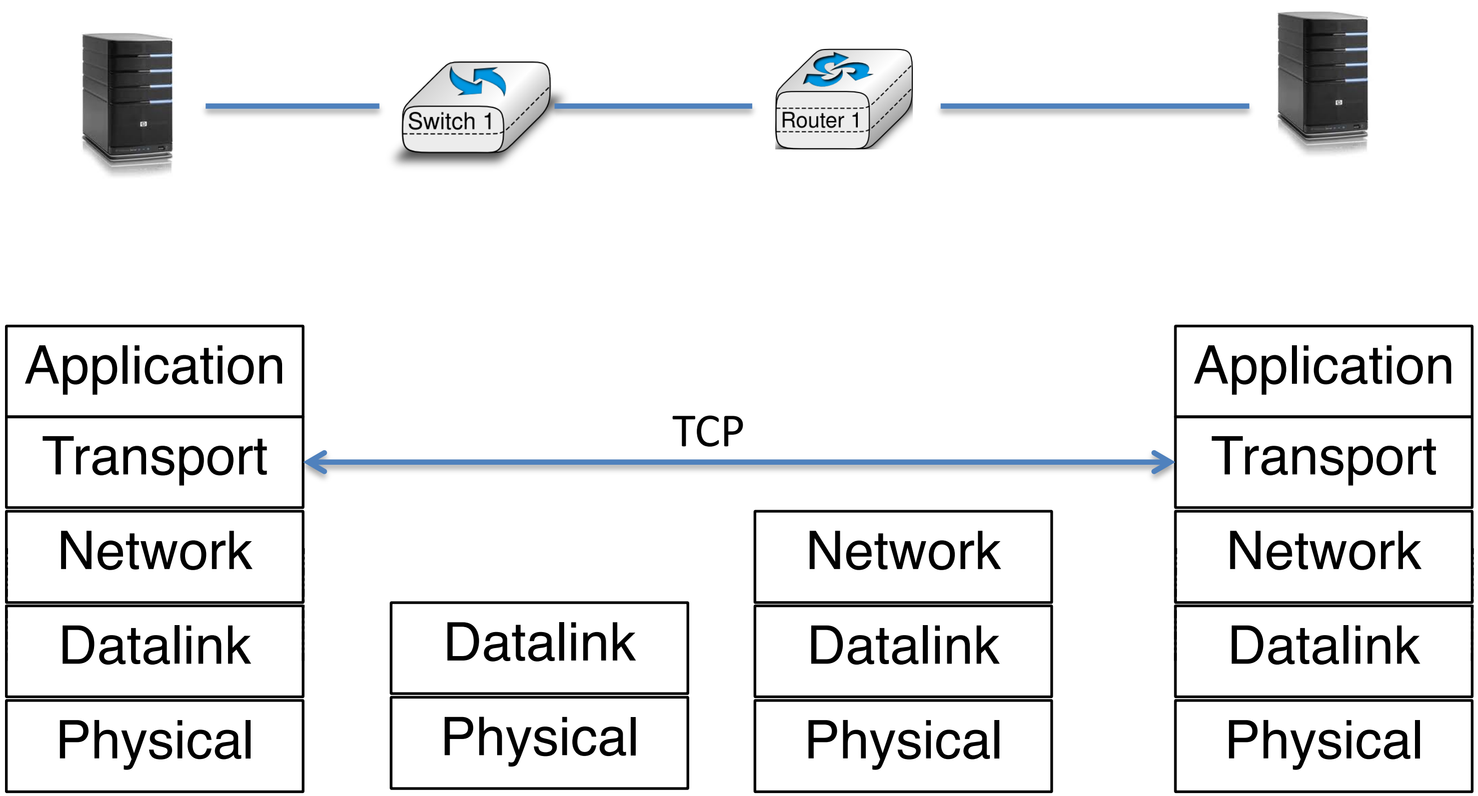
- Why Multipath TCP ?
- Barriers to Multipath TCP
- A closer look at the Multipath TCP protocol

The Internet architecture that we explain to our students



O. Bonaventure, et al. *Computer networking : Principles, Protocols and Practice*, open ebook, https://www.computer-networking.info

A typical "academic" network



In reality

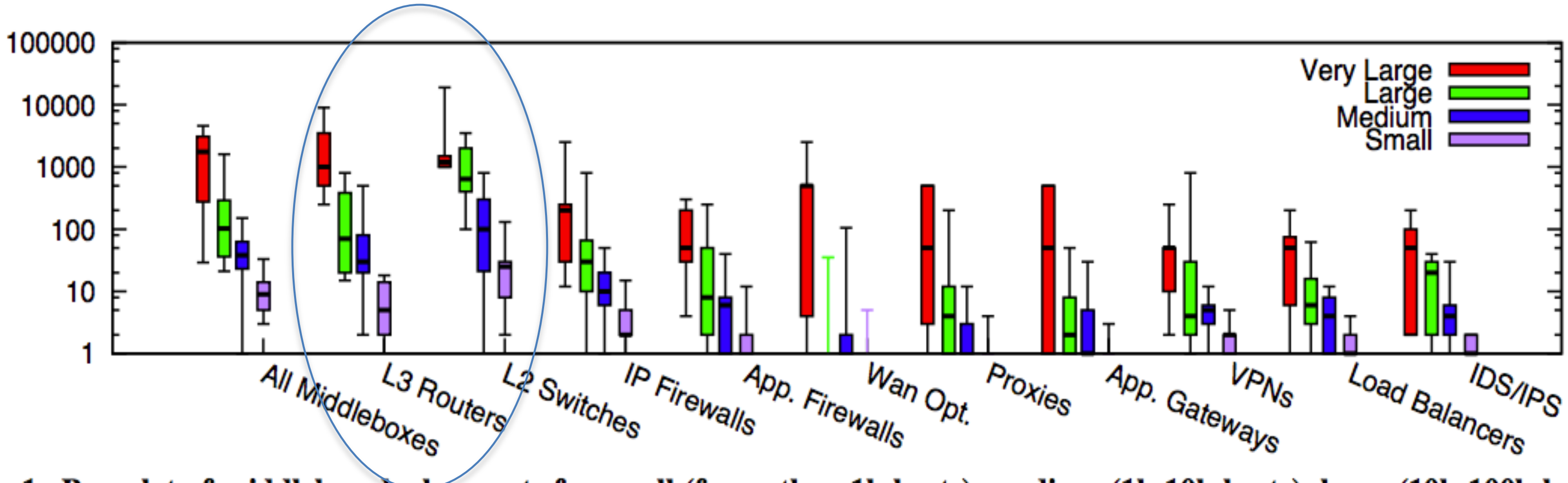


Figure 1: Box plot of middlebox deployments for small (fewer than 1k hosts), medium (1k-10k hosts), large (10k-100k hosts), and very large (more than 100k hosts) enterprise networks. Y-axis is in log scale.

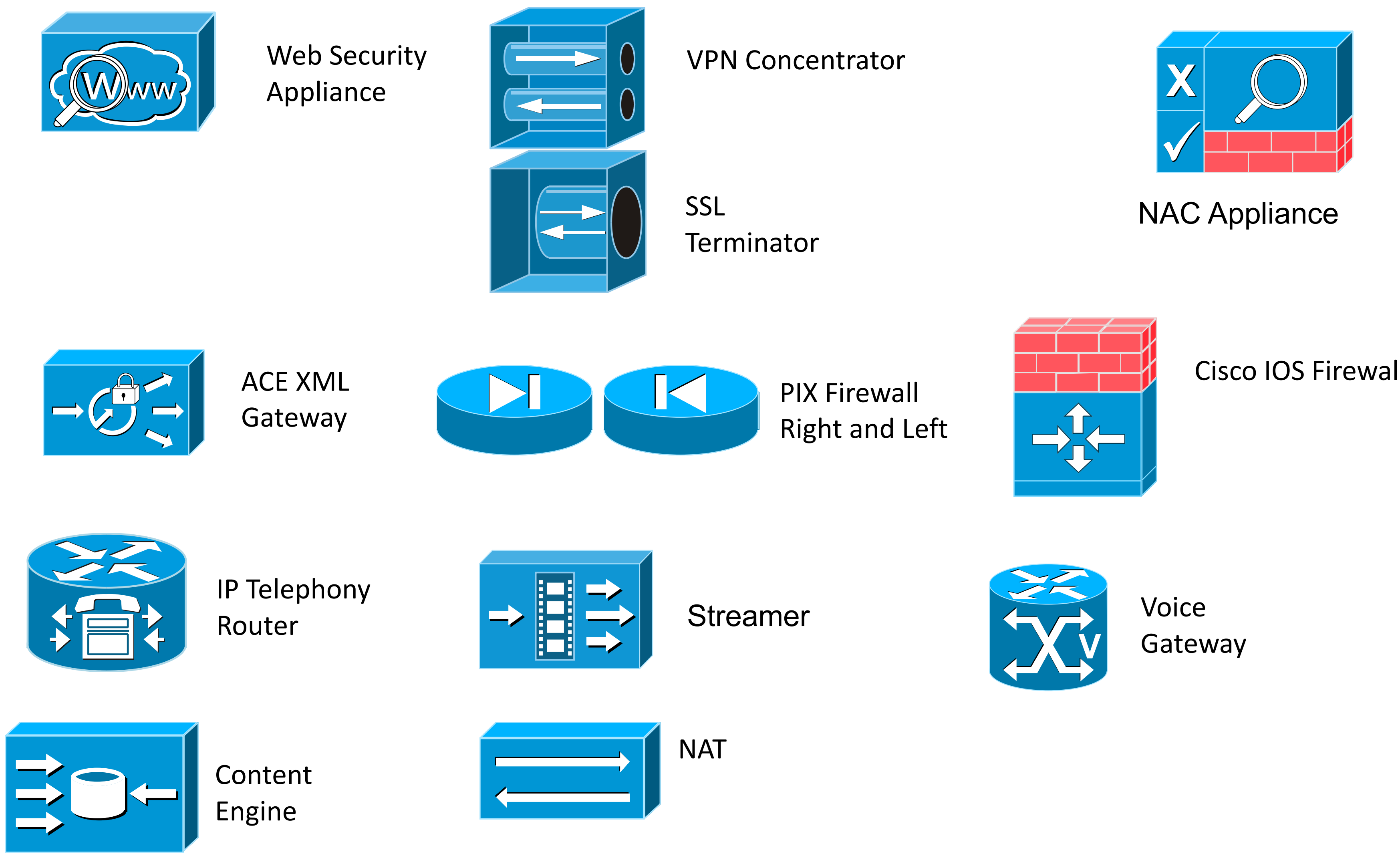
– almost as many middleboxes as routers

– various types of middleboxes are deployed

Sherry, Justine, et al. "*Making middleboxes someone else's problem: Network processing as a cloud service*." Proceedings of the ACM SIGCOMM 2012 conference. ACM, 2012.

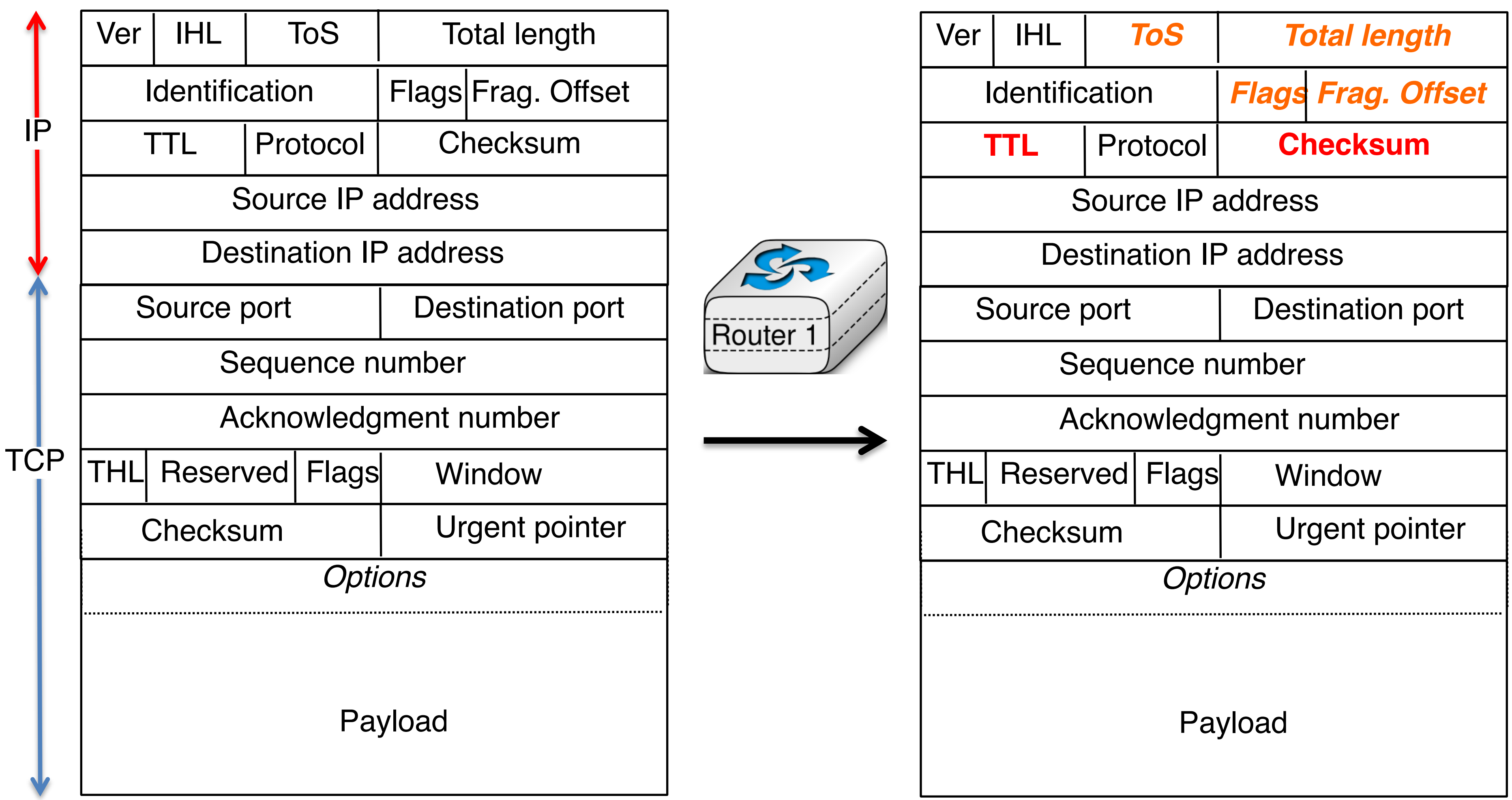
Honda, Michio, et al. "*Is it still possible to extend TCP?*" Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, 2011.

A middlebox zoo

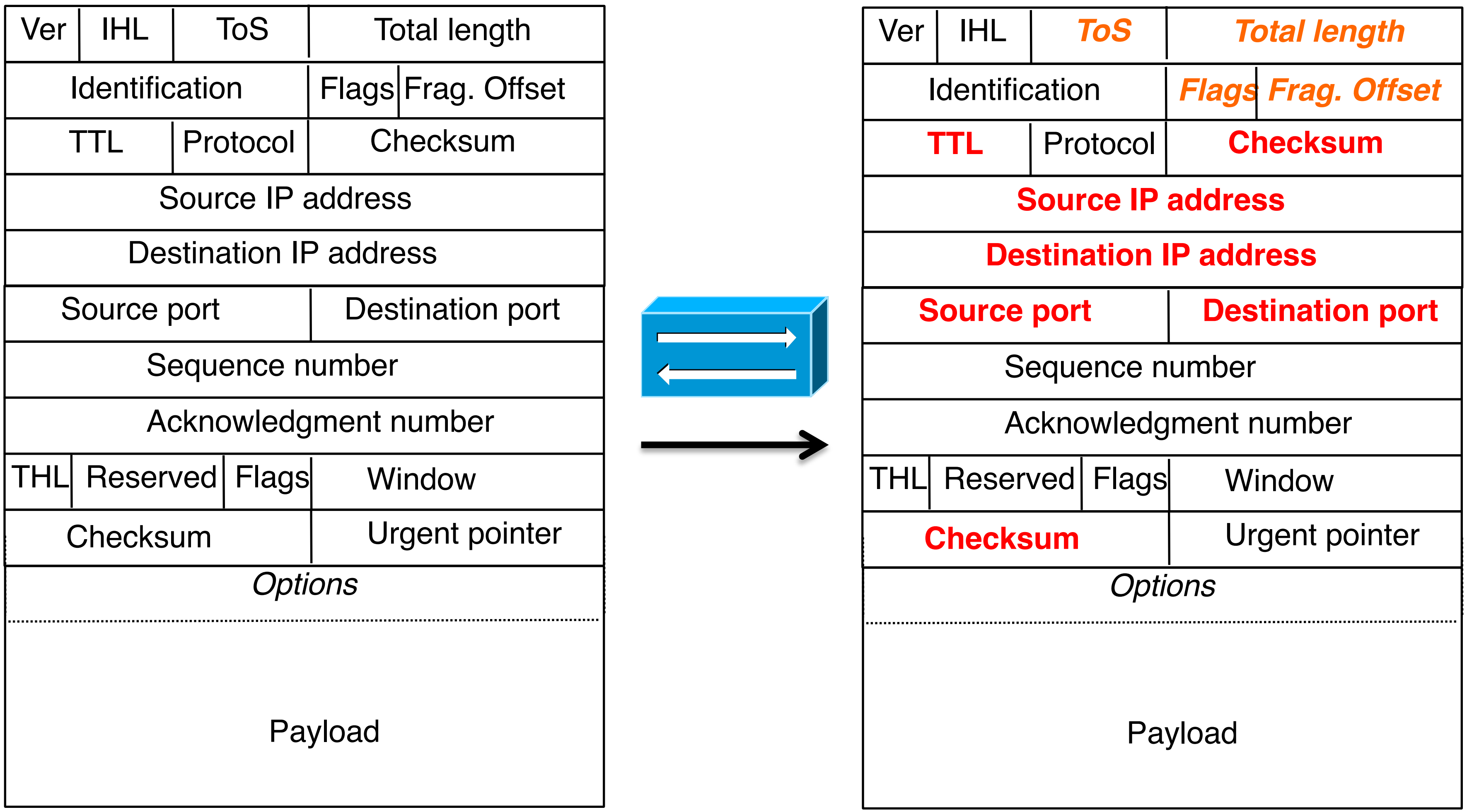


http://www.cisco.com/web/about/ac50/ac47/2.html

TCP segments processed by a router

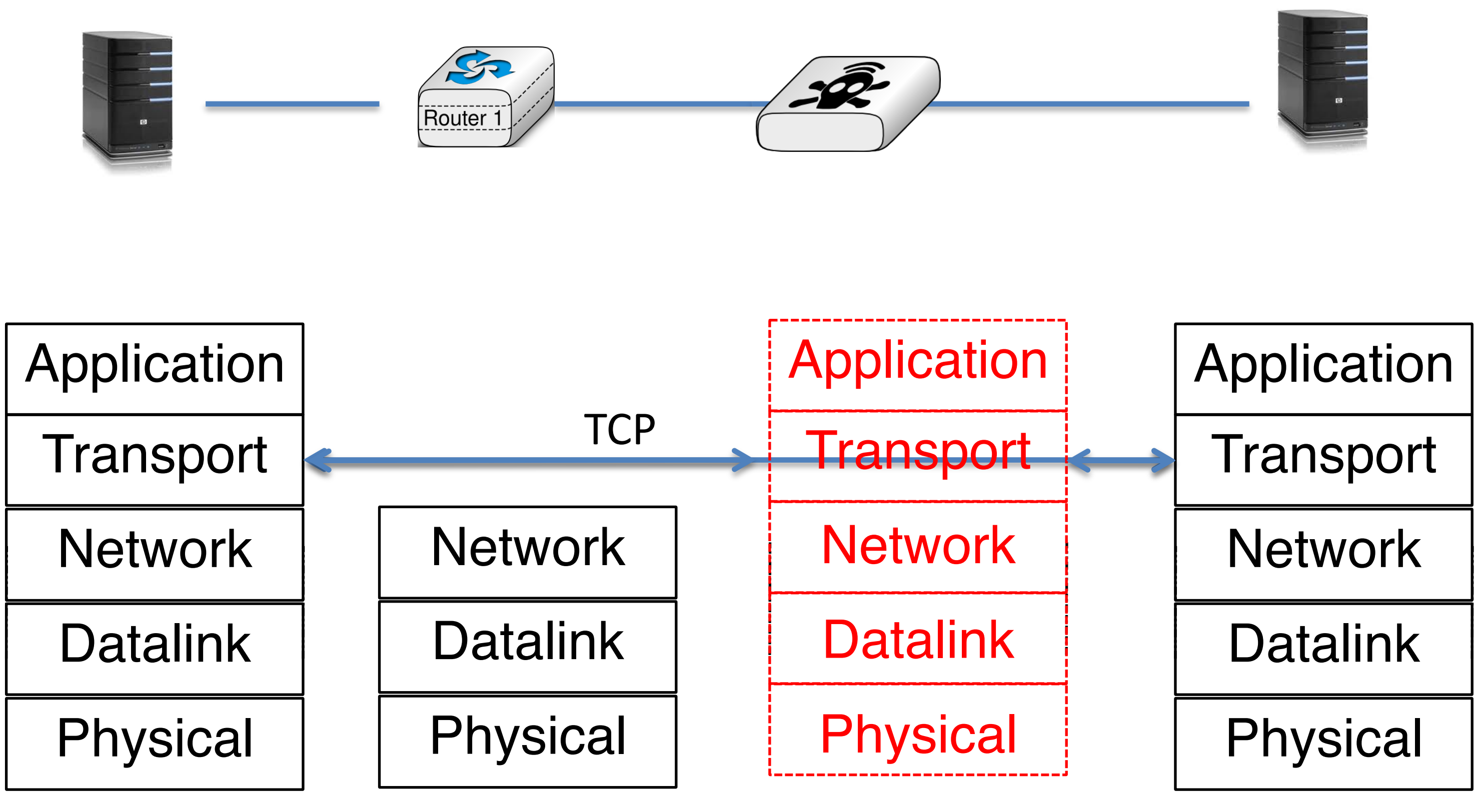


TCP segments processed by a NAT



How to model those middleboxes ?

- In the official architecture, they do not exist
- In reality...



How transparent is the Internet ?

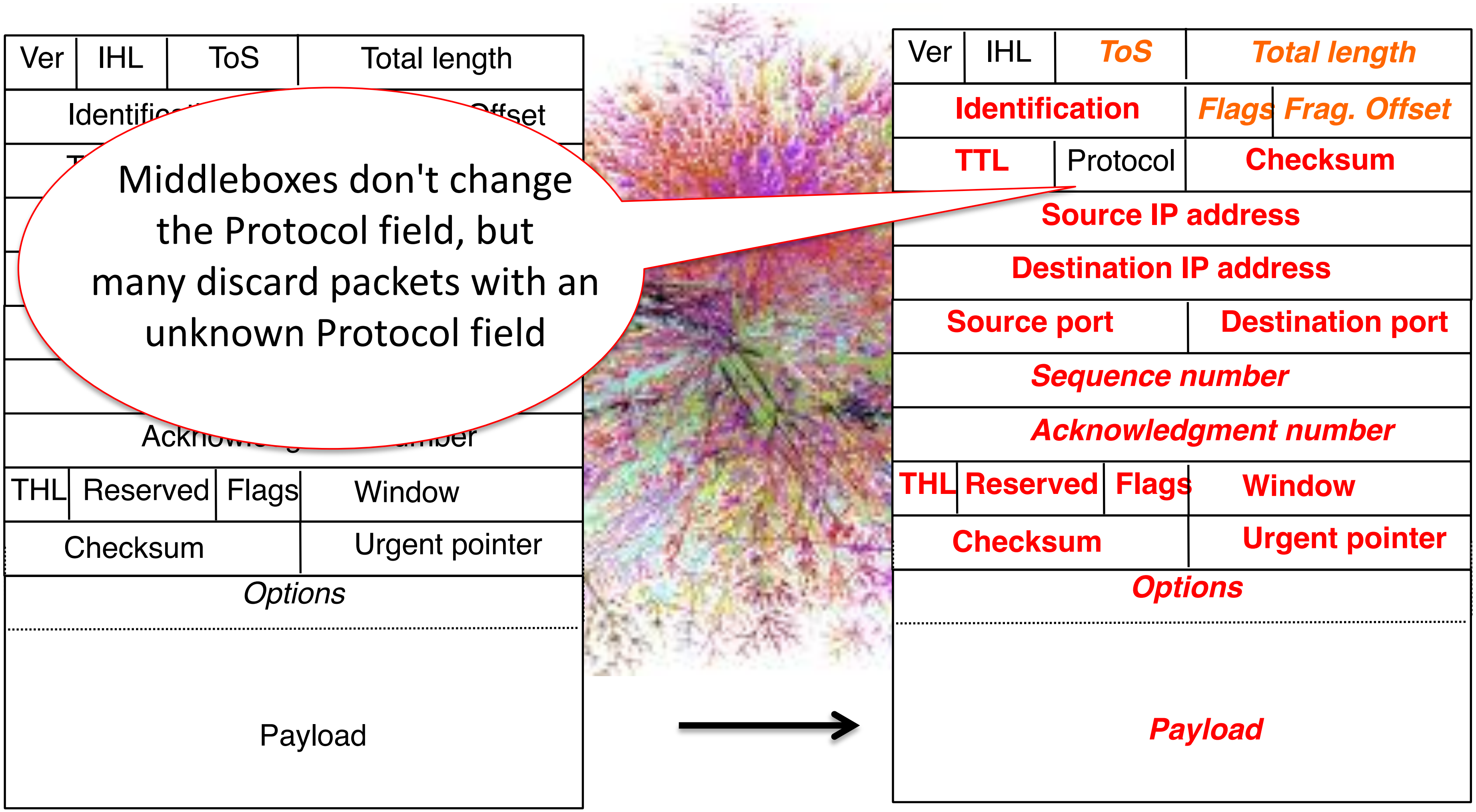
- 25th September 2010 to 30th April 2011
- 142 access networks
- 24 countries
- Sent specific TCP segments from client to a server in Japan

Table 2: Experiment Venues

Country	*Home*	*Hotspot*	*Cellular*	*Univ*	*Ent*	*Hosting*	Total
Australia	0	2	0	0	0	1	3
Austria	0	0	0	0	1	0	1
Belgium	4	0	0	1	0	0	5
Canada	1	0	1	0	1	0	3
Chile	0	0	0	0	1	0	1
China	0	7	0	0	0	0	7
Czech	0	2	0	0	0	0	2
Denmark	0	2	0	0	0	0	2
Finland	1	0	0	3	2	0	6
Germany	3	1	3	4	1	0	12
Greece	2	0	1	0	0	0	3
Indonesia	0	0	0	3	0	0	3
Ireland	0	0	0	0	0	1	1
Italy	1	0	0	0	1	0	2
Japan	19	10	7	3	2	0	41
Romania	1	0	0	0	0	0	1
Russia	0	1	0	0	0	0	1
Spain	0	1	0	1	0	0	2
Sweden	1	0	0	0	0	0	1
Switzerland	2	0	0	0	0	0	2
Thailand	0	0	0	0	2	0	2
U.K.	10	4	4	2	1	1	22
U.S.	3	4	4	0	4	2	17
Vietnam	1	0	0	0	1	0	2
Total	49	34	20	17	17	5	142

Honda, Michio, et al. "*Is it still possible to extend TCP?*" Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, 2011.

End-to-end transparency today



Agenda

- Why Multipath TCP ?
- Barriers to Multipath TCP
- A closer look at the Multipath TCP protocol

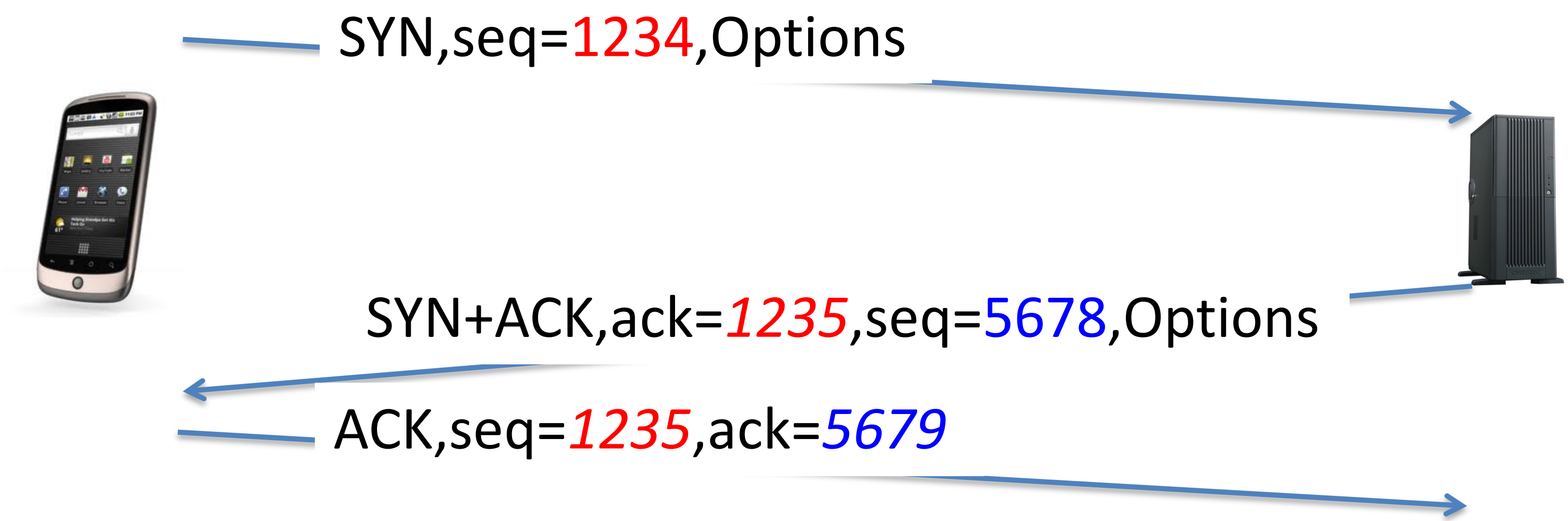
Design objectives

- Multipath TCP is an *evolution* of TCP

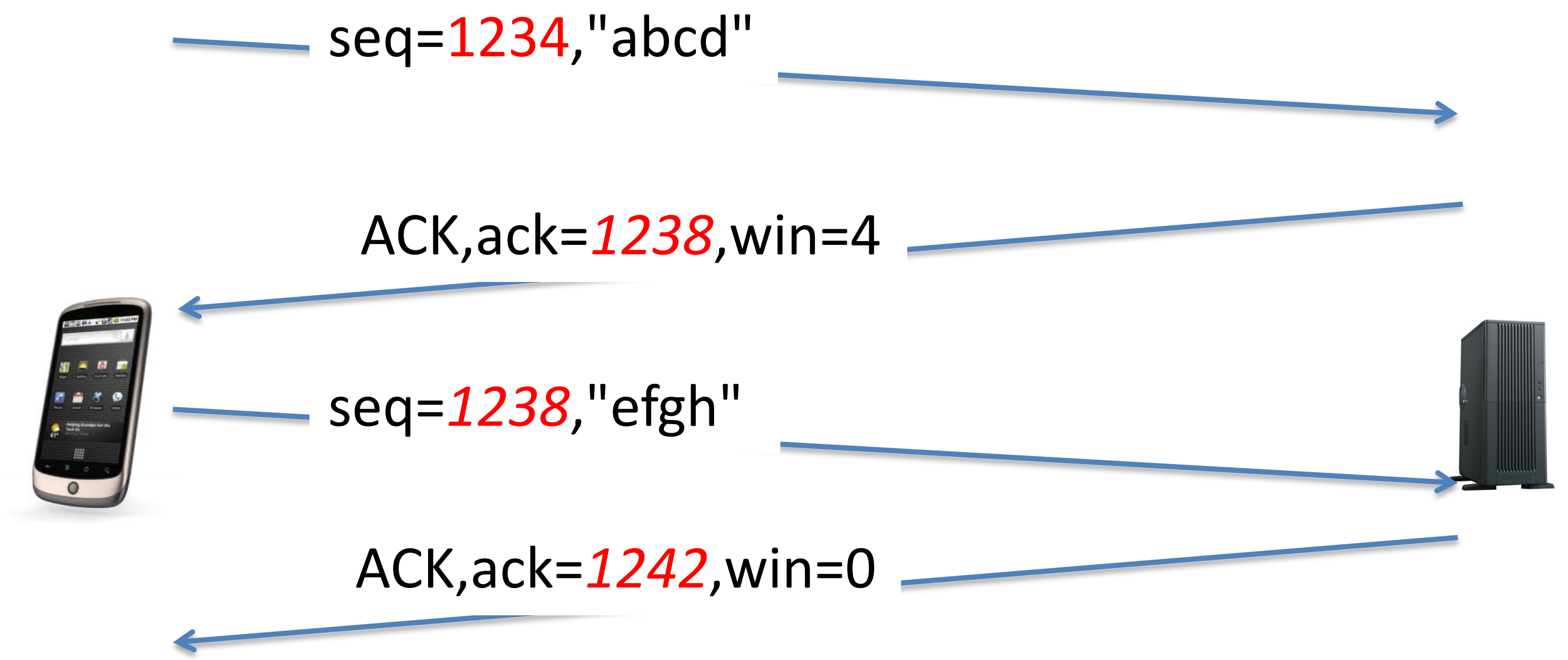
- Design objectives
  - Support unmodified applications
  - Work over today's networks (IPv4 and IPv6)
  - Works in all networks where regular TCP works

TCP Connection establishment

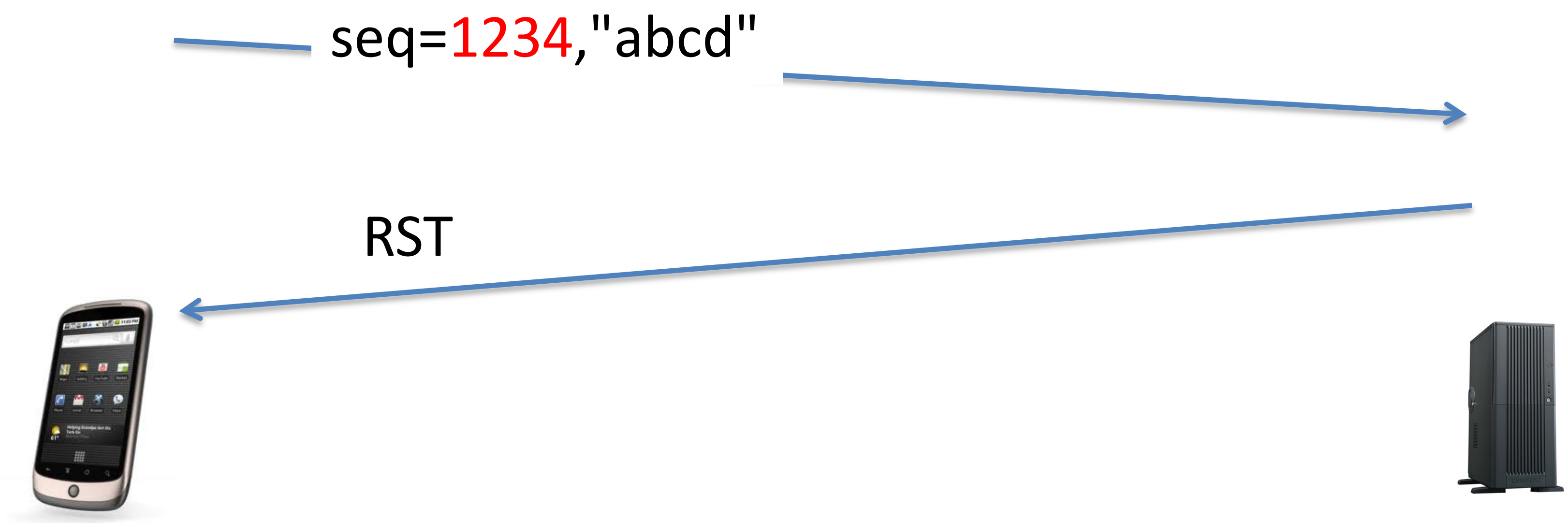
- Three-way handshake



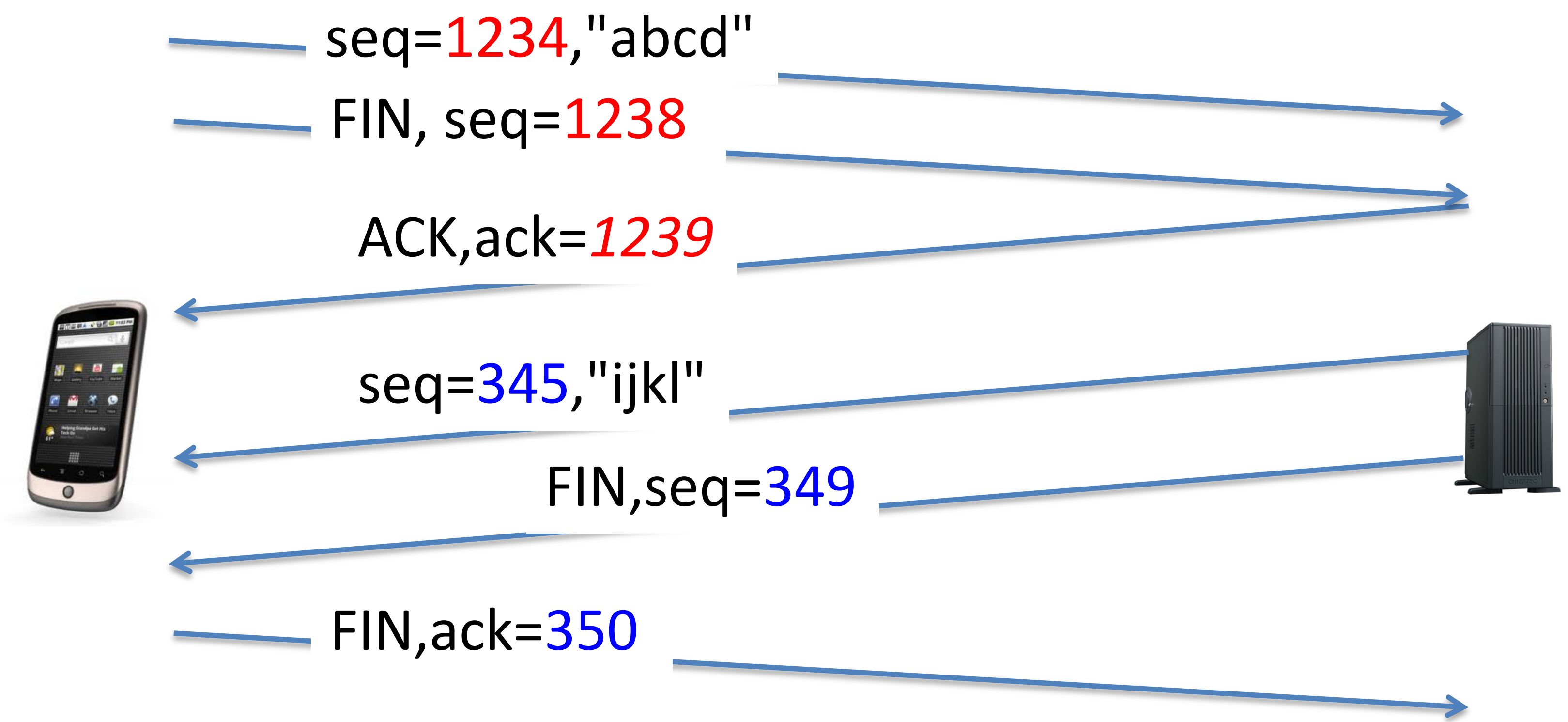
Data transfer



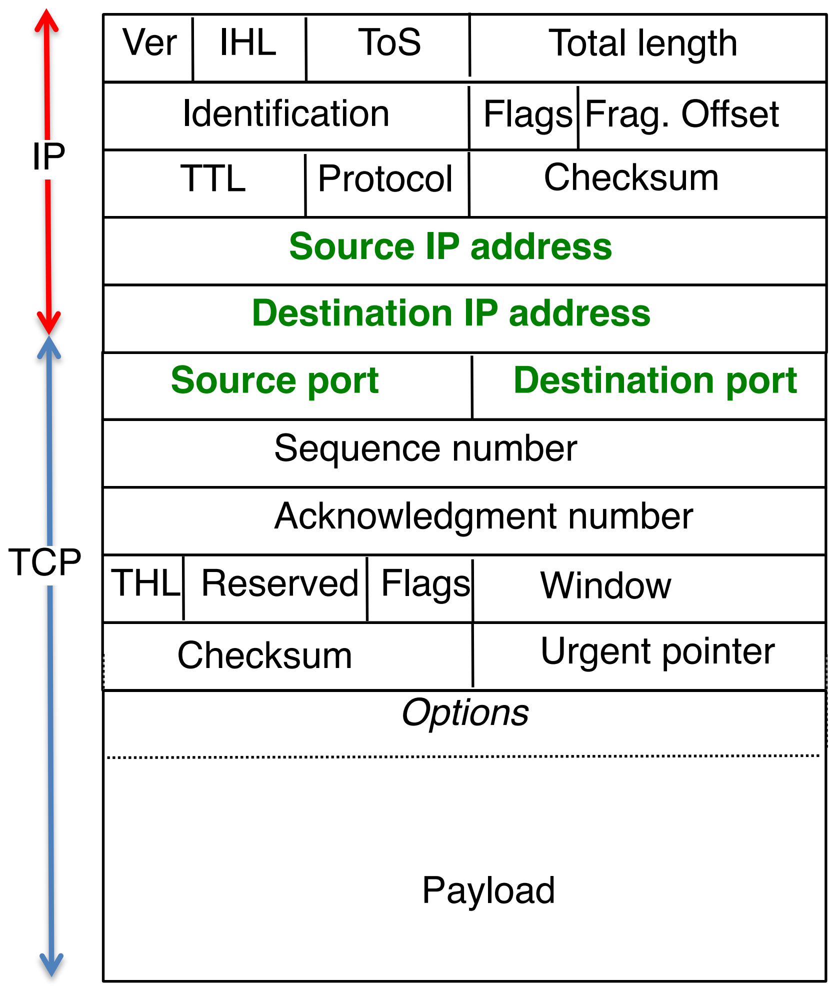
Connection release



Connection release



Identification of a TCP connection

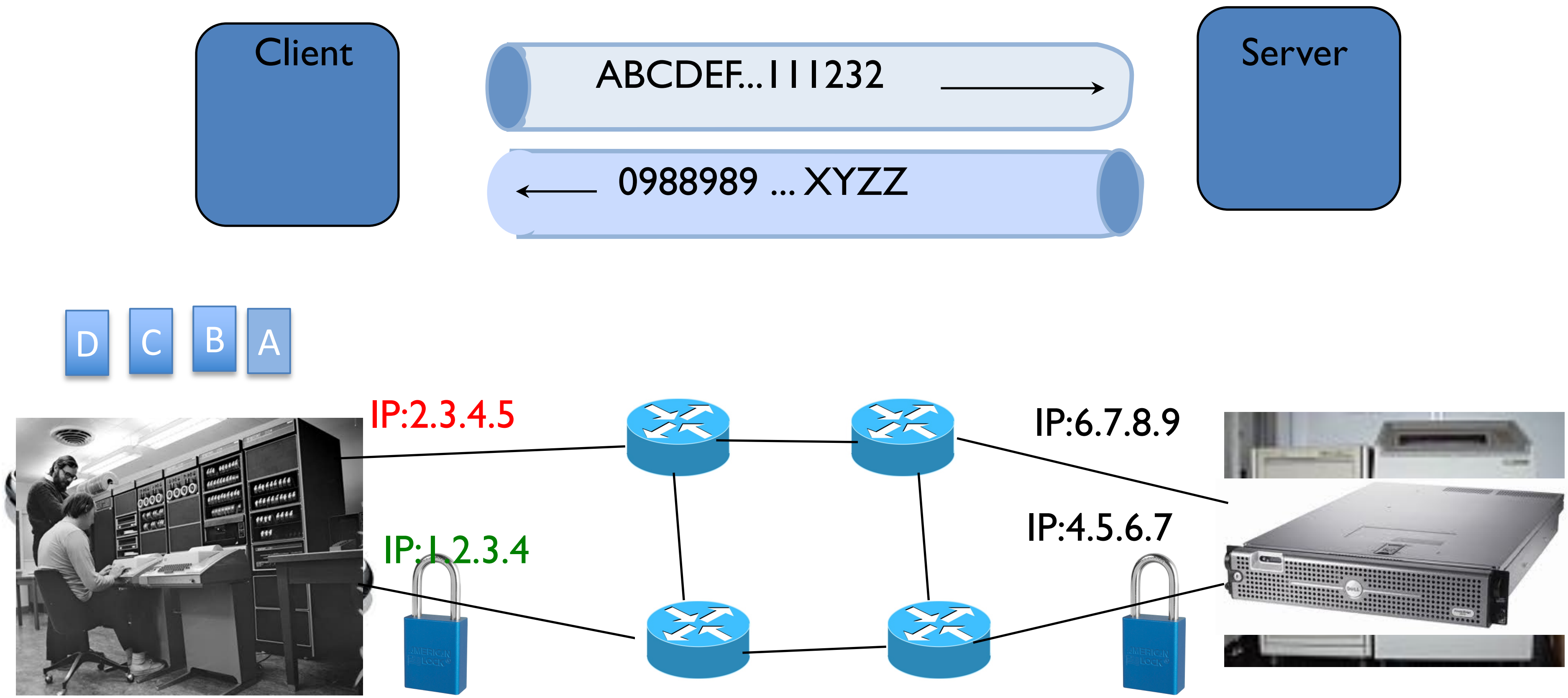


Four tuple

- IP_{source}
- IP_{dest}
- $Port_{source}$
- $Port_{dest}$

All TCP segments contain the four tuple

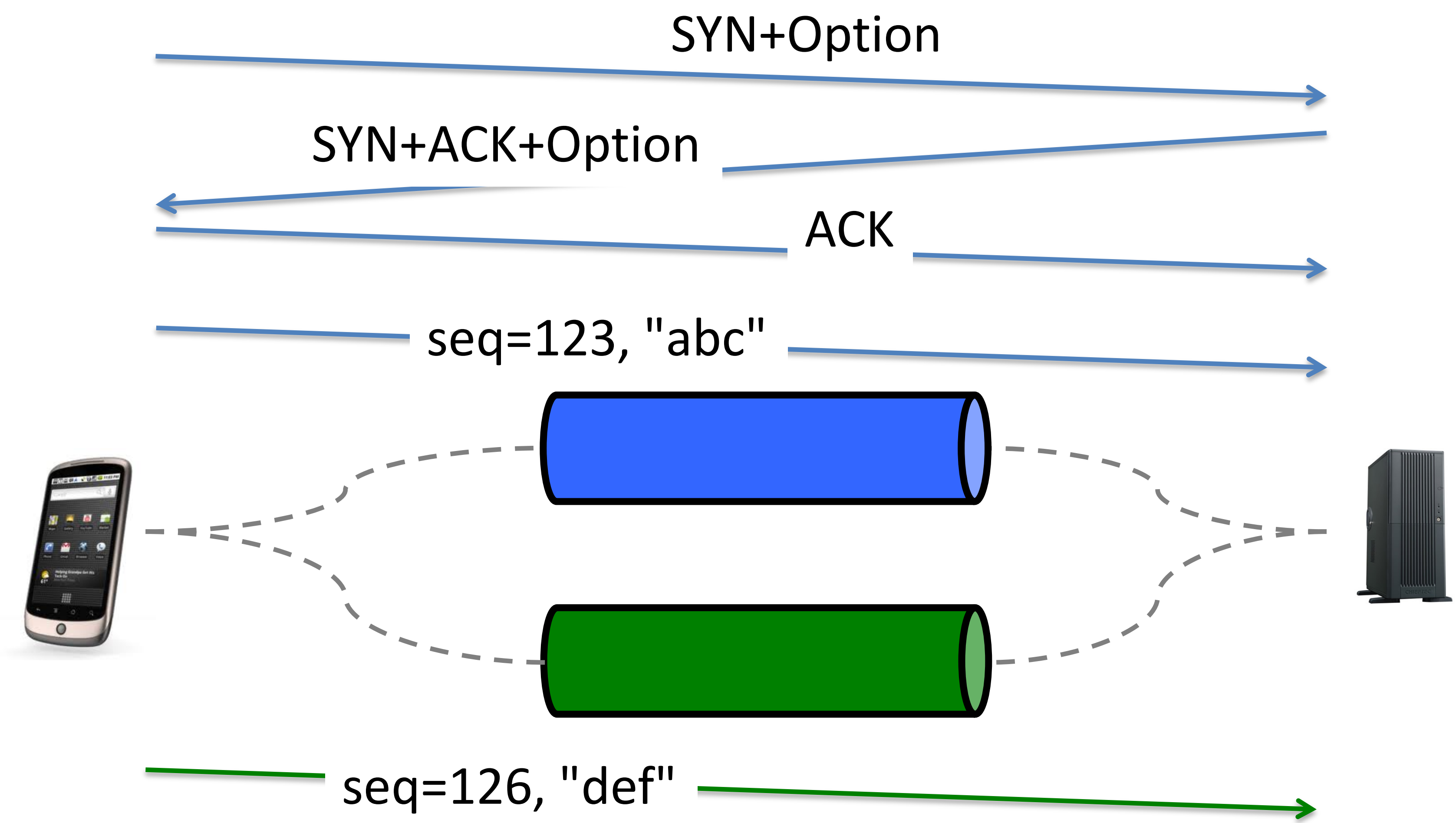
The *new* bytestream model



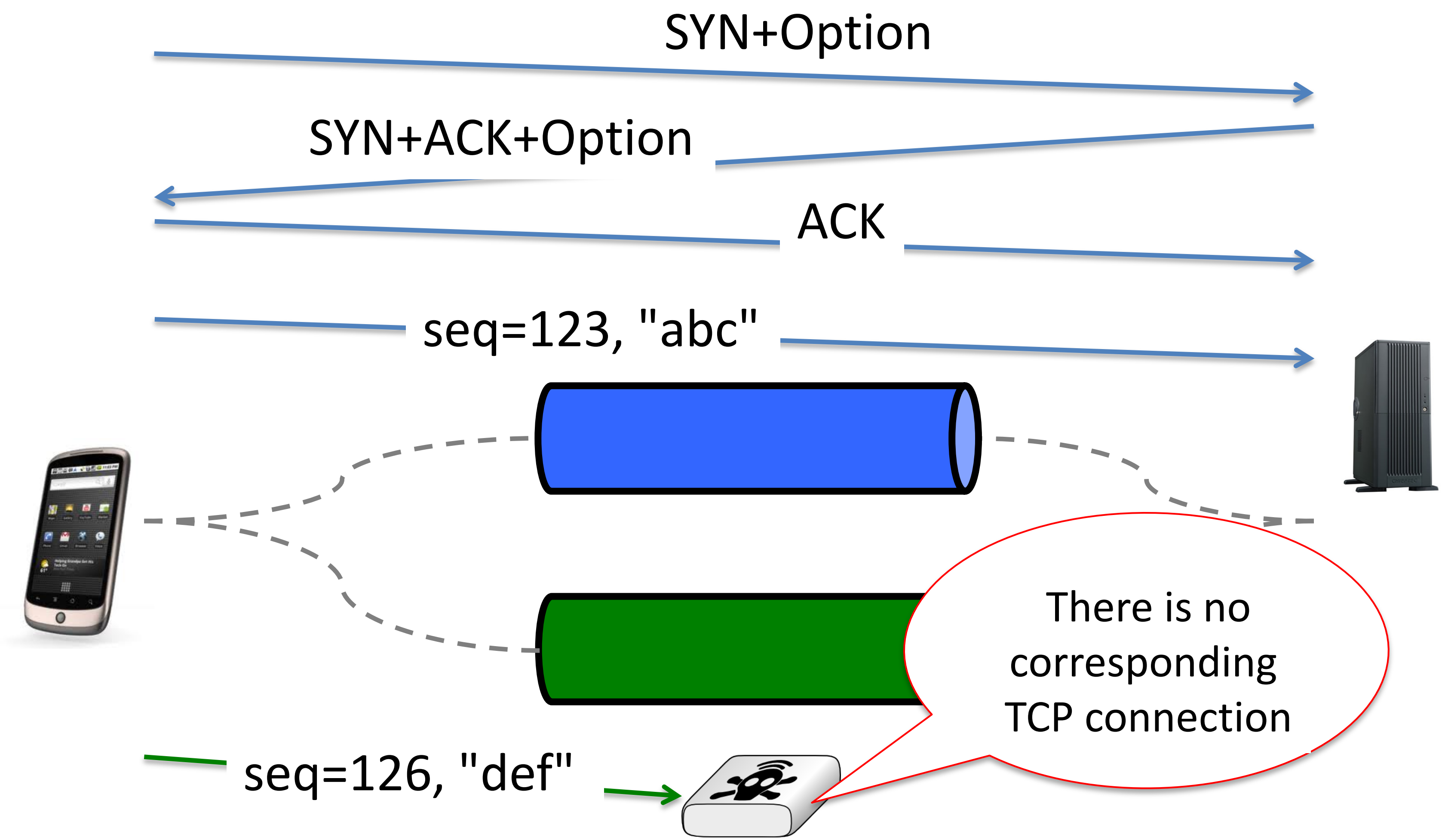
The Multipath TCP protocol

- Control plane
  - How to manage a Multipath TCP connection that uses several paths ?
- Data plane
  - How to transport data ?
- Congestion control
  - How to control congestion over multiple paths ?

A naïve Multipath TCP



A naïve Multipath TCP
In today's Internet ?

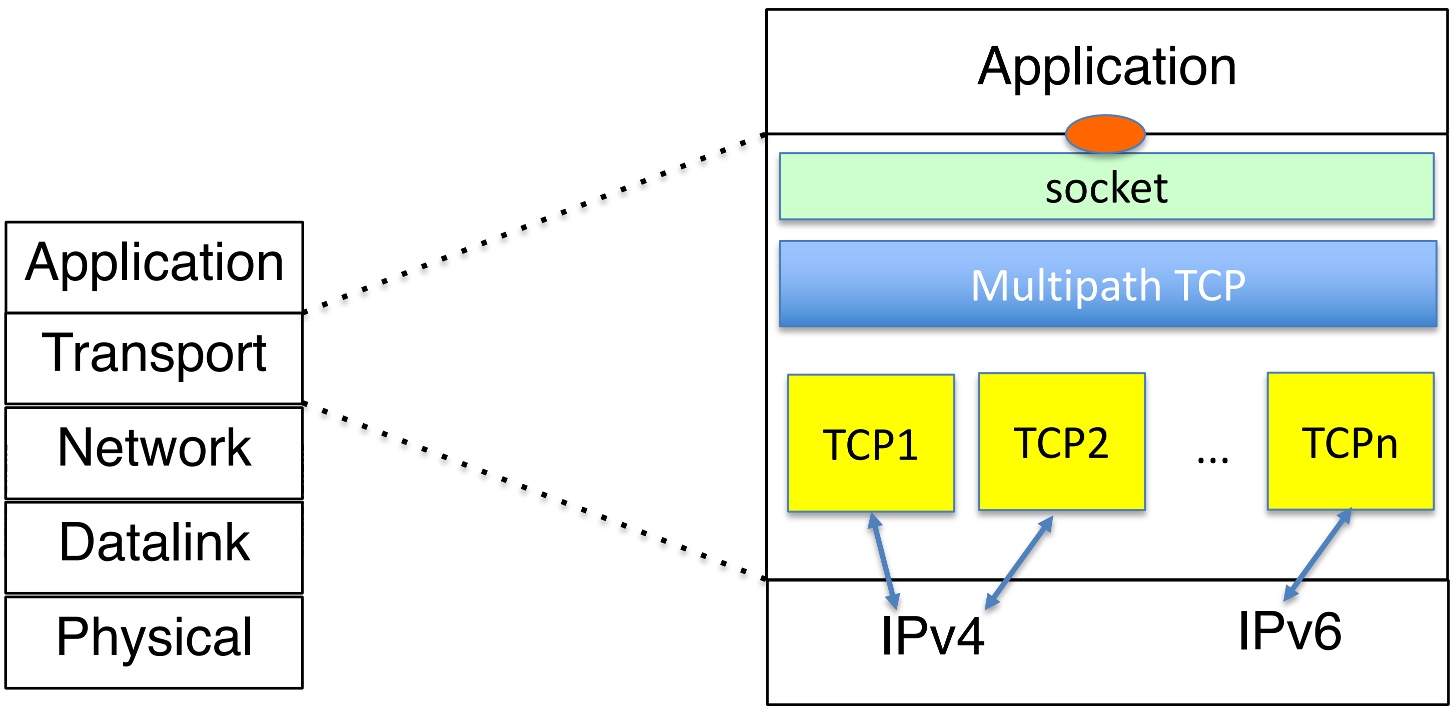


Design decision

– *A Multipath TCP connection is composed of one or more regular TCP subflows that are combined*

- Each host maintains state that glues the TCP subflows that compose a Multipath TCP connection together
- Each TCP subflow is sent over a single path and appears like a **regular TCP** connection along this path

Multipath TCP and the architecture

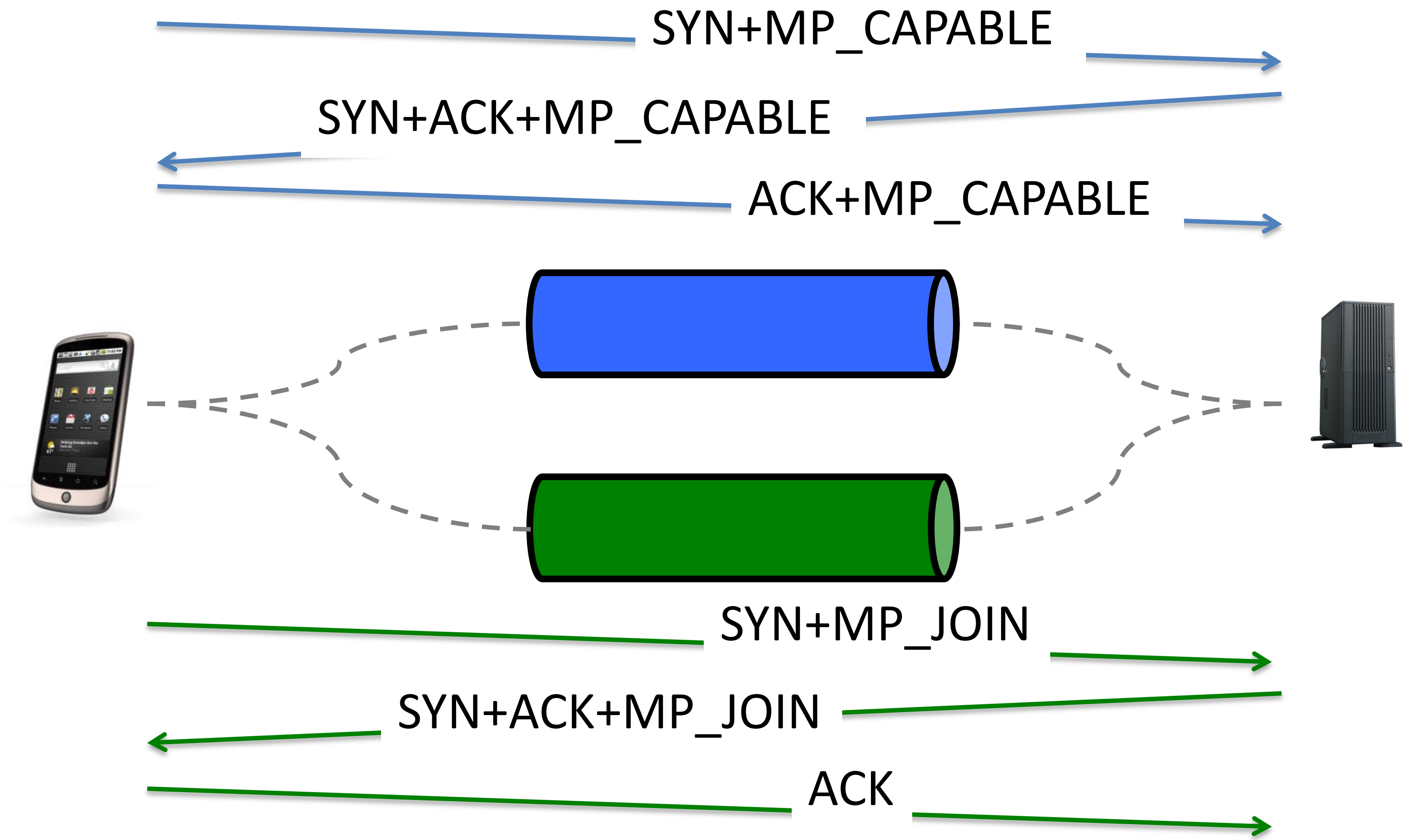


A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath TCP development", RFC6182 2011.

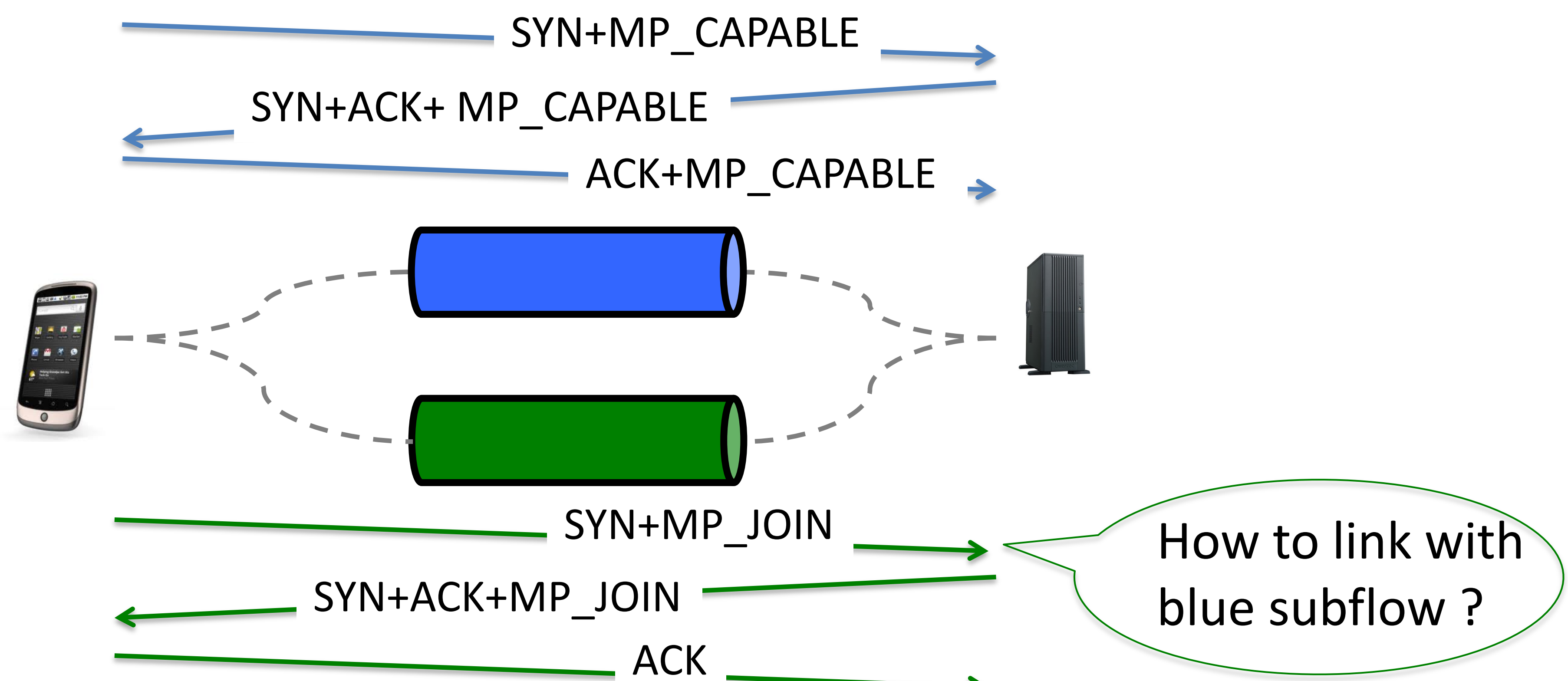
A *regular* TCP connection

- What is a *regular* TCP connection ?
  - It starts with a three-way handshake
    - SYN segments may contain special options
  - All data segments are sent in sequence
    - There is no gap in the sequence numbers
  - It is terminated by using FIN or RST

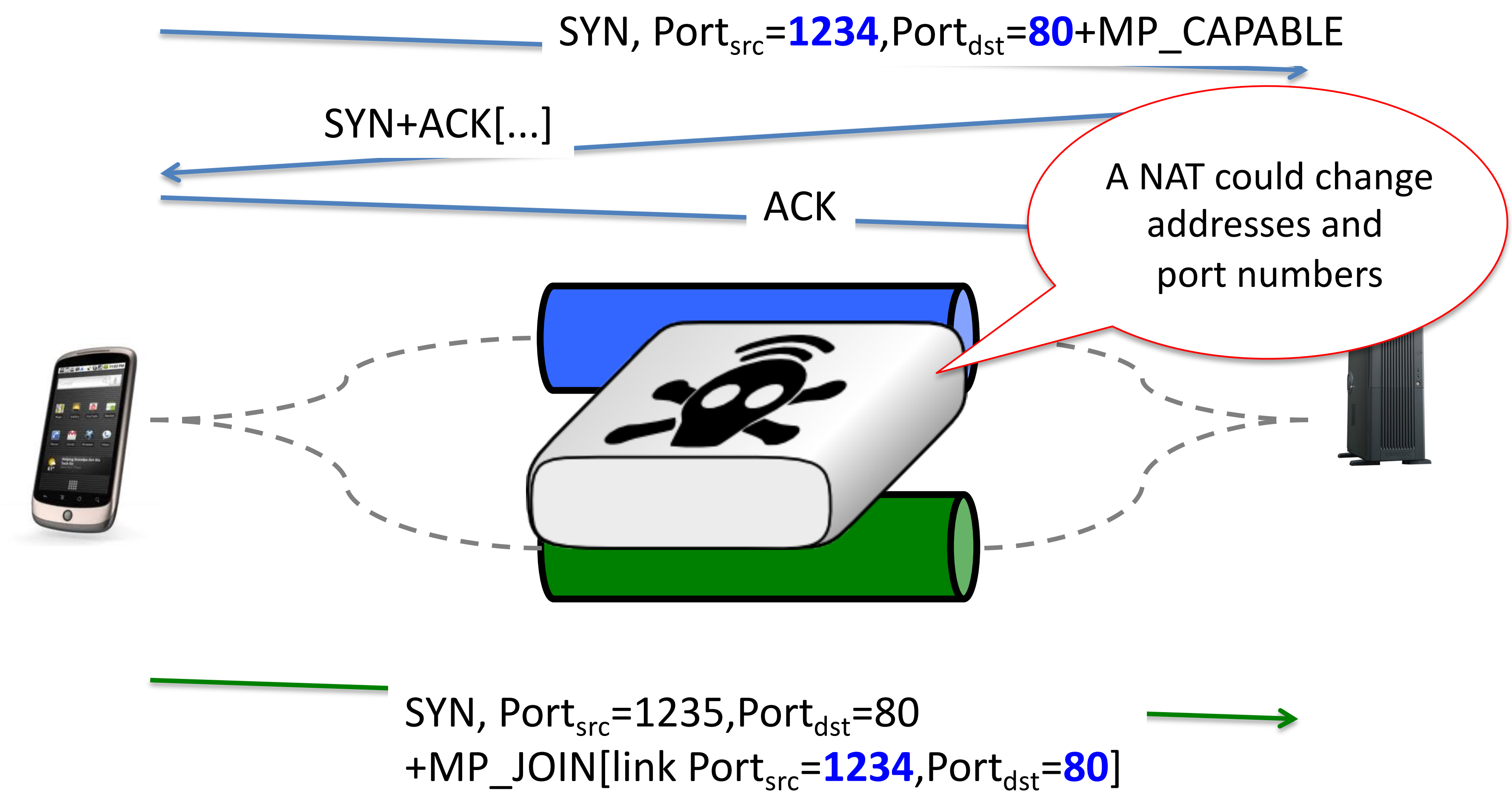
Multipath TCP



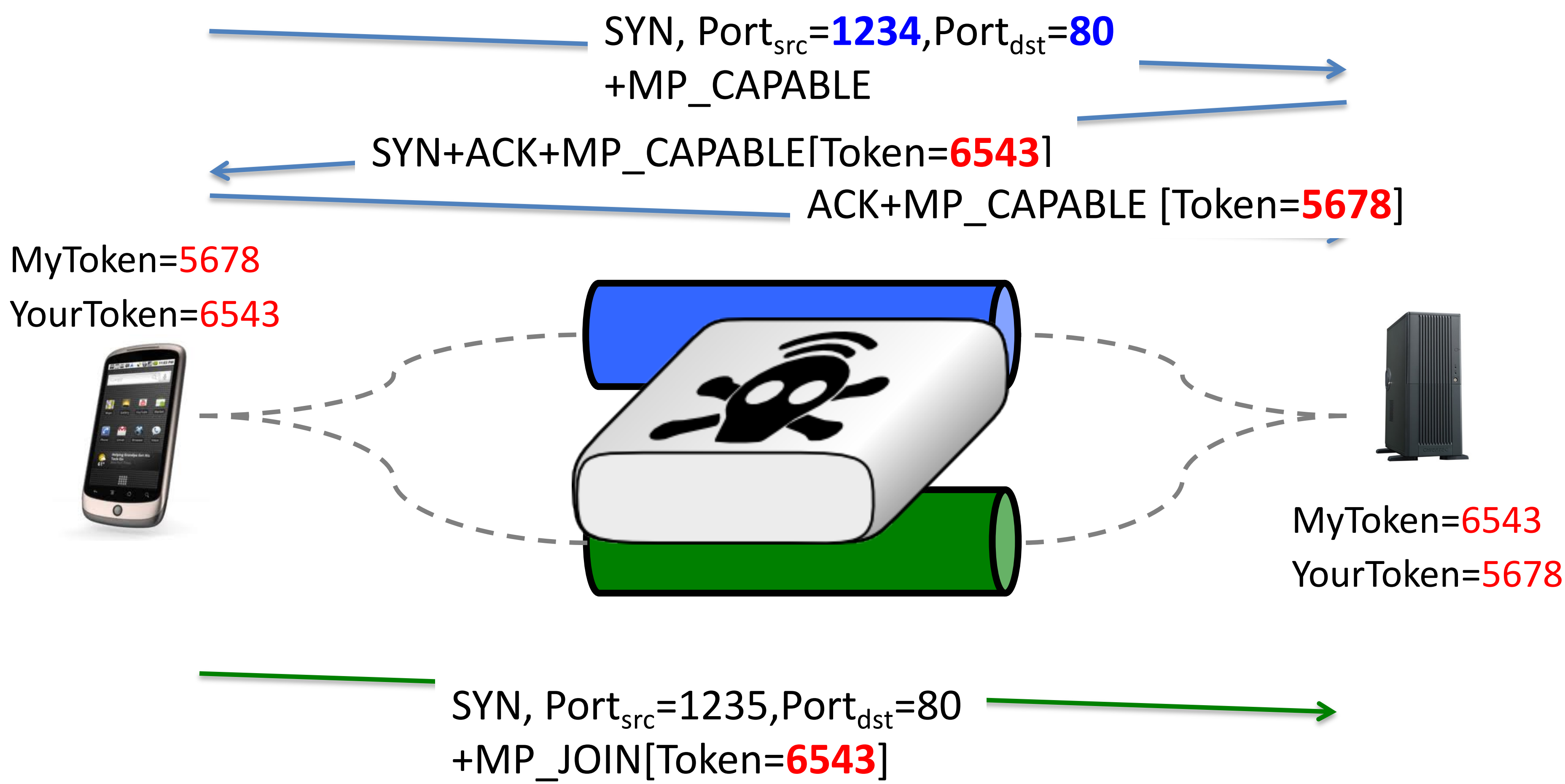
How to combine two TCP subflows ?



How to link TCP subflows ?



How to link TCP subflows ?

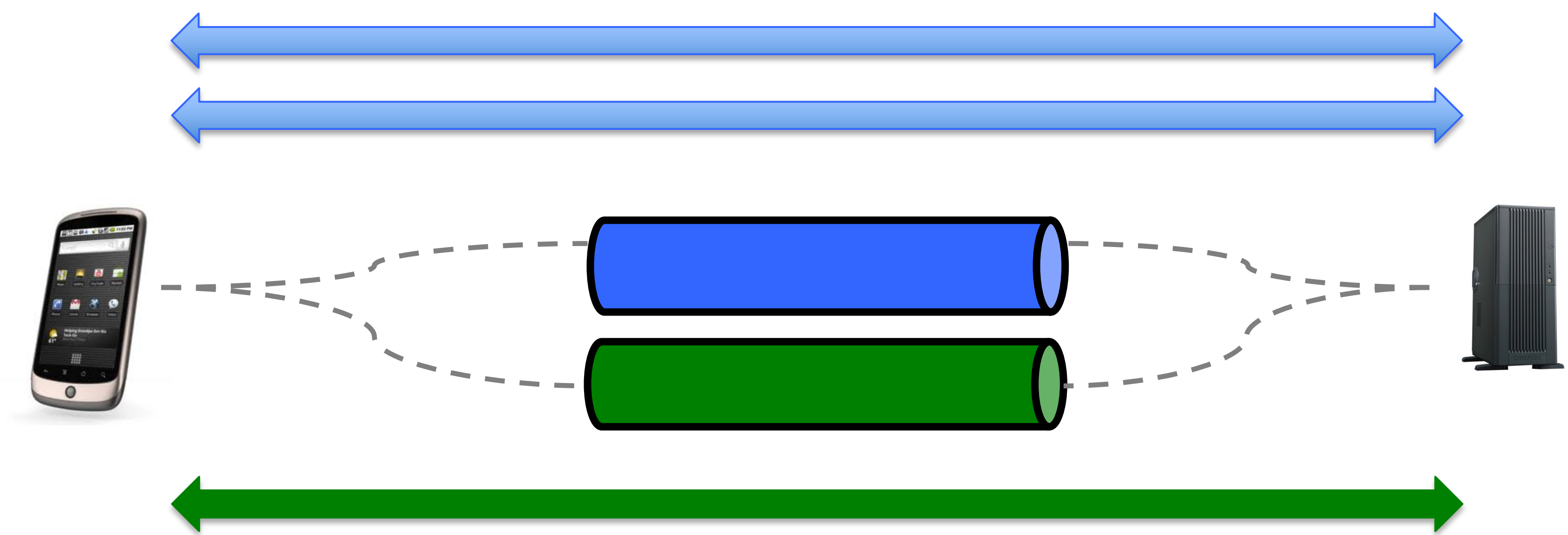


TCP subflows

- Which subflows can be associated to a Multipath TCP connection ?
  - At least one of the elements of the four-tuple needs to differ between two subflows
    - Local IP address
    - Remote IP address
    - Local port
    - Remote port

Subflow agility

- Multipath TCP supports
  - addition of subflows
  - removal of subflows

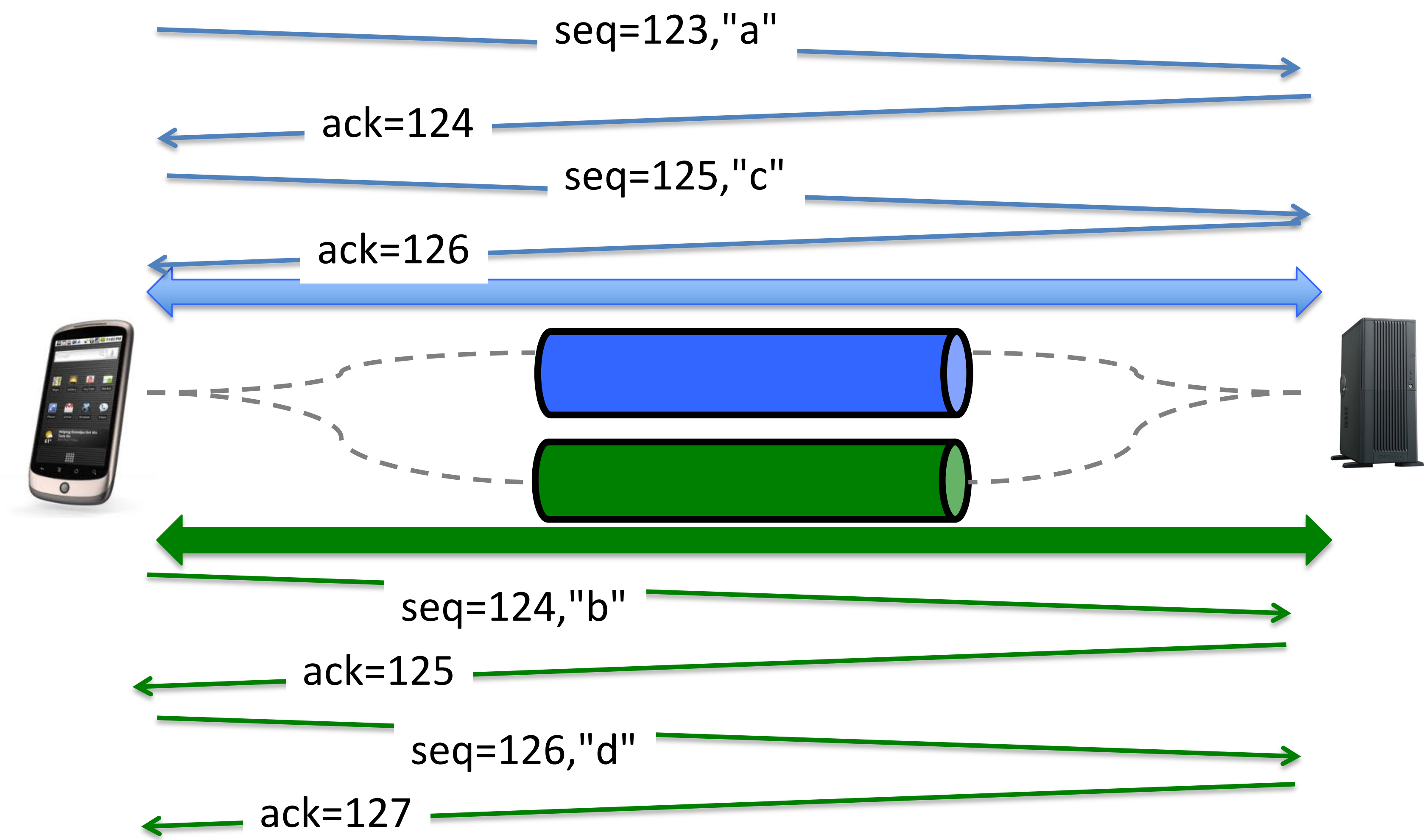


The Multipath TCP protocol

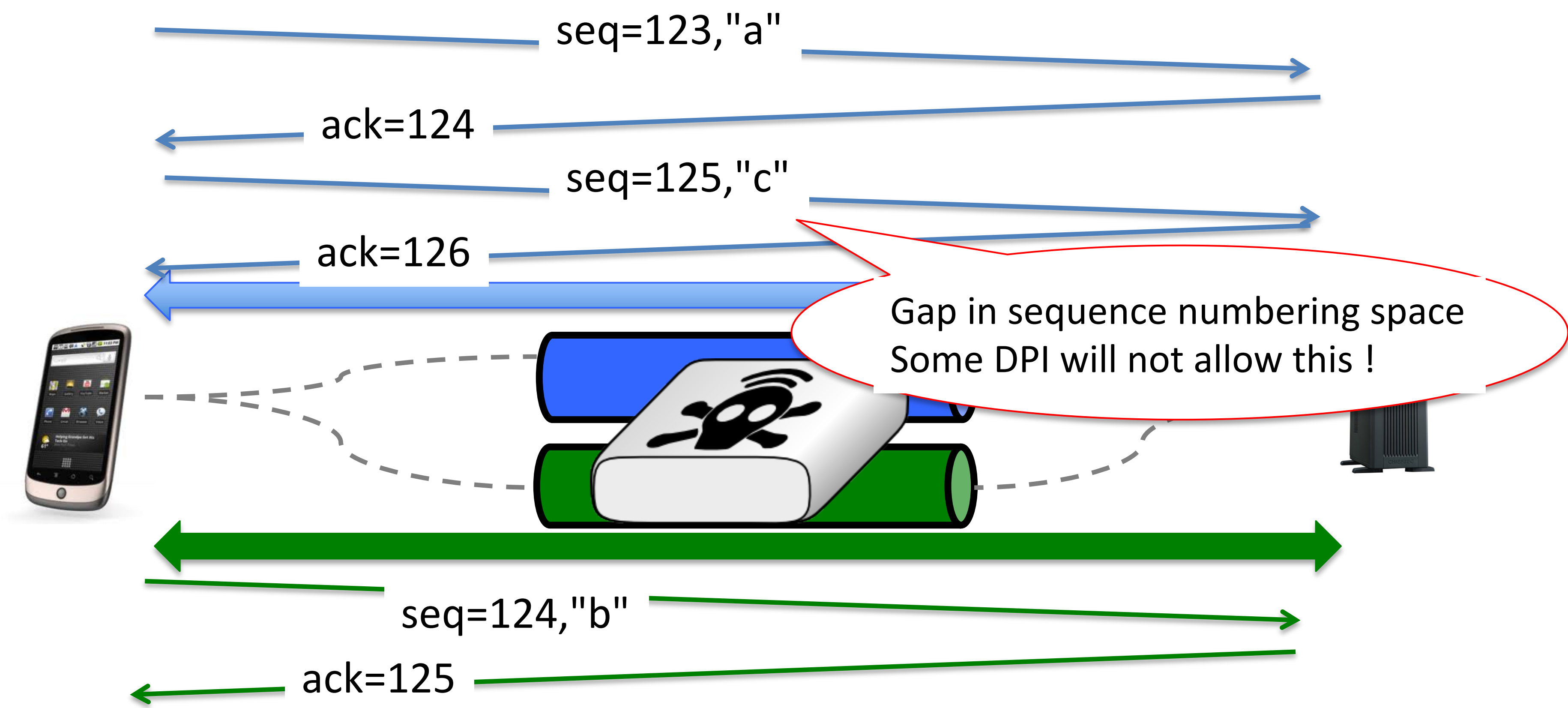
- Control plane
  - How to manage a Multipath TCP connection that uses several paths ?
- **Data plane**
  - How to transport data ?

- Congestion control
  - How to control congestion over multiple paths ?

How to transfer data ?

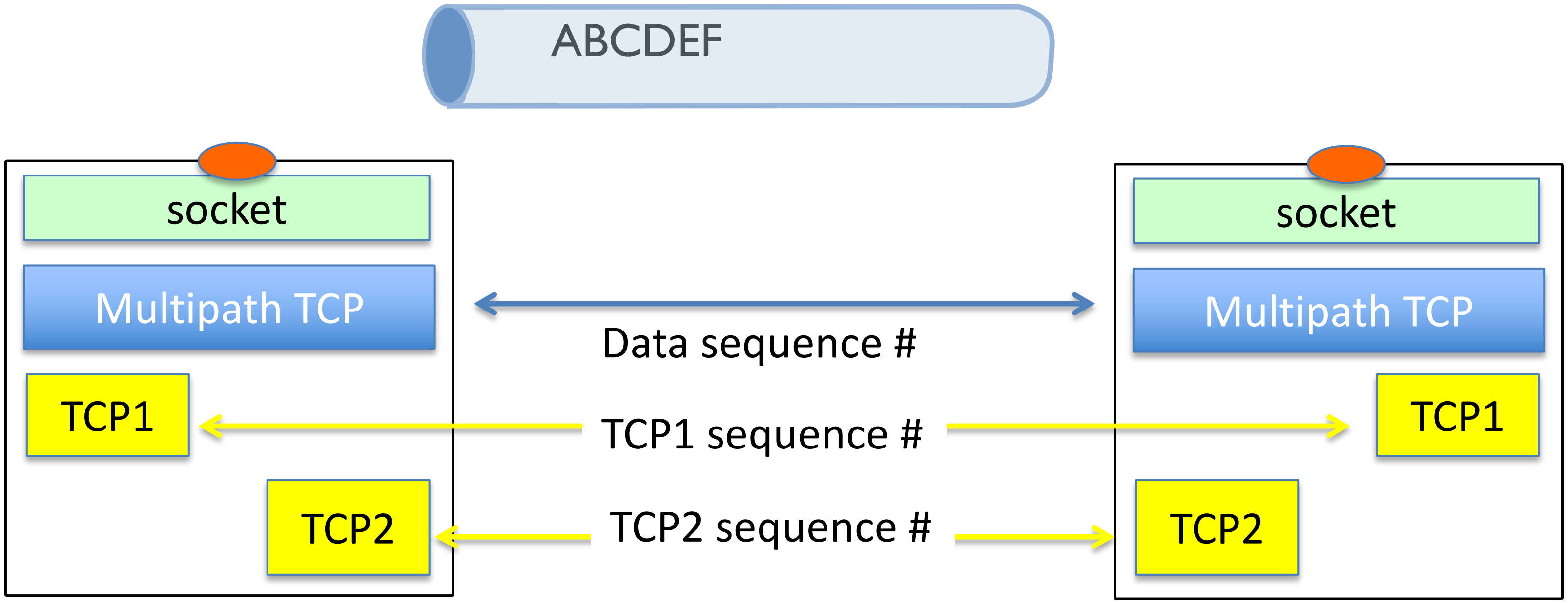


How to transfer data in today's Internet ?

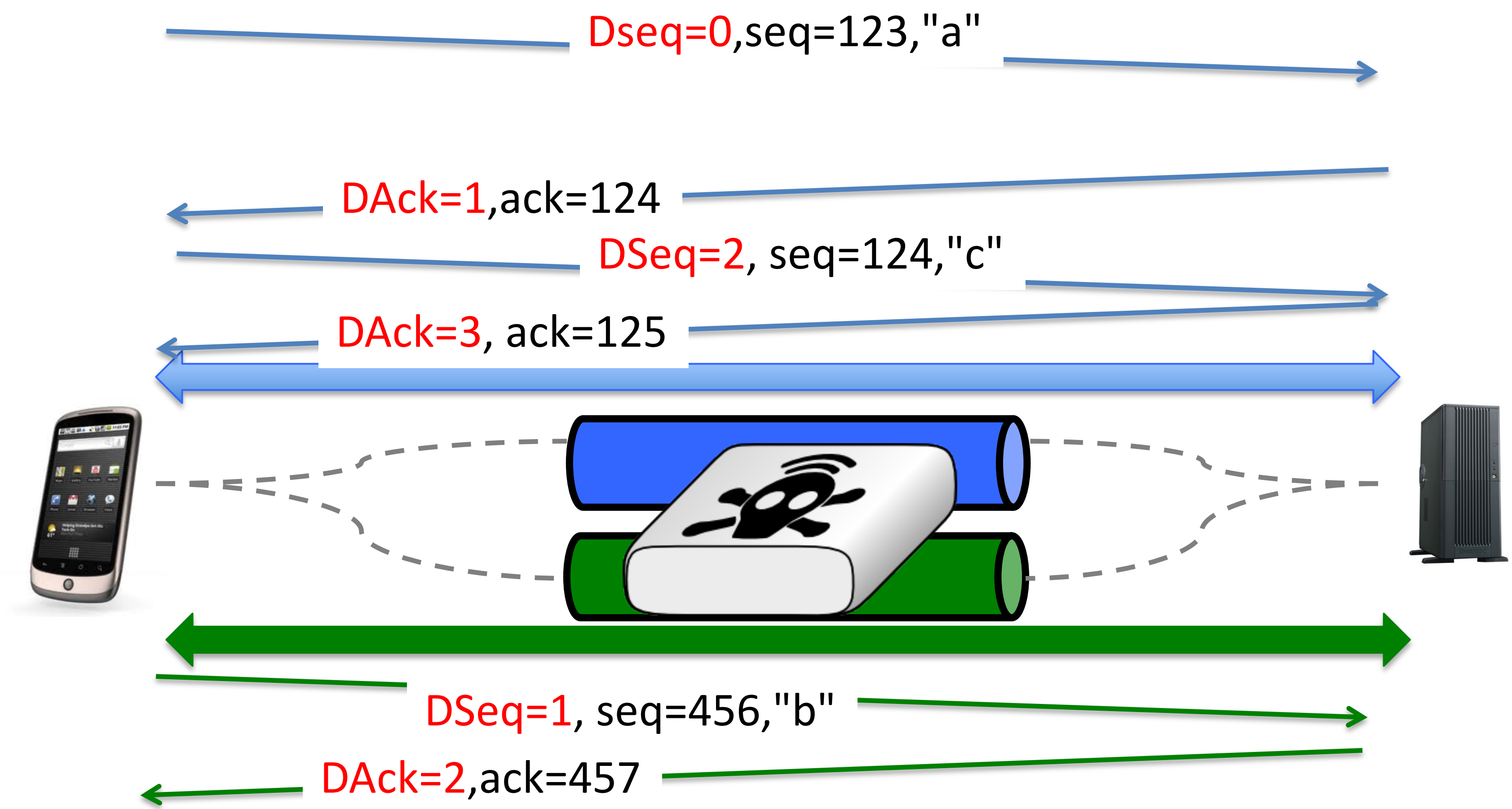


Multipath TCP Data transfer

- Two levels of sequence numbers

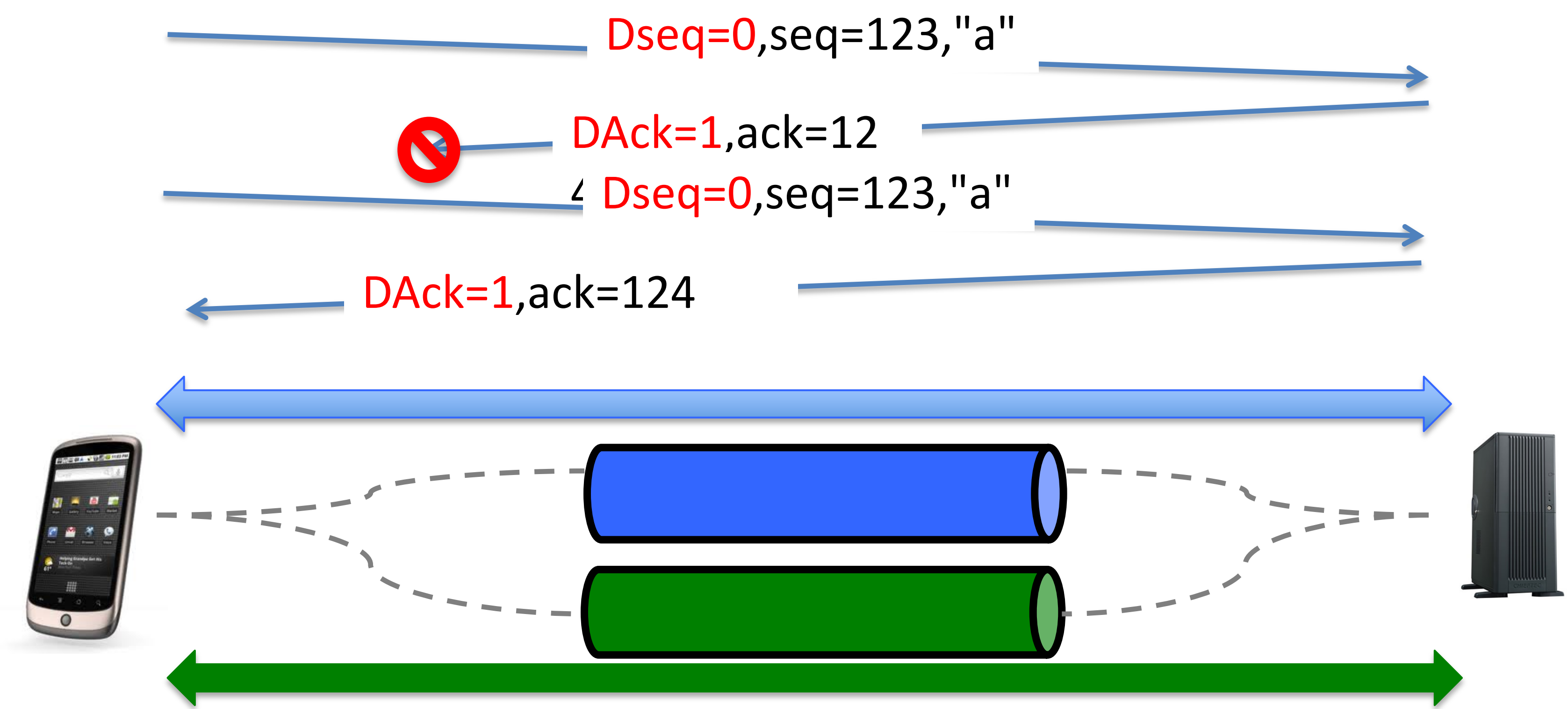


Multipath TCP
Data transfer



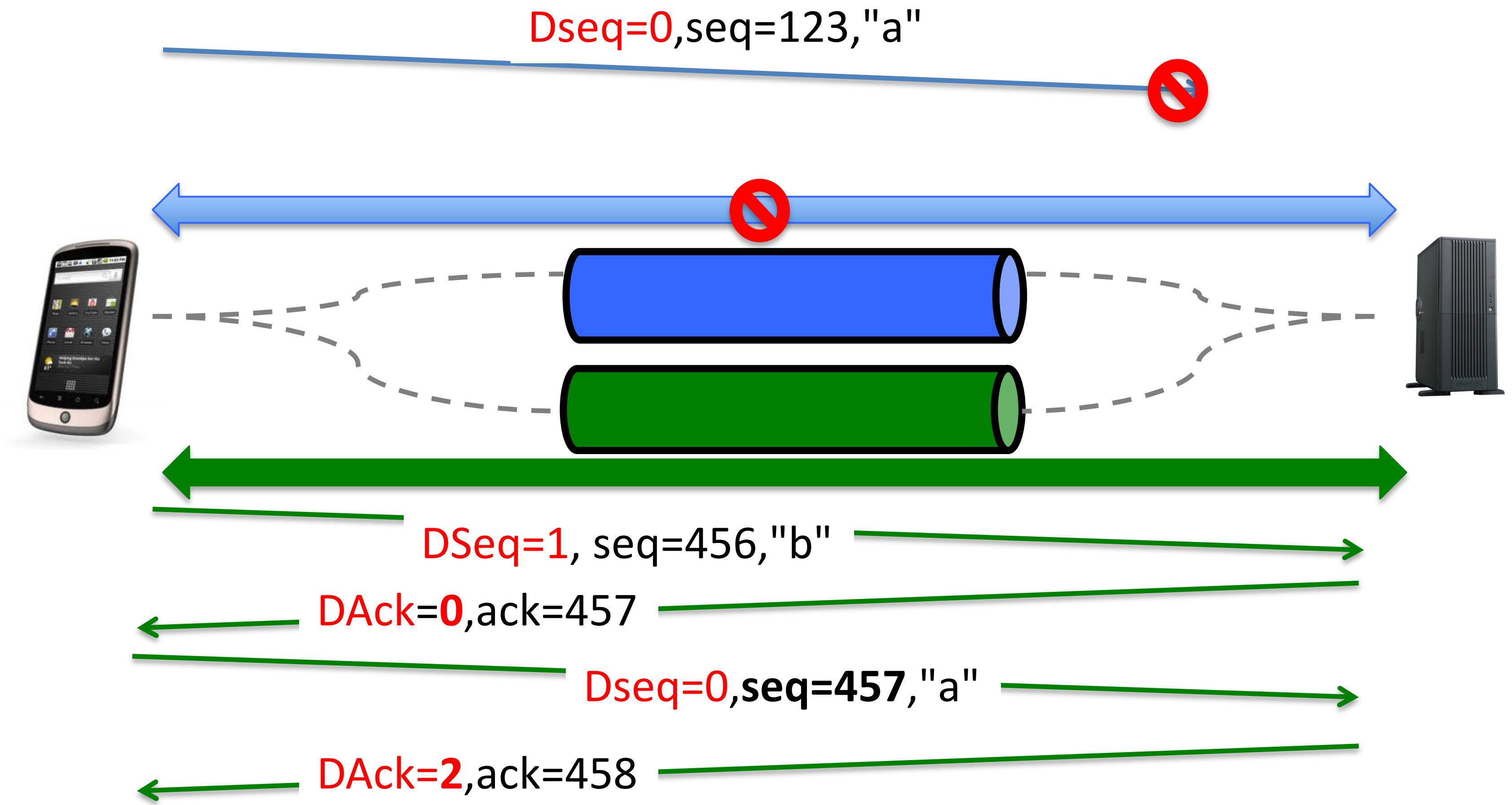
Multipath TCP
How to deal with losses ?

- Data losses over one TCP subflow
  - Fast retransmit and timeout as in regular TCP



Multipath TCP

- What happens when a TCP subflow fails ?



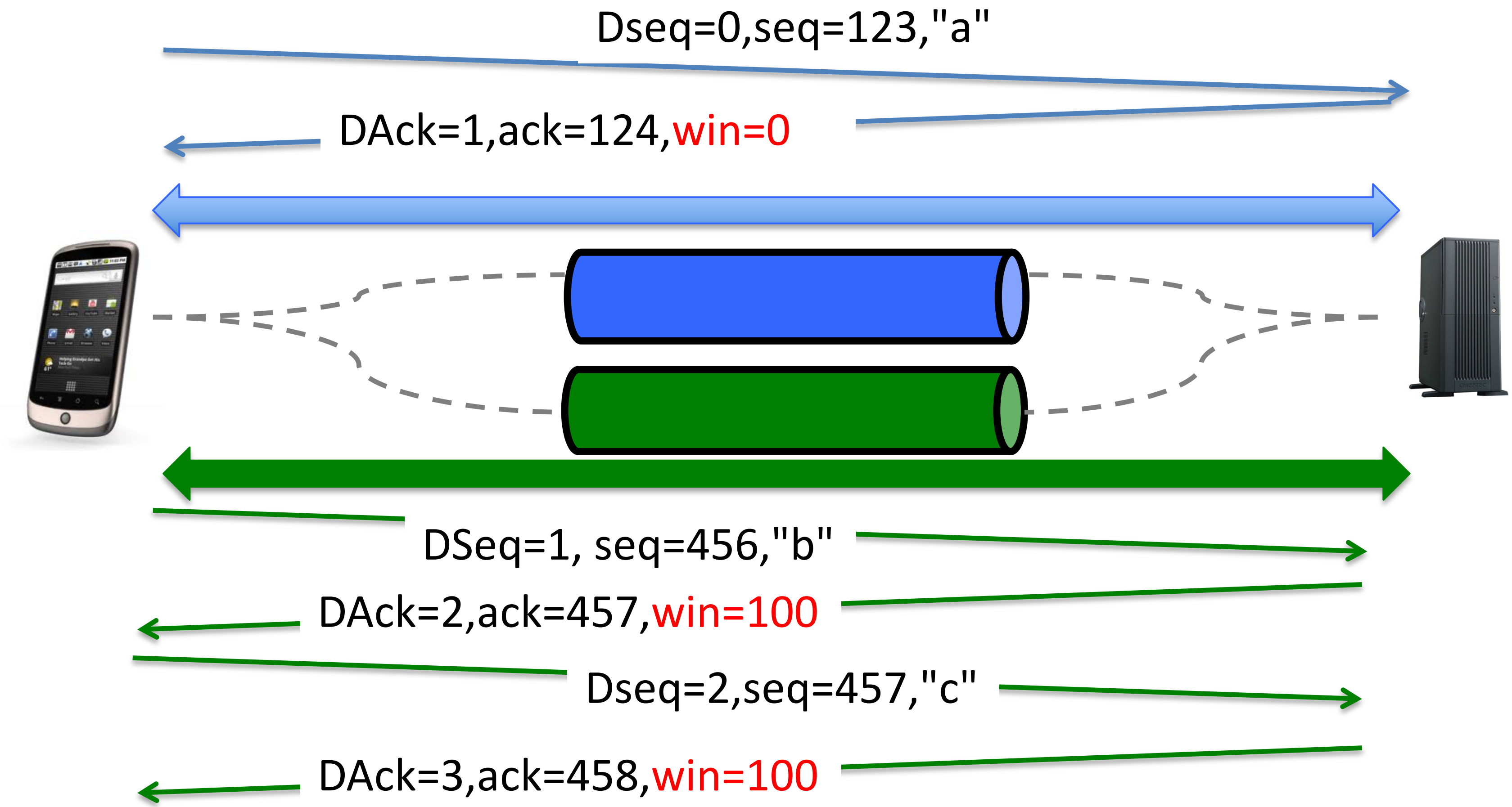
Retransmission heuristics

- Heuristics used by current Linux implementation
  - Fast retransmit is performed on the same subflow as the original transmission
  - Upon timeout expiration, reevaluate whether the segment could be retransmitted over another subflow
  - Upon loss of a subflow, all the unacknowledged data are retransmitted on other subflows

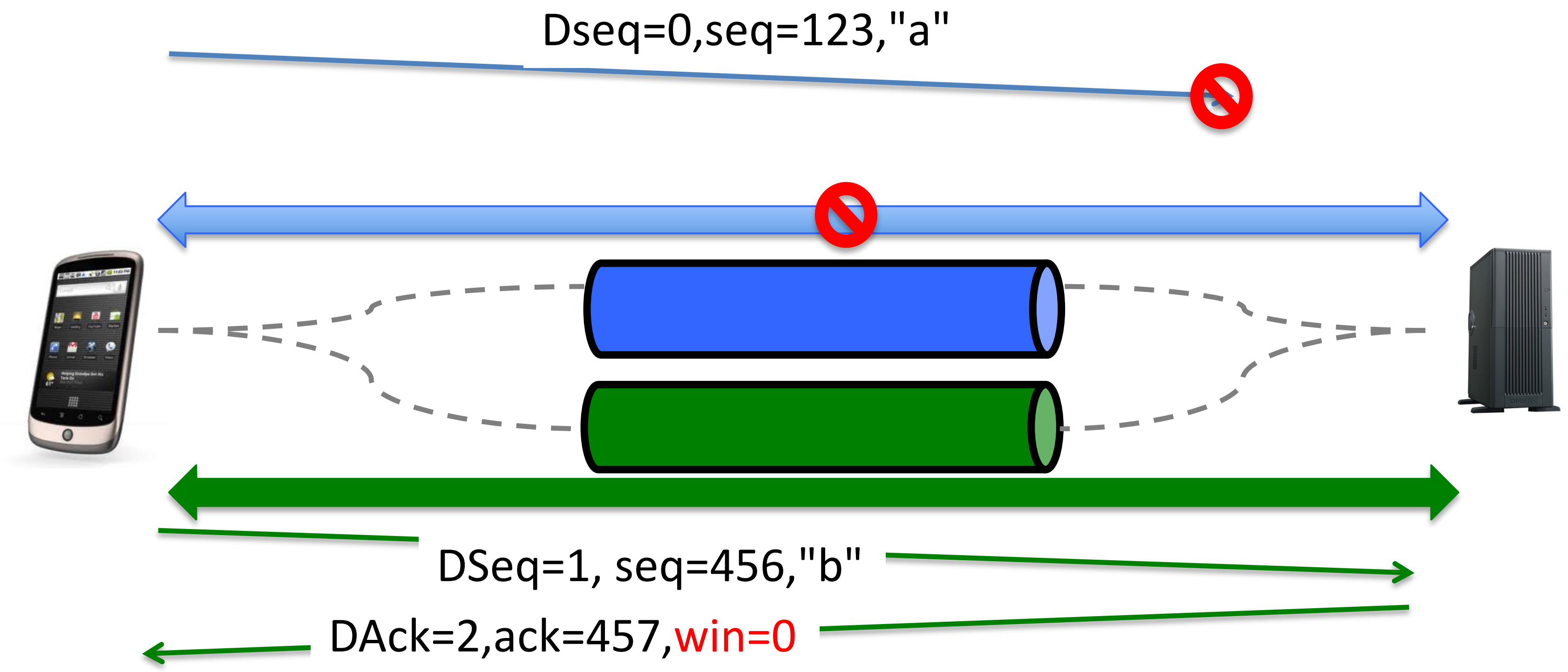
Flow control

- How should the window-based flow control be performed ?
  - Independant windows on each TCP subflow
  - A single window that is shared among all TCP subflows

Independant windows

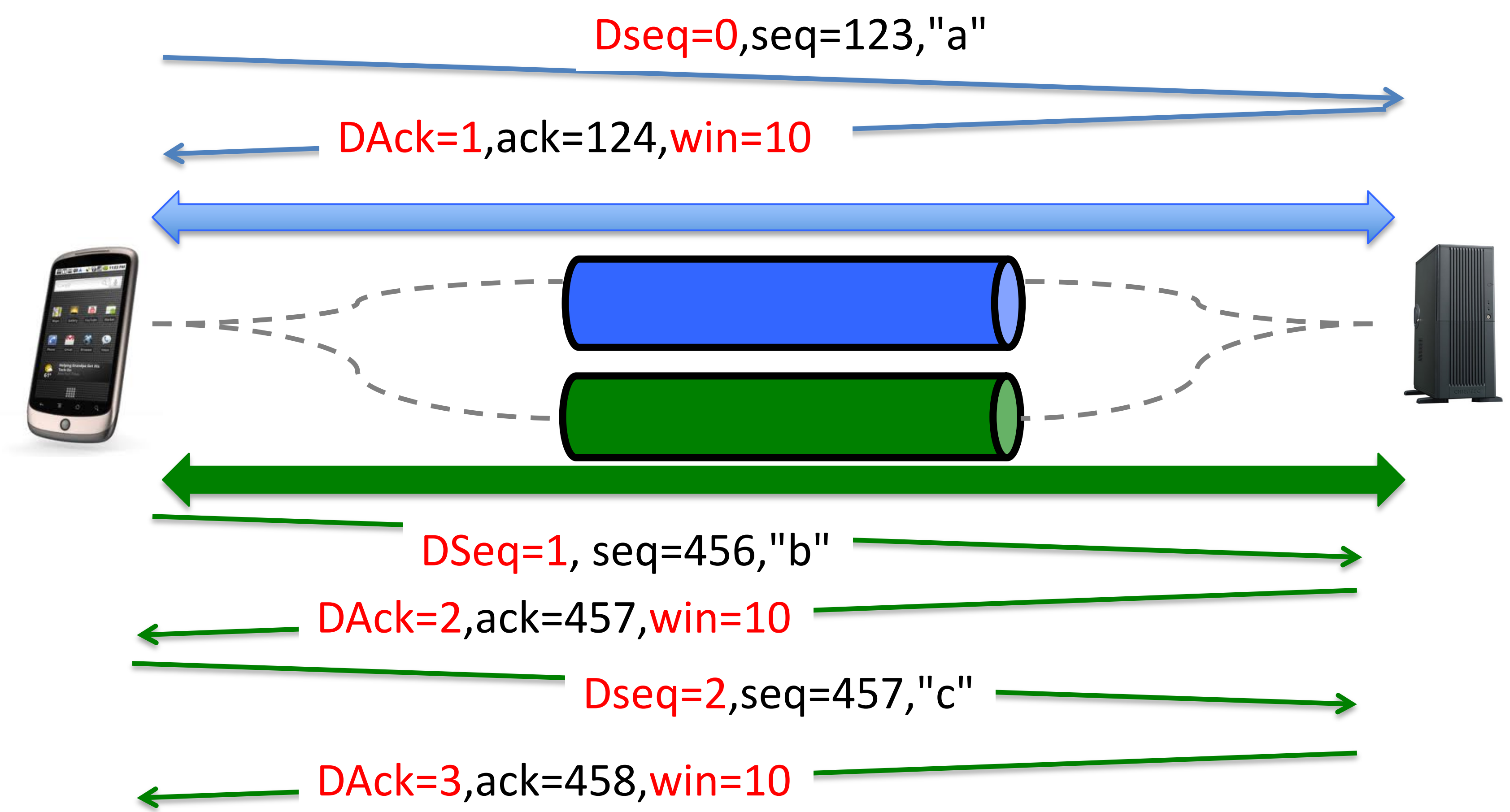


Independant windows possible problem

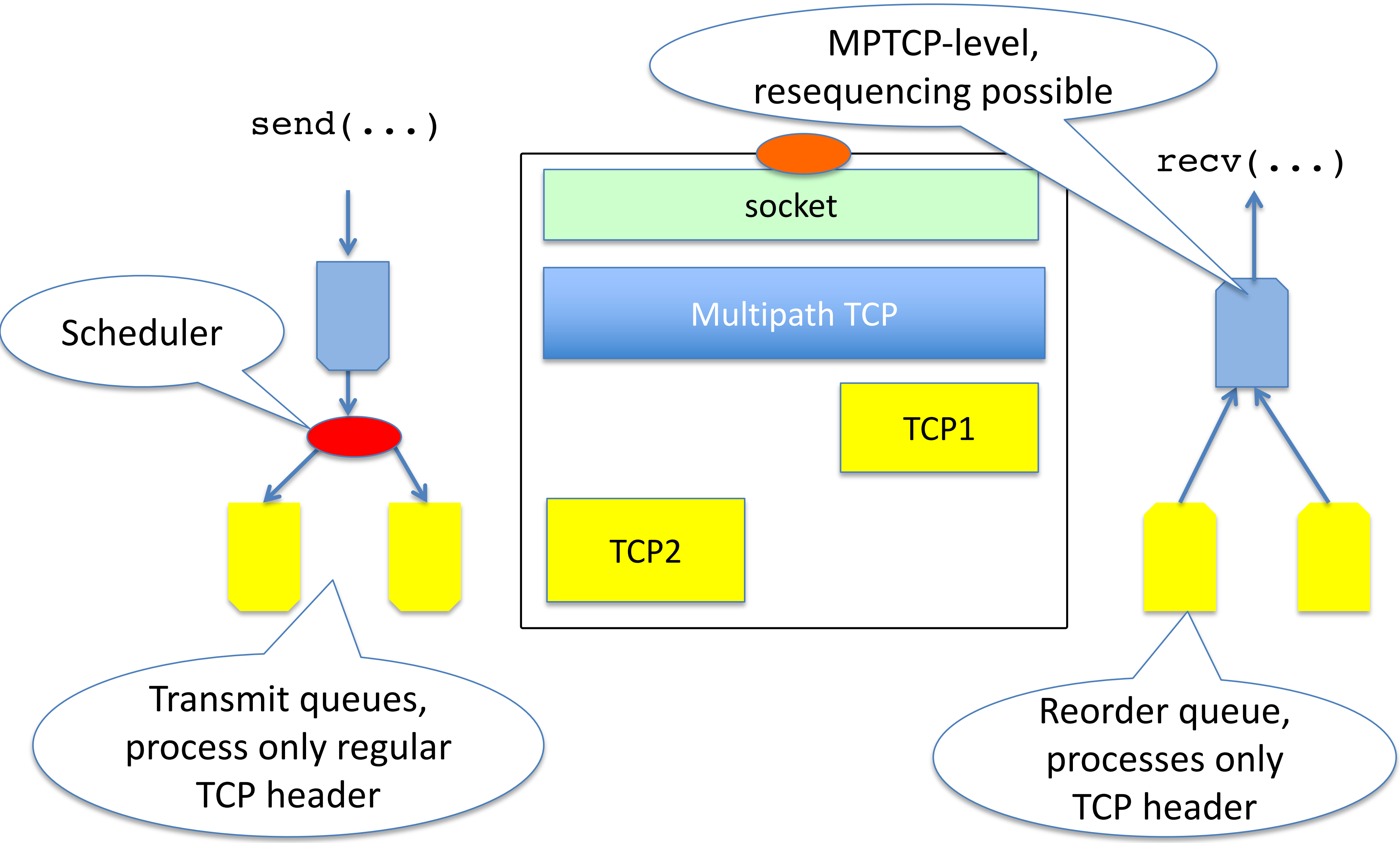


- Impossible to retransmit, window is already full on green subflow

A single window shared by all subflows



Multipath TCP buffers

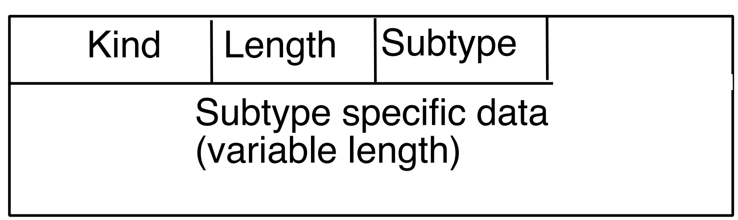


Other types of middlebox interference

- Your favorite high-speed NIC
  - GRO: Consecutive TCP segments can be merged
  - TSO: A large TCP segment will be split in segments
  - DSS is a mapping from DataSeq to SeqNum for a specific number of bytes
- Application Level Gateway “modifies” payload
  - Adding/Removing bytes is a problem
  - DSS includes optional checksum to detect this and reset affected subflow or fallback TCP

Multipath TCP option

- A single option type
  - to minimise the risk of having one option accepted by middleboxes in SYN segments and rejected in segments carrying data



Data Sequence Signal option

A = Data ACK present
a = Data ACK is 8 octets
M = mapping present
m = DSN is 8

Cumulative Data ack

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------+----------------------+
|     Kind      |    Length     |Subtype| (reserved) |F|m|M|a|A|
+---------------+---------------+-------+----------------------+
|          Data ACK (4 or 8 octets, depending on flags)        |
+--------------------------------------------------------------+
|   Data Sequence Number (4 or 8 octets, depending on flags)   |
+--------------------------------------------------------------+
|              Subflow Sequence Number (4 octets)              |
+-------------------------------+------------------------------+
|  Data-level Length (2 octets) |      Checksum (2 octets)     |
+-------------------------------+------------------------------+
```

Length of mapping, can extend beyond this segment

Computed over data covered by entire mapping + pseudo header

The Multipath TCP protocol

- Control plane
  - How to manage a Multipath TCP connection that uses several paths ?
- Data plane
  - How to transport data ?

Congestion control

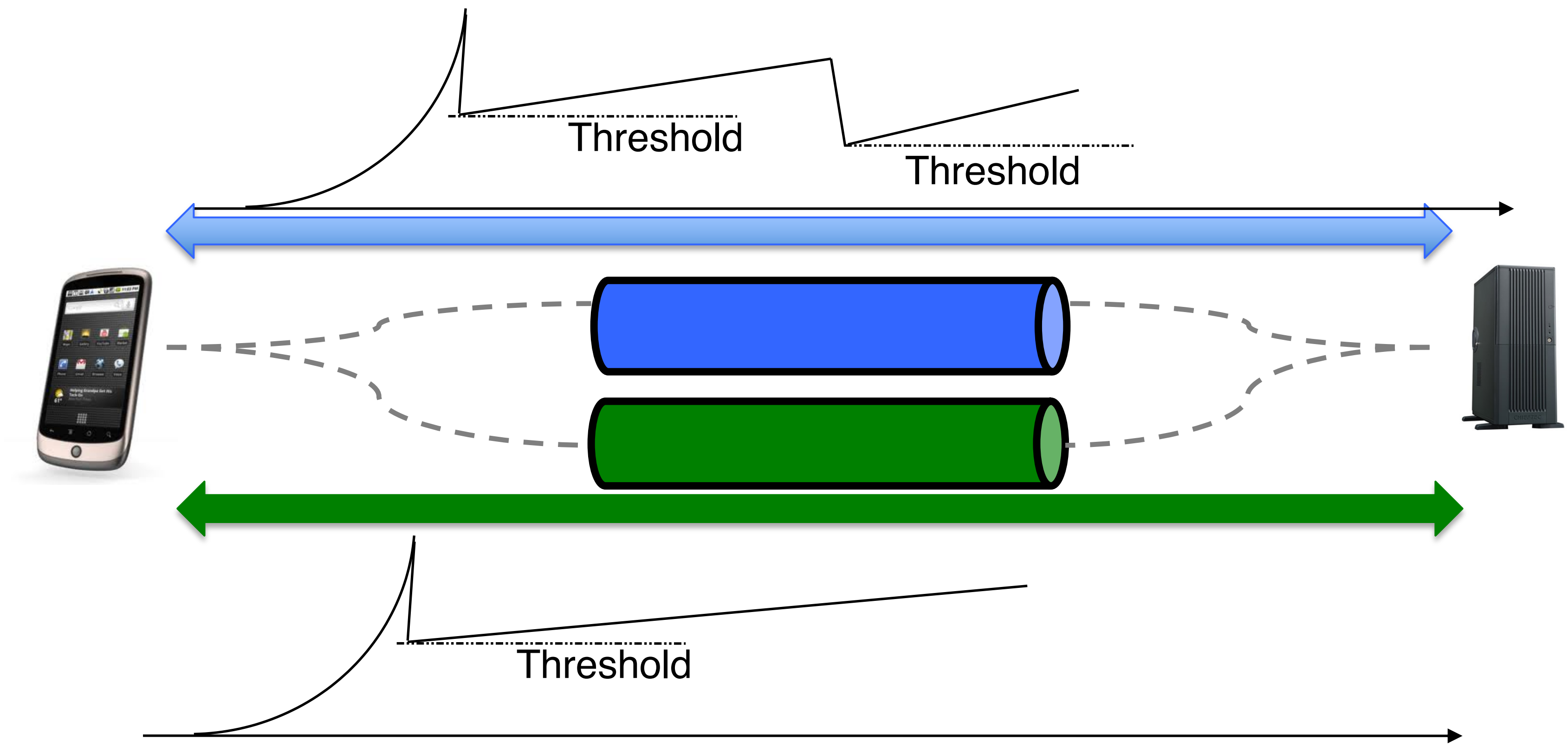
  - How to control congestion over multiple paths ?

AIMD in TCP

- Congestion control mechanism
  - Each host maintains a *congestion window (cwnd)*
  - No congestion
    - Congestion avoidance (**additive increase**)
      - increase *cwnd* by one segment every round-trip-time
  - Congestion
    - TCP detects congestion by detecting losses
    - Mild congestion (fast retransmit – **multiplicative decrease**)
      - *cwnd*=*cwnd*/2 and restart congestion avoidance
    - Severe congestion (timeout)
      - *cwnd*=1, set slow-start-threshold and restart slow-start

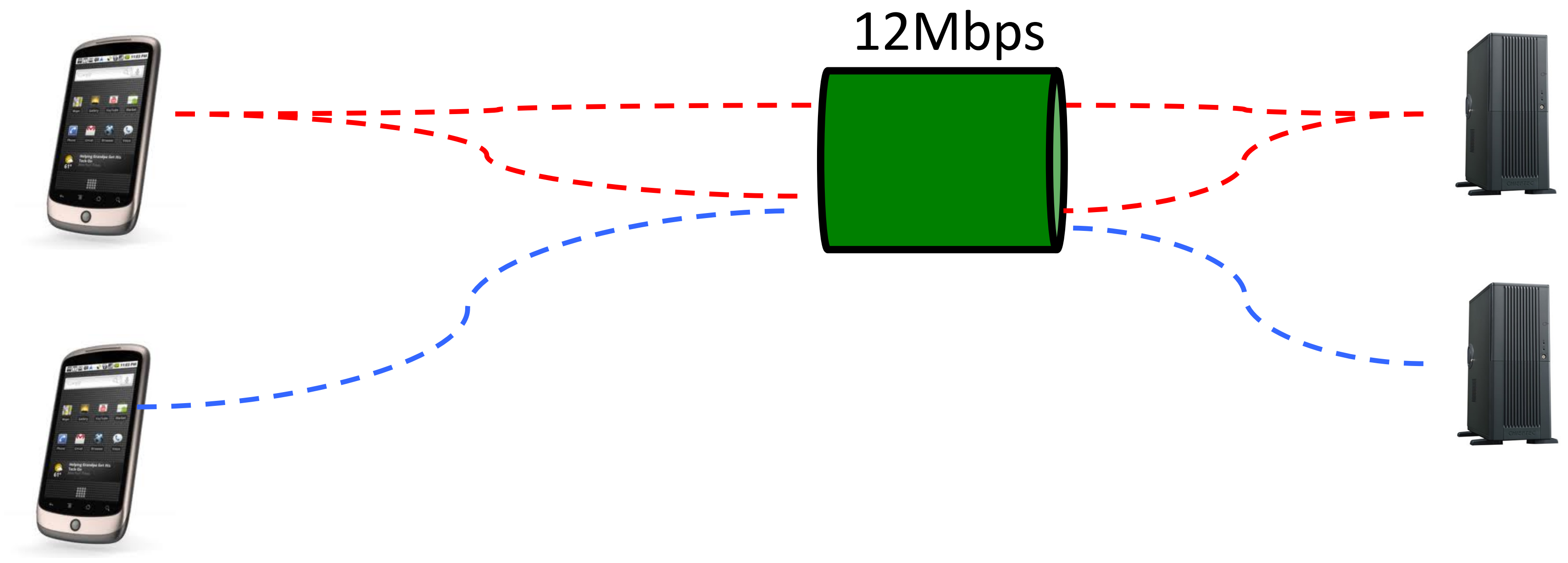
Congestion control for Multipath TCP

- Simple approach
  - independant congestion windows



Independent congestion windows

- Problem



Coupled congestion control

- Congestion windows are coupled
  - congestion window growth cannot be faster than TCP with a single flow
  - Coupled congestion control aims at **moving traffic away from congested path**

Linked increases congestion control

- Algorithm
  - Multiplicative Decrease
    - For each loss on path r, $cwin_r = cwin_r/2$
  - Additive increase

$$cwin_r = cwin_r + \min\left(\frac{\max\left(\frac{cwnd_i}{(rtt_i)^2}\right)}{\left(\sum_i \frac{cwnd_i}{rtt_i}\right)^2}, \frac{1}{cwnd_r}\right)$$

D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," NSDI'11: Proceedings of the 8th USENIX conference on Networked systems design and implementation, 2011.

Other Multipath-aware congestion control schemes

R. Khalili, N. Gast, M. Popovic, U. Upadhyay, J.-Y. Le Boudec , MPTCP is not Pareto-optimal: Performance issues and a possible solution, Proc. ACM Conext 2012

Y. Cao, X. Mingwei, and X. Fu, “Delay-based Congestion Control for Multipath TCP,” ICNP2012, 2012.

T. A. Le, C. S. Hong, and E.-N. Huh, “Coordinated TCP Westwood congestion control for multiple paths over wireless networks,” ICOIN '12: Proceedings of the The International Conference on Information Network 2012, 2012, pp. 92–96.

T. A. Le, H. Rim, and C. S. Hong, “A Multipath Cubic TCP Congestion Control with Multipath Fast Recovery over High Bandwidth-Delay Product Networks,” *IEICE Transactions*, 2012.

T. Dreibholz, M. Becke, J. Pulinthanath, and E. P. Rathgeb, “Applying TCP-Friendly Congestion Control to Concurrent Multipath Transfer,” Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, 2010, pp. 312–319.

Multipath TCP packet schedulers

- When several subflows are active, the packet scheduler selects the subflow where each packet is sent
  - Default scheduler is to prefer subflow having the lowest round-trip-time
    - Well adapted to NewReno like congestion control
  - Other schedulers have been proposed
    - Round Robin, Priority
    - Schedulers that schedule packets by estimating the time required to reach the receiver to avoid reodering

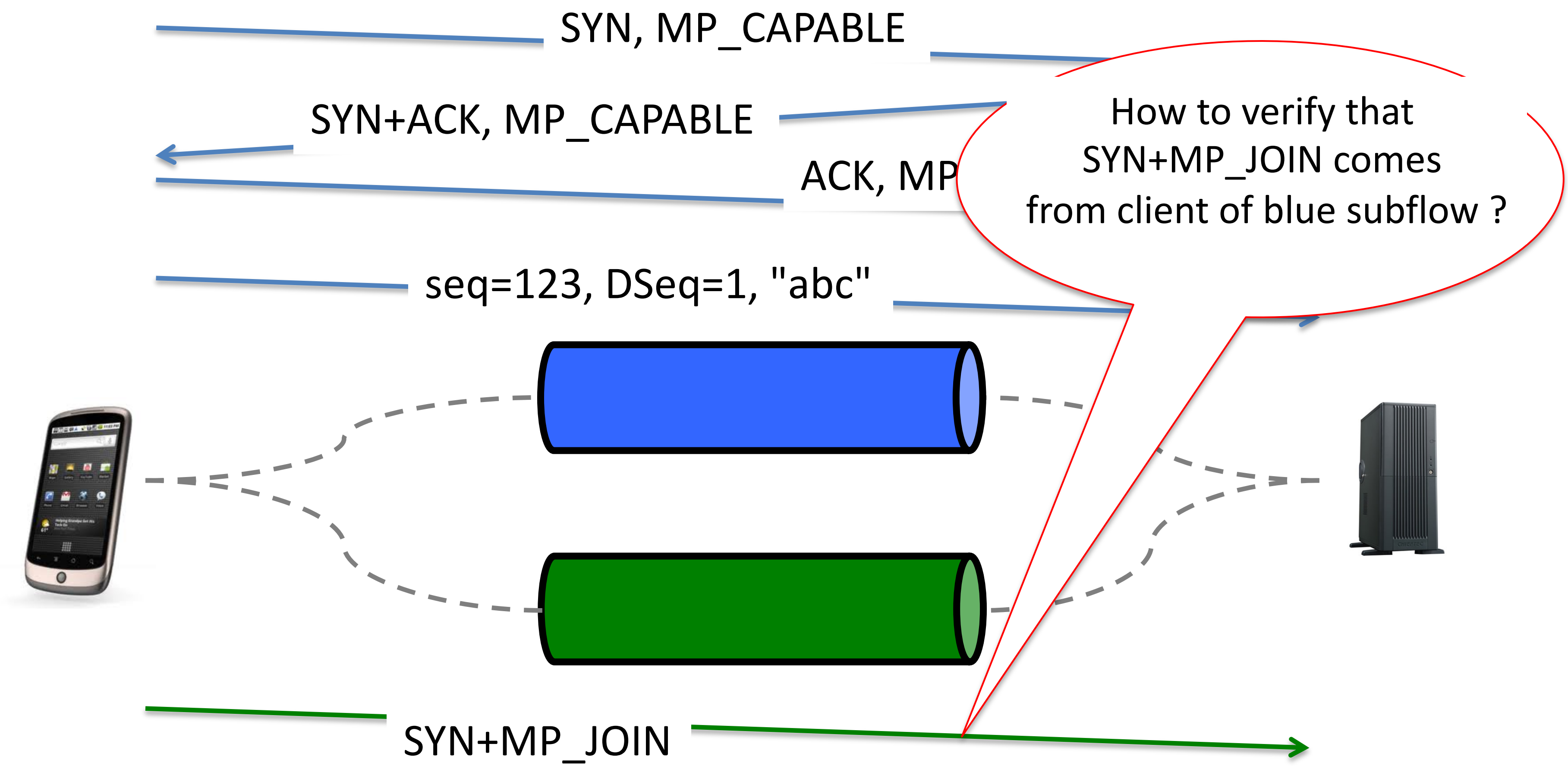
https://tools.ietf.org/html/draft-bonaventure-iccrg-schedulers

The Multipath TCP protocol

- **Control plane**
  - How to manage a Multipath TCP connection that uses several paths ?
- Data plane
  - How to transport data ?
- Congestion control
  - How to control congestion over multiple paths ?

Multipath TCP
Connection establishment

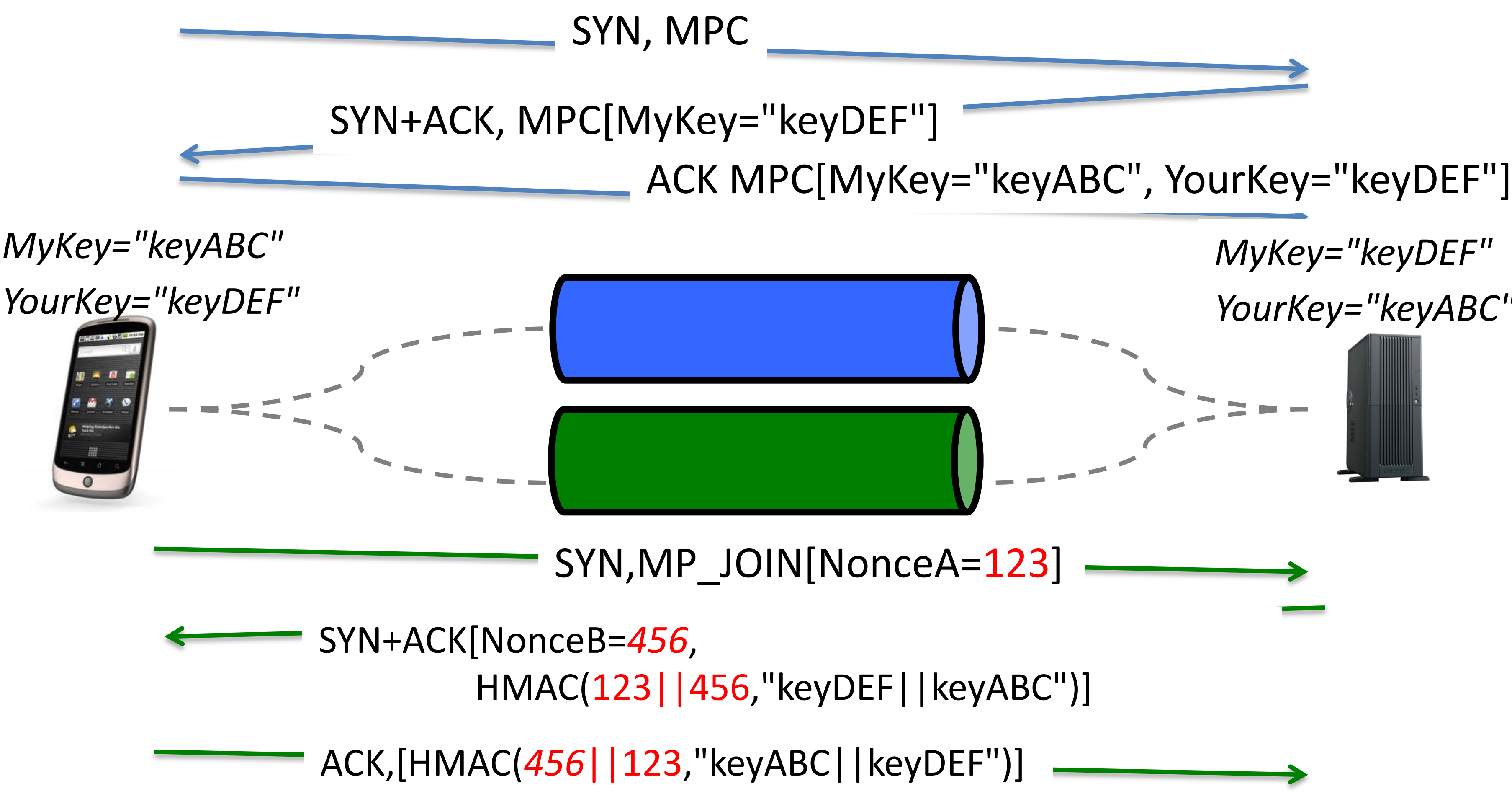
- Principle



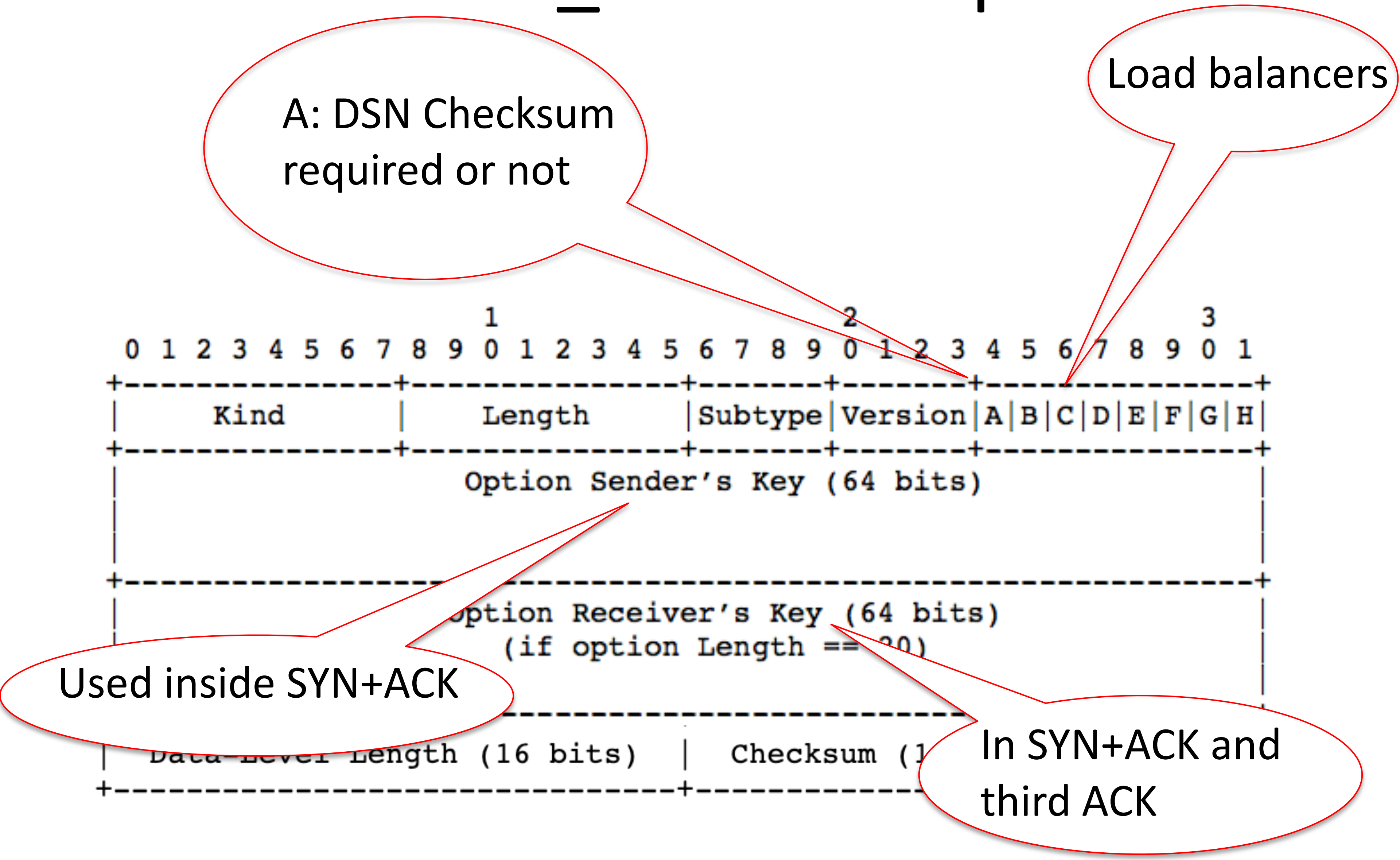
« Authenticating » Multipath TCP subflows

- Main goal
  - Prevent attacks by off-path attackers
- Principles
  - Each host announces a key during initial handshake
    - keys are exchanged in clear
  - When establishing a subflow, use HMAC + key to authenticate subflow

Key exchange



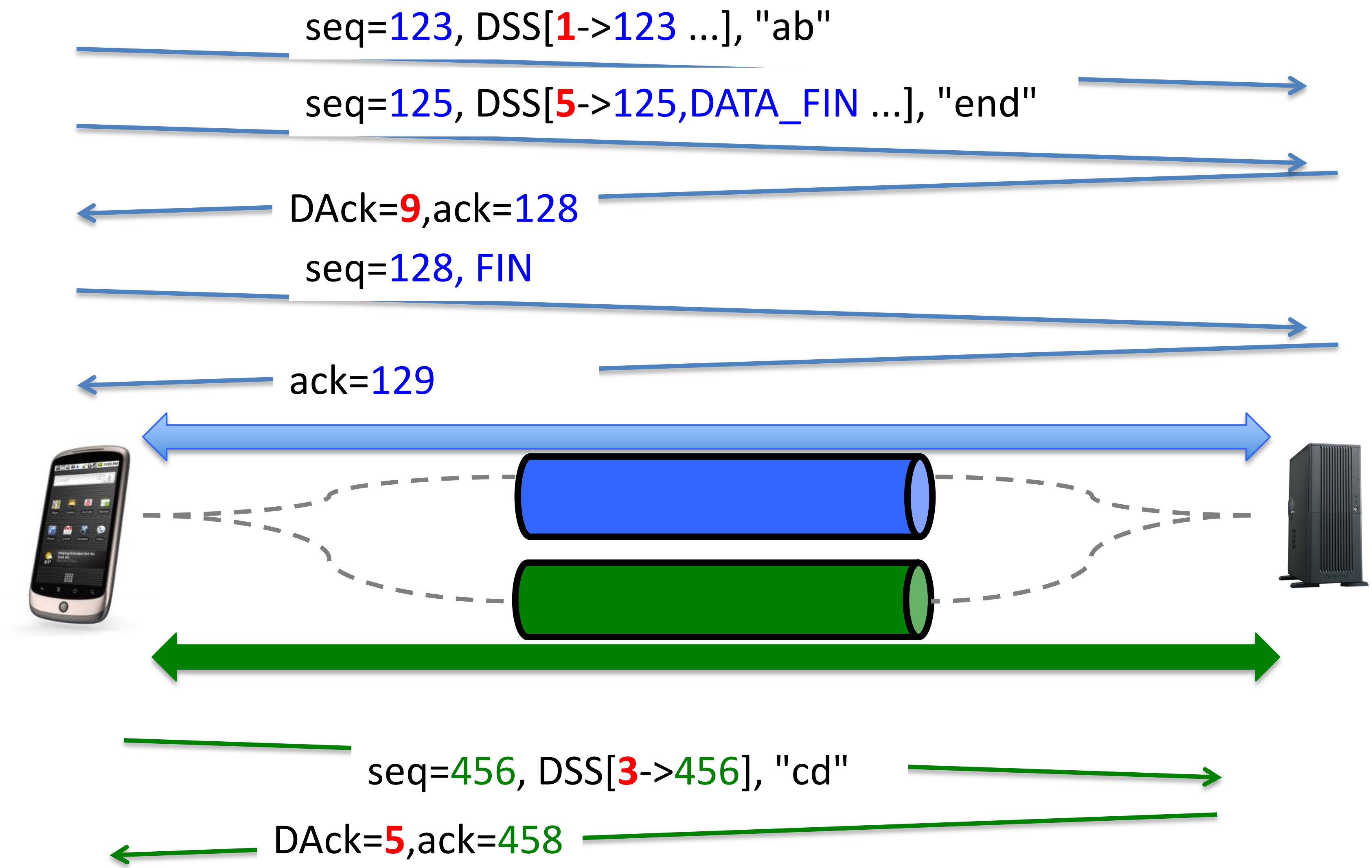
The MP_CAPABLE option



The Multipath TCP control plane

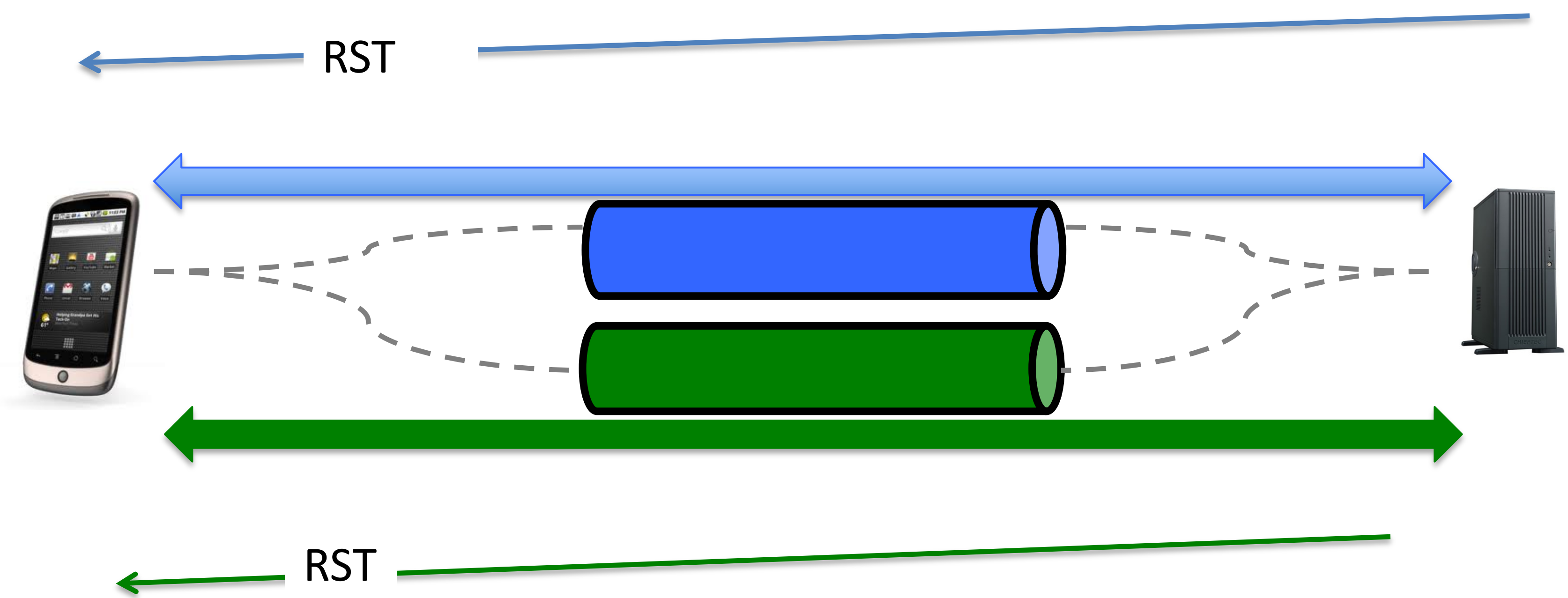
- Connection establishment in details
- Closing a Multipath TCP connection
- Address dynamics

Closing a Multipath TCP connection



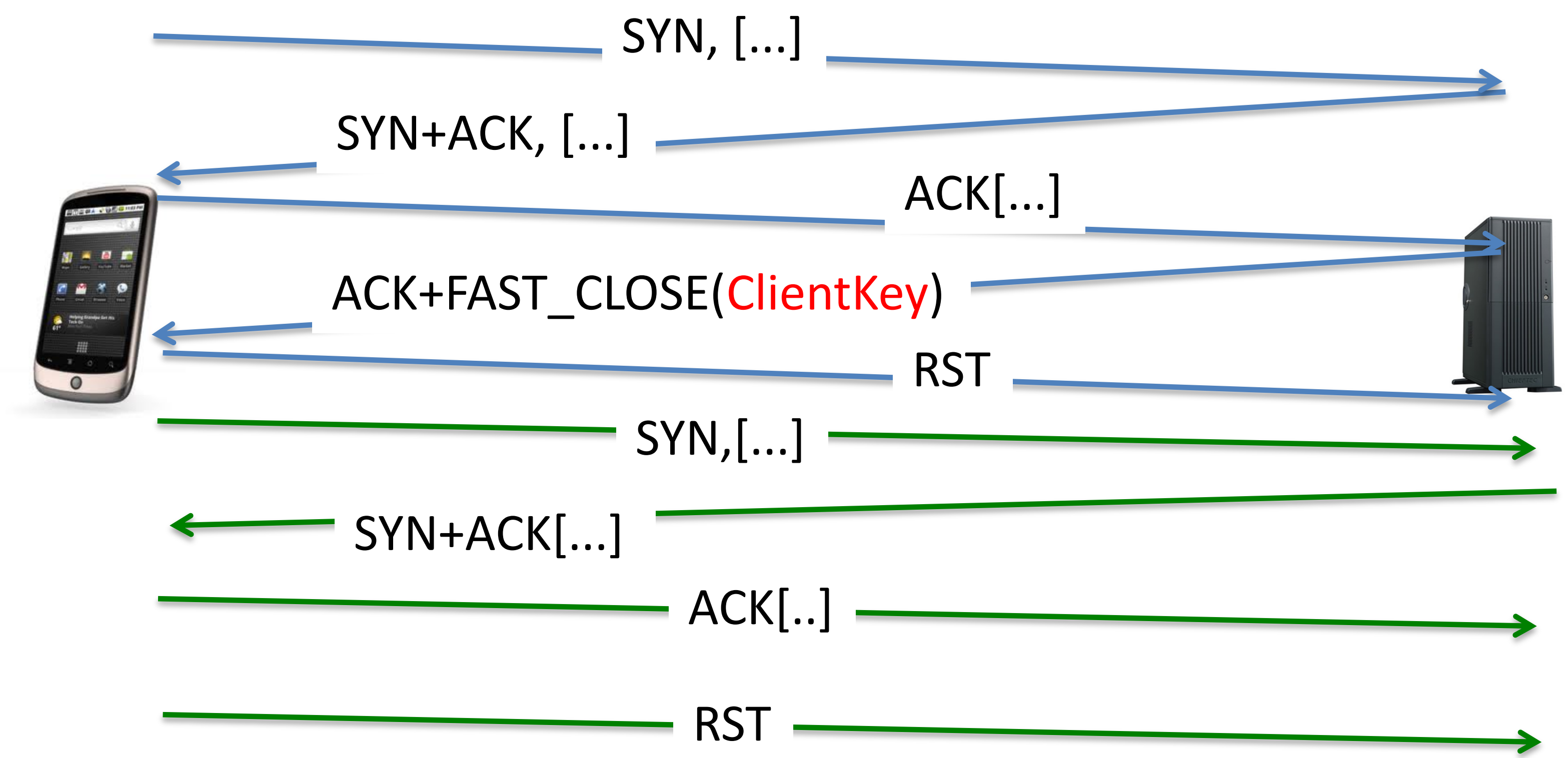
Quickly closing a Multipath TCP connection

- How to quickly close a Multipath TCP connection ?
  - By closing all subflows ?



Closing a Multipath TCP connection

- FAST Close

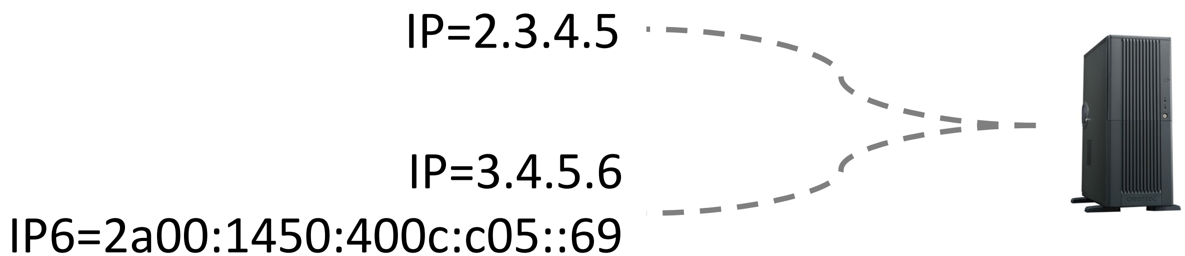


The Multipath TCP control plane

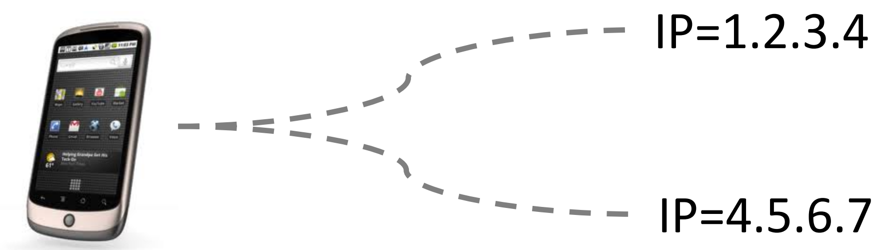
- Connection establishment in details
- Closing a Multipath TCP connection
- Address dynamics

Multipath TCP
Address dynamics

- How to learn the addresses of a host ?

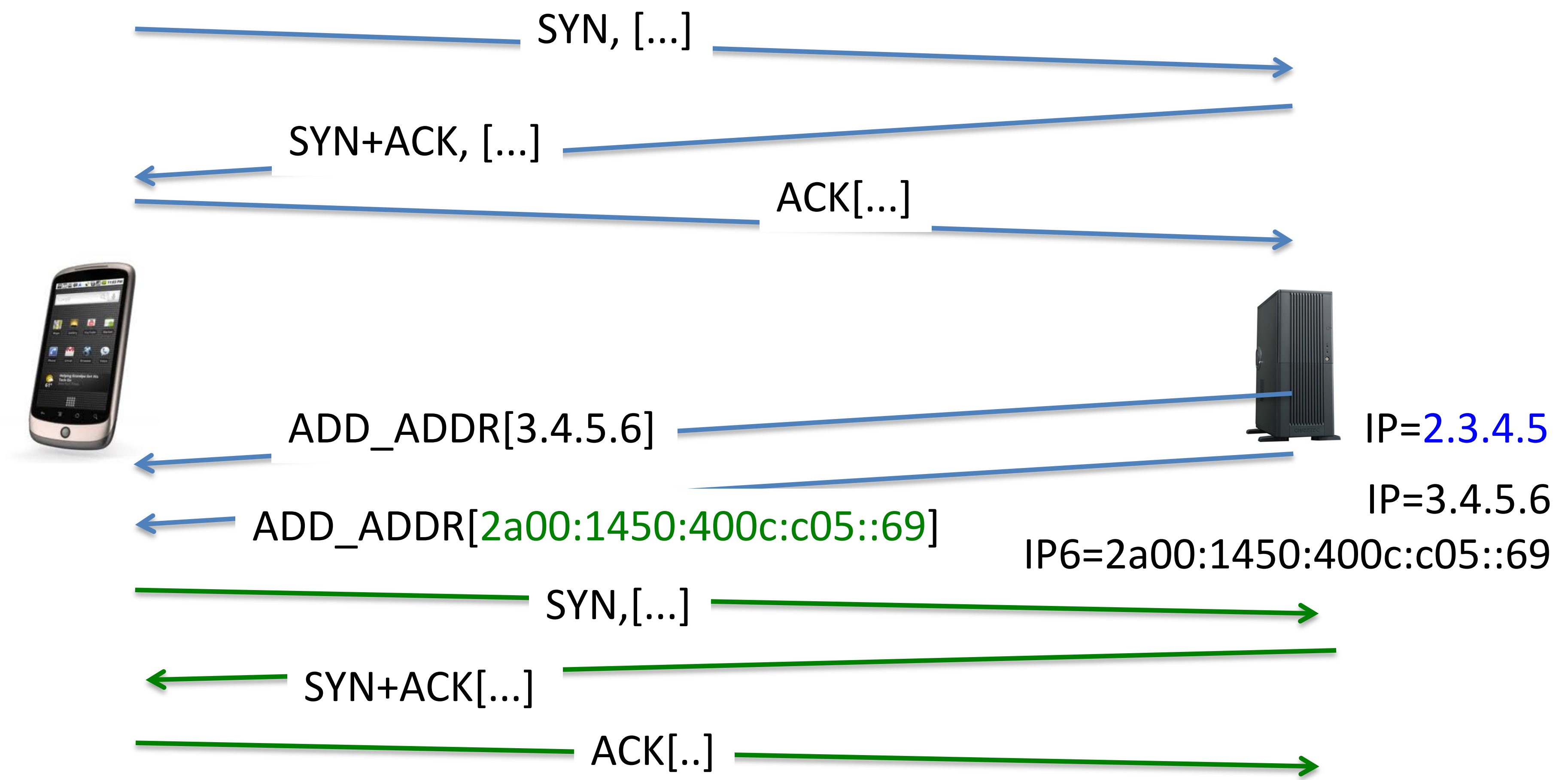


- How to deal with address changes ?



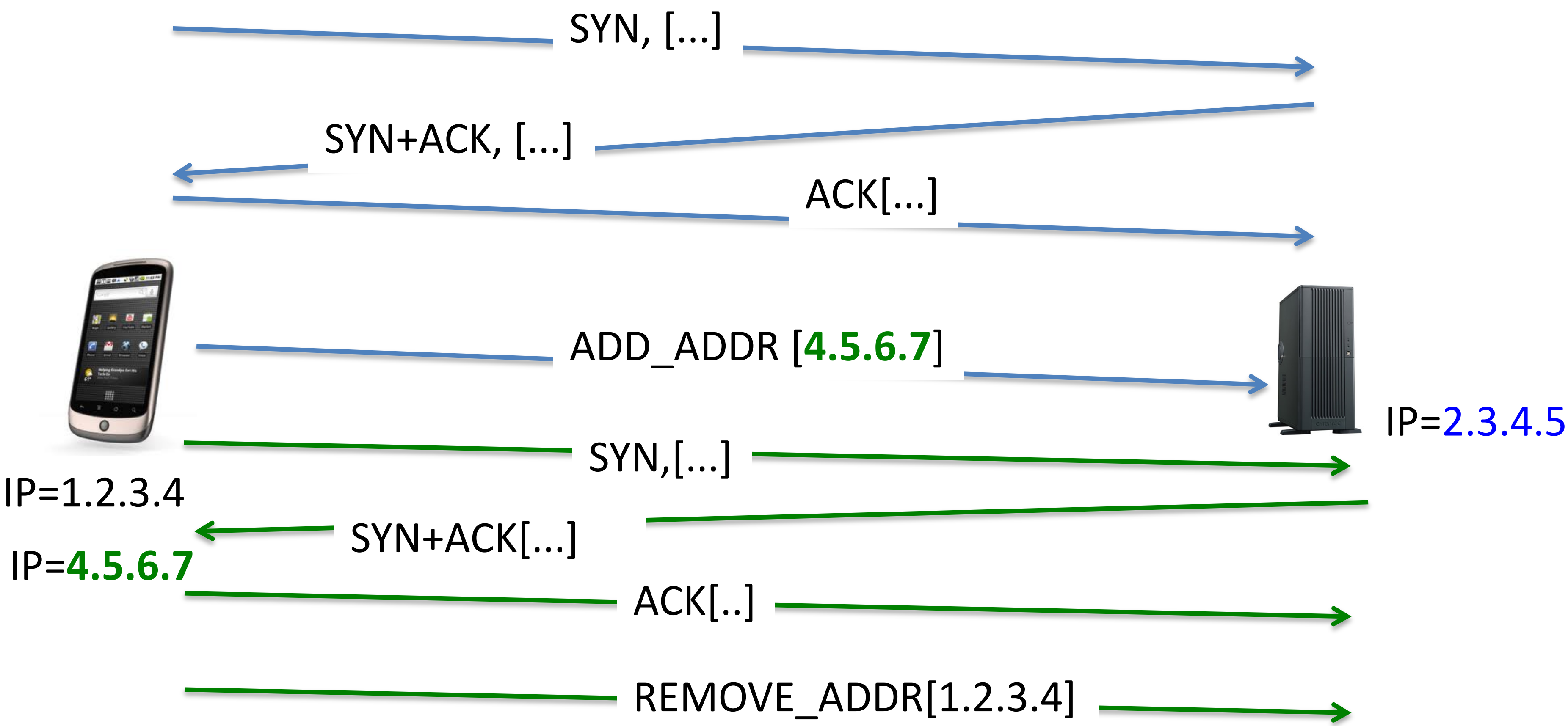
Address dynamics

- Basic solution : multihomed server

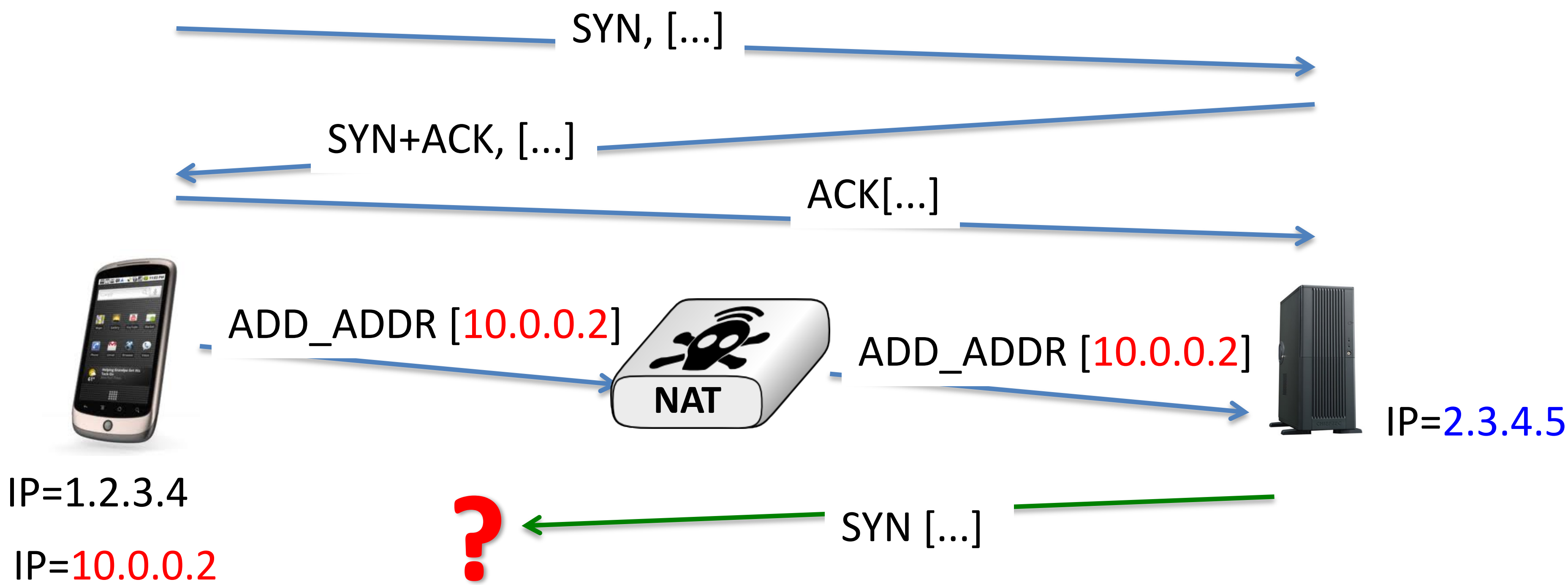


Address dynamics

- Basic solution : mobile client



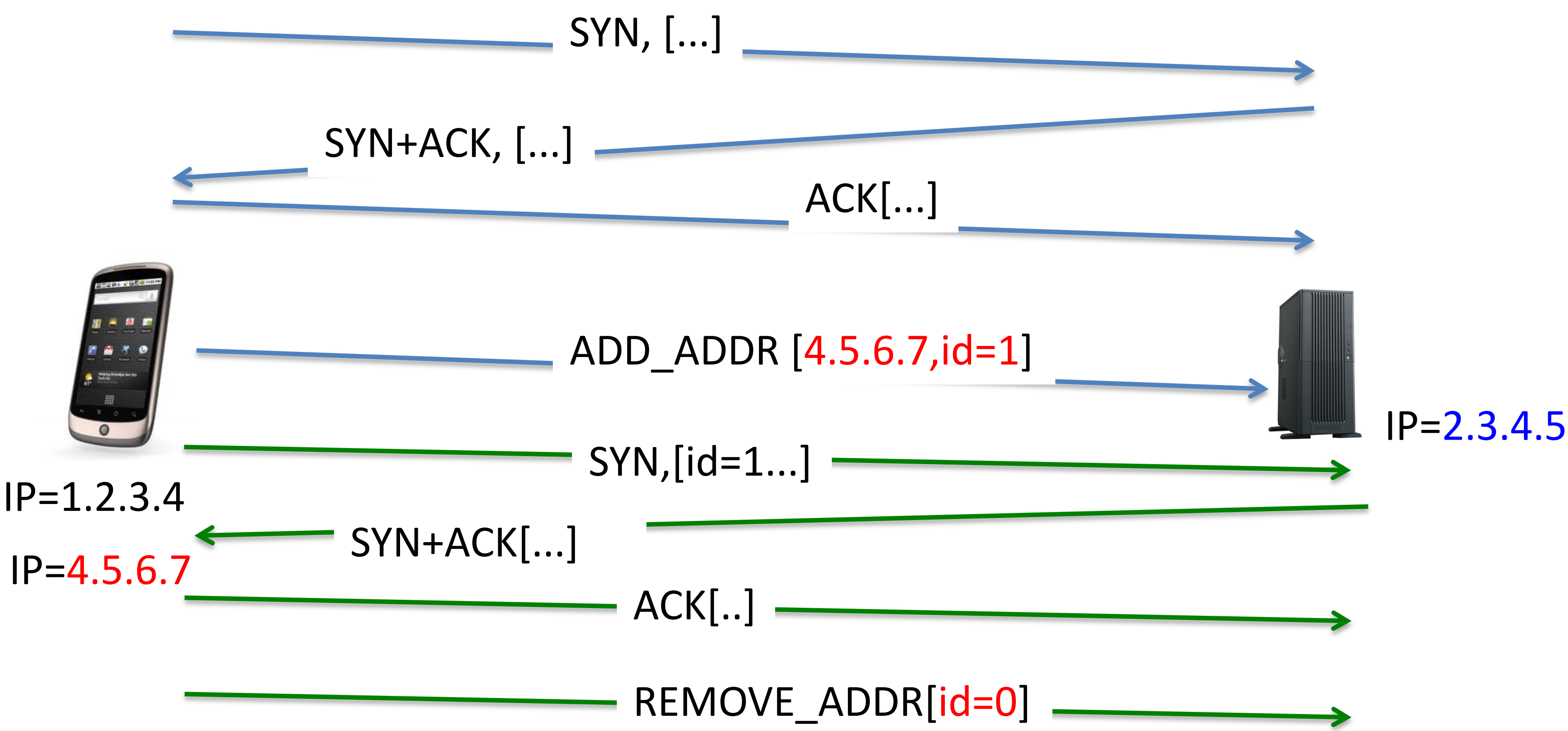
Address dynamics in today's Internet



Address dynamics with NATs

- Solution
  - Each address has one identifier
    - Subflow is established between id=0 addresses
  - Each host maintains a list of <address,id> pairs of the addresses associated to an MPTCP endpoint
  - MPTCP options refer to the address identifier
    - ADD_ADDR contains <address,id>
    - REMOVE_ADDR contains <id>

Address dynamics



The Multipath TCP Path Manager

- Leverages the information about
  - Local IP addresses and interface state
  - Remote IP addresses
- Decides to create and terminate subflows
- Sample path managers
  - Full mesh : one subflow between each pair addr.
  - Ndiffports: for single-homed hosts, n parallel flows
  - Use case specific path managers

Conclusion

- Multipath TCP is a reality
  - Many scientific papers (see references)
  - Due to the middleboxes, the protocol is more complex than initially expected
  - Multipath TCP works over today's Internet
    - Large scale deployments
- What's next ?
  - Ongoing effort to integrate it in the mainline Linux kernel for wider deployment

Hands-on

https://github.com/qdeconinck/sigcomm20_mptp_tutorial

- First check that the VM is working correctly

```
# The first `vagrant up` invocation fetches the vagrant box and runs the provision scrip
# It is likely that this takes some time, so launch this command ASAP!
# The following `vagrant reload` command is required to restart the VM with the Multipat
$ vagrant up; vagrant reload
# Now that your VM is ready, let's SSH it!
$ vagrant ssh
```

Once done, you should be connected to the VM. To check that your VM's setup is correct, let's run the following commands inside the VM

```
$ cd ~; ls
# iproute-mptcp  mininet  minitopo  oflops  oftest  openflow  picotls  pox  pquic
$ uname -a
# Linux ubuntu-bionic 4.14.146.mptcp #17 SMP Tue Sep 24 12:55:02 UTC 2019 x86_64 x86_64
```

Hands-on : bandwidth aggregation

1. Observing the Bandwidth Aggregation when Using Multiple Paths

One of the use cases of multipath transport protocols is to aggregate the bandwidths of the available paths. To demonstrate this, let's consider a simple, symmetrical network scenario.

```
   |-------- 20 Mbps, 40 ms RTT ---------|
Client                                  Router --------- Server
   |-------- 20 Mbps, 40 ms RTT ---------|
```

This scenario is described in the file `01_multipath/topo`. With this network, we will compare two `iperf` runs. The first consists in a regular TCP transfer between the client and the server. To perform this experiment, `ssh` into the vagrant VM and then type the following commands

```
$ cd /tutorial/01_multipath
$ mprun -t topo -x xp_tcp
```

```
$ mprun -t topo -x xp_mptcp
```

Comparing packet schedulers

Case 1: request/response traffic from client perspective

```
   |-------- 100 Mbps, 40 ms RTT --------|
Client                                Router --------- Server
   |-------- 100 Mbps, 80 ms RTT --------|
```

Let's consider a simple traffic where the client sends requests every 250 ms (of 10 KB, a size inferior to an initial congestion window) and the server replies to them. The client computes the delay between sending the request and receiving the corresponding response. To perform the experiment with the Lowest RTT scheduler, run the following command under folder `/tutorial/02_scheduler/reqres`:

```
$ mprun -t topo -x reqres_rtt
```

When inspecting the `msg_client.log` file containing the measured delays in seconds, you can notice that all the delays are about 40-50 ms. Because the Lowest RTT scheduler always prefer the faster path, and because this fast path is never blocked by the congestion window due to the application traffic, the data only flows over the fast path.

To perform the same experiment using the Round-Robin packet scheduler, runs:

```
$ mprun -t topo -x reqres_rr
```

Comparing packet schedulers (2)

```
   |-------- 20 Mbps, 30 ms RTT ---------|
Client                                Router --------- Server
   |-------- 20 Mbps, 100 ms RTT --------|
```

On this network, the client will perform a HTTP GET request to the server for a file of 10 MB. The experiences files are located in the folder `/tutorial/02_scheduler/http`. In the remaining, we assume that each host uses a (fixed) sending (resp. receiving) window of 1 MB.

First perform the run using regular TCP. Single-path TCP will only take advantage of the upper path (the one with 30 ms RTT).

```
$ mprun -t topo -x http_tcp
```

Have a look at the time indicated at the end of the `http_client.log` file, and keep it as a reference.

Now run any of the following lines using Multipath TCP

```
# Using Lowest RTT scheduler
$ mprun -t topo -x http_rtt
# Using Round-Robin scheduler
$ mprun -t topo -x http_rr
```

Path managers

The path manager is the multipath algorithm that determines how subflows will be created over a Multipath TCP connection. In the Linux kernel implementation, we find the following simple algorithms:

- `default` : a "passive" path manager that does not initiate any additional subflow on a connection
- `fullmesh` : the default path manager creating a subflow between each pair of (IP client, IP server)
- `ndiffports` : over the same pair of (IP client, IP server), creates several subflows (by default 2) by modifying the source port.

Notice that in Multipath TCP, only the client initiates subflows. To understand these different algorithms, consider the following network scenario first.

```
Client ----- 25 Mbps, 20 ms RTT ------ Router --------- Server
```

Let us first consider the difference between the `fullmesh` and the `ndiffports` path managers. Run the associated experiments (running an iperf traffic) and compare the obtained goodput. Then, have a look at their corresponding PCAP files to spot how many subflows were created for each experiment.

```
$ mprun -t topo_single_path -x iperf_fullmesh
$ mprun -t topo_single_path -x iperf_ndiffports
```

References

- The Multipath TCP protocol
  – https://www.multipath-tcp.org
  – http://tools.ietf.org/wg/mptcp/

A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for Multipath TCP development", RFC6182 2011.

A. Ford, C. Raiciu, M. J. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC6824, 2013

A. Ford, C. Raiciu, M. J. Handley, and O. Bonaventure, C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC8684, 2020

C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath TCP," NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, 2012.

Multipath TCP : RFCs

- RFC6181: Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses
- RFC6182: Architectural Guidelines for Multipath TCP Development
- RFC6356: Coupled Congestion Control for Multipath Transport Protocols
- RFC6824: TCP Extensions for Multipath Operation with Multiple Addresses (v0)
- RFC6897: Multipath TCP (MPTCP) Application Interface Considerations
- RFC7430: Analysis of Residual Threats and Possible Fixes for Multipath TCP (MPTCP)
- RFC8041: Use Cases and Operational Experience with Multipath TCP
- RFC8684: TCP Extensions for Multipath Operation with Multiple Addresses (v1)

Implementations

- Linux
  - Patches to different kernel versions supporting v0
    - http://www.multipath-tcp.org
    - https://github.com/multipath-tcp
  - Ongoing effort by RedHat, Apple, intel and Tessares to include Multipath TCP v1 in mainline Linux
    - https://github.com/multipath-tcp/mptcp_net-next/wiki
- Apple
  - Included in iOS and MacOS
- FreeBSD (not updated)
  - http://caia.swin.edu.au/urp/newtcp/mptcp/

Multipath TCP tutorials and surveys

– Paasch, C., & Bonaventure, O. (2014). Multipath tcp. *Communications of the ACM*, *57*(4), 51-57.
– Raiciu, C., Iyengar, J., & Bonaventure, O. (2013). Recent advances in reliable transport protocols. *SIGCOMM ebook on Recent Advances in Networking*.
– Bonaventure, O., Handley, M., & Raiciu, C. (2012). An overview of Multipath TCP. *; login:*, *37*(5), 17.
– Li, M., Lukyanenko, A., Ou, Z., Ylä-Jääski, A., Tarkoma, S., Coudron, M., & Secci, S. (2016). Multipath transmission for the internet: A survey. *IEEE Communications Surveys & Tutorials*, *18*(4), 2887-2925.
– Polese, M., Chiariotti, F., Bonetto, E., Rigotto, F., Zanella, A., & Zorzi, M. (2019). A survey on recent advances in transport layer protocols. *IEEE Communications Surveys & Tutorials*, *21*(4), 3584-3608.

Use cases

- Bonaventure, O., & Seo, S. (2016). Multipath TCP deployments. *IETF Journal*, *12*(2), 24-27.
- C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/WiFi handover with multipath TCP," CellNet '12: Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design, 2012.
- De Coninck, Q., Baerts, M., Hesmans, B., & Bonaventure, O. (2016). Observing real smartphone applications over multipath TCP. *IEEE Communications Magazine*, *54*(3), 88-93.
- Nikravesh, A., Guo, Y., Qian, F., Mao, Z. M., & Sen, S. (2016, October). An in-depth understanding of multipath TCP on mobile devices: Measurement and system design. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (pp. 189-201).
- C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. J. Handley, "Improving datacenter performance and robustness with multipath TCP," *ACM SIGCOMM* 2011.
- Duchene, F., & Bonaventure, O. (2017, October). Making multipath tcp friendlier to load balancers and anycast. In 2017 IEEE 25th International Conference on Network Protocols (ICNP) (pp. 1-10). IEEE.
- Keukeleire, N., Hesmans, B., & Bonaventure, O. (2020). Increasing Broadband Reach with Hybrid Access Networks. *IEEE Communications Standards Magazine*, *4*(1), 43-49.

Middleboxes and Multipath TCP

M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda, "Is it still possible to extend TCP?," IMC '11: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, 2011.

Hesmans, B., Duchene, F., Paasch, C., Detal, G., & Bonaventure, O. (2013, December). Are TCP extensions middlebox-proof?. In *Proceedings of the 2013 workshop on Hot topics in middleboxes and network function virtualization* (pp. 37-42).

Detal, G., Hesmans, B., Bonaventure, O., Vanaubel, Y., & Donnet, B. (2013, October). Revealing middlebox interference with tracebox. In *Proceedings of the 2013 conference on Internet measurement conference* (pp. 1-8).

Multipath congestion control

– Background

D. Wischik, M. Handley, and M. B. Braun, “The resource pooling principle,” *ACM SIGCOMM Computer …*, vol. 38, no. 5, 2008.

F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. ACM SIGCOMM CCR, 35, 2005.

P. Key, L. Massoulie, and P. D. Towsley, “Path Selection and Multipath Congestion Control,” INFOCOM 2007. 2007, pp. 143–151.

– Coupled congestion control

C. Raiciu, M. J. Handley, and D. Wischik, “Coupled Congestion Control for Multipath Transport Protocols,” *RFC*, vol. 6356, Oct. 2011.

D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, “Design, implementation and evaluation of congestion control for multipath TCP,” NSDI'11: Proceedings of the 8th USENIX conference on Networked systems design and implementation, 2011.

Multipath congestion control

R. Khalili, N. Gast, M. Popovic, U. Upadhyay, J.-Y. Le Boudec , MPTCP is not Pareto-optimal: Performance issues and a possible solution, Proc. ACM Conext 2012

Y. Cao, X. Mingwei, and X. Fu, “Delay-based Congestion Control for Multipath TCP,” ICNP2012, 2012.

T. A. Le, C. S. Hong, and E.-N. Huh, “Coordinated TCP Westwood congestion control for multiple paths over wireless networks,” ICOIN '12: Proceedings of the The International Conference on Information Network 2012, 2012, pp. 92–96.

T. A. Le, H. Rim, and C. S. Hong, “A Multipath Cubic TCP Congestion Control with Multipath Fast Recovery over High Bandwidth-Delay Product Networks,” *IEICE Transactions*, 2012.

T. Dreibholz, M. Becke, J. Pulinthanath, and E. P. Rathgeb, “Applying TCP-Friendly Congestion Control to Concurrent Multipath Transfer,” Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, 2010, pp. 312–319.

Multipath TCP : schedulers

- Paasch, C., Ferlin, S., Alay, O., & Bonaventure, O. (2014, August). Experimental evaluation of multipath TCP schedulers. In *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop* (pp. 27-32).
- Frömmgen, A., Rizk, A., Erbshäußer, T., Weller, M., Koldehofe, B., Buchmann, A., & Steinmetz, R. (2017, December). A programming model for application-defined multipath TCP scheduling. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference* (pp. 134-146).
- De Coninck, Q., & Bonaventure, O. (2018, May). Tuning multipath TCP for interactive applications on smartphones. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops* (pp. 1-9). IEEE.
- Ferlin, S., Alay, Ö., Mehani, O., & Boreli, R. (2016, May). BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops* (pp. 431-439). IEEE.
- Corbillon, X., Aparicio-Pardo, R., Kuhn, N., Texier, G., & Simon, G. (2016, May). Cross-layer scheduler for video streaming over MPTCP. In *Proceedings of the 7th International Conference on Multimedia Systems* (pp. 1-12).
- O. Bonaventure, M. Piraux, M. Baerts, C. Paasch, M. Amend, Multipath schedulers, draft-bonaventure-iccrg-schedulers-00, 2020

Multipath TCP : Path management

- Hesmans, B., Detal, G., Barre, S., Bauduin, R., & Bonaventure, O. (2015, December). Smapp: Towards smart multipath tcp-enabled applications. In *Proceedings of the 11th acm conference on emerging networking experiments and technologies* (pp. 1-7).
- Hesmans, B., & Bonaventure, O. (2016, July). An enhanced socket API for Multipath TCP. In *Proceedings of the 2016 applied networking research workshop* (pp. 1-6).
- Othman, O. & Martineau, M. The Multipath TCP Daemon, https://github.com/intel/mptcpd
- Tran, V.-H., Bonaventure, O., Beyond socket options: Towards fully extensible Linux transport stacks. In *Computer Communications*, 2020