WAP performance on an end-to-end scheme

C. Ladas, R. M. Edwards AMIEE, G. Manson, e-mail: {c.ladas, r.edwards, g.manson}@shef.ac.uk
The Centre for Mobile Communications Research (C4MCR)
The University of Sheffield

Abstract: Dealing with the high bit error rates and narrow bandwidths that typify wireless communications was not within the design goals of the Internet protocols TCP and IP. The wireless application protocol (WAP) was introduced to make better use the wireless channel and was designed using a split-connection scheme where WAP is only used for the wireless interface. This paper shows and discusses the characteristics and performance attributes of the WAP protocol along with a comparison in performance with TCP on end-to-end connection schemes. Results obtained show that WAP can be more efficient than TCP/IP not only under noisy wireless conditions but also on wired connections as well.

1. Introduction

Users of the Internet are now more dependent on on-line services. IP telephony, video-on-demand and videoconferencing are all future applications that will be driven over the mobile Internet. Smart mobile devices with accessible processors and operating systems now exist and telecommunication research is providing cellular technologies to address the high bandwidth requirements that are inherent in the aforementioned services. Although bandwidth is increasing for the mobile devices, because of limited available radio resource it is unlikely that such channels will ever provide the levels that can be offered by wired connections. Besides the low capacity, a wireless interface has also inherent high Bit Error Rates (BERs), which results to significant performance degradation [1,2]. The data recovery mechanism that resides within higher protocol layers (TCP) takes no account of any other media than wired networks. Consequently packet loss is attributed only to congestion and the flow-control mechanism effectively constricts the data flow of the session. For these reasons the Wireless Application Protocol (WAP)[3] was introduced, which provides more efficient recovery and less bandwidth requirements.

The first specification of WAP was released in 1998[3] and proposed a two-part connection scheme (split connection) for accessing web content as shown in Figure 1a. The WAP protocol is used between the Mobile Node (MN) and a WAP gateway (WG), which forms the first point of ingress to the Internet, and a regular HTTP/TCP connection is further used between the WG and the Content Server (CS). The role of the WG is to bind WAP together with the Internet world.

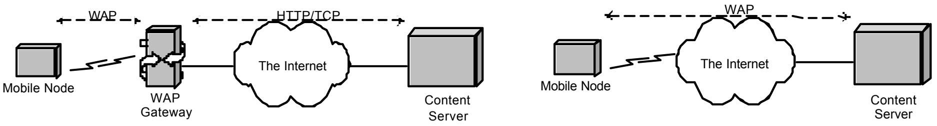

Figure 1 - WAP browsing setup, (a) with gateway, (b) end-to-end

After leveraging the first implementations it was realised that aside from typical web browsing through the WG, end-to-end connections might be required either for security reasons or for specific applications as illustrated in Figure 1b. To provide such service it is required that both ends (the MN and the CS) have a WAP protocol stack and use the IP protocol for network data delivery.

This paper focuses on the "end-to-end" scheme of WAP and a comparison with TCP over lossy links. Furthermore, we present an in depth analysis of the characteristics of WAP that are germane.

1.1 Structure of this Paper

In section 2, the wireless session protocol (WSP)[4] and the wireless transaction protocol (WTP)[5] are discussed with particular reference to the way in which they are designed to optimise wireless communication. Section 3 then presents an experimental comparison of WAP to the more conventional HTTP/TCP. In both cases end-to-end connections are assumed. Interesting and significant results are then shown in section 3.2 along with the experimental set-up and statements relating to the necessary assumptions made. Finally, section 4 presents detailed conclusions.

2. Improvements for the wireless interface

Typically, a WAP protocol stack consists of five distinct protocol layers. The stack itself sits on top of a network protocol that handles data delivery between a MN and the WG. Classical implementations run on top of IP which itself is run on a wireless bearer. The significant performance improvements demonstrated by WAP can be attributed to several mechanisms. The most important of these are due to WTP that provides reliable transport between the end nodes. Also WSP contributes somewhat by giving reduced overhead when compared with HTTP, by using a binary rather than a textual form. Further enhancement is gained from the Wireless Application Environment (WAE), which contains the service payload that is compacted. WAE is not further considered in this study since this work focuses on the comparison of the communication protocols.

2.1 Wireless Session Protocol

Wireless Session Protocol (WSP)[4] is the session layer protocol of the WAP protocol stack and also incorporates features for requesting and receiving resources in a similar way to HTTP. There are two modes of operation; *connection-less* and *connection oriented*. Connectionless refers to the type of session where each transaction is totally independent, whilst the WTP is not used implying unreliable message delivery.

In the second mode (mode-2), a connection-oriented session is established using a handshake between the client and the server (or WAP gateway). After a successful handshake transmission or receipt of data begins. Since the connection-oriented mode uses WTP, the upper layers are provided with an interface that can reliably transfer any type of message, restricted only in size[1]. The WSP messages are therefore mapped to WTP transactions that handle the reliability and consistency of the end-to-end delivery of a message. Here, since we focus on performance issues, only connection-oriented sessions will be investigated further. The request/reply mechanism of WSP is similar to HTTP but the data is in a binary-compacted form rather than textual. This is the case for HTTP. In this way WSP achieves a compaction rate (in some cases it reaches 75%), which contributes to the overhead reduction when compared with HTTP. The operation of WSP is illustrated in Figure 2.

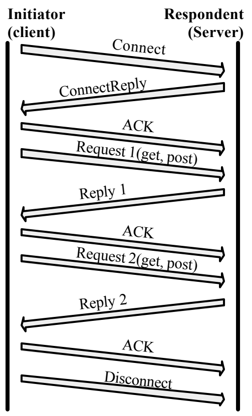

Figure 2 - typical WSP session

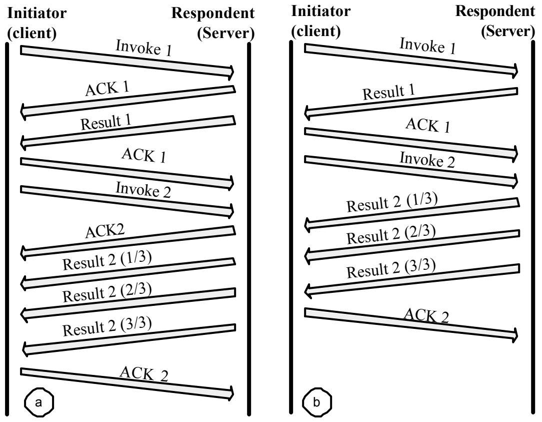

Figure 3 - typical WTP Class-2 transactions (a) explicit, (b) implicit

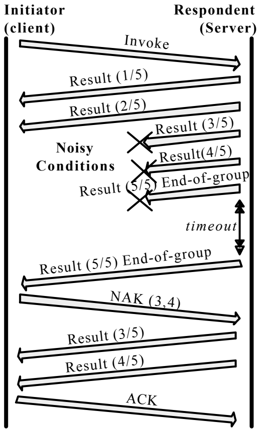

Figure 4 - WTP packet loss recovery

2.2 Wireless Transaction Protocol

The Wireless Transaction Protocol (WTP)[5] is the module responsible for handling reliable in-order message delivery, as well as fragmentation and reassembly for the ones that exceed the size of the network path (Maximum Transfer Unit). There are three classes of operation for this protocol; class-0 which is used to send packets to the network without acknowledgements, class-1 which requires an acknowledgement for every transmitted message and class-2 which is similar to class-1 with an additional reliable reply message. Furthermore, the class-2 of WTP can operate in two modes; explicit and implicit. In the first case every invocation has to be acknowledged before a result is sent, whereas in the second case (implicit mode) a result packet is considered as an acknowledgement of the invocation packet. Acknowledging the invocation message

---

[1] The WTP protocol may fragment a message in up to 256 fragments due to a single octet in its header. This octet indicates the segment number. The WAP Forum recommends that the Maximum Transfer Unit is 1400 octets implying that the maximum message size can be 256 × 1400 ≈350KB.

prevents the sender's timer from expiring in cases that the request requires additional time to be processed. Figure 3 presents the operation of Class-2 transactions in explicit and implicit mode.

Flow control in cases of fragmented messages, is performed by sending fragments in groups. Every group of packets requires only one acknowledgement, and this acknowledges the entire group. The last packet of each group contains a flag that indicates the end of the group in order for the receiver to know when to send an acknowledgment. At this stage it should be noted that the size of each group is implementation specific, and usually depends on the link characteristics and the device memory. Congestion control has not been anticipated, so implementations have to be conservative in order to avoid unnecessary packet retransmission and buffer flooding with associated data loss. If the end-of-group packet is received whilst intermediate packets are missing, the receiver sends a negative acknowledgement (NAK), which explicitly indicates the missing packets to the sender. This operation is repeated until the entire group is received and an acknowledgment is sent. In the case where a timeout occurs, only the last packet of the group is retransmitted in order to trigger the NAK mechanism that will precisely designate the missing packets (see Figure 4). In this way, the WTP protocol minimises the number of unnecessary retransmissions. Thus increases in efficiency and reduced latencies are provided by the protocol.

3. Performance comparison

3.1 Experiment setup

An experiment was designed to compare the performance of WAP in an end-to-end deployment with a typical HTTP/TCP scheme as shown in Figure 5. Elements are a content server, running on Microsoft Windows 2000, a web server and a WAP protocol stack[2] connected via Ethernet to a Linux box running NISTNet WAN emulator[6]. The purpose of this box is to route packets between the two networks involved and discard packets according to a certain pre-specified ratio. The WAN emulator is connected to the Packet Sniffer, which is used for packet tracing and measuring and also hosts serial connections from other nodes. The client is connected to the Packet Sniffer via a serial cable that is set to a 9.6kbps data rate and emulates a GSM dial-up connection.

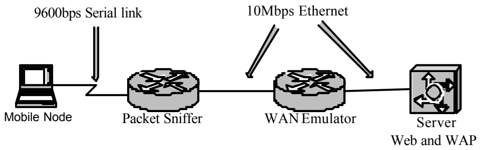

Figure 5 - Testbed setup

To measure TCP performance, a simple application was implemented which places an HTTP request over TCP to the server, to which the server replies with a binary file. The Maximum Transfer Unit (MTU) for both protocols is set to 1500 octets in order to be comparable and close to the 1400 octet WAP recommendation. It is important to note that the TCP implementation that we have employed uses selective acknowledgements (SACK), a feature which seeks to recover multiple segment losses effectively. For the WAP experiments, a similar program was implemented, which places WSP over WTP requests to the server. The content server runs a web server on top of the TCP/IP and WAP protocol stacks in order to serve the clients. The WTP protocol was configured to run in explicit acknowledgement mode (see above). For operational performance comparison, the payload data is considered to be a raw data stream that cannot be compacted. A complete measurement included a connection setup, request/reply and connection teardown/disconnect. The file sizes used for this experiment varied in size from 1000 octets to 63500 octets with a steps of 1500 octets. Experiments were performed without packet drops in order to measure the performance of the protocols under noise-less conditions and with a high rate of packet drop (10% packet drop for both directions).

3.2 Results

At start it was necessary to judge the impact of the various overheads for the two protocol stacks. This was relevant to both the possible fragmentation issues in WAP and the differing acknowledgement schemes between WTP (groups of packets) and, TCP (individual packets in most cases). Therefore, an analytical calculation of efficiency for both protocol stacks was undertaken to confirm that WAP protocol overheads were not a significant factor in reducing the download time of a file. From the results presented in Figure 6 it is gathered that the efficiency gain for small files can reach up to 5%, whereas for large ones (above 20Kbytes) it converges to 1%. This analytical result confirms that WAP is always more efficient in terms of protocol overhead.

---

[2] The WAP protocol stack has been designed and implemented at the C4MCR laboratories

Using the test-bed described above, the goodput comparison of WAP and HTTP/TCP is presented in Figure 7. This illustrates a number of important facts about the performance of the WAP and HTTP/TCP stacks.

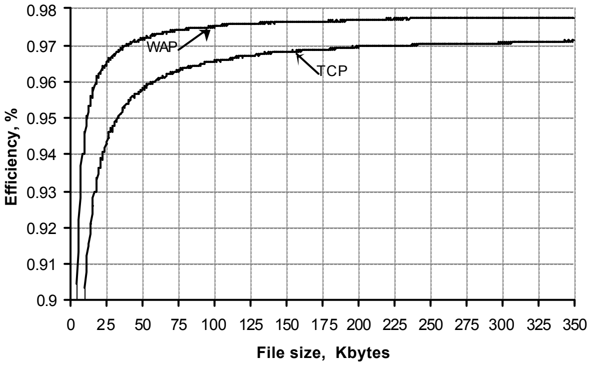

Figure 6 - efficiency of WAP and HTTP/TCP browsing

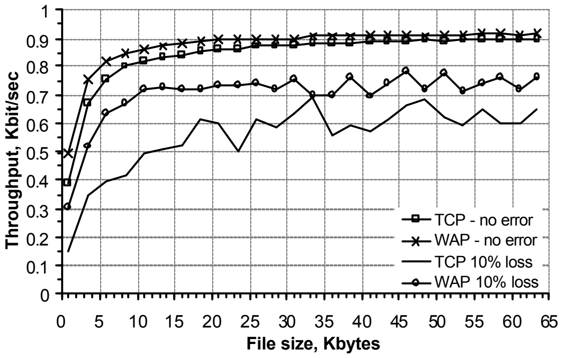

Figure 7 - goodput comparison under noisy conditions

For the error-less comparison WAP achieves higher goodput than the typical HTTP/TCP. This confirms expectation from the previously presented efficiency figures in Figure 6. This implies that even for a wired network WAP increases the performance of the network in terms of throughput.

For the lossy comparison, the recovery mechanism of WTP improves the goodput of the connection radically at rates of between 16% and 107%. These depend on the file size. It can also be seen from Figure 7 that WTP performs very well for small files that is particularly pertinent with mobile devices. Conversely, TCP exhibits relatively long delays when recovering if it uses small windows. Such delays can be attributed to congestion and packet losses. Moreover, when considering the flow of the sessions and looking in Figure 7 at the knee of the curves for the 10% loss comparison, it can be seen that WTP becomes stable more quickly at 12Kb file size with better channel efficiency as against TCP, that due to its congestion-avoidance algorithms, stabilises only after file a sizes of at least 35Kb.

4. Conclusions

This paper has summarised the basic functional principles of the WAP protocol stack with special emphasis to the WTP and WSP protocols. Furthermore, we have schematically presented the recovery mechanism and shown how unnecessary retransmission is minimised. Results presented illustrate how the wireless resource can be more efficiently unutilised. A comparison of the two topical protocol stacks is shown that in a simulated lossy environment performance gains can be achieved using WAP as against TCP. Enabling of selective acknowledgement in TCP does not negate this result. It is therefore proposed that for a slow and lossy links WAP can significantly increase efficiency for end-to-end data transfers.

Acknowledgements

The first author would like to thank Dialogue Communications (www.dialogue.co.uk) for funding this work.

References

[1] B.S. Bakshi, P. Krishna, N.H. Vaidya, D.K.Pradhan, “Improving performance of TCP over wireless networks”, Proceedings of the 17th International Conference on Distributed Computing Systems (Cat. No.97CB36053). IEEE Comput. Soc. Press, Los Alamitos, CA, USA; 1997; xvii+596 pp. p.365-73.

[2] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, R.H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links”, IEEE/ACM Transactions on Networking (TON), v.5 n.6, p.756-769, 12-01-1997

[3] The WAP Forum, "WAP Architecture Specification", Version 30-Apr-1998, http://www.wapforum.org

[4] The WAP Forum, "Wireless Session Protocol", Version 30-Apr-1998, http://www.wapforum.org

[5] The WAP Forum, "Wireless Transaction Protocol", Version 30-Apr-1998, http://www.wapforum.org

[6] National Institute of Standards and Technology, “NIST Net WAN emulator”, version 2.5, http://www.antd.nist.gov/itg/nistnet/