<Hewlett Packard Laboratories, Bristol>

SWAP and TCP performance

Jean Tourrilhes, HPLB

23 March 98

1 Introduction

The SWAP protocol that we have proposed [4] the HRFWG is designed to carry TCP/IP traffic. Of course, we would never had proposed such a protocol if the performance with TCP/IP was not the one expected, and great care has been taken for a smooth integration with TCP/IP [6].

However, at the last Network Committee of the HRFWG, some concerns have been raised about the performance of TCP over SWAP and people have suggested that we investigate using the SNOOP protocol [7].

The SNOOP protocol is already well known to us, and previous research has concluded that SNOOP was useless on top of SWAP [5]. I will summarise our findings below.

At the same time, I will explain how we can tune a SWAP implementation for optimal TCP performance.

2 SWAP philosophy

The original target of SWAP (in its CSMA/CA part) it to offer to the higher layer an interface as similar as possible to Ethernet. This is done with the following feature :

- Ethernet channel access (CSMA/CA) - asynchronous broadcast shared medium
- Ethernet plug and play - no configuration, no coordination
- Ethernet flexible packet size - through SWAP frames and fragmentation
- Ethernet Framing - easy mapping Ethernet to SWAP
- Ethernet reliability - MAC level retransmission reduces losses to better than 10^{-9} Bit Error Rate in almost any cases within the usable range

Offering a interface similar in every way to an Ethernet interface (framing, behaviour...) guarantee that SWAP could be integrated transparently in any computing platform through a simple driver. TCP/IP and many other protocol such as NetBeui and IPX have been designed over Ethernet and optimised to it, so they will offer a maximum level of performance and unrestricted services over SWAP.

On the other hand, the TDMA service offers a point to point service, like a wire or a serial link. The real-time retransmission mechanism of SWAP and the DECT management offers a reliability comparable to a wire. The SWAP connection is seen exactly as any point to point connection (parallel cable, phone modem...), with very similar characteristics.

This, of course, doesn't offer as much flexibility and performance as an Ethernet interface. The model in this case is to put PPP over this link and TCP/IP over PPP. All the work of putting TCP/IP on top of DECT can in fact be directly translated to SWAP.

3 The SNOOP protocol - How useful it is ?

The SNOOP paper [7] of the students of Berkeley describes different means to improve the performance of TCP data transfer over wireless links.

The main problem of wireless links is that the losses caused by channel errors are interfering with the congestion avoidance algorithm built in the TCP protocol. TCP is built as an end to end reliable connection, and retransmission mechanisms are implemented in order to recover from packet losses on the lower layers.

But, TCP was designed for wired medium, which have a very high reliability (Ethernet - 10^{-9} Bit Error Rate), and all losses are assumed to be due to congestion (either saturation of the medium or overflow of buffers in the routers or switches). The wireless medium has a much higher probability of failure (10^{-5} Bit Error Rate at the sensitivity - limit of range). Those packet losses on the wireless link are perceived by TCP and assumed to be due to congestion of the network, so TCP slow down in order to decrease this imaginary congestion and under-utilise the wireless medium.

In the paper, there is different class of algorithm presented to deal with this problem, some changing the behaviour of TCP, some splitting the TCP connection, and Link Layer retransmission schemes. Link Layer retransmission scheme offers the advantage of being transparent to TCP/IP, and according to their paper to offer the best performance [7].

A link layer retransmission scheme attempts to detect the packet losses at the link level and to retransmit packets before TCP notices their loss, giving to TCP the illusion of a reliable link.

In their paper, they describe and compare several link layers schemes. Their basic Link Layer scheme (LL - in chapter 3.2 of [7]) just attempts to detect losses of packets and retransmits them. Their best scheme, the SNOOP protocol, is also a link layer scheme, adding to LL some ack filtering for better performance.

The LL scheme is one of the best scheme, but not as good as SNOOP. They explain that the only cause of the bad performance of their LL scheme (see chapter 4.2 of [7]) is that "... link layer protocols that do not attempt in-order delivery across the (e.g., LL) cause packets to reach the TCP receiver out of order. This leads to the generation of duplicate acknowledgements by the TCP receiver, which cause the sender to invoke fast retransmission and recovery...". The SNOOP protocol is used to compensate that effect.

The SWAP protocol implements MAC level acknowledgement and retransmission (so, SWAP implement a Link Level retransmission). But, the SWAP scheme is not the same as LL and is not studied in their paper. Basically, the SWAP scheme is the same as LL, but based on a Stop and Go mechanism (instead of an infinite window). The Stop and Go mechanism guarantee the "in order" delivery of packets over the link (the sender makes sure to get the packet across before attempting to send the next one in sequence).

Because it uses a Stop and Go mechanism, SWAP doesn't suffer from the same deficiency as LL, and offers the same performance as SNOOP. On the other hand, SWAP doesn't have to mess with TCP and works equally well with non-TCP transports.

The same authors warn as well that Link Level retransmission, because it adds delay to some packets, could interfere with TCP timers and round trip estimate. The SWAP scheme use ACK packet embedded in the transmission frame and therefore detects losses immediately (in a matter of microseconds) and attempts retransmission before sending any other packet. The net effect is that a retransmission just adds the delay of contending for the medium and retransmitting the packet (a few milliseconds, maximum 5 ms on a idle network). On the other hand, the SNOOP protocol uses a timer of 200 ms to detect losses, so retransmissions are attempted only after a minimum of 200 ms (chapter 3.2 of [7]).

The result is that the increase of delay of retransmitted packets caused by the SWAP scheme is 2 orders of magnitude lower than SNOOP. This helps to offer TCP a more uniform round trip variation and less latency over the link, so is much more transparent to TCP and offers a better performance. This also insures that SWAP will work with version of TCP that would have a lower granularity of the timers than the one used in this study (most TCP stacks are going towards low granularity timer to perform well on ATM and Gigabit Ethernet).

SWAP, because it uses MAC level retransmission, never looses a packet (provided enough retransmissions are done), so the main problem described in the article doesn't apply to us. Furthermore, as SWAP never generates packets out of order, SNOOP is not needed in the case of SWAP. SWAP timers are much smaller than SNOOP timers, so the performance of the SWAP scheme is better than the SNOOP protocol.

4 Tuning SWAP for optimal TCP/IP performance

TCP/IP is a very complex protocol, and its interaction with the MAC layer are never easy to model and simulate. The aim of the SWAP protocol is to offer a premium TCP/IP performance, but a lot of implementation factors have to be taken into consideration.

This section describe how to maximise the performance of TCP over the SWAP protocol and the effects that may reduce the performance of TCP transmissions.

4.1 The TCP/IP stack

A good TCP/IP performance requires a good TCP/IP stack. This sounds like an evidence, but nowadays a lot of TCP/IP stacks are still not implemented totally correctly and lack the optimisations found in modern Unix TCP/IP stack.

A good TCP/IP stack must implement :

- Coalescence, and good management of it
- Fast Recovery and Fast Retransmission
- Good round trip estimate with low granularity timers
- Efficient transmission window management
- Clever acknowledgement strategy (less acknowledgements messages)
- SACK (Selective Acknowledgements)
- Low overhead (minimise the number of management or useless packets)

The impact of the TCP/IP stack quality is especially important for bad link layer. For example I noticed that the Linux TCP/IP stack was giving me a 30 % performance increase compared to another common TCP/IP stack while testing a common wireless LAN product.

SWAP is a good link layer, shielding TCP/IP from most of the effect on the radio, so the quality of the TCP/IP stack is less of a problem. Furthermore, SWAP has been designed to offer good performance whatever the quality of the TCP/IP stack, by offering a link having characteristics and behaviour as similar as possible as Ethernet. But, obviously, optimal performance requires a good TCP/IP stack.

4.2 The SWAP implementation

The implementation of the SWAP protocol, network adapter, hardware interface with the PC and driver are also a concern. Modern Ethernet cards offer a better performance than the first generations because the implementation has been greatly optimised.

A few rules are :

- No packet losses or corruptions between the baseband and the OS
- Enough throughput and low latency for the hardware interface
- Low latency in the MAC, the network adapter and the driver
- Enough buffers in the adapter to compensate host latencies, especially in reception
- Good implementation of the channel access mechanism
- Good RSSI and state channel assessment implementation

In fact, a good implementation could reuse the techniques and interfaces of modern Ethernet cards. The fact that we are using 802.3 compatible frames [6] allow us to be able to reuse the network adapter engine and interface specification of any Ethernet card.

4.3 Packet timeout tuning

The retransmission algorithm of the SWAP protocol doesn’t use a limit on the number of retransmission but a timeout. Tuning this timeout can help to get better performance.

The rationale is that the retransmission algorithm is constrained by two factors : first to get enough reliability to shield TCP from link losses, and second to limit the increase of delay to avoid interfering with TCP timers.

TCP performs its own retransmission when it detects a timeout on the network. The granularity of TCP timer is usually 100 ms, and on a single link network the timeout, based on the round trip estimate, is only a few times that figure. So, no retransmission on the physical layer should exceed this boundary, because at this point TCP performs its own retransmission.

At the same time, there is little point on putting an upper limit to the number of retransmissions. Fading varies upon the time and doesn’t have Gaussian properties, so the packet will get through when the channel characteristic is optimal, not after a certain number of trials. The protocol should try as much as possible to get the packet through, to avoid TCP believing that there is an imaginary congestion. This is why we have set a packet timeout (TxRxPacketTimeout) instead of a limit of retransmission.

In SWAP, the retransmission mechanism can take advantage of the short dwell. In 100 ms, the transmitter is able to try transmission on at least 5 different channels. A delay of 100 ms also enables the transmitter to defeat most fade periods.

If the medium is lightly loaded, in 100 ms, we can perform up to 20 retransmission which ensure a very good reliability. At the sensitivity (limit of range), the Bit Error Rate is 10^{-5}, so the resulting the Mac Failure Rate of a 512 B fragment is 10^{-28} ! For a heavily loaded network (which is not the target of SWAP), the protocol might only be able to perform a few of them (5 to 8 retransmissions - this still gives SWAP a very good reliability and much better than a 10^{-9} Bit Error Rate).

Having a variable number of retransmission dependant on the load of the network is in fact acceptable : when the network is lightly loaded, few people are using the network, so the high number of retransmission (using network capacity) is not a penalty ; and when the load is high, the packet might be dropped after a few attempts, signalling to TCP that there is a congestion (this congestion is real - and this is the usual way to signal this to TCP).

My simulations of the SWAP protocol [4] have confirmed those results. On a two node network (with both nodes fully loaded), with any interferer present in the band (microwave oven, Frequency Hopping or Direct Sequence interferer), the protocol was losing less than one packet per 25 s up to the 43 m range.

The value of 100 ms for the timeout has been chosen as a “good compromise”, results of our analysis and experience, but some experimentation and simulations on the prototype might allow to tune this parameter for a better performance. A correct value of this parameter should allow a smooth and transparent transition between the SWAP and the TCP retransmission mechanisms.

4.4 SWAP protocol enhancements

The SWAP protocol can be modified in a number of different way to improve the performance [6]. We have designed a number of schemes totally transparent to the SWAP protocol, increasing only slightly the complexity of the MAC and offering some specific performance improvements (note that some of those scheme are applicable to 802.11 as well).

The implementers are free to implement any enhancement to the protocol, as long as they make sure to still comply to the SWAP specification and interoperate with other SWAP devices. But usually, those enhancements work only if the other parts of the implementation are correctly optimised and the TCP performance already good, and it offers only a limited additional gain.

5 References

[1] *IEEE 802.11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications.* IEEE.

[2] *Radio equipment and systems (RES) ; High PErformance Radio Local Area Networks (HIPERLAN), Type 1, Functional specification.* ETSI.

[3] ETR 015. *DECT Reference Document.* March 1991, ETSI.

[4] *DECTplus proposal to the Home RF Working Group.* Hewlett-Packard.

- Part 3 : *DECTplus architectural overview.*
- Part 4 : *DECTplus common air interface specification.*
- Part 5 : *System performance, simulation of DECTplus.*

[5] Jean Tourrilhes. *Wireless Networks and TCP/IP.* Internal memo. Hewlett-Packard.

[6] Jean Tourrilhes. *Swap protocol improvements.* Internal memo. Hewlett-Packard.

[7] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz. *A comparison of mechanisms for improving TCP performance over wireless links.* Proc. of ACM SIGCOM ‘96.