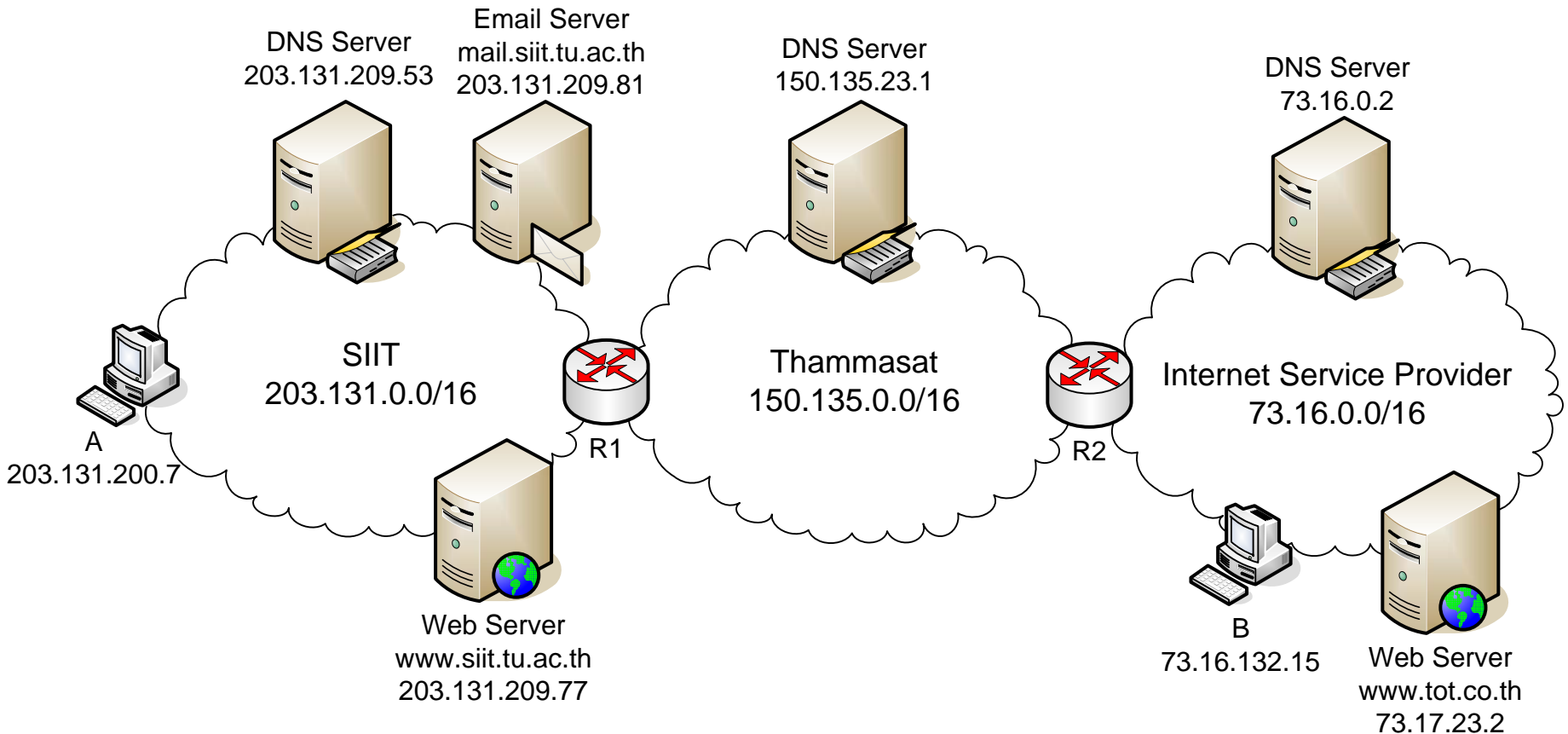


Case Study: SIIT Internet Applications

Dr Steve Gordon
ICT, SIIT

Internet Applications at SIIT



DNS Example

- Scenario:
 - A user on Host A browses to the website www.tot.co.th/index.html
- Assumptions:
 - Each DNS server knows about the URLs/IP addresses in its DNS cache
 - Each DNS server knows the IP address of the “next” DNS server
 - Each host knows the IP address of its local DNS server
 - Host A has no DNS cache (yet)
- Actions:
 1. Since Host A does not know the IP address for www.tot.co.th it uses DNS to find out. Host A sends a DNS message to the SIIT DNS server
 2. The SIIT DNS server does not know about www.tot.co.th so it sends a DNS request to the Thammasat DNS server
 3. The Thammasat DNS server knows the IP address of www.tot.co.th (it is 73.17.23.2). The Thammasat DNS server sends a response back to the SIIT DNS server
 4. The SIIT DNS server stores the value in its DNS cache for later use (not shown in the figure), and sends the response back to Host A
- Result:
 - Host A now knows the IP address of www.tot.co.th so it can send an IP datagram to this web server. Host A will store the IP address for www.tot.co.th in its local DNS cache



DNS Example

DNS Cache

| | |
|--------------------|----------------|
| www.siiit.tu.ac.th | 203.131.209.77 |
| www.tu.ac.th | 150.135.23.17 |
| www.google.com | 66.45.130.3 |

DNS Cache

| | |
|--------------------|----------------|
| www.siiit.tu.ac.th | 203.131.209.77 |
| www.tu.ac.th | 150.135.23.17 |
| www.tot.co.th | 73.17.23.2 |

DNS Cache

| | |
|---------------|---------------|
| www.yahoo.com | 11.16.34.6 |
| www.tu.ac.th | 150.135.23.17 |
| www.tot.co.th | 73.17.23.2 |

DNS Server
203.131.209.53

DNS Server
150.135.23.1

DNS Server
73.16.0.2

1. What is the IP address
for www.tot.co.th?

2. What is the IP address
for www.tot.co.th?

3. IP address for
www.tot.co.th is
73.17.23.2

4. IP address for
www.tot.co.th is
73.17.23.2

A
203.131.200.7

Web Server
www.tot.co.th
73.17.23.2

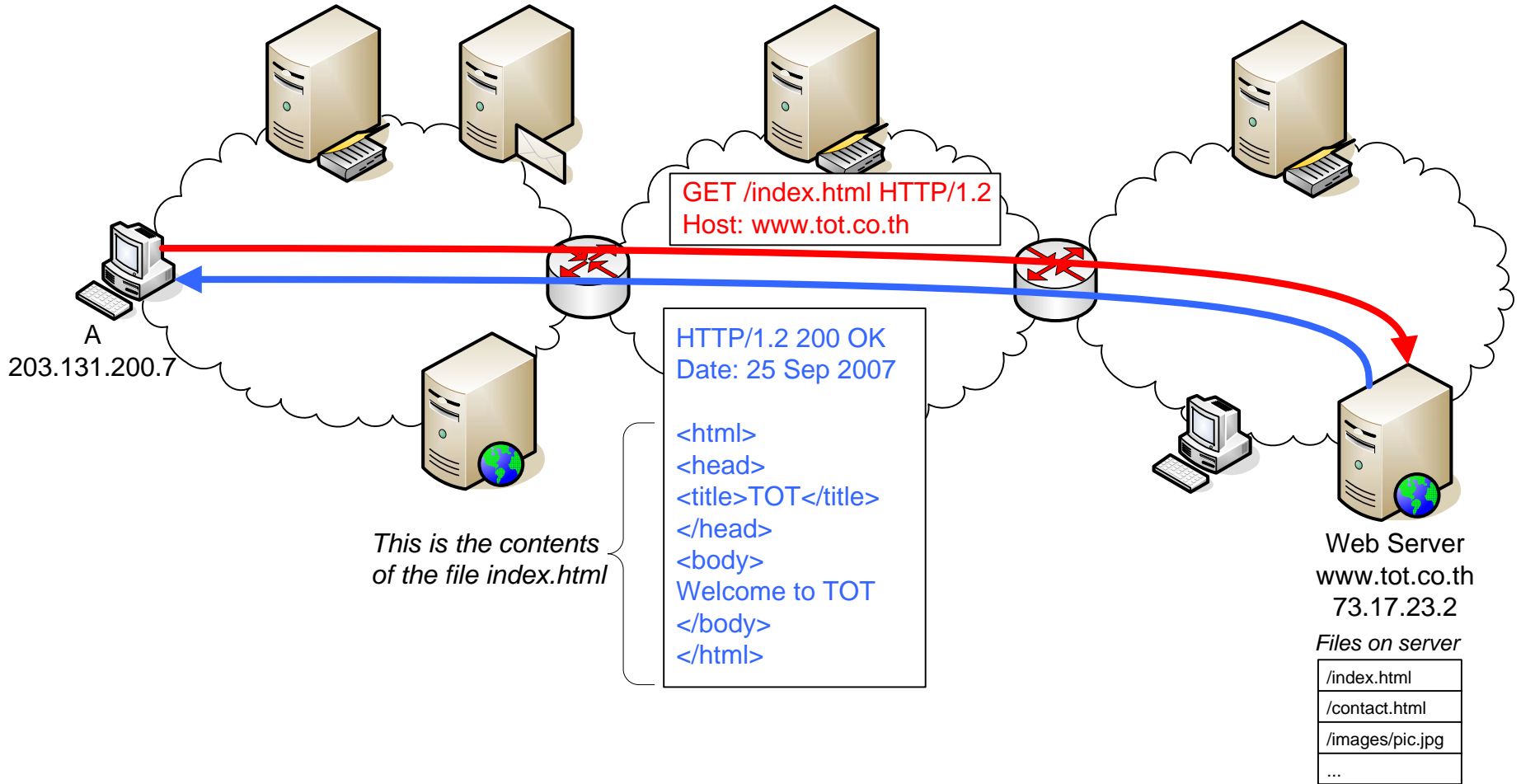
Note: the messages shown in the boxes 1, 2, 3 and 4 are not the real DNS messages; they just describe the meanings of the messages

HTTP Example

- Scenario:
 - Host A has browsed to the URL www.tot.co.th/index.html
- Assumptions:
 - Host A knows the IP address of www.tot.co.th (see DNS Example)
 - A TCP connection has been established from Host A to www.tot.co.th
- Actions:
 1. Host A sends a HTTP request message to the web server. The request asks to GET the file “/index.html”
 2. The TOT web server sends a HTTP response message, containing some header information and the actual content of the index.html file
- Result:
 - The web browser on Host A will display the index.html, of course formatting the output according to the HTML standard



HTTP Example

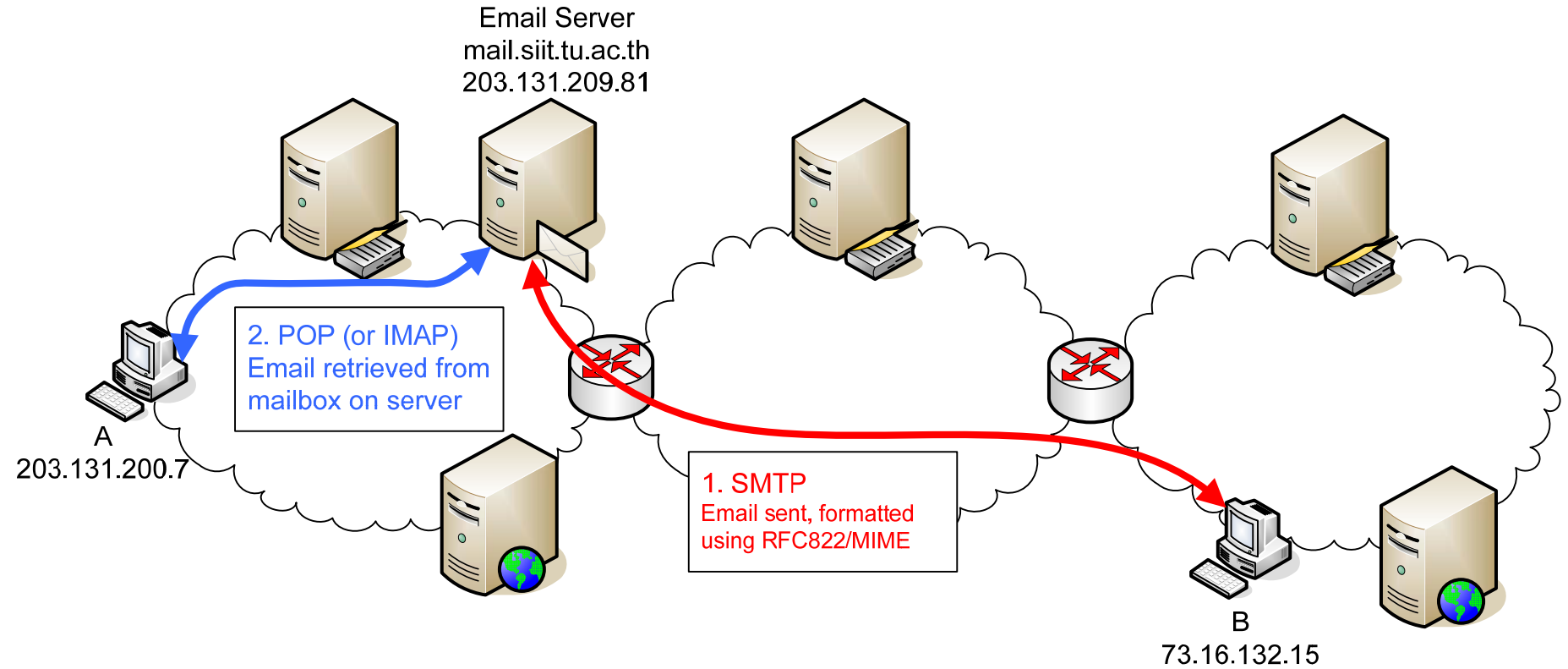


Email Example

- Scenario:
 - User on Host B sends an email to steve@siit.tu.ac.th
- Assumptions:
 - Host B knows the IP address of the email server at siit.tu.ac.th is 203.131.209.81 (DNS was used – although more complex than what we have covered in class!)
- Actions:
 1. After the user on Host B writes the email, they hit “Send” and Host B sends the email to the SIIT Email server using SMTP (the exact details of the SMTP exchange are not shown)
 2. The next day, user “steve” logs on to Host A and opens his email client. The email client uses POP (or alternatively IMAP) to retrieve the email to Host A (again, the details of POP/IMAP are not shown)
- Result:
 - An email was sent from Host B to Host A, via the email server



Email Example



Web Access Details

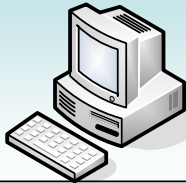
- The following slide shows the details of the packets sent between source and destination during the web request
 - At the Source (Host A):
 1. The web browser, using HTTP, generates the HTTP Request message and sends it to TCP
 2. TCP creates the TCP segment by adding a header and follows its rules (e.g. flow control, retransmission) before sending the segment to IP. Important fields in the TCP header are:
 1. Source Port: 50134 – chosen by the web browser/operating system
 2. Destination Port: 80 – default for a web server
 3. IP adds the IP header and sends the datagram to the data link layer. Important fields in the IP header are:
 1. Source IP: 203.131.200.7 – Host A's address
 2. Destination IP: 73.17.23.2 – Server's address, obtained via DNS
 3. Protocol number: 6 – TCP
 4. The data link layer and physical layer (implemented in the Network Interface Card) follow their rules to send the IP datagram
 - Across the network:
 5. The IP datagram passes through the networks, via routers (we don't show details here)



Web Access Details

- At the Destination (Server):
 6. The network interface card receives the bits and passes an IP datagram up to IP
 7. IP realises that this datagram is destined to this computer, so removes the IP header and passes the remaining up to TCP
 6. IP knows to send to TCP because of the Protocol number field (6 = TCP)
 8. TCP follows its rules (such as sending an ACK – not shown here), removes the TCP header and passes the remaining up to the Web Server application using HTTP
 6. TCP knows to send to the Web Server because the Destination Port is 80, the default for all web servers
 9. The Web Server reads the HTTP GET request and determines the file /index.html was requested. The server reads the /index.html file and returns it to the Web browser via a HTTP Response. The details of the response are shown on slide 10.

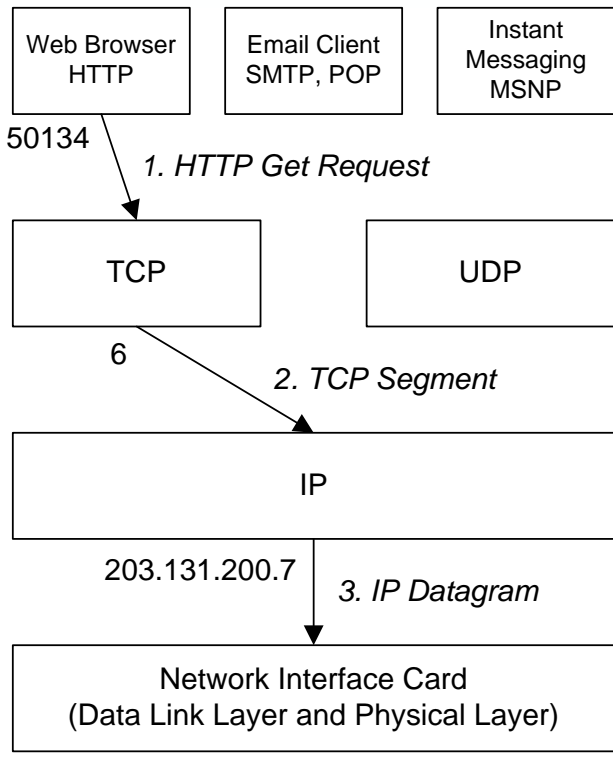




Host A
203.131.200.7



Server
www.tot.co.th
73.17.23.2



HTTP Get Request

GET /index.html ...
Host: www.tot.co.th

TCP Segment

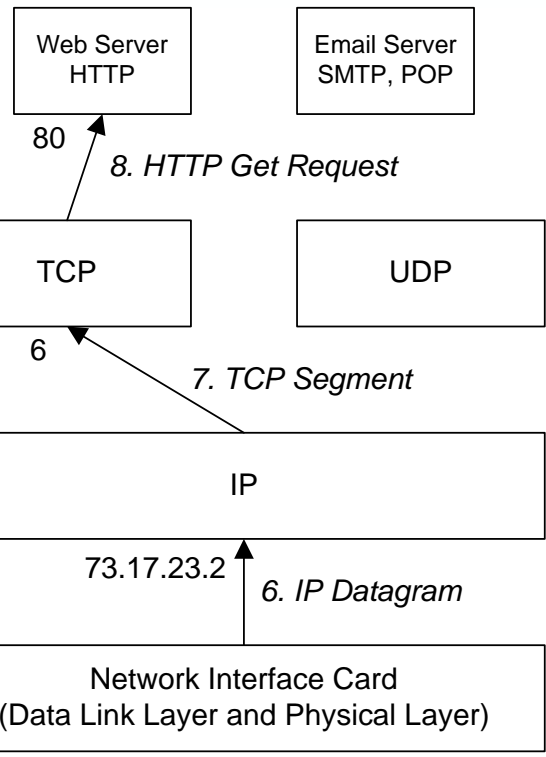
| | |
|-----|--|
| TCP | GET /index.html ... Host: www.tot.co.th |
|-----|--|

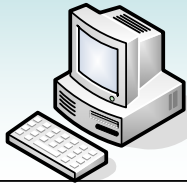
Source Port: 50134
Dest Port: 80

IP Datagram

| | | |
|----|-----|--|
| IP | TCP | GET /index.html ... Host: www.tot.co.th |
|----|-----|--|

Source IP: 203.131.200.7
Dest IP: 73.17.23.2
Protocol: 6 (TCP)

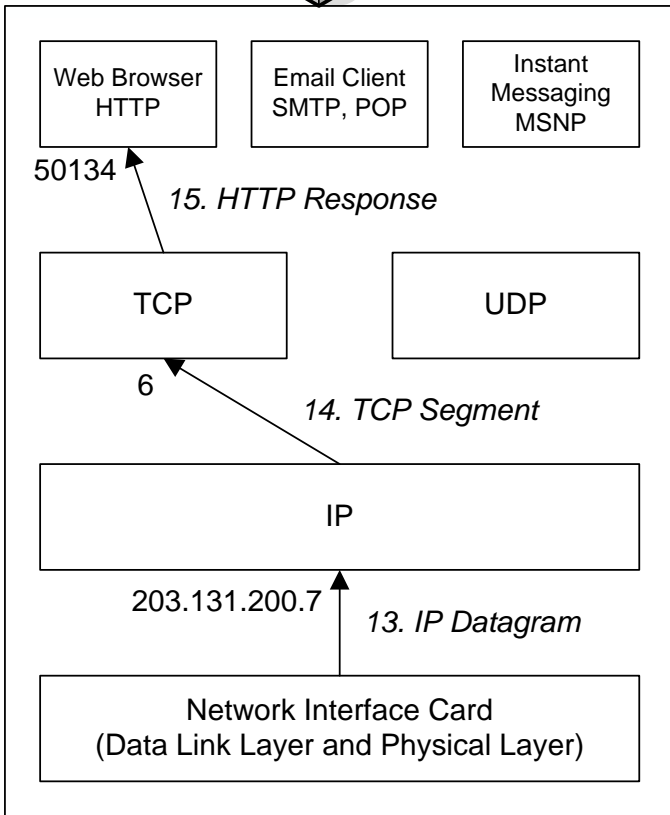




Host A
203.131.200.7



Server
www.tot.co.th
73.17.23.2



HTTP Response

```

HTTP/1.2 200 OK
Date: 25 Sep 2007

<html>
<head>
...

```

TCP Segment

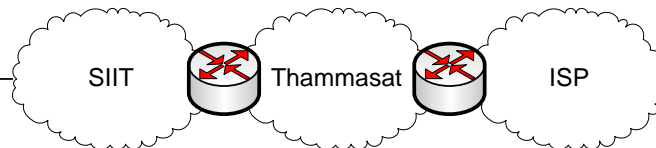
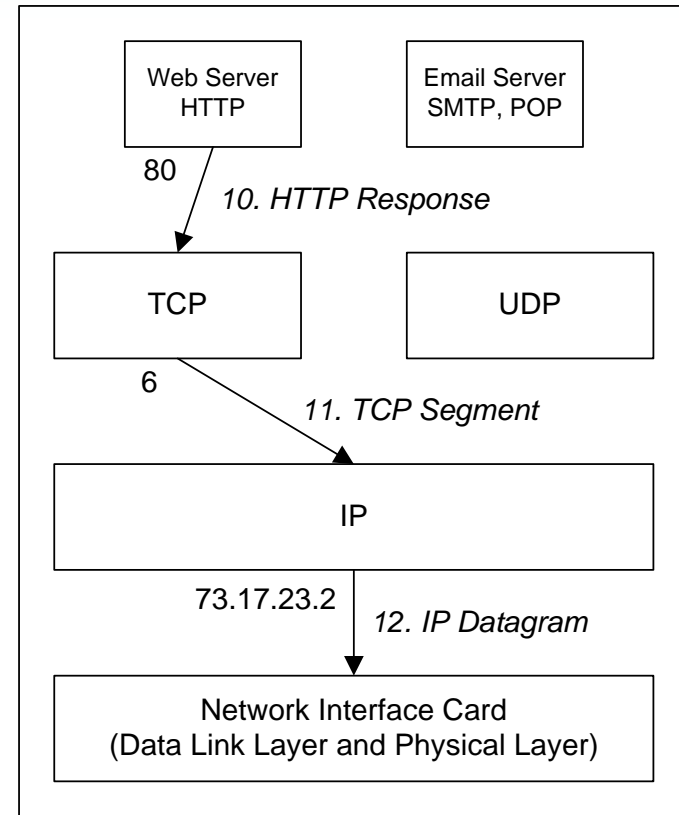
| | |
|-----|------------------------|
| TCP | HTTP/1.2 200 OK ... |
|-----|------------------------|

Source Port: 80
Dest Port: 50134

IP Datagram

| | | |
|----|-----|------------------------|
| IP | TCP | HTTP/1.2 200 OK ... |
|----|-----|------------------------|

Source IP: 73.17.23.2
Dest IP: 203.131.200.7
Protocol: 6 (TCP)

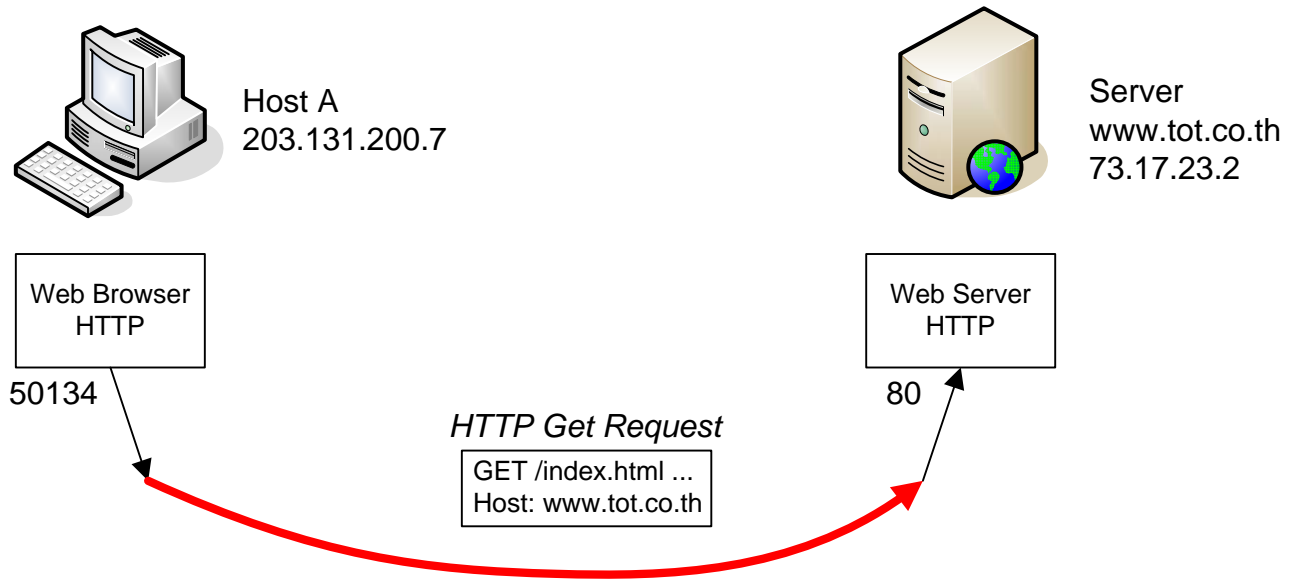


Protocols and Layers

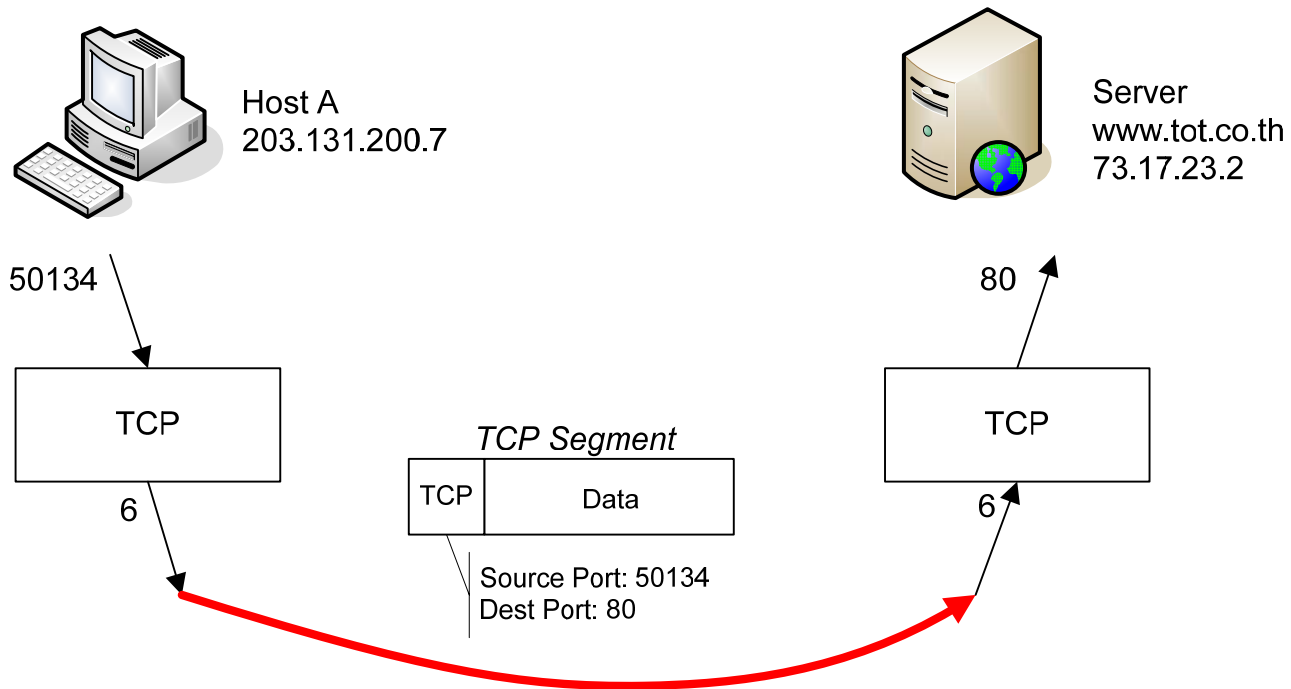
- The web access example illustrates the layering principle of communication networks
 - By separating functionality into layers, each layer can be relatively simple, but together all the layers achieve a very difficult communications task
 - The following three slides illustrate the viewpoint from the top three layers:
 - Application layer
 - From HTTP's point of view, a HTTP Request is sent from Web Browser to Web Server
 - » HTTP does not perform any retransmissions, routing, flow control, congestion control, etc.
 - Transport layer
 - From TCP's point of view, it receives Data from an application and sends a TCP segment to the destination computer, which passes the Data on to the destination application
 - » TCP does not perform any routing, and is not concerned with the purpose of the Data, i.e. the application
 - Network layer
 - From IP's point of view, it receives Data from a transport protocol and sends an IP datagram through the network, via routers, until it arrives at the destination computer, which passes the Data on to the destination transport protocol
 - » IP does not perform any retransmissions, flow control, and is not concerned with the purpose of the Data, i.e. the application or whether it is data or ack



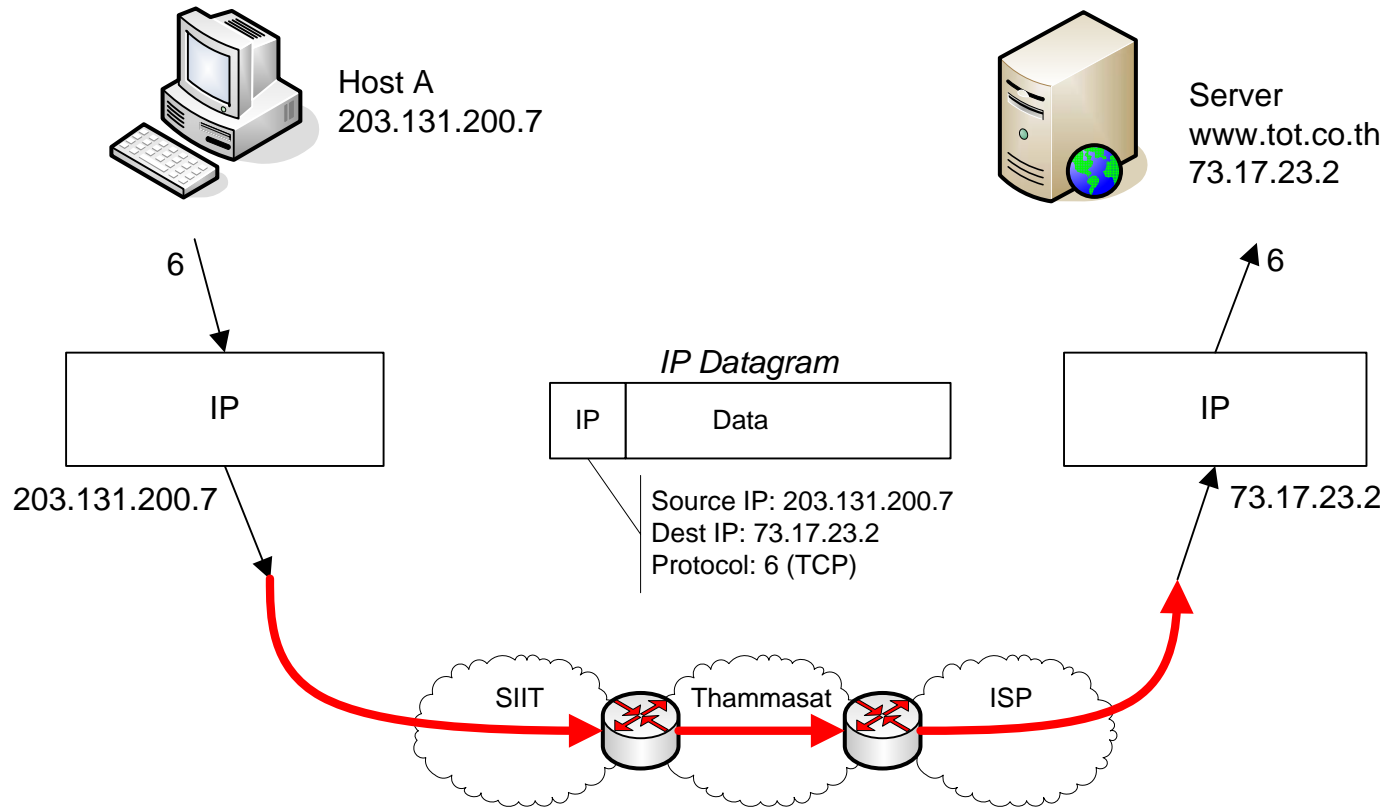
HTTP View Point



TCP View Point



IP View Point



Complete Data Transfer

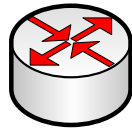
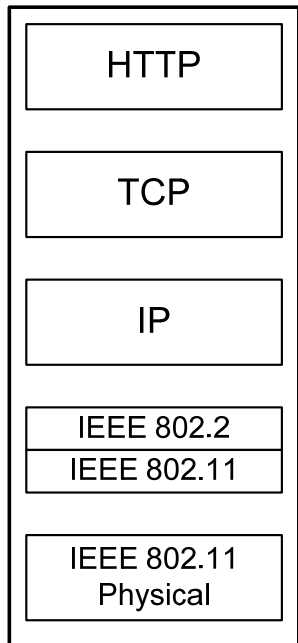
- The previous web access details focussed on HTTP, TCP and IP data transfer. However it did not show the complete data transfer, in particular, the following were missing:
 - TCP connection setup
 - TCP ACKs
 - Data passing through the routers
- The following 6 slides show these steps in detail
 - However, we do not show the final TCP ACK and TCP Connection Close
- The final slide summarises the sequence of messages sent in the data transfer



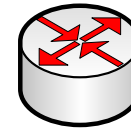
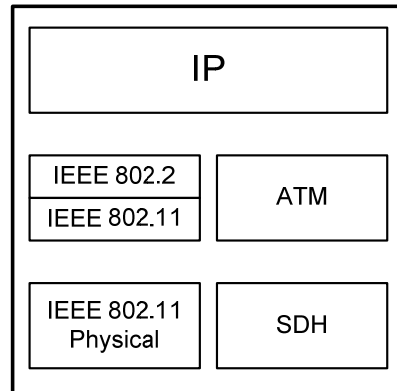
Protocol Stacks



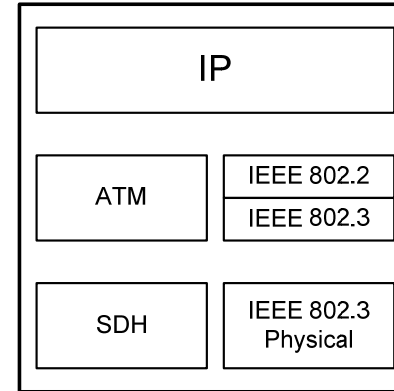
Host A



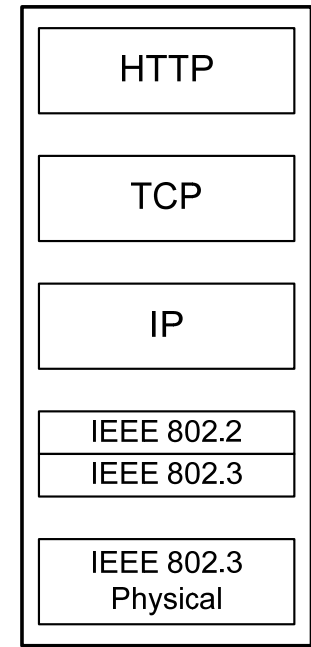
Router 1



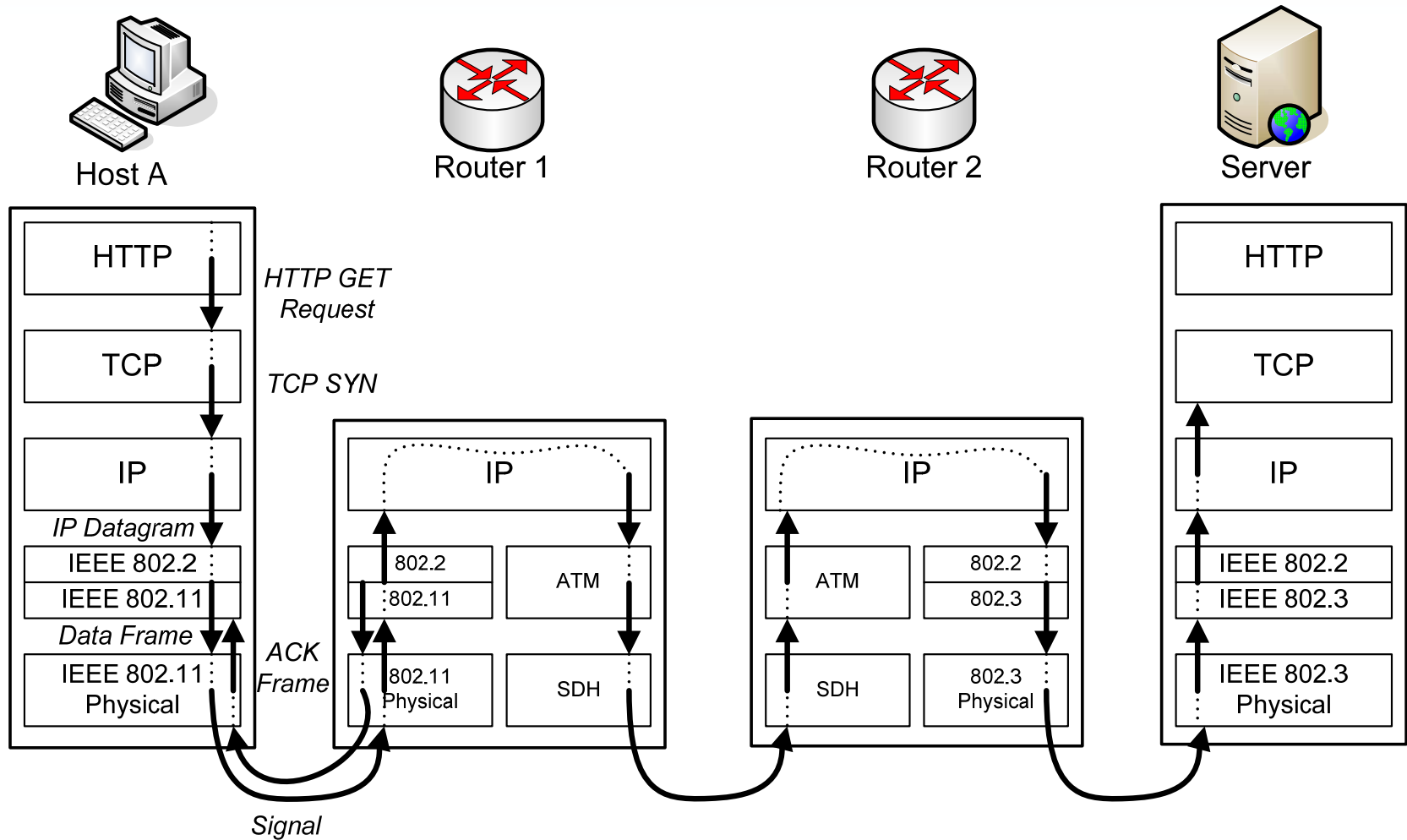
Router 2



Server



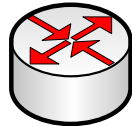
Step 1: HTTP Request



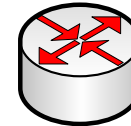
Step 2: TCP ACK+SYN



Host A



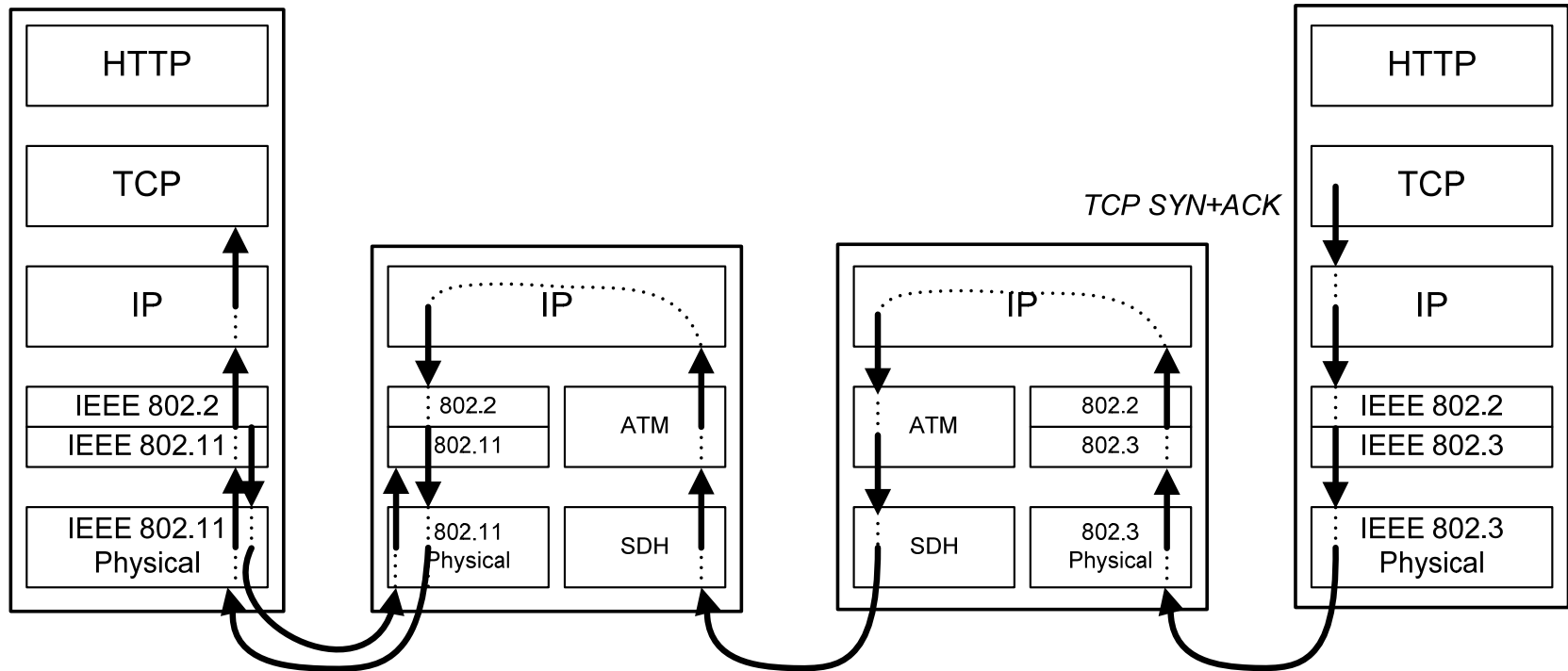
Router 1



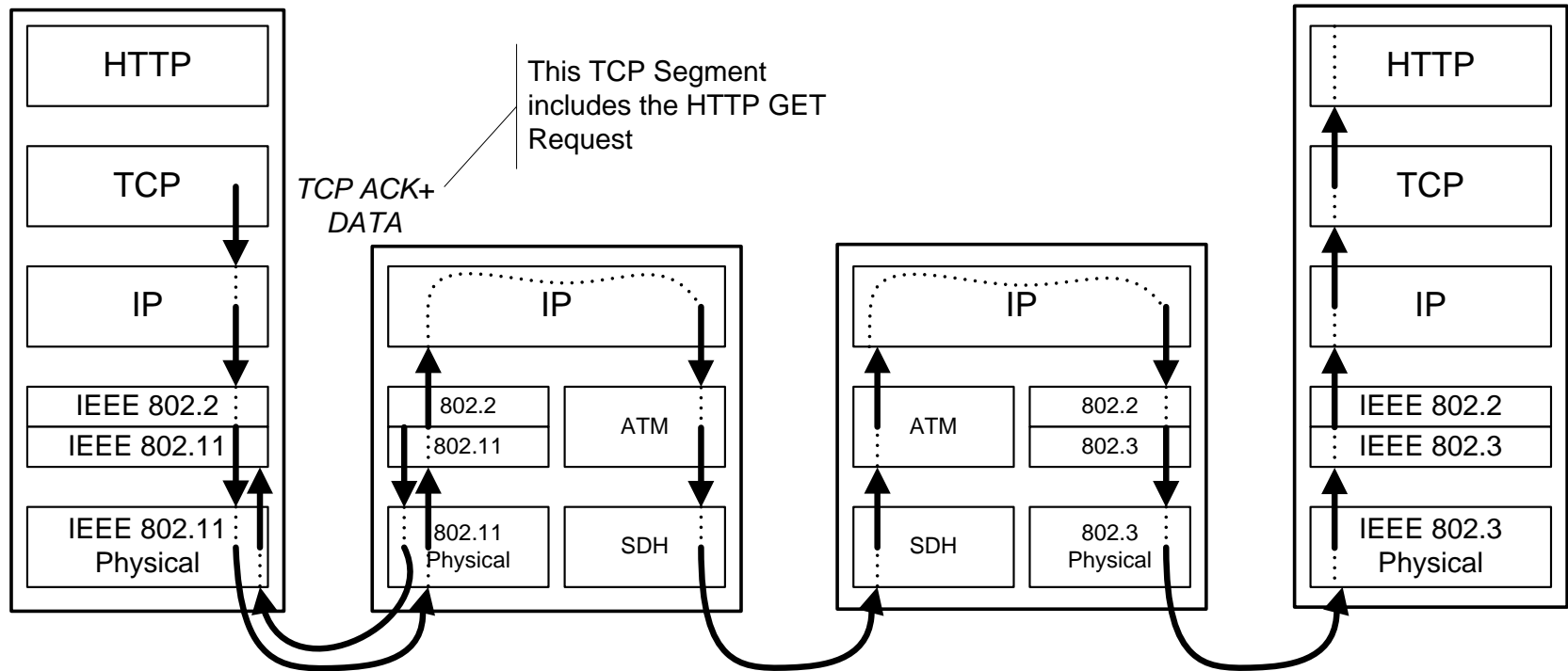
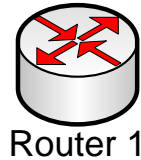
Router 2



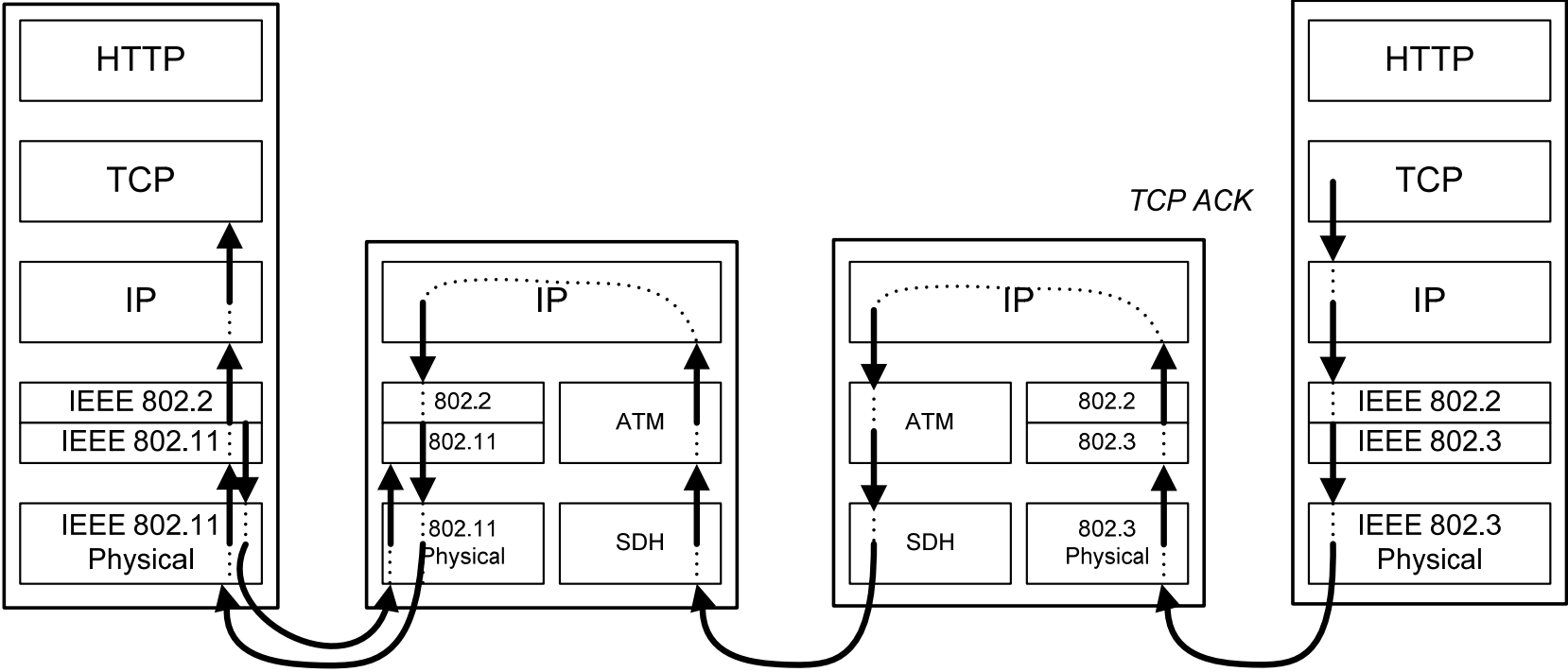
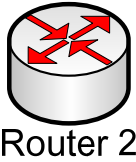
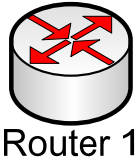
Server



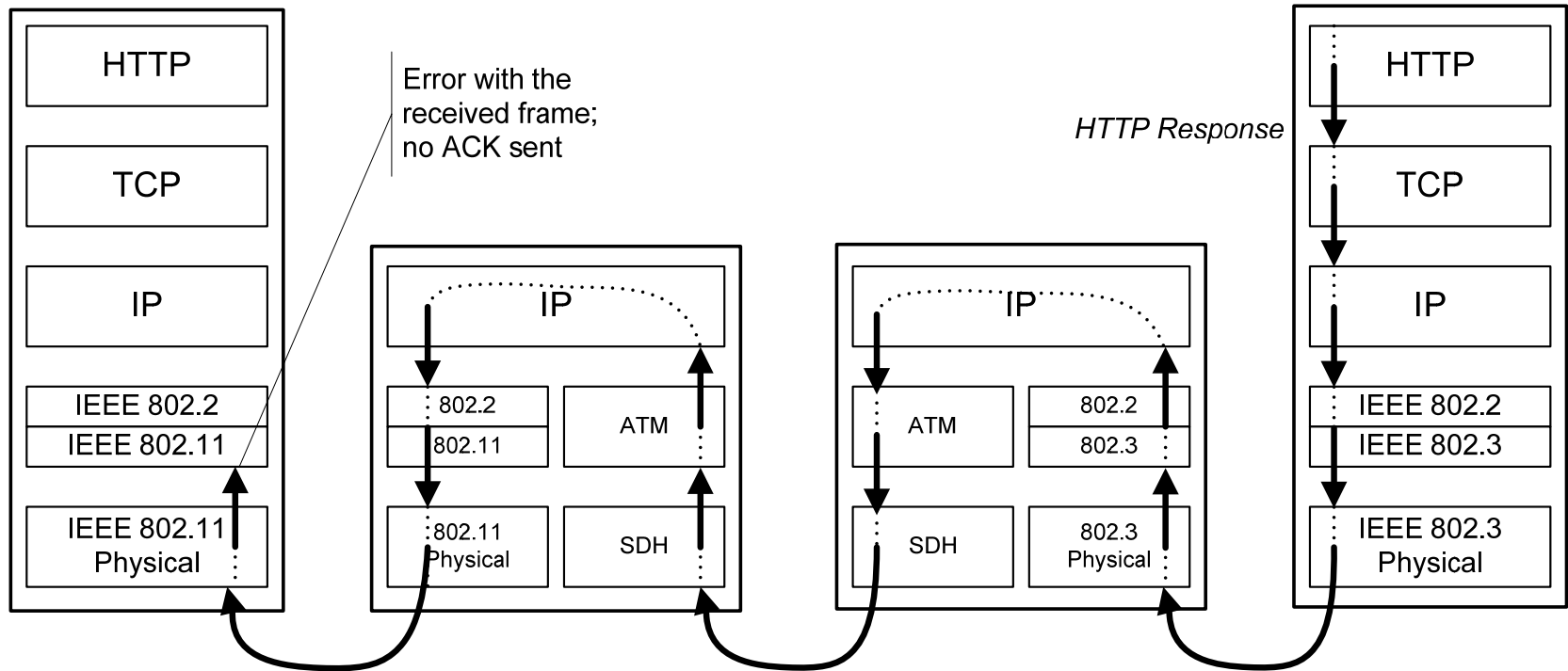
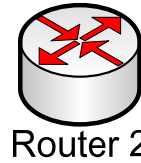
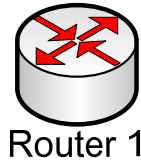
Step 3: TCP ACK+DATA



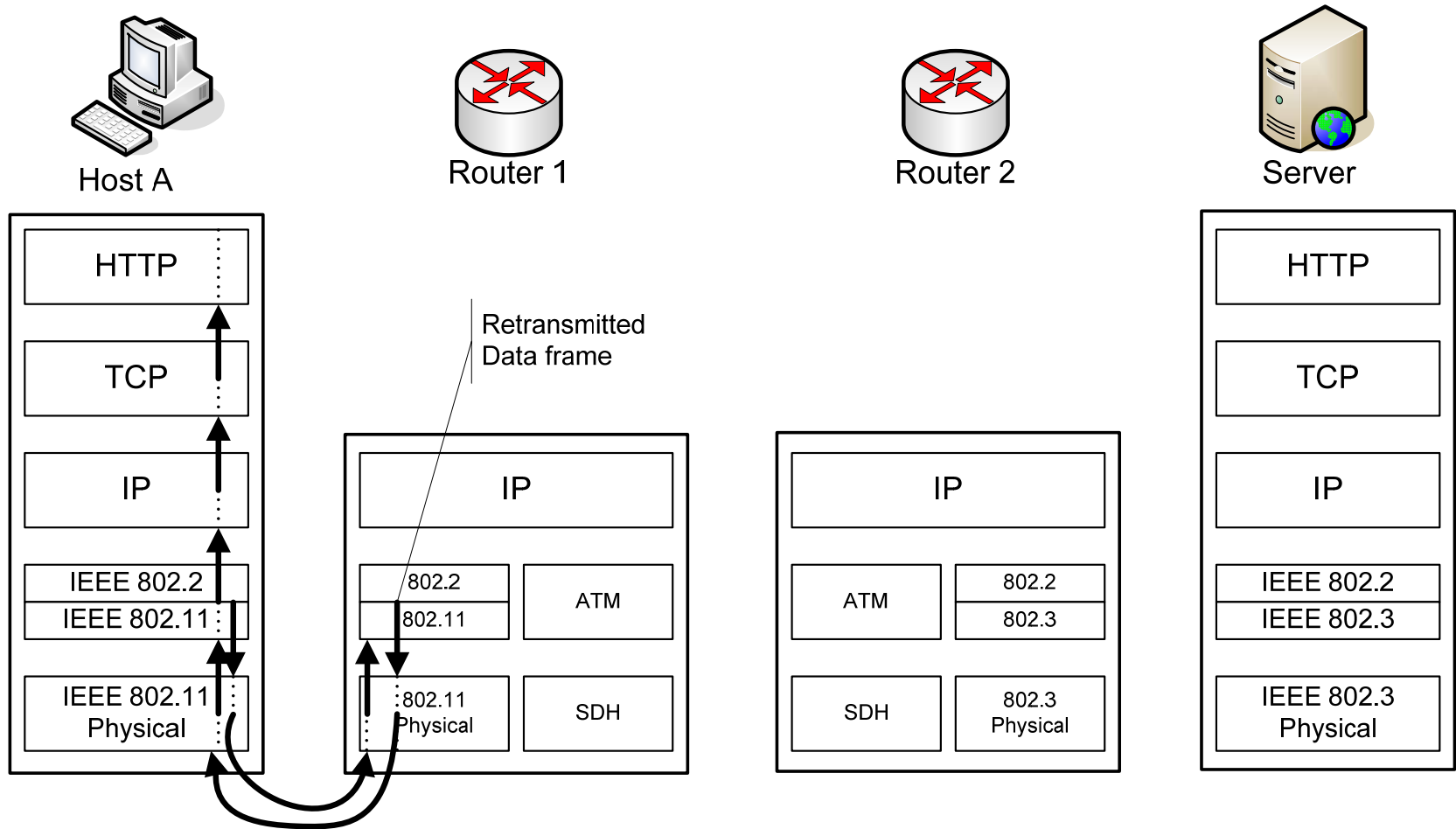
Step 4: TCP ACK



Step 5: HTTP Response



Step 6: Retransmitted Frame



Sequence of Messages

