Topic 8

Wireless Web

Outlines:

- HTTP and HTML
  - Why they are not suitable for the wireless web?
- WAP
- i-Mode

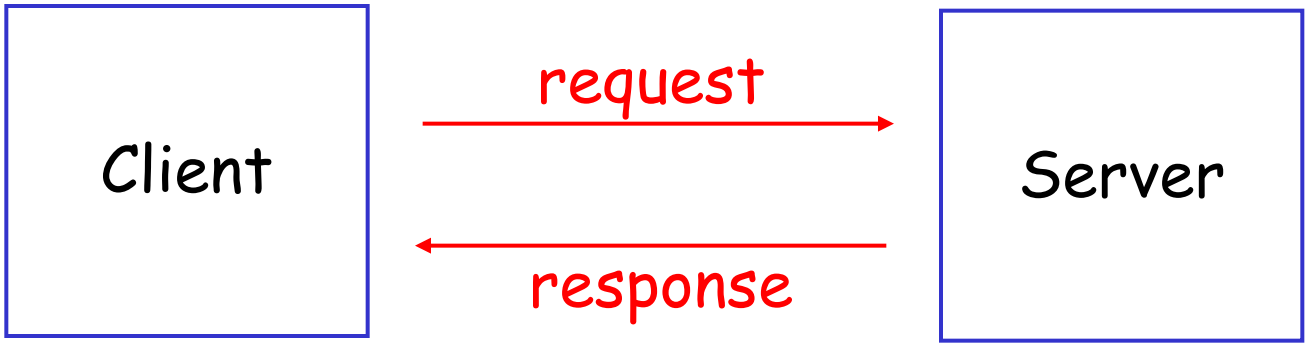
HTTP and HTML

World Wide Web

- Application-Layer Protocol:
  - HyperText Transfer Protocol (HTTP)
  - Two versions http/1.0, http/1.1
  - Currently, http1.1 is used by most implementations
- Language
  - HyperText Markup Language (HTML)

HTTP Transaction

- An HTTP transaction consists of
  - an HTTP request issued by a client, and
  - an HTTP response sent from a server.



- All HTTP transactions are stateless.
  - server maintains no information about past client requests

Example: HTTP messages

Request to Server, port 80

```
GET / HTTP/1.0
```

HTTP messages are in ASCII
(a human-readable format)

Response from Server

```
HTTP/1.1 200 OK
Date: Tue, 12 Nov 2004 06:33:50 GMT
Server: Apache/1.3.20 (Unix)
Last-Modified: Fri, 05 Oct 2003 06:10:55 GMT
Content-Length: 195
Content-Type: text/html
Connection: close

<HTML>
<HEAD><TITLE> Tezpur University Homepage </TITLE></HEAD>
<BODY><IMG SRC="abc.gif">
<IMG SRC="www.tezu.ernet.in/tezpur_univ.jpg"> </BODY>
</HTML>
```

Trying out http for yourself

1. Telnet to your favorite Web server:

```
telnet www.tezu.ernet.in 80
```

Opens TCP connection to port 80 (default http server port) Anything typed in sent to port 80 at www.tezu.ernet.in

2. Type in a GET http request:

```
GET / HTTP/1.0
```

By typing this in (hit carriage return twice), you send this minimal (but complete) GET request to http server

3. Look at response message sent by http server!

TCP Transport Service

- HTTP uses TCP:
  1. client initiates TCP connection to server, port 80
  2. server accepts TCP connection from client
  3. http messages exchanged between browser (http client) and Web server (http server)
  4. TCP connection closed

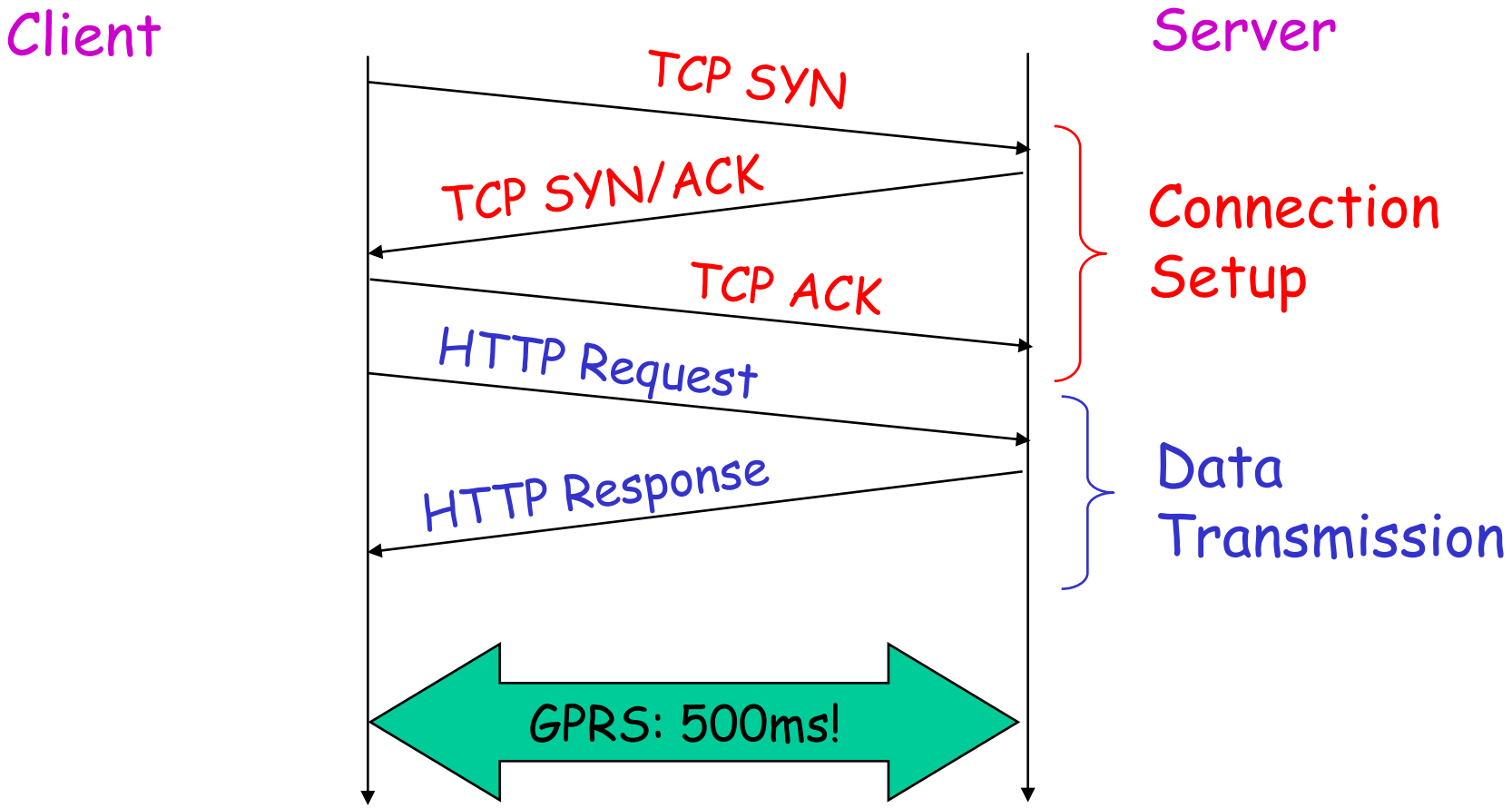
Problems with HTTP

- Protocol headers are unnecessarily large
  - Headers are readable for humans and in ASCII.
- Protocol headers are sometimes redundant
  - Many information fields are transferred over and over again with each request because HTTP is stateless

Problems with HTTP

- Uncompressed content transfer
  - Unless the applications compress content (e.g. GIF and JPEG images).
- Using TCP
  - huge overhead per request (3-way-handshake) compared with the content (e.g. a GET request)
  - slow-start problem (mentioned in the previous lecture on TCP for Wireless Networks)

TCP Connection Setup Overhead



Problems with HTTP

- DNS lookup by client
  - Each time a browser reads a hyperlink reference to a new server it has to resolve the logical name into an IP address.
  - This requires an additional request to a DNS server over the wireless link.
- Caching: Although useful in many cases, caching is disabled/not possible in many cases.
- POSTing: Sending content from a client to a server can cause additional problems.

HTML and Mobile Devices

- HTML
  - designed for computers with high-resolution color display, mouse, hard disk
  - typically, web pages optimized for design, not for communication
- Mobile devices
  - often only small, low-resolution displays, very limited input interfaces
  - e.g., without additional mechanisms, large high-resolution pictures would be transferred to a mobile phone with a low-resolution display, causing high costs

Problems with HTML

- It delivers far too much information to be adequately handled by mobile devices.
- Plug-ins for proprietary content format are not available for all platforms.

Approaches for Wireless Access

- Image Scaling: Picture with a true color, high-resolution can be scaled down to fewer colors, lower resolution, or just to the title of the picture.
- Content Transformation: Special converter to translate documents to plain text
- Content Extraction/Semantic compression: Keywords, Abstract from documents
- Special language and protocols: *replace HTTP and HTML with HDTP and HDML (???)*
- Push Technologies for content delivery from server
- Integrated into servers or Gateways

System Architectures for Wireless Access

- Integration of Caching into Web Browser: Only caching of already transferred contents. Does not perform pre-fetching.
- Additional application supporting browsing: Support pre-fetching, caching and disconnected service. Not transparent to browser. (e.g. WebWhacker)
- Client Proxy (Transparent): Pre-fetch and cache contents according to some strategies. (e.g. CaubWeb)
- Network Proxy: for adaptive content transformation, pre-fetch and cache contents.(e.g. TranSend, Digestor)
- Client and Network Proxy: better coordination between Client and Network proxies for pre-fetching, caching. .(e.g. WebExpress)

System Architectures for Wireless Access

- Client and Network Proxies with special transmission protocol: Content transfer can be further optimized. Online compression, replacement of HTTP and TCP with protocols better adapted to mobility and wireless access of the client. (E.g. Mowgli data service between client and network proxies)

WAP

(Wireless Application Protocol)

Wireless Application Protocol (WAP)

- A standard for bringing Internet content and services to mobile devices.

- WAP is *not* a service or a product but a protocol for delivering applications.

WAP Forum

- To avoid many islands of incompatible solutions, WAP Forum was founded in June 1997 by Ericsson, Motorola, Nokia, and Unwired Planet (renamed to Openwave).

WAP 1.0	**Apr 1998**
WAP 1.1	**May 1999**
WAP 1.2	**Nov 1999**
WAP 1.2.1	**Jun 2000**
WAP 2.0	**July 2001**

We will focus on WAP 1.x first.

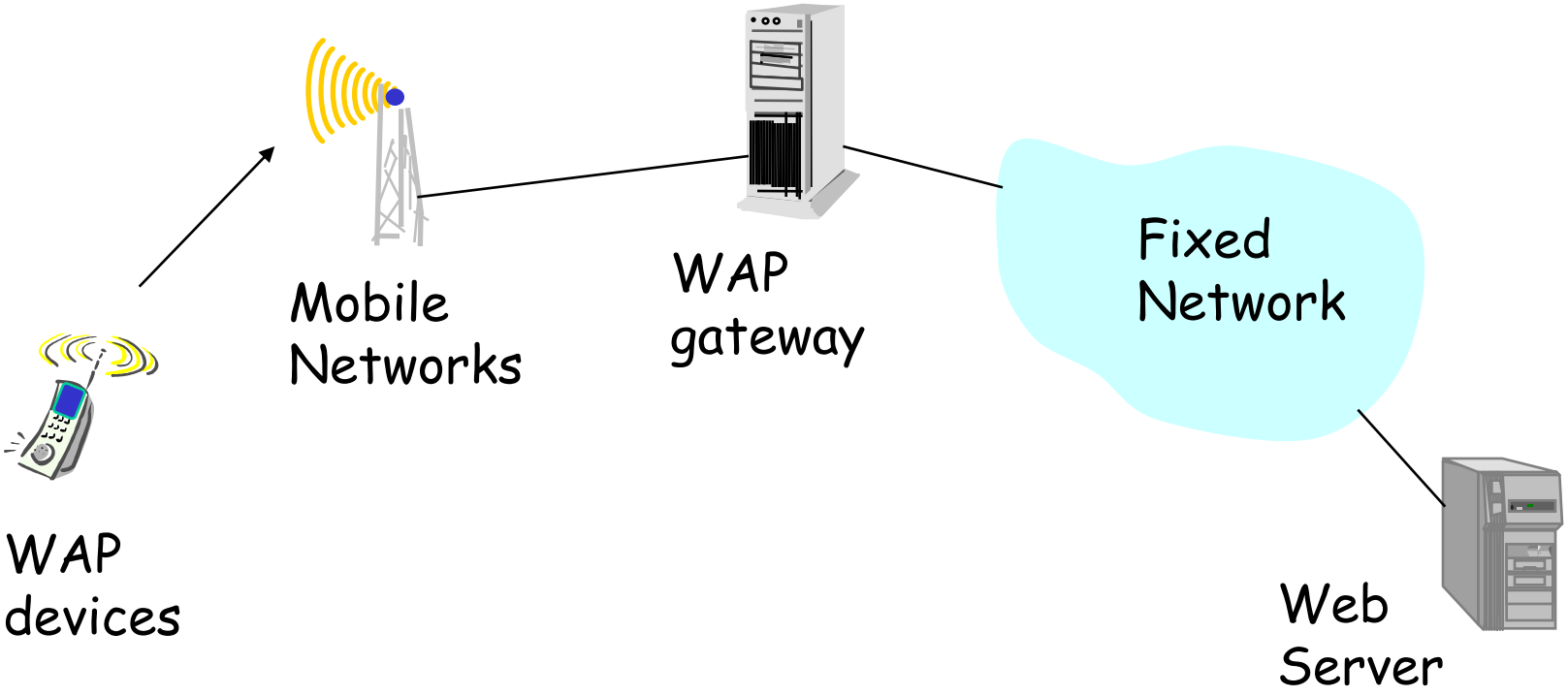
Key Aspects of WAP

- It can run on any network
  - e.g. GSM, CDMA, 3G networks
- It can run on any kind of device
  - e.g. cell phones, PDAs, and laptops.
  - achieves device-independence through the Wireless Application Environment (WAE).
- WAP defines a new format, the Wireless markup Language (WML)
  - designed for efficient content delivery

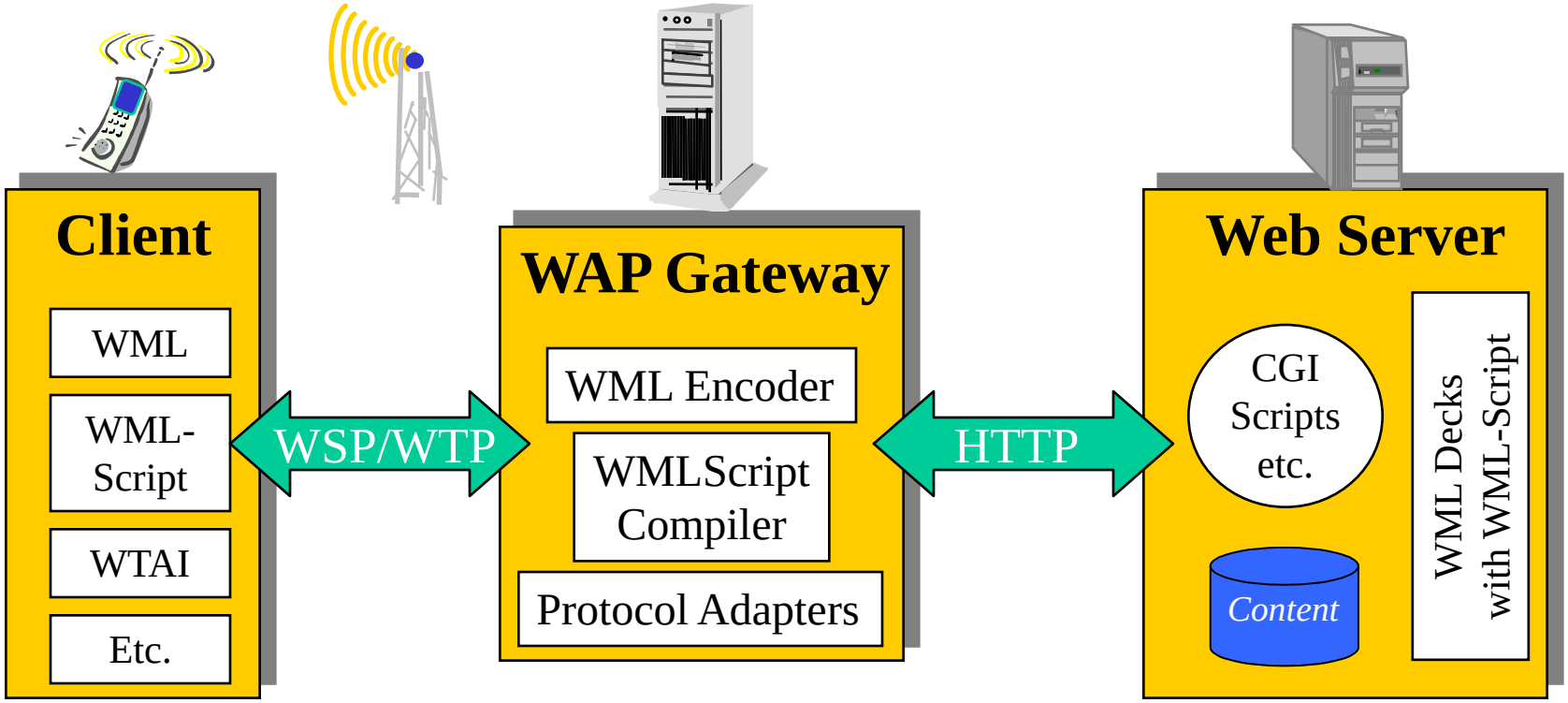
WAP-Enabled Devices

- WAP-enabled devices consists of
  - A micro-browser
    - An embedded software that enables users to view information
  - A method for users to input data
    - e.g. number buttons on a mobile phone

Entities in a WAP Model

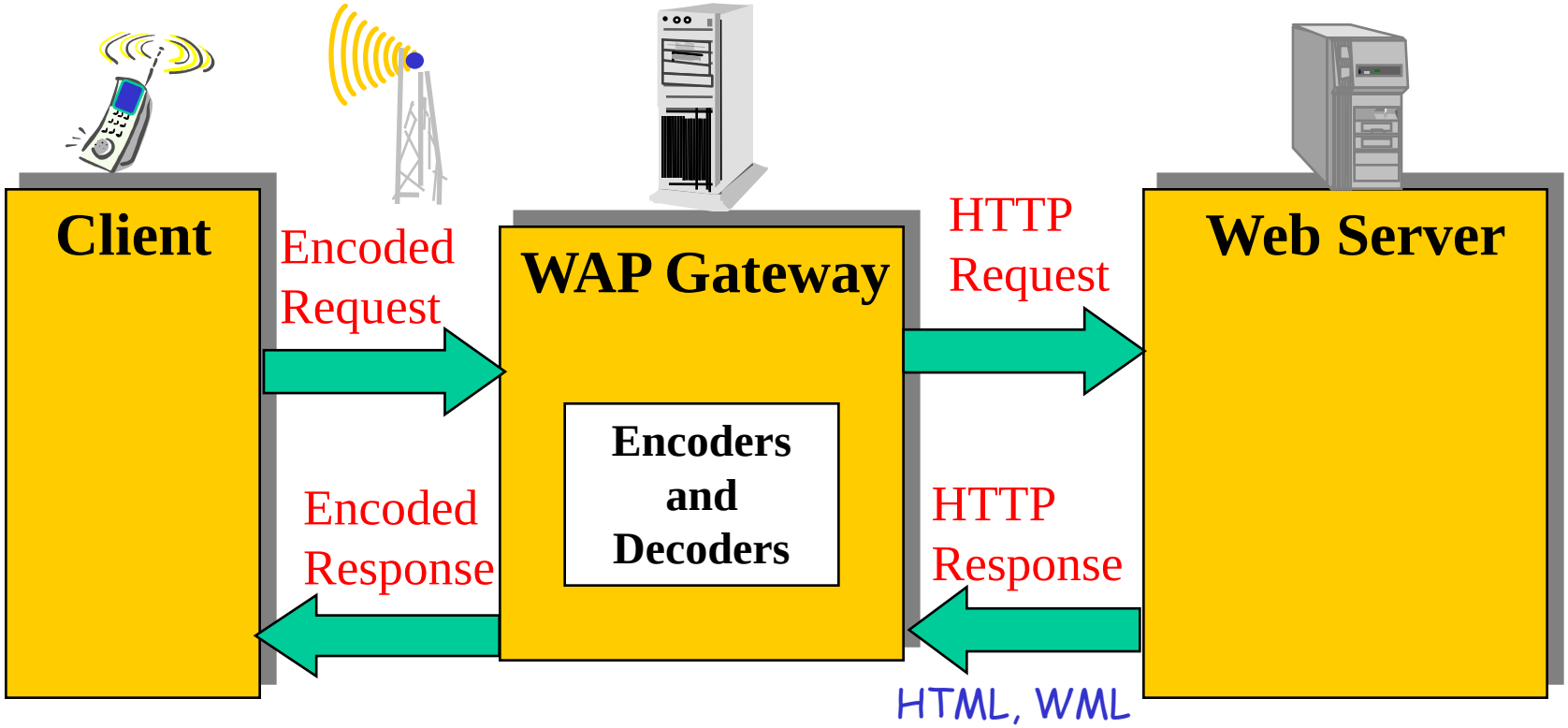


WAP Architecture



In principle, HTML is translated to WML for transmission to the client.

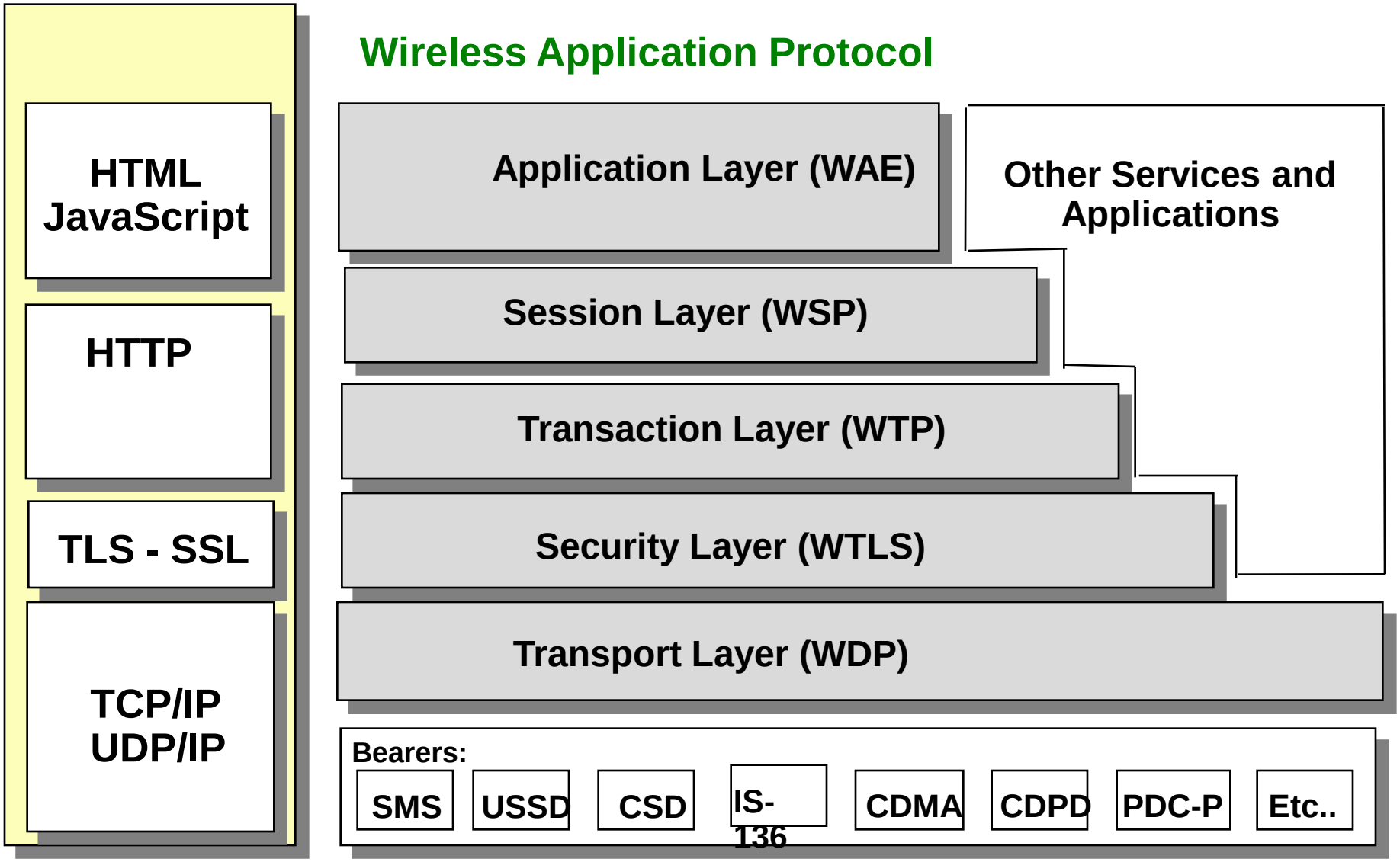
In practice, however, …



Protocol Stack

- Five Layers
  - Transport, security, transaction, session and application
- Applications can use only a part of the architecture.
  - e.g. if an application does not require security, it can use directly the service of the transport layer

Internet and WAP Protocol Stacks



1. Wireless Datagram Protocol (WDP)

- Offers a consistent datagram transport service independent of the underlying bearer (GSM, CDMA, etc.).

- Offers more or less the same services UDP does.
  - e.g. application addressing by port numbers.

2. Wireless Transport Layer Security (WTLS)

- Provides security features
  - Privacy: data cannot be understood by intermediate parties
  - Data integrity: data has not been modified
  - Authentication: author of a message can be identified
- Cryptographic algorithms optimized for mobile devices of low processing power and limited memory.

2. Wireless Transport Layer Security (WTLS) (cont'd….)

- Provides security between a WAP-enabled device and a WAP gateway.
- Additional mechanisms are needed for end-to-end security.
  - e.g. a user accesses his bank account using WAP.

3. Wireless Transaction Protocol (WTP)

- Support for transaction-oriented services (e.g. web browsing)
  - A transaction is defined as a request with its response.
- A lightweight protocol suitable for implementation in mobile devices.

3. Wireless Transaction Protocol (WTP)(cont'd….)

- Three classes of transaction service offered to upper layer:
  - Class 0: Unreliable one-way request
    - No retransmission if the sent message is lost
  - Class 1: Reliable one-way request
    - Message is resent if no ACK received.
  - Class 2: Reliable two-way request-response
    - A data request is sent and a result is received which finally is ACK'ed by the sender.

4. Wireless Session Protocol (WSP)

- Capabilities of suspending and resuming a session
  - Assume a mobile device is disconnected. The user is able to continue operation at exactly the point where the device was switched off.
- Content encoding
  - Defines the efficient binary encoding for the content it transfers.

5. Wireless Application Environment (WAE)

- Wireless Markup Language (WML)
  - Analogy to HTML
  - It can be binary encoded by the WAP gateway in order to save bandwidth
- WMLScript
  - Enhance services written in WML
  - Similar to the role of JavaScript

Wireless Markup Language (WML)

- A WML document is made up multiple cards.
  - Each card represents a screen of information.
- Cards can be grouped together in a deck.
  - A WML deck is similar to an HTML page.
- Web servers deliver web pages one at a time but WAP delivers a deck of related cards.
  - There is no waiting for the next screen (card) to display, which differs from the web, where clicking on a new link typically means waiting for the server to deliver that page.

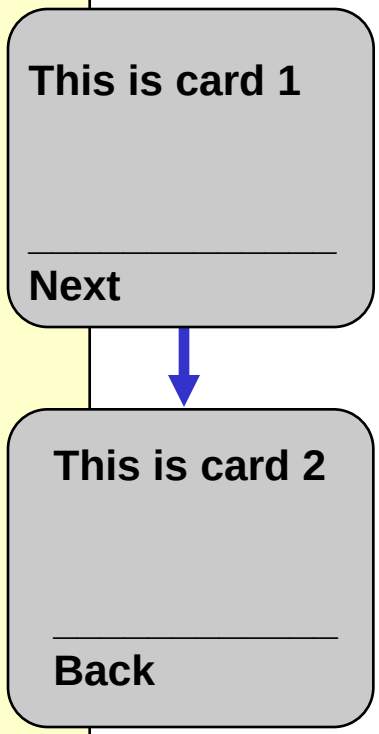
Example: A Deck of Cards

```
<WML>

    <CARD id="card1">
        <DO TYPE="ACCEPT" LABEL="Next">
            <GO href="#card2"/>
        </DO>
        <p> This is card 1</p>
    </CARD>


    <CARD id="card2">
        <DO TYPE ="ACCEPT" LABEL="Back">
            <GO href="#card1"/>
        </DO>
        <p> This is card 2</p>
    </CARD>

</WML>
```



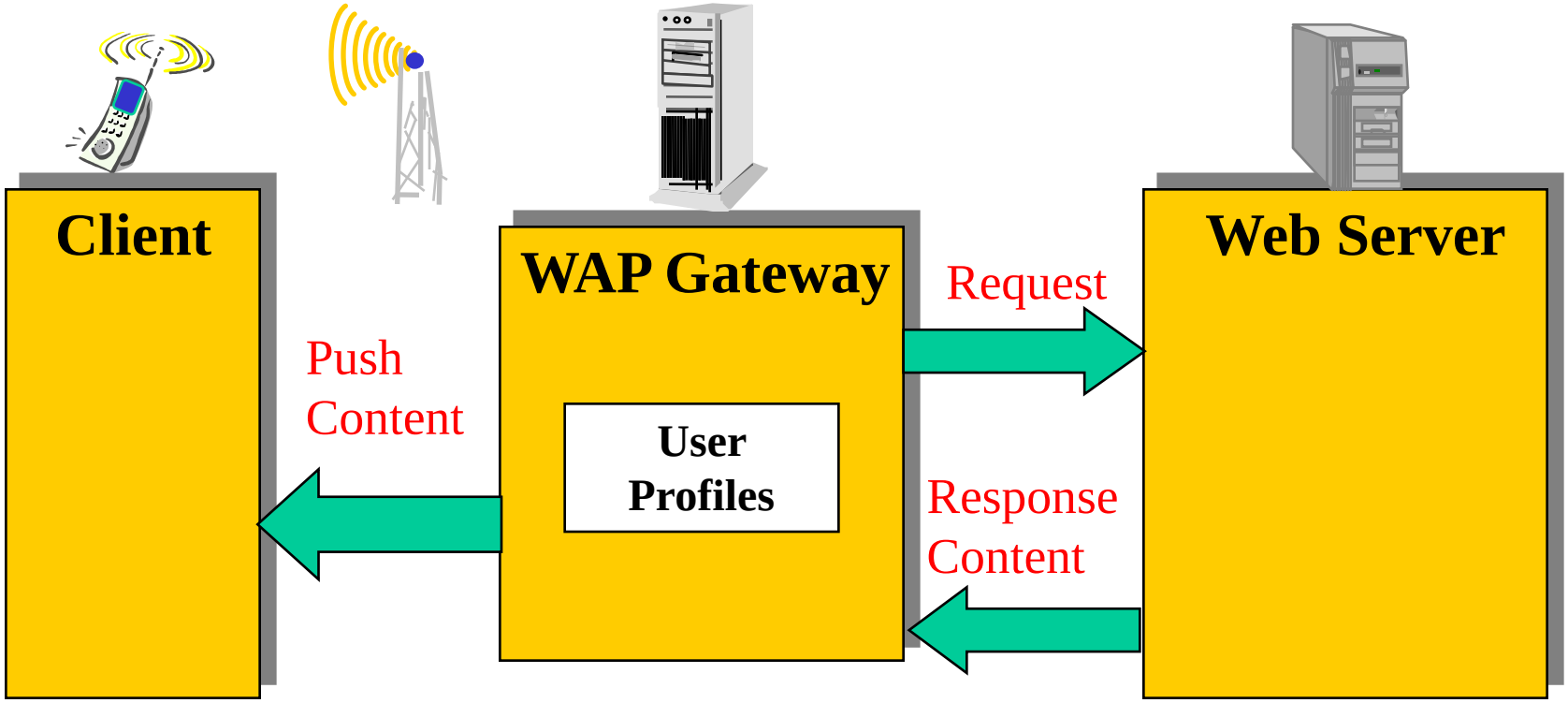
WMLScript

- A complement to WML
- Provides a general scripting capability (like JavaScript)
  - Example: before user input is sent to a server, WMLScript can check the validity and save bandwidth and latency in case of an error.

Push Architecture

- WAP 1.2 introduces a push architecture (2000)
- Clients pulling content from servers are typical for today's web.
- In a push context, the server initiates the message transfer, *not* the client.
  - Useful for services such as online auctions or stock trading, where it's important for users to receive information whenever something interesting happens.

WAP Push Architecture



WAP 2.0

- WAP 2.0 was published by the WAP Forum in July 2001.
- Backward compatible with WAP 1.x.
- Disadvantage of WAP 1.x
  - It uses WML, which is incompatible with HTML.
  - Developers have to create content independently of HTML.
- WAP 2.0 supports XHTML.

WAP 2.0

- It uses a cut-down version of XHTML with end-to-end HTTP, dropping the gateway and custom protocol suite used to communicate with it.
- Spin-off technologies, such as Multimedia Messaging Service (MMS), a combination of WAP and SMS, have further driven the protocol.
- However, today, WAP use has largely disappeared. Most modern handset internet browsers now support full HTML, CSS, and most of Javascript, and do not need to use any kind of WAP markup for webpage compatibility.
- The list of handsets supporting HTML is extensive, and includes all Android handsets, all Blackberry devices, all versions of the iPhone handset, all devices running Windows Phone, and many Nokia handsets.

What is XHTML?

- Extensible HyperText Markup Language (XHTML) is the World Wide Web Consortium (W3C)'s standardized web document format.
- XHTML is based on XML and is ultimately meant to replace HTML as the accepted Web page format.
- This enable WAP to smoothly integrate into mainstream web technology.

XHTML Basic

- The W3C has defined XHTML profiles.
- WAP 2.0 developers use the XHTML Basic profile
  - A subset of all available XHTML features.
- Developers can use XHTML Basic to create fully featured web sites, but it's also simple enough to scale down to a mobile device's limited resources.

i-Mode

i-Mode Service

- Introduced in Japan by NTT DoCoMo in Feb 1999.
- A big success with more than 30 million users.
- Example services:
  - email, reserve tickets, find a restaurant, check bank balance, transfer money, view train schedules and city maps, picture exchange.

Compact HTML (cHTML)

- i-mode uses cHTML
- cHTML is a smaller, slightly modified version of HTML
  - Optimized for low bandwidth and limited client resources.
- Similar to HTML
  - Easy to create
  - Viewable using common Web browsers.

Example

```
<HTML>
    <HEAD>
    <TITLE> My cHTML Page </TITLE>
    </HEAD>

    <BODY>
        <IMG SRC="icon.gif">
        <P> This is some text. </P>
        <A HREF="index.html"> A Link </A>
    </BODY>

</HTML>
```

Parameters and Tags

- cHTML has its own parameters and tags.
- Example 1:
  - `<a href="next.html" accesskey="1"> Next </a>`
  - Users press "1" on their keypad to go to the Next link.
- Example 2:
  - `<a href="tel:555-0123"> Call Frank </a>`
  - Users can open a voice connection to the phone number provided from a cHTML page.

i-Mode vs WAP

- i-Mode is a big success.
- WAP 1.x is usually considered a failure.

- Why?
  - Many reasons…
  - We take a technological point of view…

WML vs cHTML

- WAP 1.x uses WML
  - incompatible with HTML.
  - difficult to create.
- i-mode uses cHTML
  - a subset of HTML
  - easy to create.

Circuit Switching vs Packet Switching

- WAP started with GSM (circuit-switched)
  - A connection had either to be permanently open to support web browsing (which is expensive) or a new connection has to be established each time content was loaded (which is time consuming).
- i-mode is packet-switched
  - users can take as much time as they want to look at a web site, and they are charged only for the amount of data transferred.

References

- J. Schiller, *Mobile communications*, PEA, 2nd edition, 2004.
- F. P. Coyle, *Wireless Web: a manager's guide*, Addison-Wesley, 2001.
- K. Read and F. Maurer, "Developing mobile wireless applications," pp. 81-86, *IEEE Internet Computing*, Jan/Feb 2003.