UNIT – V MOBILE TRANSPORT LAYER AND SUPPORT FOR MOBILITY


TRADITIONAL TCP

Mechanisms that influence the efficiency of TCP in a mobile environment

- Congestion control
- Slow start
- Fast retransmit/fast recovery
- Implications on mobility

Congestion control

- TCP has been designed for fixed networks with fixed end-systems
- Hardware and software are mature enough to ensure reliability of data
- The probable reason for a packet loss in a fixed network is a temporary overload some point in the transmission path, i.e., a state of congestion at a node
- The packet buffers of a router are filled and the router cannot forward the packets fast enough
- The only thing a router can do in this situation is to drop packets
- The sender notices the missing acknowledgement for the lost packet and assumes a packet loss due to congestion
- Retransmitting the missing packet and continuing at full sending rate would now be unwise, as this might only increase the congestion.

Slow start

- The behavior TCP shows after the detection of congestion is called slow start
- The sender always calculates a congestion window for a receiver.
- The start size of the congestion window is one segment (TCP packet).
- This scheme doubles the congestion window every time the acknowledgements come back, which takes one round trip time (RTT) like 1, 2, 4, 8 etc.
- This is called the exponential growth of the congestion window in the slow start mechanism.
- The exponential growth stops at the congestion threshold.
- As soon as the congestion window reaches the congestion threshold, further increase of the transmission rate is only linear by adding 1 to the congestion window each time the acknowledgements come back
- Linear increase continues until a time-out at the sender occurs due to a missing acknowledgement, or until the sender detects a gap in transmitted data
  - the sender sets the congestion threshold to half of the current congestion window
  - The congestion window itself is set to one segment

Fast retransmit/fast recovery

Fast Retransmit

- a receiver sends acknowledgements only if it receives any packets from the sender.
- Receiving acknowledgements from a receiver also shows that the receiver continuously receives something from the sender.
- The gap in the packet stream is not due to severe congestion, but a simple packet loss due to a transmission error.
- The sender can now retransmit the missing packet(s) before the timer expires.
- This behavior is called fast retransmit
Fast Recovery
- The receipt of acknowledgements shows that there is no congestion to justify a slow start.
- The sender can continue with the current congestion window.
- The sender performs a fast recovery from the packet loss.
- This mechanism can improve the efficiency of TCP dramatically.

Implications on mobility
- TCP concludes a congestion situation from a missing acknowledgement:
  - Typically wrong in wireless networks, here we often have packet loss due to transmission errors.
  - Mobility itself can cause packet loss, if e.g. a mobile node roams from one access point (e.g. foreign agent in Mobile IP) to another while there are still packets in transit to the wrong access point and forwarding is not possible.
- The performance of an unchanged TCP degrades severely:
  - TCP cannot be changed fundamentally due to the large base of installation in the fixed network.
    - TCP for mobility has to remain compatible.
  - The basic TCP mechanisms keep the whole Internet together.

CLASSICAL TCP IMPROVEMENTS
- Indirect TCP (I-TCP)
- Snooping TCP
- Mobile TCP
- Fast retransmit/fast recovery
- Transmission/time-out freezing
- Selective retransmission
- Transaction-oriented TCP

Indirect TCP (I-TCP)

Socket and state migration after handover of a mobile host
- I-TCP segments a TCP connection into a
  - fixed part: Standard TCP is used
  - wireless part: optimized TCP protocol
- splitting of the TCP connection at, e.g., the foreign agent into 2 TCP connections, no real end-to-end connection any longer
- hosts in the fixed part of the net do not notice the characteristics of the wireless part

**Advantages**

- no changes in the fixed network necessary, no changes for the hosts (TCP protocol) necessary, all current optimizations to TCP still work
- transmission errors on the wireless link do not propagate into the fixed network
- simple to control, mobile TCP is used only for one hop between, e.g., a foreign agent and mobile host
- therefore, a very fast retransmission of packets is possible, the short delay on the mobile hop is known

**Disadvantages**

- loss of end-to-end semantics, an acknowledgement to a sender does now not any longer mean that a receiver really got a packet, foreign agents might crash
- higher latency possible due to buffering of data within the foreign agent and forwarding to a new FA

**Snooping TCP**

- the foreign agent buffers all packets with destination mobile host and additionally ‘snoops’ the packet flow in both directions to recognize acknowledgements
- buffering enable the FA to perform a local retransmission in case of packet loss on the wireless link
- **Transparent extension of TCP within the foreign agent**
  - buffering of packets sent to the mobile host
  - lost packets on the wireless link (both directions!) will be retransmitted immediately by the mobile host or foreign agent, respectively (so called “local” retransmission)
  - the foreign agent therefore “snoops” the packet flow and recognizes acknowledgements in both directions, it also filters ACKs
  - changes of TCP only within the foreign agent

**Integration of the MAC layer**

- the MAC layer often has similar mechanisms to those of TCP
- thus, the MAC layer can already detect duplicated packets due to retransmissions and discard them

**Problems**

- snooping TCP does not isolate the wireless link as good as I-TCP
- snooping might be useless depending on encryption schemes

**Advantages**

- The end-to-end TCP semantic is preserved
• The correspondent host does not need to be changed; most of the enhancements are in the foreign agent
• It does not need a handover of state as soon as the mobile host moves to another foreign agent.
• It does not matter if the next foreign agent uses the enhancement or not

Disadvantages
• Snooping TCP does not isolate the behavior of the wireless link as well as ITP
• Using negative acknowledgements between the foreign agent and the mobile host assumes additional mechanisms on the mobile host.
• All efforts for snooping and buffering data may be useless if certain encryption schemes are applied end-to-end between the correspondent host and mobile host

Mobile TCP
• Special handling of lengthy and/or frequent disconnections
• M-TCP splits as I-TCP does
  o unmodified TCP fixed network to supervisory host (SH)
  o optimized TCP SH to MH
• Supervisory host
  o no caching, no retransmission
  o monitors all packets, if disconnection detected
    ▪ set sender window size to 0
    ▪ sender automatically goes into persistent mode
  o old or new SH reopen the window
• Advantages
  o maintains semantics, supports disconnection, no buffer forwarding
• Disadvantages
  o loss on wireless link propagated into fixed network
  o adapted TCP on wireless link

Fast retransmit / fast recovery
• Change of foreign agent often results in packet loss
  o TCP reacts with slow-start although there is no congestion
• Forced fast retransmit
  o as soon as the mobile host has registered with a new foreign agent, the MH sends duplicated acknowledgements on purpose
  o this forces the fast retransmit mode at the communication partners
  o additionally, the TCP on the MH is forced to continue sending with the actual window size and not to go into slow-start after registration
• Advantage
  o simple changes result in significant higher performance
• Disadvantage
  o further mix of IP and TCP, no transparent approach

Transmission / time-out freezing
• Mobile hosts can be disconnected for a longer time
  o no packet exchange possible, e.g., in a tunnel, disconnection due to overloaded cells or mux. with higher priority traffic
  o TCP disconnects after time-out completely
• TCP freezing
  o MAC layer is often able to detect interruption in advance
  o MAC can inform TCP layer of upcoming loss of connection
  o TCP stops sending, but does now not assume a congested link
  o MAC layer signals again if reconnected
• Advantage
  o scheme is independent of data
• Disadvantage
  o TCP on mobile host has to be changed, mechanism depends on MAC layer
Selective retransmission

- TCP acknowledgements are often cumulative
  - ACK n acknowledges correct and in-sequence receipt of packets up to n
  - if single packets are missing quite often a whole packet sequence beginning at the gap has to be retransmitted (go-back-n), thus wasting bandwidth
- Selective retransmission as one solution
  - RFC2018 allows for acknowledgements of single packets, not only acknowledgements of in-sequence packet streams without gaps
  - sender can now retransmit only the missing packets
- Advantage
  - much higher efficiency
- Disadvantage
  - more complex software in a receiver, more buffer needed

Transaction-oriented TCP

- TCP phases
  - setup, data transmission, connection release
  - using 3-way-handshake needs 3 packets for setup and release, respectively
  - thus, even short messages need a minimum of 7 packets!
- Transaction oriented TCP
  - RFC1644, T-TCP, describes a TCP version to avoid this overhead
  - connection setup, data transfer and connection release can be combined
  - thus, only 2 or 3 packets are needed
- Advantage: efficiency
- Disadvantage
  - requires changed TCP
  - mobility not longer transparent

<table>
<thead>
<tr>
<th>Approach</th>
<th>Mechanism</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indirect TCP</td>
<td>Splits TCP connection into two connections</td>
<td>Isolation of wireless link, simple</td>
<td>Loss of TCP semantics, higher latency at handover, security problems</td>
</tr>
<tr>
<td>Snooping TCP</td>
<td>Snoops data and acknowledgements, local retransmission</td>
<td>Transparent for end-to-end connection, MAC integration possible</td>
<td>Insufficient isolation of wireless link, security problems</td>
</tr>
<tr>
<td>M-TCP</td>
<td>Splits TCP connection, chokes sender via window size</td>
<td>Maintains end-to-end semantics, and long term and frequent disconnections</td>
<td>Bad isolation of wireless link, processing overhead due to bandwidth management, security problems</td>
</tr>
<tr>
<td>Fast retransmit / fast recovery</td>
<td>Avoids slow-start after roaming</td>
<td>Simple and efficient</td>
<td>Mixed layers, not transparent</td>
</tr>
<tr>
<td>Transmission / time-out freezing</td>
<td>Freezees TCP state at disconnection, resumes after reconnection</td>
<td>Independent of content, works for longer interruptions</td>
<td>Changes in TCP required, MAC dependent</td>
</tr>
<tr>
<td>Selective retransmission</td>
<td>Retransmits only lost data</td>
<td>Very efficient</td>
<td>Slightly more complex receiver software, more buffer space needed</td>
</tr>
<tr>
<td>Transaction-oriented TCP</td>
<td>Combines connection setup/release and data transmission</td>
<td>Efficient for certain applications</td>
<td>Changes in TCP required, not transparent, security problems</td>
</tr>
</tbody>
</table>
WORLD WIDE WEB

- Protocol (HTTP, Hypertext Transfer Protocol) and language (HTML, Hypertext Markup Language) of the Web have not been designed for mobile applications and mobile devices, thus creating many problems!
- Typical transfer sizes
  - HTTP request: 100-350 byte
  - responses avg. <10 kbyte, header 160 byte, GIF 4.1kByte, JPEG 12.8 kbyte, HTML 5.6 kbyte
  - but also many large files that cannot be ignored
- The Web is no file system
  - Web pages are not simple files to download
  - static and dynamic content, interaction with servers via forms, content transformation, push technologies etc.
  - many hyperlinks, automatic loading and reloading, redirecting
  - a single click might have big consequences!

Hypertext transfer protocol

- HTTP is a stateless, lightweight, application level protocol for data transfer between servers and clients.
- The first version, HTTP/1.0, never became a formal standard due to too many variant implementations
- HTTP/1.1 is the standard currently used by most
- An HTTP transaction consists of an HTTP request issued by a client and an HTTP response from a server
- Stateless means that all HTTP transactions are independent of each other.
- HTTP does not ‘remember’ any transaction, request, or response.
- This results in a very simple implementation without the need for complex state machines.

Example

```
GET / HTTP/1.1
Host: www.inf.fu-berlin.de
```

The server might answer with something similar to the following (the response):

```
HTTP/1.1 200 OK
Date: Wed, 30 Oct 2002 19:44:26 GMT
Server: Apache/1.3.12 (Unix) mod_per1/1.24
ETag: "2d8190-2322-3dbfdbaf"
Accept-Ranges: bytes
Content-Length: 8994
Content-Type: text/html

<html>
<head>
```

HTTP 1.0 and Mobility

Characteristics

- stateless, client/server, request/response
- needs a connection oriented protocol (TCP), one connection per request
- primitive caching and security

Problems

Bandwidth and delay

- designed for large bandwidth (compared to wireless access) and low delay
- big and redundant protocol headers (readable for humans, stateless, therefore big headers in ASCII)
- uncompressed content transfer
- using TCP
  - huge overhead per request (3-way-handshake) compared with the content, e.g., of a GET request
  - slow-start problematic
- DNS lookup by client causes additional traffic
Caching
- quite often disabled by information providers to be able to create user profiles, usage statistics etc.
- dynamic objects cannot be cached
  - numerous counters, time, date, personalization, ...
- mobility quite often inhibits caches
- Security problems
  - how to use SSL/TLS together with proxies?
- today: many user customized pages, dynamically generated on request via CGI, ASP, ...

POSTing (i.e., sending to a server)
- can typically not be buffered, very problematic if currently disconnected

Hyper Text Markup Language (HTML) and Mobile Devices
- HTML
  - designed for computers with “high” performance, color high-resolution 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 (small touch-pads, soft-keyboards)
- Additional “features”
  - animated GIF, Java AWT, Frames, ActiveX Controls, Shockwave, movie clips, audio, ...
  - many web pages assume true color, multimedia support, high-resolution and many plug-ins
- Web pages ignore the heterogeneity of end-systems!
  - e.g., without additional mechanisms, large high-resolution pictures would be transferred to a mobile phone with a low-resolution display causing high costs

Approaches toward WWW for mobile devices
- Image scaling
  - picture can be scaled down to fewer colors, lower resolution, or to just the title of the picture.
  - The user can then decide to download the picture separately.
  - Clipping, zooming, or detail studies can be offered to users if they are interested in a part of the picture.
- Content transformation
  - Postscript or portable document format (PDF) into plain text before transmitting
- Content extraction / semantic compression
  - headline extraction, automatic abstract generation
- Special languages and protocols
  - HDML (handheld device markup language):
    - simple language similar to HTML requiring a special browser
    - HDTP (handheld device transport protocol): transport protocol for HDML
- Push technologies
  - Instead of pulling content from a server, the server could also push content to a client
  - This avoids the overhead of setting up connections for each item, but is only useful for some content
    - e.g. news, weather information, road conditions, where users do not have to interact much
- Application gateways, enhanced servers
  - simple clients, pre-calculations in the fixed network
  - compression, filtering, content extraction
  - automatic adaptation to network characteristics
- Problems
  - proprietary approaches, require special enhancements for browsers
  - heterogeneous devices make approaches more complicated

HTTP Ver 1.1 Improvements
- Connection re-use
  - use the same TCP connection for several requests and responses
  - Persistent connections are default in 1.1 (keep-alive option in 1.0)
  - pipelining : multiple requests
- Caching enhancements
A cache may now also store cacheable responses to reduce response time and bandwidth for future, equivalent responses.

- semantic transparency
- A special tag allows for the identification of content and helps to determine if two different URIs map to the same content
- defines a large set of cache-control directives
  - public/private, max-age, no-cache

**Bandwidth optimization**
- supports compression, negotiation of compression parameters and different compression styles (hop-by-hop or end-to-end).
- allows for partial transmission of objects (like images)

**Security**
- comprises further mechanisms to check message integrity and to authenticate clients, proxies, and servers

**Cookies**
- can set up a long-term “session” by storing state upon request

### Mobile WWW System Architecture

- client/server system
- Caching is a major topic in the web client/server scenario

**Integrated browser enhancement**

- Pre-fetching, caching, off-line use
  - e.g. Internet Explorer

**Additional application supporting browsing**

- Pre-fetching, caching, off-line use
  - e.g. original WebWhacker

**Client proxy as browser support**

- Pre-fetching, caching, off-line use
  - e.g., Caubweb, TeleWeb, Weblicator, WebWhacker, WebEx, WebMirror

**Network proxy as browser support**

- adaptive content transformation for bad connections, pre-fetching, caching
  - e.g., TranSend, Digestor
Client and network proxy as browser support
- combination of benefits plus simplified protocols
  - e.g., MobiScape, WebExpress

Client and network proxy with special transmission protocol
- adaptive content transformation for bad connections, pre-fetching, caching
  - e.g., Mowgli

WIRELESS APPLICATION PROTOCOL

- Goals
  - deliver Internet content and enhanced services to mobile devices and users (mobile phones, PDAs)
  - independence from wireless network standards
  - open for everyone to participate, protocol specifications will be proposed to standardization bodies
  - applications should scale well beyond current transport media and device types and should also be applicable to future developments

- Platforms
  - e.g., GSM (900, 1800, 1900), CDMA IS-95, TDMA IS-136, 3rd generation systems (IMT-2000, UMTS, W-CDMA)

- Forum
  - WAP Forum, co-founded by Ericsson, Motorola, Nokia, Unwired Planet
  - further information http://www.wapforum.org

WAP - scope of standardization

- Browser
  - “micro browser”, similar to existing, well-known browsers in the Internet

- Script language
  - similar to Java script, adapted to the mobile environment

- WTA / WTAI
  - Wireless Telephony Application (Interface): access to all telephone functions

- Content formats
  - e.g., business cards (vCard), calendar events (vCalendar)

- Protocol layers
  - transport layer, security layer, session layer etc.

- Working Groups

WAP Forum / Open Mobile Alliance Specifications must be (ISERS)

- Interoperable
  - allowing terminals and software from different vendors to communicate with networks from different providers;

- Scaleable
  - protocols and services should scale with customer needs and number of customers;

- Efficient
provision of QoS suited to the characteristics of the wireless and mobile networks;
- Reliable
  - provision of a consistent and predictable platform for deploying services
- Secure
  - preservation of the integrity of user data, protection of devices and services from security problems.

## WAP Architecture

### Components and interface of the WAP 1.x architecture

![WAP Architecture Diagram]

- GSM: Global System for Mobile communication
- HSCSD: High-Speed Circuit Switched Data
- GPRS: General Packet Radio Service
- WDP: Wireless Datagram Protocol
- WCMP: Wireless Control Message Protocol
- T-SAP: Transport Layer Service Access Point
- WTLS: Wireless Transport Layer Security
- SEC-SAP: SECurity SAP
- WTP: Wireless Transaction Protocol
- TR-SAP: Transaction SAP
- WSP: Wireless Session Protocol
- S-SAP: Session-SAP
- WAE: Wireless Application Environment
- WML: Wireless Markup Language

### Bearer services

- The basis for transmission of data is formed by different bearer services
- Examples:
  - SMS, HSCSD, GPRS, CDPD, IS-136, PHS
- No special interface has been specified between the bearer service and the next higher layer

### Transport layer

- Has wireless datagram protocol (WDP) and the additional wireless control message protocol (WCMP)
- Offers a bearer independent, consistent datagram-oriented service
- Communication is done transparently over one of the available bearer services
- Transport Layer Service Access Point (T-SAP) is the common interface to be used by higher layers independent of the underlying network

### Security layer

- Wireless transport layer security protocol WTLS offers its service at the security SAP (SEC-SAP).
- WTLS is based on the transport layer security
- optimized for use in wireless networks with narrow-band channels
- offer data integrity, privacy, authentication, and (some) denial-of-service protection

Transaction Layer
- uses wireless transaction protocol (WTP)
- offers a lightweight transaction service at the transaction SAP (TR-SAP).
- This service efficiently provides reliable or unreliable requests and asynchronous transactions

Session layer
- uses wireless session protocol (WSP)
- offers two services at the session-SAP (S-SAP),
  - one connection-oriented
  - one connectionless if used directly on top of WDP.
- A special service for browsing the web (WSP/B) has been defined that offers
  - HTTP/1.1 functionality,
  - long-lived session state,
  - session suspend and resume,
  - session migration
  - other features needed for wireless mobile access to the web.

Application layer
- wireless application environment (WAE)
- offers a framework for the integration of different www and mobile telephony applications
- offers many protocols and services with special service access points

Integration of WAP components
- To be able to browse these pages or additional pages with handheld devices, a wireless markup language (WML) has been defined in WAP.
- Special filters within the fixed network can now translate HTML into WML, web servers can already provide pages in WML, or the gateways between the fixed and wireless network can translate HTML into WML.
- These gateways also act as proxies for web access
- WML is additionally converted into binary WML for more efficient transmission
- a special gateway can be implemented to access traditional telephony services via binary WML.
- This wireless telephony application (WTA) server translates, e.g., signaling of the telephone network (incoming call etc.) into WML events displayed at the handheld device.

Wireless Datagram Protocol (WDP)
- Protocol of the transport layer within the WAP architecture
  - uses directly transports mechanisms of different network technologies
  - offers a common interface for higher layer protocols
  - allows for transparent communication using different transport technologies
- Goals of WDP
  - create a worldwide interoperable transport system with the help of WDP adapted to the different underlying technologies
  - transmission services such as SMS in GSM might change, new services can replace the old ones

WDP service primitives
- WDP offers source and destination port numbers used for multiplexing and demultiplexing of data respectively
• The service primitive to send a datagram is TDUnitdata.req with
  o the destination address (DA), destination port (DP), Source address (SA), source port (SP), and user
data (UD) as mandatory parameters
• The T-DUnitdata.ind service primitive indicates the reception of data
• error is indicated with the T-DError.ind service primitive
  o error code (EC) is returned indicating the reason for the error to the higher layer

**Wireless Control Message Protocol (WCMP)**

• provides error handling mechanisms for WDP
• Typical WCMP messages are
  o destination unreachable (route, port, address unreachable),
  o parameter problem (errors in the packet header),
  o message too big,
  o reassembly failure
  o echo request/reply

**Wireless Transport Layer Security (WTLS)**

• Goals
  o data integrity
    ▪ prevention of changes in data
  o privacy
    ▪ prevention of tapping
  o authentication
    ▪ creation of authenticated relations between a mobile device and a server
  o protection against denial-of-service attacks
    ▪ protection against repetition of data and unverified data
• WTLS
  o is based on the TLS (Transport Layer Security) protocol (former SSL, Secure Sockets Layer)
  o optimized for low-bandwidth communication channels

**WTLS Establishing A Secure Session**

• The first step is to initiate the session with the SEC-Create primitive.
Parameters are source address (SA), source port (SP) of the originator, destination address (DA), destination port (DP) of the peer.

The originator proposes:
- a key exchange suite (KES) (e.g., RSA, DH, ECC),
- a cipher suite (CS) (e.g., DES, IDEA)
- a compression method (CM)

The peer answers with parameters for the sequence number mode (SNM),
- the key refresh cycle (KR),
- the session identifier (SID) (which is unique with each peer),
- the selected key exchange suite (KES'), cipher suite (CS'), compression method (CM').

The peer also issues a SEC-Exchange primitive.
- This indicates that the peer wishes to perform public-key authentication with the client, i.e., the peer requests a client certificate (CC) from the originator.

The originator answers with its certificate and issues a SEC-Commit.req primitive.
- This primitive indicates that the handshake is completed.

**WTLS Datagram Transfer**

- After setting up a secure connection between two peers, user data can be exchanged.
- This is done using the simple SEC-Unitdata primitive.

**Wireless Transaction Protocol**

- **Goals**
  - Different transaction services, offloads applications
    - Application can select reliability, efficiency
  - Support of different communication scenarios
    - Class 0: unreliable message transfer
    - Class 1: reliable message transfer without result message
    - Class 2: reliable message transfer with exactly one reliable result message
  - Supports peer-to-peer, client/server and multicast applications
  - Low memory requirements, suited to simple devices (< 10kbyte)
  - Efficient for wireless transmission
    - Segmentation/reassembly
    - Selective retransmission
    - Header compression
    - Optimized connection setup (setup with data transfer)

- The three service primitives offered by WTP are:
  - TR-Invoke to initiate a new transaction,
  - TR-Result to send back the result of a previously initiated transaction
  - TR-Abort to abort an existing transaction

- The PDUs exchanged between two WTP entities for normal transactions are the:
  - Invoke PDU
  - Ack PDU
  - Result PDU

- A special feature of WTP is its ability to provide a user acknowledgement or, alternatively, an automatic acknowledgement by the WTP entity.
WTP Class 0

- Class 0 offers an unreliable transaction service without a result message.
- The transaction is stateless and cannot be aborted.
- The service is requested with the TR-Invoke.req primitive.
- Parameters are the source address (SA), source port (SP), destination address (DA), destination port (DP).
- the A flag the user of this service can determine, if the responder WTP entity should generate an acknowledgement or if a user acknowledgement should be used.
- The WTP layer will transmit the user data (UD) transparently to its destination.
- The class type C indicates here class 0.
- the transaction handle H provides a simple index to uniquely identify the transaction
  - an alias for the tuple (SA, SP, DA, DP)
- The WTP entity at the responder then generates a TR-Invoke.ind primitive
- H’ is the local handle for the transaction on the responder’s side.

![WTP Class 0 Diagram](image)

WTP Class 1

![WTP Class 1 Diagram](image)

WTP class 2: Basic transaction, no user acknowledgement

![WTP Class 2 Diagram](image)
WTP class 2: Basic transaction, with user Acknowledgement

WTP class 2 transaction with “hold on”, no user acknowledgement

Wireless Session Protocol (WSP)

- Goals
  - HTTP 1.1 functionality
  - Request/reply, content type negotiation, ...
  - support of client/server, transactions, push technology
  - key management, authentication, Internet security services
  - session management (interruption, resume,...)

- Services
  - session management (establish, release, suspend, resume)
  - capability negotiation
  - content encoding

- WSP/B (Browsing)
  - HTTP/1.1 functionality - but binary encoded
  - exchange of session headers
  - push and pull data transfer
  - asynchronous requests
server address (SA), client address (CA), client header (CH), requested capabilities (RC), server header (SH), negotiated capabilities (NC), reason R

WSP/B session suspension and resume

WSP/B session termination
**WSP/B completed transaction**

- Client transaction identifier CTID, method M, request URI RU, server transaction identifier STID, status (S), response header (RH), response body (RB)

**Push primitives**

- A server can push data towards a client if allowed
- Push header (PH), push body (PB), server push identifier (SPID), client push identifier (CPID)

**WSP/B non-confirmed push**

- S-Push.ind (PH, PB)
- Push PDU
- WTP Class 0 transaction

**WSP/B confirmed push**

- S-ConfirmedPush.ind (CPID, PH, PB)
- ConfPush PDU
- S-ConfirmedPush.req (SPID, PH, PB)
- S-ConfirmedPush.res (CPID)
- S-ConfirmedPush.cnf (SPID)
- WTP Class 1 transaction
Wireless Application Environment (WAE)

- Goals
  - network independent application environment for low-bandwidth, wireless devices
  - integrated Internet/WWW programming model with high interoperability

- Requirements
  - device and network independent, international support
  - manufacturers can determine look-and-feel, user interface
  - considerations of slow links, limited memory, low computing power, small display, simple user interface (compared to desktop computers)

- Components
  - architecture: application model, browser, gateway, server
  - WML: XML-Syntax, based on card stacks, variables, ...
  - WMLScript: procedural, loops, conditions, ... (similar to JavaScript)
  - WTA: telephone services, such as call control, text messages, phone book, ... (accessible from WML/WMLScript)
  - content formats: vCard, vCalendar, Wireless Bitmap, WML, ...

WAE logical model

Wireless Markup Language (WML)

- WML follows deck and card metaphor
  - WML document consists of many cards, cards are grouped to decks
  - a deck is similar to an HTML page, unit of content transmission
  - WML describes only intent of interaction in an abstract manner
  - presentation depends on device capabilities

- Features
  - text and images
  - user interaction
  - navigation
  - context management

WML Example

```xml
<WML>
  <CARD>
    <DO TYPE="ACCEPT">
      <GO URL="#card_two"/>
    </DO>
    This is a simple first card!
    On the next you can choose ...
  </CARD>
  <CARD NAME="card_two">
    ... your favorite pizza:
  </CARD>
</WML>
```
WMLScript

- Complement to WML
- Provides general scripting capabilities
- Features
  - validity check of user input
  - check input before sent to server
  - access to device facilities
    - hardware and software (phone call, address book etc.)
  - local user interaction
    - interaction without round-trip delay
  - extensions to the device software
    - configure device, download new functionality after deployment

Example:

```wmlscript
function pizza_test(pizza_type) {
    var taste = "unknown";
    if (pizza_type == "Margherita") {
        taste = "well... ";
    }
    else {
        if (pizza_type == "Vulcano") {
            taste = "quite hot";
        }
    }
    return taste;
}
```

Wireless Telephony Application (WTA)

- Collection of telephony specific extensions
- Extension of basic WAE application model
  - content push
    - server can push content to the client
    - client may now be able to handle unknown events
  - handling of network events
    - table indicating how to react on certain events from the network
  - access to telephony functions
    - any application on the client may access telephony functions

Example

- calling a number (WML)
  `wtai://wp/mc;07216086415`
- calling a number (WMLScript)
  `WTAPublic.makeCall("07216086415");`
WTA logical architecture

WTA example: voice message

WAP push architecture with proxy gateway

Comments & Feedback

Thanks to my family members who supported me while I spent hours and hours to prepare this.
Your feedback is welcome at GHCRajan@gmail.com