Velammal College of Engineering and Technology (Autonomous), Madurai

Department of Electronics and Communication Engineering

UNIT V MOBILE TRANSPORT AND APPLICATION LAYER

Mobile TCP- Types of TCP, WAP, Architectures - WDP, WTLS, WTP, WSP, WAE, WTA Architecture, WML.

Mobile TCP (M-TCP)

Mobile TCP is a specialized extension of the traditional TCP protocol, designed to function efficiently in mobile and wireless network environments. Unlike wired networks, where packet loss mostly happens due to congestion, mobile networks experience losses because of fading signals, mobility, handoffs, interference, and temporary disconnections. Traditional TCP misinterprets these wireless losses as congestion and unnecessarily reduces the transmission rate, which greatly reduces performance. Mobile TCP attempts to overcome these limitations by adjusting the behavior of TCP to suit mobility.

In essence, Mobile TCP retains the familiar end-to-end semantics of TCP but introduces mechanisms that help the protocol respond appropriately to the conditions of a wireless environment. One of the key features of M-TCP is its ability to maintain the connection even when the mobile device temporarily disconnects or moves between cells. During such events, Mobile TCP "freezes" the sender's transmission window so that the sender does not reduce its congestion control variables. When the device reconnects, the same congestion window is restored, allowing communication to resume smoothly without the slow start phase. This approach ensures that mobility-related disruptions do not impact application performance.

Another important aspect of Mobile TCP is its support for efficient handovers. As a mobile user moves from one base station to another, the protocol notifies the sender about the change so that the congestion window remains stable. By separating wireless-specific behaviors from wired network assumptions, M-TCP significantly improves throughput, reduces delay, and enables seamless communication in environments like MANETs, VANETs, cellular networks, and Wi-Fi-based systems.

Types of TCP Used in Mobile/Wireless Networks

Mobile environments require customized TCP variants to handle the unique challenges of wireless communication. Each variant attempts to solve a specific problem, such as high error rates, variable bandwidth, or frequent disconnections.

1. Indirect TCP (I-TCP)

Indirect TCP divides the end-to-end connection into two separate parts: one between the fixed host and the base station, and the other between the base station and the mobile device. This structure allows the wireless section to be optimized independently. For example, faster retransmissions and shorter timeouts can be applied only to the wireless segment without affecting the wired portion. Although this improves performance substantially, the drawback is that the original end-to-end semantics of TCP are broken because the base station becomes an intermediary in the communication.

2. Snooping TCP

Snooping TCP keeps the traditional end-to-end connection intact but uses the base station as a smart observer. The base station "snoops" on packets going back and forth, caching them and retransmitting lost ones quickly. It detects lost acknowledgments and corrects errors before the sender even notices. This technique minimizes the impact of wireless losses on the sender and avoids unnecessary reduction in the congestion window.

3. Mobile TCP (M-TCP)

M-TCP is specifically designed to handle long disconnections and frequent handovers. When the mobile device is about to disconnect, it signals the base station, which in turn tells the sender to pause. The key idea is to prevent the sender from reacting to the disconnection as if it were congestion. Once the device reconnects, the communication continues immediately from where it stopped.

4. Freeze TCP

Freeze TCP allows the mobile device to inform the TCP sender proactively about upcoming signal degradation. It freezes the sender's state—congestion window, timers, and sequence numbers—so that the sender does not enter slow start unnecessarily. When connectivity is restored, the sender resumes transmission without delay, resulting in smoother performance in highly mobile situations.

5. TCP with Explicit Loss Notification (ELN)

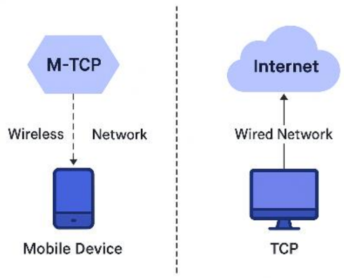
ELN introduces a mechanism where intermediate nodes or the receiver can mark packets that were lost due to wireless errors, not congestion. When the sender receives such a notification, it retransmits lost packets without reducing its congestion window. This method is effective in environments where wireless errors are frequent but the network is not congested.

6. TCP Westwood / Westwood+

TCP Westwood estimates the available bandwidth based on the rate of received acknowledgments. During packet loss, instead of blindly reducing its window, it adjusts it according to the measured bandwidth. This makes it better suited for mobile and wireless networks where bandwidth fluctuates frequently due to movement and interference.

WAP (Wireless Application Protocol)

WAP is a communication protocol designed to enable mobile phones and handheld devices to access Internet-based services. Before modern smartphones became mainstream, mobile devices had extremely limited processing power, small screens, and low bandwidth. Traditional web protocols like HTTP and HTML were too heavy for these devices. WAP emerged as a solution to provide an Internet-like experience that was optimized for wireless networks.



WAP uses a layered architecture similar to the OSI model, allowing it to adapt to different wireless technologies such as GSM, CDMA, GPRS, and SMS-based data services. At the application layer, WAP introduced WML (Wireless Markup Language), which functioned similarly to HTML but was lighter and specifically designed for small screens. WML pages were highly compressed to allow fast loading even on low-speed networks.

The protocol stack includes the Wireless Session Protocol (WSP), which acts like a simplified form of HTTP, enabling session management with much less overhead. The Wireless Transaction Protocol (WTP) provides lightweight transaction services similar to TCP but optimized for unreliable wireless links. Security is handled by the Wireless Transport Layer Security (WTLS), a version of TLS modified to reduce processing requirements. At the bottom of the stack, the Wireless Datagram Protocol (WDP) adapts WAP services to different wireless bearers, creating a uniform interface for higher layers.

Architectures - WDP, WTLS, WTP, WSP, WAE, WTA

1. Wireless Datagram Protocol (WDP)

The **WDP** forms the lowest layer of the WAP architecture and acts as the adaptation layer between the wireless bearer networks and the upper WAP layers. It offers a uniform datagram service to all higher layers regardless of the underlying network technology. In a wireless environment where different carriers use various bearers such as SMS, USSD, CSD, CDMA, or GPRS, application portability becomes difficult. WDP overcomes this by masking the differences between these networks. By providing standard addressing mechanisms and a consistent interface, it allows WAP applications to function seamlessly over any bearer service. Essentially, WDP plays the role of a transport-independent communication layer, ensuring that upper protocols remain unaffected by the limitations, variation, or characteristics of individual wireless networks.

Features:

- Provides a **uniform interface** to the upper layers regardless of the underlying wireless network (GSM, CDMA, SMS, GPRS).
- Supports **port addressing** similar to UDP.
- Allows **transparent data transmission** over various wireless bearer services.
- Handles **segmentation and reassembly** of data packets.

- Provides **minimal overhead** to support low-bandwidth wireless channels.
- Offers **bearer independence**, allowing applications to run across multiple network types.
- Ensures **basic error detection** using checksums at the datagram level.

Applications

- Used as the **transport layer** for all WAP communication.
- Suitable for **messaging services** over SMS, GPRS, USSD.
- Enables **WAP browsing** on low-bandwidth mobile networks.
- Supports **IoT devices** needing lightweight data transfer.

Advantages

- Provides **bearer independence** across GSM, CDMA, and GPRS.
- Very **low overhead**, ideal for slow wireless links.
- Allows **uniform addressing** of applications via ports.
- Supports **error detection** and basic segmentation.

Disadvantages

- Does not ensure **reliable delivery** (similar to UDP).
- No built-in **flow control or congestion control**.
- Requires higher layers for reliability and security.

2. Wireless Transport Layer Security (WTLS)

The **WTLS** is a specialized security layer designed specifically for wireless communication constraints. It provides privacy, data integrity, and authentication in an environment where bandwidth is limited, processing power is restricted, and latency is high. Modeled after TLS but optimized for mobile devices, WTLS supports smaller message sizes, compressed certificates, and efficient cryptographic operations. It protects the communication link between the mobile device and the WAP gateway from threats such as eavesdropping, tampering, and unauthorized access. Given that wireless communication is inherently more vulnerable than wired communication, WTLS plays a crucial role by ensuring secure transactions, safe browsing, and encrypted transmission of sensitive information. The protocol is lightweight, fast, and robust enough to provide strong security without overburdening mobile devices.

Features

- Provides **data integrity, confidentiality, and authentication** for wireless communication.
- Optimized for **low-bandwidth and high-latency** mobile networks.
- Smaller message sizes compared to TLS to suit wireless constraints.
- Supports **certificate-based authentication** and dynamic key refreshing.
- Protects against **common network attacks** (replay, man-in-the-middle, spoofing).
- Includes **compression mechanisms** before encryption for efficiency.
- Offers **class-based security levels** depending on application need.

Applications

- Used for **secure WAP browsing** and mobile banking.
- Protects **mobile e-commerce transactions**.
- Ensures confidentiality in **wireless email/SMS gateways**.

Advantages

- Provides **encryption, authentication, and integrity**.
- Optimized for **low bandwidth and high latency** networks.
- Smaller overhead than standard TLS.
- Protects against **replay, spoofing, and MITM attacks**.

Disadvantages

- Certificate management may be **complex** on small devices.
- Adds **processing overhead** for encryption.
- Gateway-based architecture may compromise **end-to-end security**.

3. Wireless Transaction Protocol (WTP)

The **WTP** is responsible for providing reliable transaction support over unreliable wireless networks. Unlike traditional TCP, which maintains heavy connection-oriented communication, WTP offers efficient transaction mechanisms designed specifically for wireless environments. It can support reliable or unreliable requests depending on the needs of the application. WTP performs essential functions such as duplicate detection, retransmission of lost packets, and acknowledgment handling, all while minimizing the overhead associated with TCP. Because wireless channels often experience high error rates and variable delays, WTP ensures dependable message exchange without the burden of maintaining full connection states. Thus, it allows faster and more efficient operations, enabling mobile devices to handle tasks such as browsing, form submissions, and data transfers with reduced latency and improved reliability.

Features

- Ensures **reliable message delivery** where necessary, unlike UDP.
- Supports three transaction types:
    i. Unreliable one-way request
    ii. Reliable one-way request
    iii. Reliable two-way request
- Provides **lightweight semantics**, reducing processing overhead.
- Offers **duplicate detection** to prevent repeated transactions.
- Handles **acknowledgment and retransmission** mechanisms.
- Supports **asynchronous transactions** to enhance performance.
- Reduces latency by using **compact control messages**.

Applications

- Used for **interactive browsing** requiring reliable transactions.
- Supports **mobile payment** confirmation messages.
- Used in **download services**, mobile forms, and data entry.
- Provides reliability for **push services**.

Advantages

- Lightweight and faster than TCP.

- Supports **reliable** and **unreliable** transactions.
- Lower delay due to optimized signaling.
- Ensures **duplicate detection** and efficient retransmission.

Disadvantages

- Does not offer **full TCP functionalities** (no congestion control).
- Limited for applications requiring **large data transfer**.
- Needs WDP and WSP for full features.

4. Wireless Session Protocol (WSP)

The **WSP** functions as the session layer of the WAP stack and closely resembles HTTP, but it is adapted to suit the limitations of mobile communication. WSP supports long-lived sessions, enabling a mobile device to suspend and resume communication sessions without restarting the connection. This reduces the overhead caused by repeated handshakes and connection setups that occur in wireless networks. It also provides mechanisms for content negotiation, capability detection, and compressed header transmission, making it highly efficient in handling web-style communication over narrow wireless channels. WSP ensures that the interaction between WAP applications and servers is smooth, structured, and optimized. It acts as a bridge between the application layer and underlying transaction services, managing all session-related responsibilities for efficient data exchange.

Features:

- Enables content negotiation (capabilities, preferences, encoding).
- Supports long-lived sessions between mobile client and server.
- Provides session suspend/resume to handle mobility and signal loss.
- Optimized for low-bandwidth networks with reduced header size.
- Includes cache control, improving loading speed.
- Compatible with HTTP/1.1 functions but tailored for wireless use.
- Allows push and pull services, enabling content to be delivered proactively.

Applications

- Used in **WAP browsers** for session handling.
- Enables **content negotiation** between client and server.
- Supports **push services** like alerts and notifications.
- Useful for applications requiring **session continuity**.

Advantages

- Supports **long-lived sessions** for better performance.
- Reduces overhead using **compact headers**.
- Allows **suspend/resume** to handle mobility.
- Enhances responsiveness by using **caching mechanisms**.

Disadvantages

- Not suitable for **high-bandwidth media**.
- Limited compared to full HTTP functionality.
- Extra processing required for maintaining sessions.

5. Wireless Application Environment (WAE)

The **WAE** defines the topmost application environment in the WAP architecture, offering a standardized framework for developing and deploying applications on wireless devices. It includes various technologies such as WML (Wireless Markup Language), WMLScript, and device-independent formats like WBMP. WAE provides essential services and content specifications that enable developers to build applications that are lightweight, efficient, and suited for small screens and low computational capabilities. It functions as the presentation and application engine, ensuring that content delivered to mobile devices is structured appropriately and rendered correctly. WAE also integrates with telephony services, push mechanisms, and browser functionalities. Overall, it offers a complete application execution environment, giving mobile users access to interactive and dynamic services similar to traditional web browsing but optimized for wireless constraints.

Features:

- Provides an integrated **application framework** for mobile devices.

- Contains essential components such as **WML**, **WMLScript**, and **WTA**.
- Ensures **interoperability** of mobile applications across devices and networks.
- Enables **content generation** optimized for small screens and limited memory.
- Provides **standard libraries** for script execution and user interaction.
- Supports **user agent features**, including bookmarks and navigation control.
- Ensures **efficient content retrieval** using compact markup formats.

Applications

- Used to build **WML-based mobile web pages**.
- Supports **WMLScript**, enabling client-side logic.
- Allows creation of **telephony-enabled services**.
- Suitable for **news portals, email access, and utility apps**.

Advantages

- Platform-independent application environment.
- Efficient content delivery for small screens.
- Provides standardized **browser-like features**.
- Lightweight markup ensures **fast loading**.

Disadvantages

- Limited UI capabilities compared to modern HTML5.
- Restricted script functionalities in WMLScript.
- Dependent on WAP microbrowser support.

6. Wireless Telephony Application (WTA)

The **WTA** architecture extends the WAE framework to include telephony-specific capabilities. It enables the integration of mobile phone functionalities—such as call handling, messaging, phonebook access, and network service alerts—directly with WAP-based applications. Using WTA, service providers can create applications that interact with the device’s telephony features, allowing actions like click-to-call, network-triggered notifications, and automated call management. WTA supports both user-initiated and network-initiated operations, making it suitable for services such as customer care, mobile banking, location-based alerts, and emergency notifications. It tightly integrates WAP browsing with the phone's native capabilities, enhancing the overall user experience by enabling seamless coordination between web-based content and telephony functions.

Features:

- Integrates **telephony services** with WAP applications.
- Allows applications to control phone features such as **call initiation, call termination, and address book access**.
- Supports **event handling** for incoming calls, call alerts, and network notifications.
- Works with **WTAI (Wireless Telephony Application Interface)** for device-level control.
- Enables **service providers** to create advanced value-added telecom services.
- Offers **secure access** to device functions to prevent unauthorized use.
- Supports **interactive voice services**, combining telephony and web features.

Applications

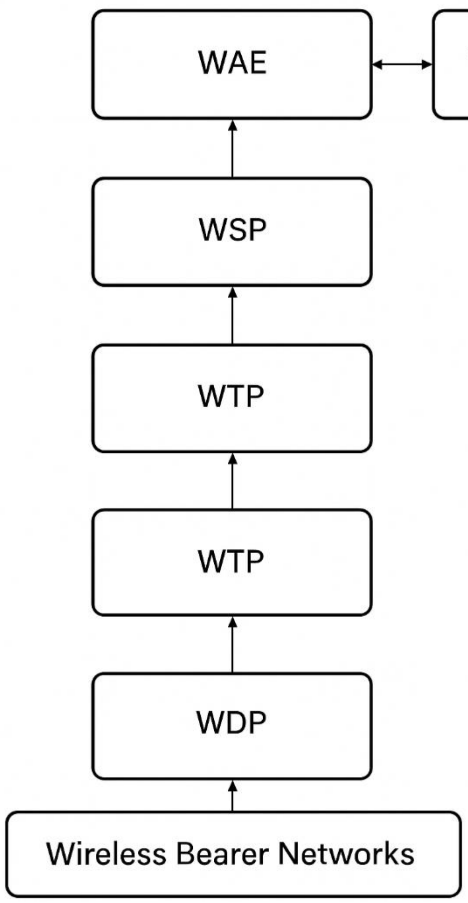
- Telecom value-added services like:
  - Call forwarding
  - Voicemail access
  - Caller ID services
- Integrates **web content with phone features**.
- Used in **service-provider menus** (SIM Toolkit style).
- Enables **interactive voice and data applications**.

Advantages

- Allows web-based control of **telephony functions**.
- Supports event-based interactions (incoming call, SMS).
- Provides advanced service creation for operators.
- Offers secure device-level access via **WTAI** APIs.

Disadvantages

- Requires device and network **support for WTA**.
- Security risks if telephony APIs are misused.
- Limited in modern smartphones due to replacement by **3G/4G/5G APIs**.



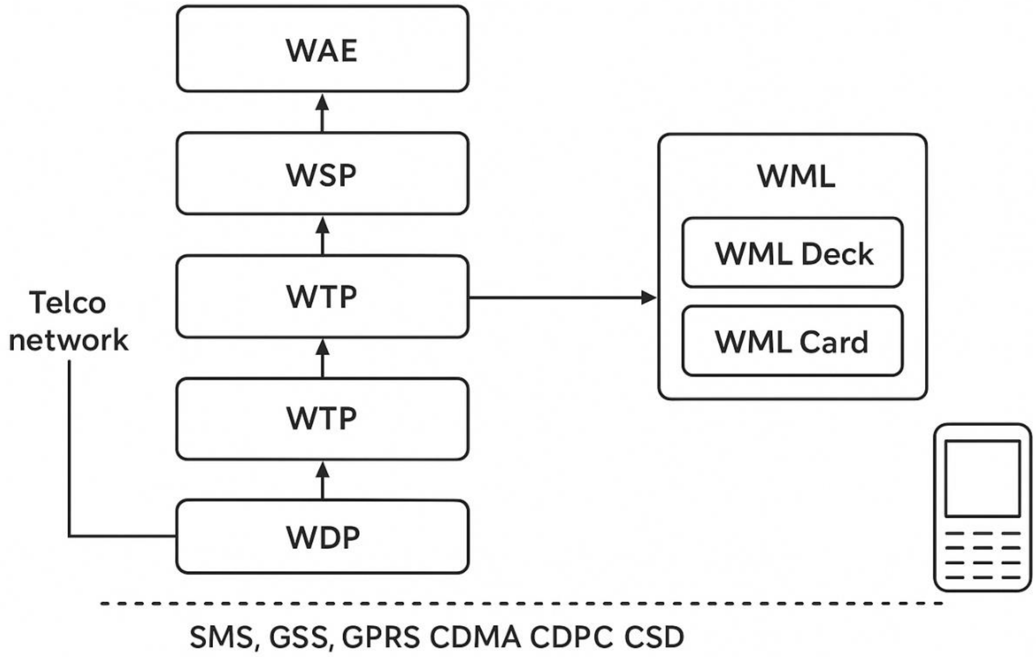
Wireless Markup Language (WML)

Wireless Markup Language (WML) is a specialized markup language developed for mobile devices operating under the WAP framework. It is based on XML and was designed to overcome the limitations of early mobile phones such as small display screens, low memory, limited processing power, and restricted input mechanisms. Unlike HTML, which is optimized for desktop browsers, WML focuses on structuring content into small, manageable units.

A WML document is organized into **cards** and **decks**. A *deck* is equivalent to a single WML file, while each *card* represents one screen or page displayed on the mobile device. Users navigate between cards through links, making WML ideal for step-by-step interactions and simple menu-based applications. This structure reduces the amount of data transmitted at one time, which is essential in low-bandwidth environments.

WML supports user interaction through forms, text entry fields, selection lists, and soft-key actions. Navigation is designed to be minimal and efficient, often relying on keypresses rather than complex input methods. The language also integrates with **WMLScript**, a lightweight scripting language similar to JavaScript but optimized for limited devices. WMLScript enables client-side validation and minor logic processing, reducing the need for frequent communication with the server.

Overall, WML serves as the presentation layer of WAP applications, ensuring that content is readable, functional, and fast-loading on small mobile devices. It played a crucial role during the early development of mobile internet by providing a standardized way to deliver interactive services before smartphones and HTML-capable browsers became widespread.



Wireless Application Protocol (WAP)

Features

- **Card and Deck Structure**: Content is divided into decks (documents) and cards (individual screens), optimizing navigation on small displays.
- **Lightweight and Efficient**: Designed for low-bandwidth networks like GSM/GPRS, making pages load fast even on slow connections.
- **XML-Based Language**: Uses structured tags similar to XML, making it easy to parse, validate, and render on mobile WAP browsers.
- **Event Handling Support**: Allows user interactions such as clicking links, entering text, or triggering actions through softkeys.
- **WMLScript Integration**: Supports simple client-side scripting for validation, logic, and calculations.
- **Support for Navigation Controls**: Built-in mechanisms for backward/forward navigation, softkeys, and hierarchical browsing.

- **Device-Friendly Layout**: Content automatically adapts to small screens, low memory, and limited input capabilities.
- **Form and Input Elements**: Includes tags for data entry, text input, selection lists, and submission.

Applications of WML

- **Early Mobile Web Browsing** - Used to deliver news, weather, email, and simple text-based websites on first-generation phones.
- **Banking and Financial Services** - Enabled secure transactions and balance inquiries through WAP-enabled handsets.
- **SMS/USSD-Based Portals** - Provided menu-driven services through telecom operators.
- **Mobile E-commerce** - Allowed simple purchasing, order status tracking, and account management.
- **Telecom Operator Services** - Integrated with WTA for call control, prepaid balance checks, and mobile service menus.
- **Enterprise Applications** - Helped companies deploy mobile field-service tools, basic data entry forms, and service status portals.

Advantages of WML

- **Optimized for Low Bandwidth** - Loads quickly even on 2G systems and low-speed wireless connections.
- **Low Processing Requirements** - Designed for mobile devices with limited memory and CPU.
- **Efficient Navigation Model** - Card-based structure reduces data transfer and boosts user experience.
- **Lightweight Code Structure** - Faster parsing and rendering compared to early HTML versions.

- **Secure Environment** - Works with WTLS for secure mobile transactions.
- **Device Independence** - WML pages render consistently across various WAP-enabled phones.

Disadvantages of WML

- **Limited Multimedia Support** - Cannot handle audio, video, or advanced graphics used in modern web applications.
- **Outdated Technology** - Replaced by HTML, XHTML, and modern smartphone browsers.
- **Restrictive User Interface** - Only allows basic layouts, text, and simple forms; no advanced styling or interactivity.
- **Requires WAP Gateway** - Content must pass through a gateway, increasing complexity and latency.
- **Not Compatible with HTML Browsers** - WML and HTML are not interchangeable, requiring separate websites for mobile and desktop.
- **Limited Screen Space Design** - Originally designed for monochrome screens, making UI expression minimal.

Comparison of Architectures

Architecture Layer	Full Form	Primary Function	Operates At	Key Features	Role in WAP
WDP	Wireless Datagram Protocol	Provides a uniform transport service over different wireless bearers	Transport Layer (Bottom of WAP stack)	Bearer independence, port addressing, low-overhead datagrams	Makes all wireless networks (GSM, CDMA, GPRS, SMS) appear identical to upper layers

WTLS	Wireless Transport Layer Security	Provides security for wireless communication	Between Transport & Session Layers	Encryption, authentication, integrity, optimized for low bandwidth	Protects WAP traffic from threats like spoofing, eavesdropping, replay attacks
WTP	Wireless Transaction Protocol	Ensures reliable message delivery & lightweight transactions	Above WDP	Supports reliable/unreliable transactions, acknowledgment, retransmission	Provides reliability similar to TCP but lighter and faster for wireless
WSP	Wireless Session Protocol	Manages sessions between client and server	Session/Presentation Layer	Long-lived sessions, suspend/resume, content negotiation	Enables efficient browsing, push/pull services, and session continuity
WAE	Wireless Application Environment	Provides application framework including WML & WMLScript	Application Layer	Deck-card model, scripting, UI controls, telecom integration	Defines how applications run and display on WAP-enabled devices
WTA	Wireless Telephony Application	Integrates telephony functions into WAP applications	Application Layer (Alongside WAE)	Access to call control, phonebook, events, alerts	Allows apps to initiate calls, handle call events, manage telecom services