

WIRELESS APPLICATION PROTOCOL (WAP)

PRESENTED BY:

D.R.Esesve
III/IV B.Tech ECE.
VITAM COLLEGE OF ENGINEERING.
DRESEVE@YAHOO.COM

VITAM COLLEGE OF ENGG.
PARVATHIPURAM.

URL: members.rediff.com/dresesve/wap.html

WIRELESS APPLICATION PROTOCOL (WAP)

D.R.ESESVE

ABSTRACT

Dresesve@rediffmail.com

The paper presents a small part of the drastically improving changes in the field of communications. This also helps in understanding the world of WAP, WIRELESS APPLICATION PROTOCOL. This is the latest development in this field which enables people access the entire Web world and many other devices through miniature devices, like cellphones, pagers etc.

The World Wide Web being the powerhouse of information the WAP has enabled access to it an easy task. The technology helps users surf the Web world by very small and hand held devices. WAP is simply a protocol, which makes the mobile telephone talk to the server, installed in the network.

Many new technologies like Wireless Telephony Application Interface are on the threshold of realization with WAP technology. This is the technology that inferred to set the start for a new era of finest sets of protocols, which are more appropriate to communicate with the hand held devices.

Nevertheless, like all the things on and in this world, even WAP has certain unidealistic features. The trouble is that the user cannot download all the entire graphics of a particular HTML built page. Moreover, the mere capacities of downloading are very less, thereby consuming huge amount of time, and to discuss there are certain such other odds about it. The WAP Forum for the first time defined an open structured architecture and certain protocols in an attempt to address some of the constraints.

*With the advent of the technologies like WAP, and certain other things like **G.P.R.S**, WAP enabled wireless Internet may replace the system of the desktop computer accessed internet. Also INDIA has also been the host for many emerging technologies in the recent past. WAP is no way exception. But to think for, are we making ourselves comfortable or complex?*

INTRODUCTION

Ever since the Web became popular with consumers, there has been a desire for wireless Internet access. The obvious solution is to connect a laptop computer to a modem and cell phone. This solution mirrors the in-home solution but used the cell phone for mobility. If one could actually get the individual devices to communicate, this form of web surfing was quite disappointing due to long response times and terrible communication paths that decimated the effective bit rate. With many Web pages having large and numerous image files, these early experiences at web surfing were quite trying on one's patience.

WAP is an emerging industry standard whose goal is to provide wireless internet-like services and information to handheld devices that have limited display and data capabilities, such as pagers, cellphones, and personal digital assistants (PDAs). The WAP Forum, a consortium whose current members are Ericsson, Nokia, Motorola, and Unwired Planet, controls the WAP spec. By mid 1998, additional companies had the chance to join the WAP Forum.

WAP recognizes that the consumer not only values the power and usefulness of the Web but also wishes to have it available while away from home or work. The concept is very simple: someone with a cell phone, pager, or PDA should be able to do limited Web surfing, for example, to check stock quotes, get restaurant information, or access bank accounts. But how can a wireless device with a small display render Web pages that are loaded with text that scrolls on and on? And who wants to wait "forever" for a huge image to be sent over the painfully slow wireless link? And how can a user follow links without a mouse? The goal of WAP is to extend the Web to handheld wireless devices by addressing and solving these difficulties.

The WAP specification is not yet finalized. A draft version, 0.9, was released to the public in early February 1998.

Technical Overview of WAP

WAP consists of wireless equivalents of HTTP and HTML. The HTTP-like component defines the communication protocol between the handheld device and a server or gateway. This component addresses characteristics that are unique to wireless devices, such as data rate and round-trip response time. For example, a

cell phone may have a maximum achievable data rate of 9.6 kbps as compared to 33 kbps in the home or 1.5 Mbps in the workplace. The HTML-like component defines new markup and scripting languages for displaying information to and interacting with the user. This component is highly focused on the limited display size and limited input devices available on small, handheld devices. For example, a typical cellphone may have only a 4x10-character display with 16-grey levels and only a numeric keypad plus up/down volume keys.

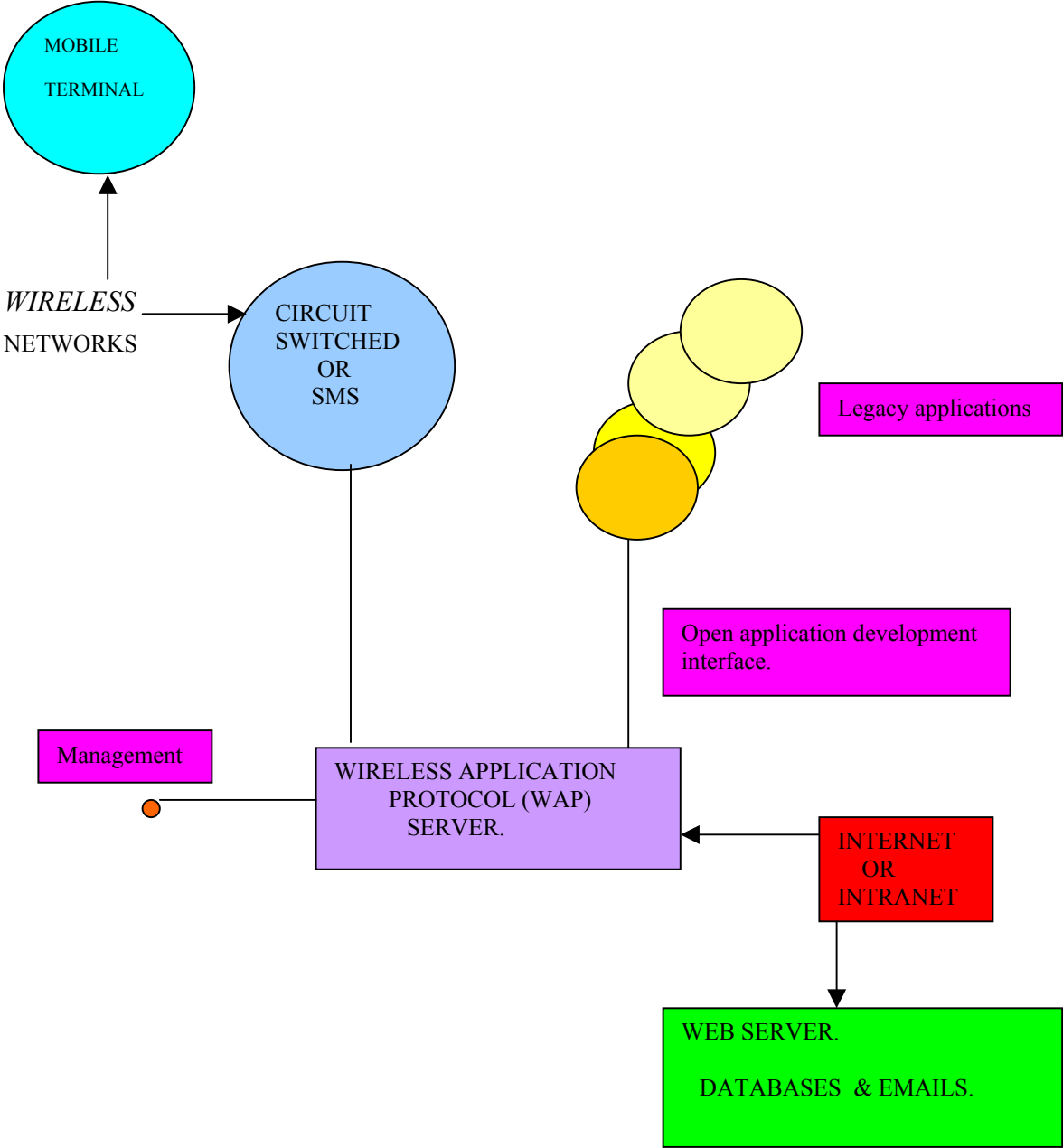
System Architecture

The basic model for WAP is that of the Web, namely a user makes a request for information using a URL. Of course, the URL may be presented to the user in the form of a hyperlink. The information is retrieved and presented to the user. Thus, at either end of the WAP system is a user and content. The user employs a WAP-compatible client device to make requests and to view the content. Two scenarios are presently envisioned for getting the content to the user.

In the first scenario, the client device is in direct communication with a "WAP Server". In this case, the client device requests content from that server, the server retrieves the content (either locally or remotely), and the content is returned to the device. In the second scenario, the client device is in communication with a "WAP Gateway". In this case, the client device requests the content from the gateway, the gateway retrieves the content, reformats it, and then the content is returned to the device. There is a subtle difference here: the WAP Server "speaks" WAP whereas the WAP Gateway translates WAP messages into another protocol, such as HTTP.

It is expected that the second scenario, using WAP Gateways, will be the initial embodiment of WAP since it will speed the deployment and industry-acceptance of WAP.

WAP SERVER AND INFORMATION TRANSFER SYSTEM



Content and Capabilities

WAP provides three broad categories of functionality for content providers: content, scripting, and telephony.

Content:

Since a WAP client device is expected to have a limited display size, the basic display unit of WAP content is a "card". A card can be thought of as one screen of information or one part of a fill-in form. In actuality, though, a card can span several client-device-screens since the client device can have a single-line display. It is then up to the client device to determine how to present the entire card as a single entity, e.g. using scrolling. A card can display text, images, hyperlinks, and input fields. Input fields can be fill-in-the-blank type, multiple-choice selections, or buttons. The client device must decide how to render these input fields, how the user navigates between the fields, and how the user enters data into or selects these fields. WAP has properly left all these things to the client device manufacturers and their human factors experts.

*As discussed, the basic display unit of WAP content is the "card". Since several cards may be required for a single application or service, WAP defines a content file as a group of cards, called a "deck". A deck has one or more cards and each card can be labeled just like name references within HTML files. When a deck is retrieved, the first card is automatically displayed unless otherwise specified in a reference field of the URL, e.g. "**foo.bar#here**".*

SCRIPTING:

WAP also defines a scripting language, WML Script, which is similar in concept and purpose to JavaScript for HTML. WML Script is created by a content provider and is stored in its own file, separate from the WML that uses it. With WML Script, a card can execute functions that verify input fields or that convey limited state information between cards in a deck or even between decks.

WML Scripts are just like functions within other high-level languages. WML Script functions are invoked by name, can accept parameters, and can return values. They can execute other WML Script functions, either in the same file or in another file, and they can execute system library functions to display input fields or to static text. WML Scripts are accessed using a URL, just like WML content itself.

TELEPHONY:

*Since it is expected that WAP will be implemented within a cellphone or other communication device, WAP has provisions for controlling the telephony aspects of these client devices. For example, a user may wish to find a local restaurant. Using a WAP cell phone, the user consults a "**yellow pages**" directory*

and finds an appropriate diner. To make a reservation, the user just "clicks" a button displayed by the WML card and the phone dials the restaurant. Obviously, this is much easier than writing down the telephone number, ending the WAP session, and dialing the number manually. The telephony control aspects of WAP are much less mature than other aspects of WAP.

WAP Layers

The WAP architecture follows the OSI layering model and consists of three major layers.

Application Layer - WAE and WTA: *Wireless Application Environment (WAE) and Wireless Telephony Application (WTA) are the top-most layers in the WAP architecture. They are the main interfaces to the client devices and specify a markup language, a scripting language, and a telephony interface. WAE and WTA impose a few, simple and basic requirements on the client device. For example, the client device must maintain a "history list" of recently visited decks, so that the user may navigate "backwards".*

WAE consists of WML and WML Script plus the WML Script virtual machine. WTA is a separate, standalone function.

WIRELESS MARKUP LANGUAGE (WML)

WML is the markup language for WAP-WAE. It is the WAP equivalent of HTML. WML formalizes the concept of cards and decks discussed in 2.2.1. Much like HTML, WML uses "tags", such as "" and "", to identify the purpose and function of blocks of text so that the client device can properly display, or render, them to the user.

WML is based on HTML and is heavily influenced by HDML, used in the Unwired Planet product. WML is XML compliant and is specified using a full Document Type Definition (DTD). The DTD allows a WML text file to be intelligently parsed and verified for correctness. In this raw, text-based form, WML is quite verbose. To make transmission of a deck more efficient, WML defines simple compression techniques such as representing all defined tags using tokens.

WML Script:

WML Script is the scripting language for WAP-WAE. It is the WAP equivalent of JavaScript. WML Script is a high level language that allows a

content provider to implement arbitrary functions that may be required by individual WML decks, for example, to verify form input prior to submitting it to a server.

As with WML, WML Script is quite verbose and a compression/compilation scheme is defined to make transmission much more efficient. Compiled WML Script consists of machine-independent byte codes, much like Java's compiled class files.

WML Script obviously requires an operating environment where the byte codes are executed in an interpreted mode. This "virtual machine" is similar to the Java Virtual Machine (JVM). WAE specifies a set of library and system functions that all client devices must implement and provide by default. This provides a basis upon which content providers may build their applications regardless of the exact client device.

WIRELESS TELEPHONY APPLICATION (WTA)

WTA, as mentioned above, provides a set of functions that allow control over the client device assuming it is a telephone device. Calls may be placed or answered. This part of WAP is still in the early stages of the specification process, so further information will not be provided at this time.

Session Layer - WSP and WTLS

***Wireless Session Protocol (WSP) and Wireless Transport Layer Security (WTLS)** is the session layer of the WAP architecture. They provide connection-based services to the application layer - WAE and WTA. Basically, a session is started, content is exchanged, and the session is later closed. Additionally, the session can be suspended and resumed. (NOTE: Although one would think that WTLS belongs in the transport layer, WAP places it in the session layer since the security context is based on WAP sessions rather than WAP transmissions.)*

Wireless Session Protocol (WSP)

WSP is the WAP equivalent of HTTP and is based on HTTP/1.1. Within HTTP and WSP is the concept of a request and a reply, each consisting of a header and body. The header is metadata - data about the data - and consists of name-value pairs that specify information about the particular request or response. The body is the payload for the WAE/WTA layer and typically consists of tokenized

WML, compiled WML Script, or images, but can also contain raw WML text. WSP, like HTTP/1.1, can convey multipart data consisting of several header-data pairs. For example, when a specific deck is requested, the server may respond with the deck, its images, and its WML Script in a multipart response. This eliminates the need for subsequent requests from the client, which delay the rendering of the deck due to the round-trip delay that would have been imposed by additional request-response exchanges.

WSP also defines a server "push" transaction where the server sends unrequested content to a client device. This may be used for broadcast messages or for services, such as news headlines or stock quotes, that may be tailored to each client device. As with other layers in WAP, WSP specifies compression techniques to provide efficient transmission of the request and response. This compression consists of one-byte tokens for "well-known" header fields plus efficient transmission of numerical data where possible, such as for date values. WSP also allows for negotiation of capabilities between the client and server, for example to determine if server "push" is supported.

WIRELESS TRANSPORT LAYER SECURITY (WTLS)

WTLS is the WAP equivalent of the HTTP SSL or TLS. Security is provided using encryption of all session data using a cryptographic technique that is negotiated when the connection is established. WTLS is not very well defined now and is currently optional in WAP.

Transport Layer - WTP and Bearer Service:

Wireless Transport Protocol (WTP) and bearer services are the protocol layer in the WAP architecture. They provide reliable transmission of WSP data packets between the client and server over a wireless link.

Wireless Transport Protocol (WTP):

WTP is the WAP equivalent of TCP or UDP. Although WTP should provide reliable communication to WSP, the current specification allows for reliable (TCP-like) or unreliable (UDP-like) communication. When the connection is unreliable, WSP (unfortunately) is responsible for retransmission to make the connection reliable.

WTP is responsible for packet segmentation and reassembly and for acknowledgement of packets and retransmission of lost, unacknowledged, or

corrupted packets. WTP numbers packets so that an at-most-once policy is effected. This ensures that a retransmitted packet is not mistaken for a new packet, which would cause duplication.

The "bottom end" of WTP consists of adaptation elements that match WTP functionality to the underlying bearer service, such as SMS or CDPD. In a sense, these adaptation elements "take up the slack" between what WTP needs and what the bearer service provides. Each client device will probably have only one adaptation element since each client device will probably have only one wireless link. It is possible, however, that a client device may have a single wireless link that it can use in different manners. For example, GSM may provide different types of data services and each would require a different adaptation element (not to mention a mechanism for the user to specify which one to use).

Bearer Services:

The bearer service is the wireless data link between the client and a server. Many different bearer services are possible: CDPD in the analog cellular system, SMS and GPRS in the GSM cellular system, and one-way (traditional) and two-way paging. Each one of these has its advantages and disadvantages in terms of maximum / typical throughput rates, round-trip delay times, and cost.

Each client device must obviously have at least one bearer service and some client devices may have several, for example, with GSM TELEPHONES.

PRACTICAL OVERVIEW OF WAP

TECHNICAL CONSIDERATIONS

From a technical perspective, WAP is very interesting and has a lot of potential. It is leveraging the industry's cumulative experiences with the Web (HTML and HTTP) by eliminating the difficulties and improving the strong points. Additionally, since WAP is targeted at handheld, wireless devices, a realm that is essentially untouched by the existing Web world, it can start with a clean slate and has thrown out the old, preconceived notions about Web services. The attention to transmission times and other considerations that are critical and unique to handheld devices is properly and justifiably focused. Savings in transmitted bytes should translate to savings in round-trip delay between a client's request and receipt of the response.

Some technical aspects and technical design decisions are inferior, however. Allowing WTP to be either reliable or unreliable is a bad choice. This forces some of the transport functions (e.g. retransmission) up into the session

layer, WSP, and breaks the OSI model. Even the tokenization of WML and of WSP messages could be detrimental if the time it takes to perform these operations is longer than the time it would have taken to just send the uncoded, raw text. Perhaps part of the negotiation between the client and server at connect time should include the option of sending raw or encoded WML. The next phase of WAP, however, will test these hypotheses and fine-tune the end-to-end performance of the system.

Market Considerations

This may be the biggest obstacle for WAP, namely, will the end-users, whether first-time or power surfers, adopt WAP into their everyday lives and make it explode the way the Web has in recent years? Several factors are at work here. First, the specification must be sufficiently developed so manufacturers can develop and sell WAP-compatible devices and WAP-specific servers and gateways. Second, content and service providers must adapt current material to the new format and must develop new applications that make use of the unique aspects of WAP.

The final applications for WAP will be an interesting development. Currently, WAP envisions specific types of services, such as stock quotes, news updates, email reading, yellow pages lookup, smart customized call forwarding. Content providers and independent software vendors have a unique way of creating new and unimagined applications for technology of this sort. The original designers of the Web never imagined the ways in which the Web is being used today. Hopefully, the same will some day be said of WAP.

Usability Considerations:

This is yet another extremely difficult question to tackle. Human factor approaches to device design are very complex. Will the user understand the paradigm? How can the information be displayed so that scrolling is intuitive? How does the user enter data into form fields using limited keypads? How are images and text rendered in a "rich" Web-like manner on small displays? Each client-device manufacturer will have its own solution to these questions. End users will certainly try all of these solutions and the market will probably stabilize on a single style of interface, much like web browsers have pretty much the same user interface.

CONCLUSION

HTML, HTTP, JavaScript, and SSL. The greatest obstacle that WAP faces is WAP is an emerging technology that has great potential. It is based heavily on existing Web technologies such as market acceptance. Will the consumer, with expectations built up from traditional Web surfing, accept the limited capability of WML decks? Second in line is the technical uncertainty of an unproven and untried system. Token compression looks promising on paper, but will the resultant reduction in transmission time be enough to warrant the complexity required by the compilers and interpreters in the client and server?

Obviously, there are many unanswered questions. These may be resolved within an amount of time as companies, both WAP Forum members and others, build and deploy systems and gain experience with this new technology. WAP complements the Web by borrowing from and building upon it, and WAP is complementary to the Web because it fills a new application space.

REFERENCES

- Wireless Application Environment Overview.
- Wireless Application Environment Specification.
- WML Script Language Specification.
- Wireless Markup Language Specification.
- Wireless Telephony Application Specification.
- Wireless Session Protocol Specification.
- Wireless Transport Layer Security Specification
- Wireless Transport Protocol Specification.

(Note: All documents are available at <http://www.wapforum.org/what/technical.htm>.)