WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

Inter-Domain Handoff Over 3G and Wi-Fi

Steve Ajemian
Bhuwan Agarwal

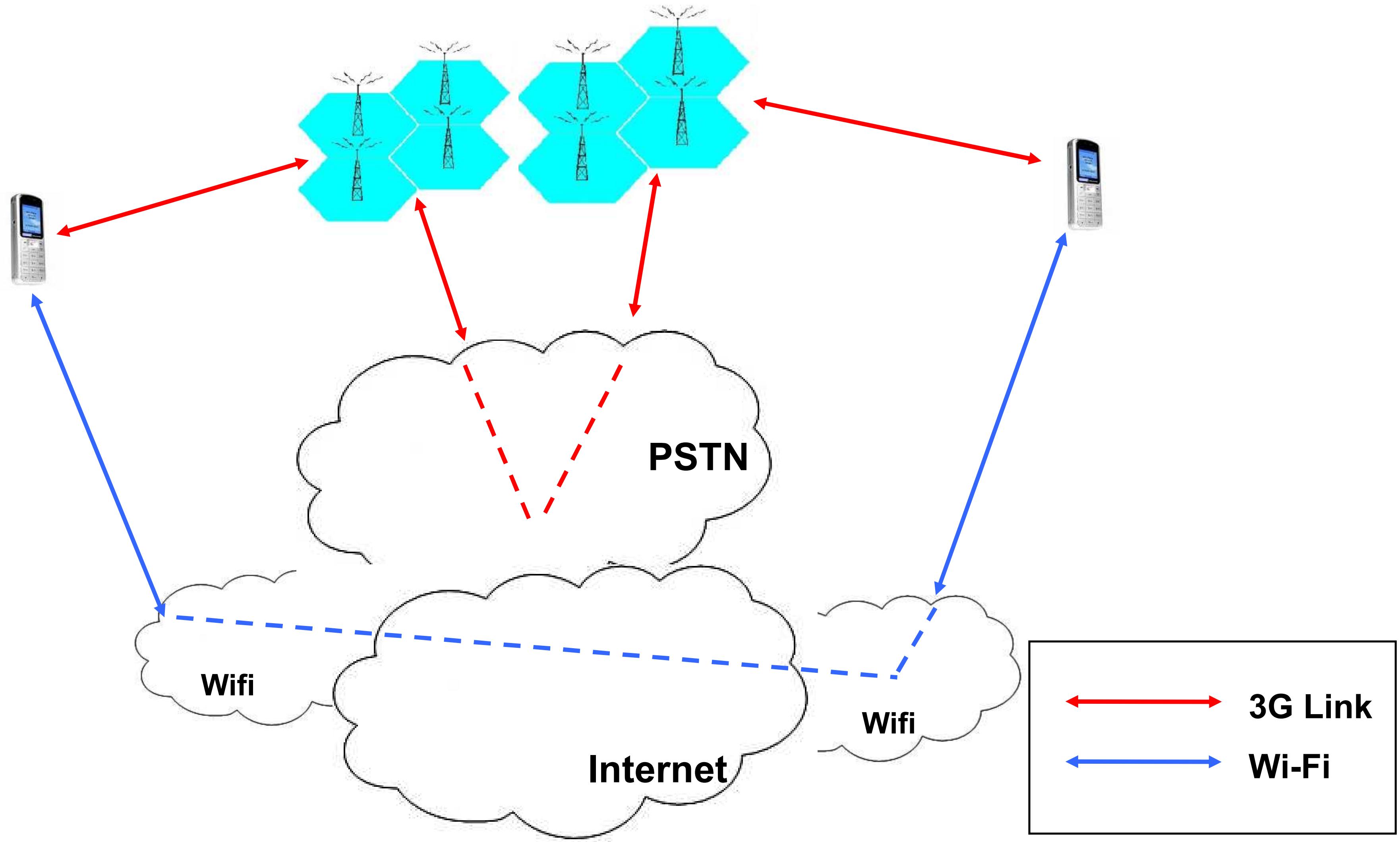
Motivation

- The goals of this project were to examine the performance of inter-domain handoff between Wi-Fi and 3G

- Growing number of mobile devices provide internet access from both domains

- Wanted to examine the feasibility of using handoff for a VOIP application

Scenario



WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

Scenario

Preconditions:

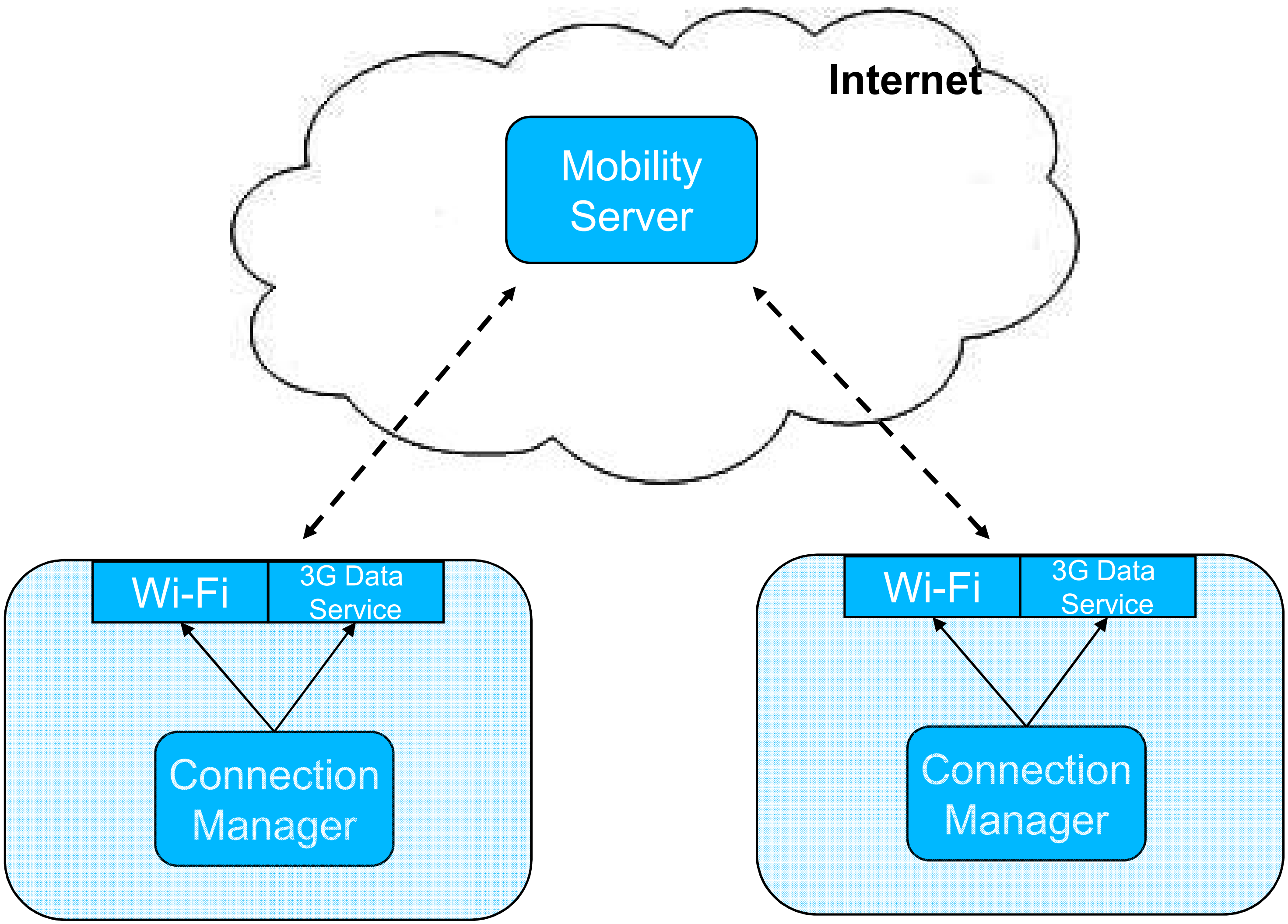
1. Both mobile terminals have both 3G data link and Wi-Fi data link
2. A “Mobility server” acts as a gateway between both parties using a static IP address.

Capability to Provide:

1. Each party will continually send packets every 20 ms to simulate a VoIP packet communicating through the Mobility Server, which relays packets to each party.
2. If one party detects a Wi-Fi access point, it will join AP and begin sending/receiving packets over the Wi-Fi interface.
3. The Mobility Server will terminate the 3G data link

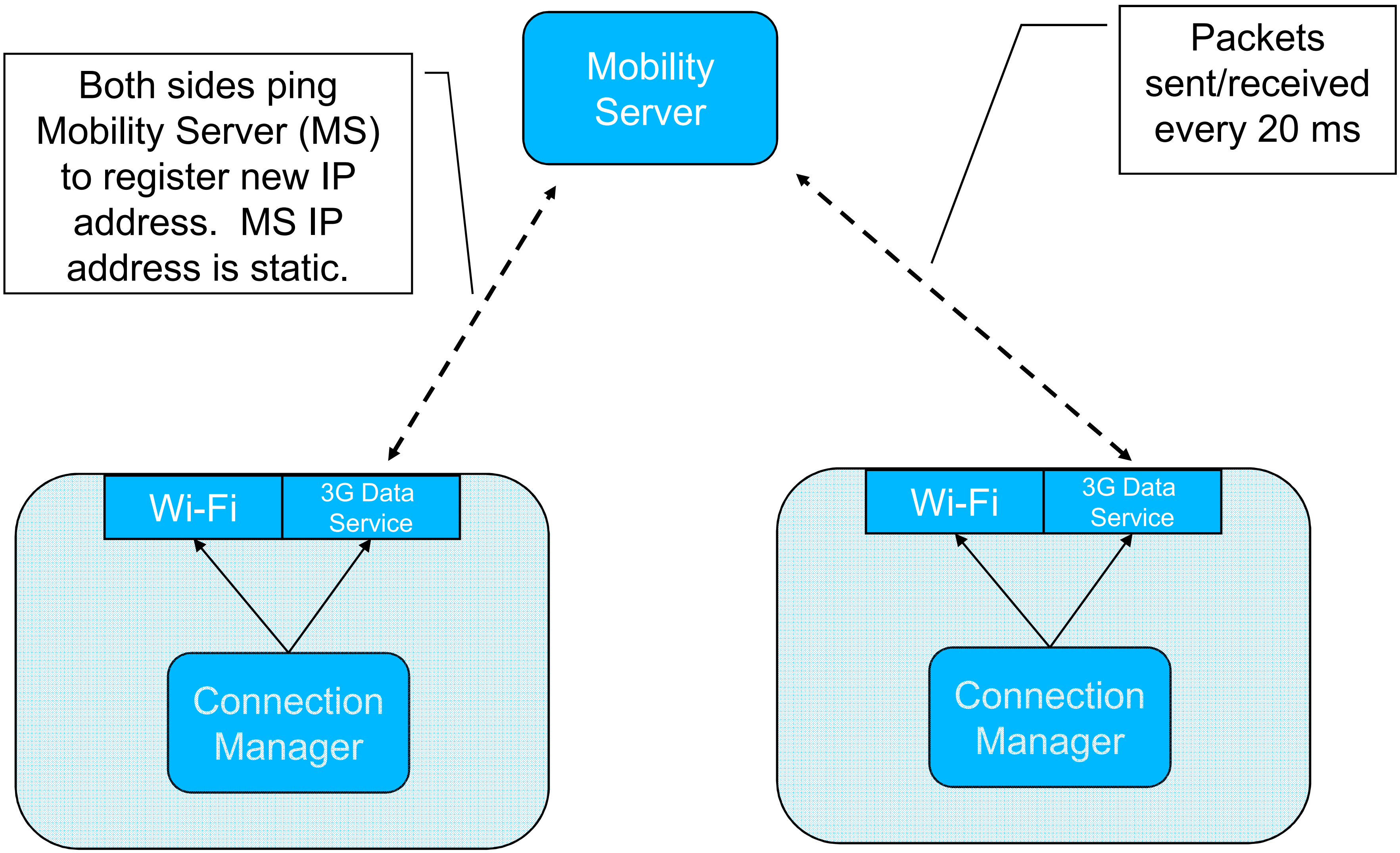
WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

System Overview

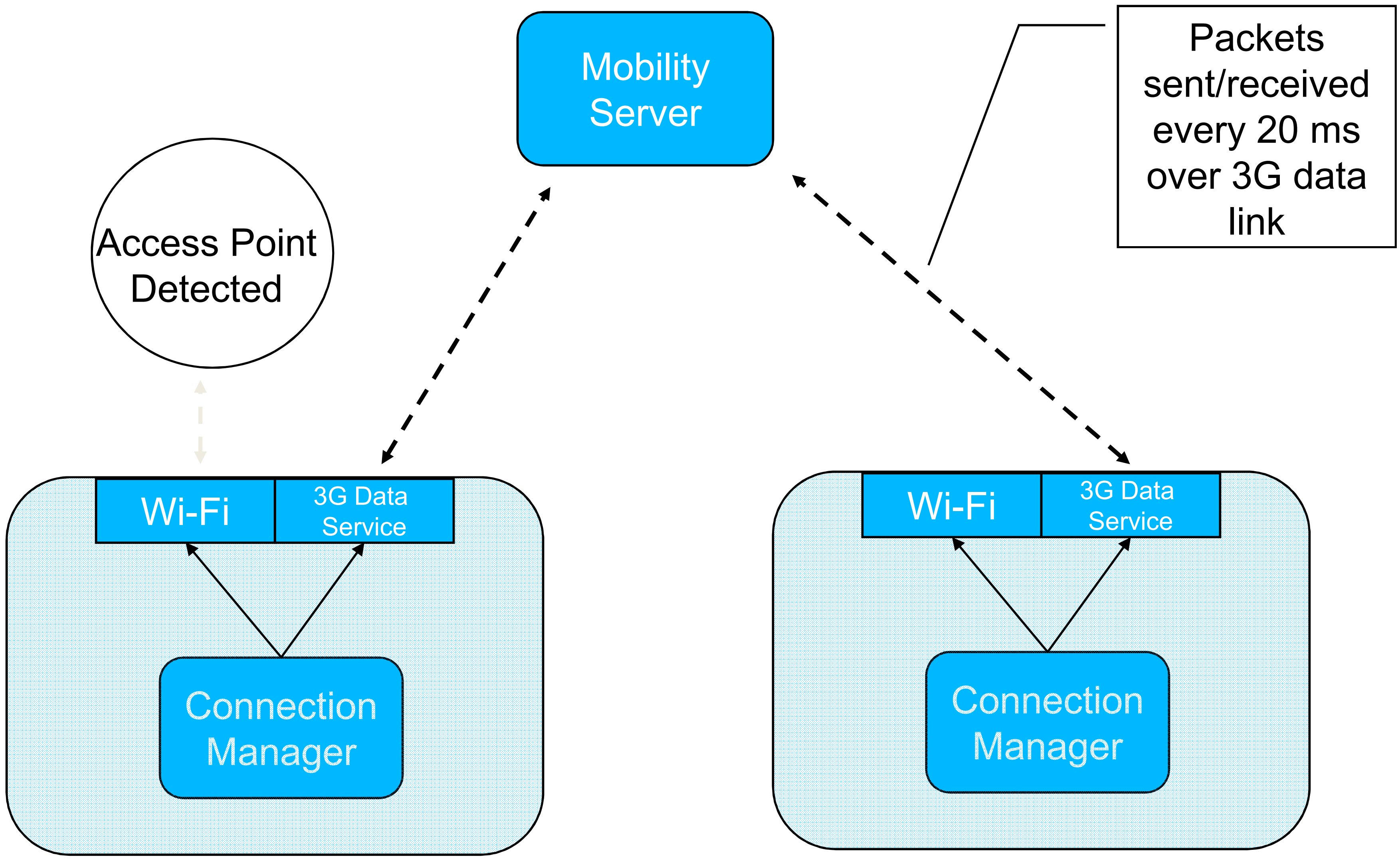


WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

Step 1

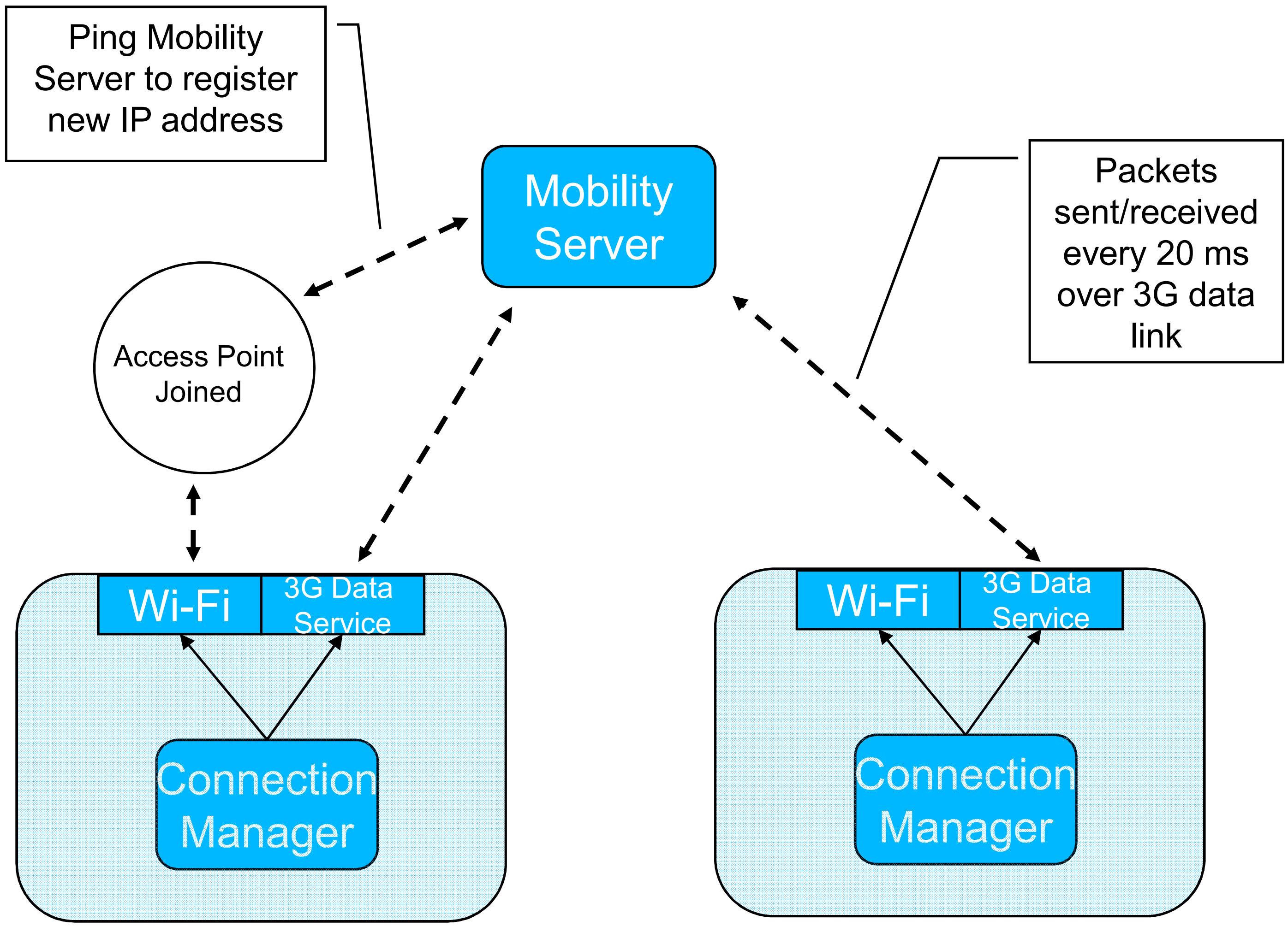


Step 2

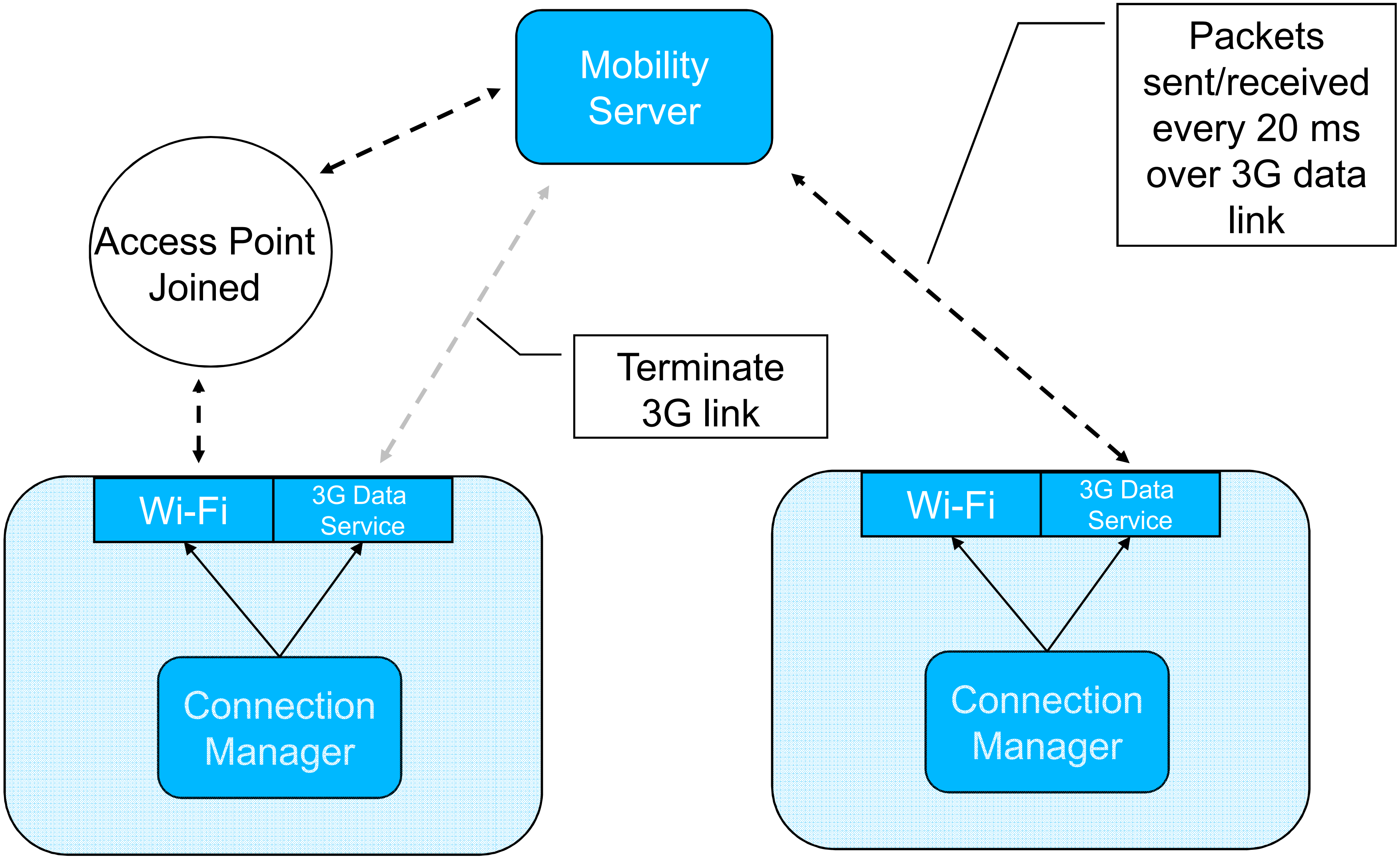

Mobility Server
Packets sent/received every 20 ms over 3G data link
Access Point Detected
Wi-Fi
3G Data Service
Connection Manager
Wi-Fi
3G Data Service
Connection Manager

WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

Step 3

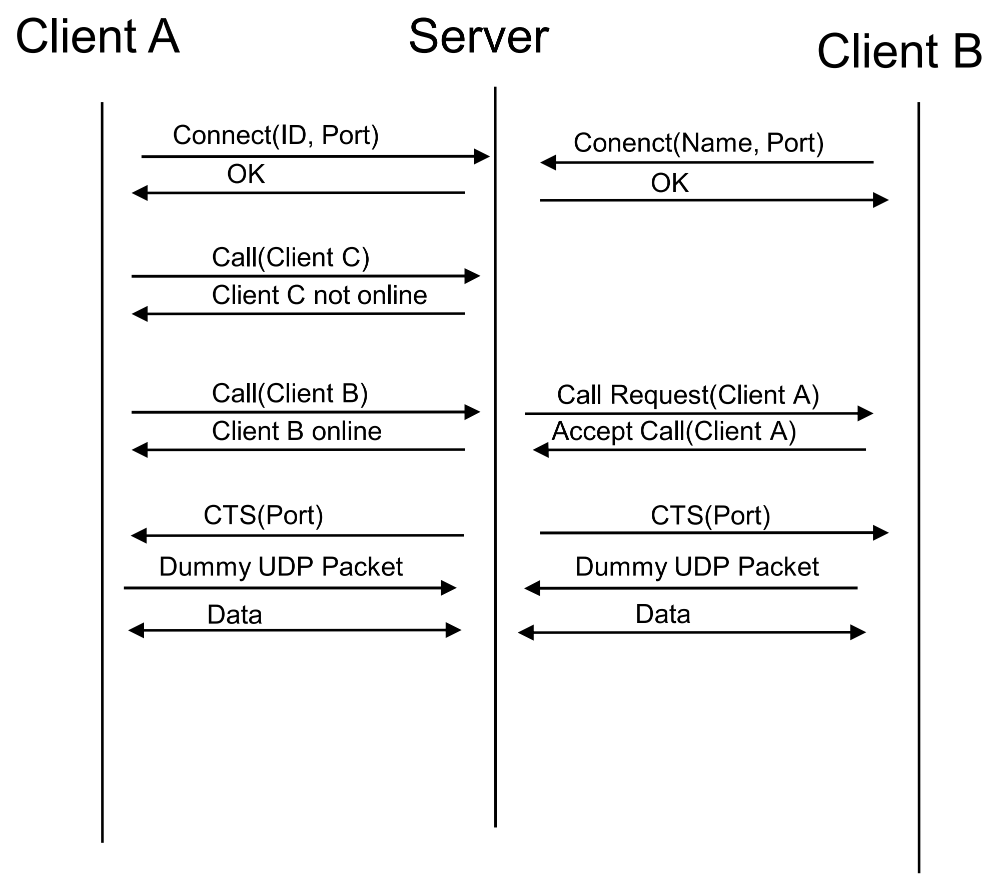


Step 4


Mobility Server
Access Point Joined
Packets sent/received every 20 ms over 3G data link
Terminate 3G link
Wi-Fi
3G Data Service
Connection Manager
Wi-Fi
3G Data Service
Connection Manager

WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

Startup



1. Client A opens a Control Connection over TCP and sends start message over TCP Control Connection, sent to the well-known port on Server.
2. Client A sends Call request to a client

   - Receiver must be online
3. Client A will wait for an OK from the server indicating that Client C has connected, at which point it will begin sending/receiving data. The CTS message will contain the port number to send UDP packets to.

Assumptions:

-If Wi-Fi is available, a Client will use that interface, otherwise 3G will be used.

-Client B will initiate startup in the same manner.

Handover

WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

Server

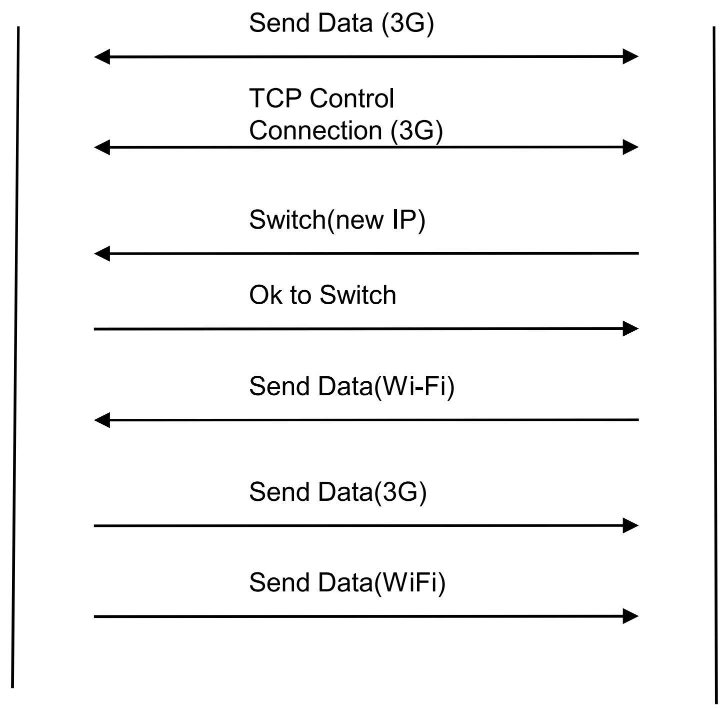
1. Server receives Switch message with new IP address

   -Continues sending data over 3G

2. Server gets UDP packet over new IP address

   - Start sending over new IP address

   -Disables previous IP address after timeout



Client B

1. Client B enters Wi-Fi
2. Opens TCP Control Connection over existing interface
3. Client sends switch message to server containing new IP
4. Continue receiving data over 3G socket

   -Start sending over Wi-Fi connection

5. Receives packets over new IP address

   -Keeps 3G socket for some time t for delayed packets (timeout), then closes connection

Assumptions:

-During handoff, both interfaces are available

-Existing connection between users

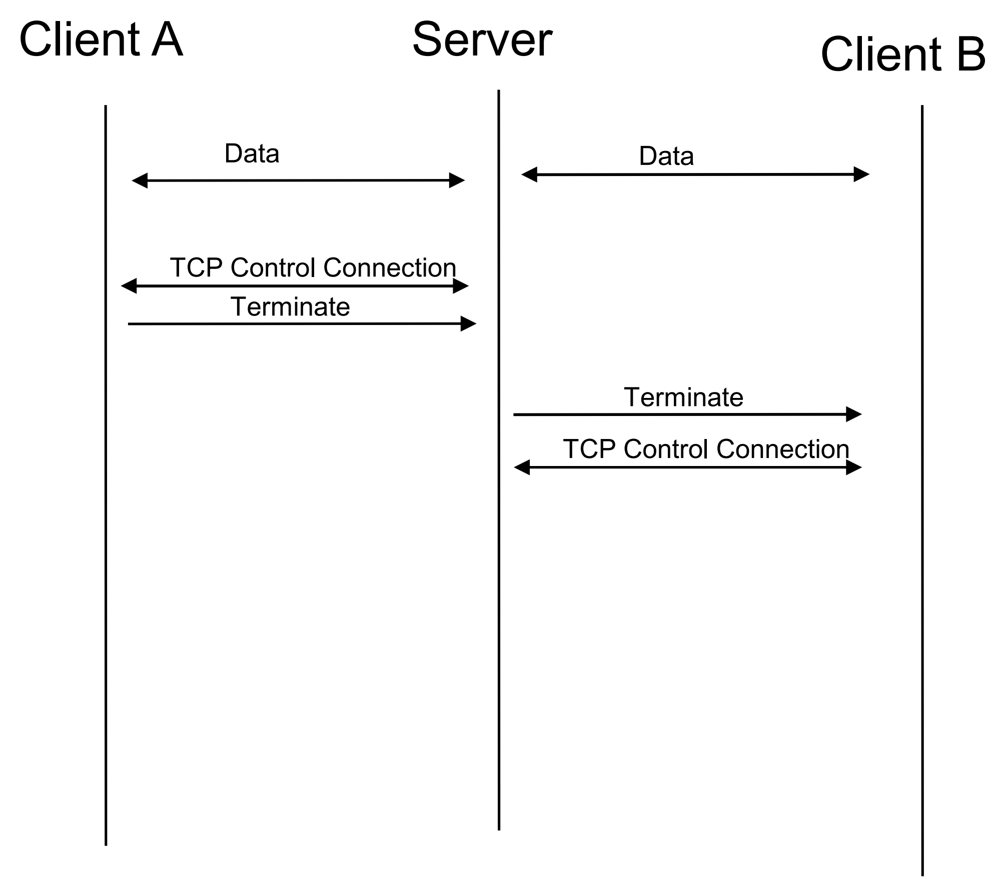
-Switch from Wi-Fi to 3G works the same way

Termination

1. Client will open a TCP connection to initiate tear-down by sending a Terminate message to the server

2. The Server will send a terminate message to the other client over UDP (sent 4 times). The Server will set a timer while waiting for a TCP connection to be opened.
    - Terminate message will be resent if timeout reached

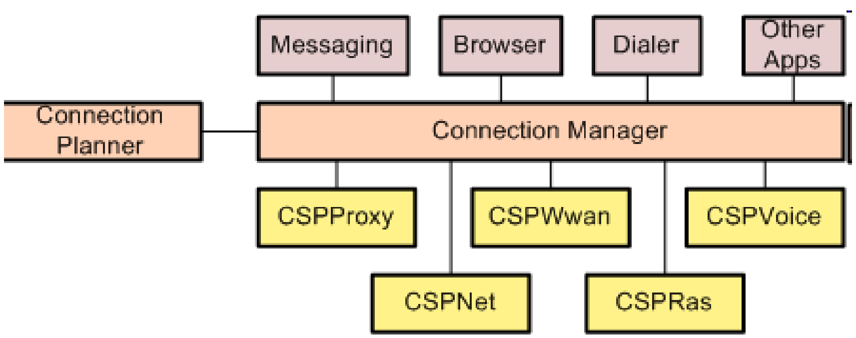
3. After each Client opens the TCP Control Connection, packets received on the UDP ports at each Client will be dropped, and the session will end.



Connection Manager

- The Connection Manager API automates the establishment and management of various types of network connections for applications sending UDP or TCP traffic. When an application requests a connection, the Connection Manager establishes a connection using an "optimal" connection type. Applications are configured to specify a connection name and a network name.

- Connection Manager creates a connection to an interface that the user specifies. However, once the connection is made, it will persist throughout the process or thread on subsequent creation of UDP/TCP sockets.



Source:
http://www.codeproject.com/KB/mobile/ConnectionManager.aspx?display=Print

WHITING SCHOOL OF ENGINEERING
JOHNS HOPKINS UNIVERSITY

Issues

- The use of Connection Manager forced us to use threading to create sockets on different interfaces (Wifi/3G).

- The use of the UDPClient class forced us to use blocking receive calls to receive UDP traffic.
  - Windows Mobile does not support the specification of timeouts during blocking receive calls
  - This forced us to create separate threads for all of our UDP receiving sockets using an asynchronous "beginreceive" function provided from built-in Windows Mobile dll. Beginreceive recursively calls a receive each time a packet is received.
  - Resulted in unpredictable behavior

- Throughput on Device conducting two handoffs (one handoff from 3G to Wifi, and another handoff from Wifi to 3G:
  - Packets 1 – 500: 3.76 packets/sec
  - Packets 500 – 1000: 4.31 packets/sec
  - Packets 1000-1500: 4.85 packets/sec

- Expected throughput closer to 50 packets/sec (based on 20 ms send interval)
  - Some sort of delay is occurring in our Receive thread
  - We think it might be in the recursive receive call, however, the literature explaining the operation of this function is vague