

TCP Header Compression according to Van Jacobson via AX.25

*Gunter Jost, DK7WJ/K7WJ
Lichtenbergstrasse 77, D-64289 Darmstadt, Germany
Translated by: Don Rotolo, N2IRZ
c/o RATS, PO Box 93, Park Ridge NJ 07656 USA*

Abstract:

The Van Jacobsen scheme for TCP/IP header compression is briefly introduced, and an implementation of this system under FlexNet is described and discussed.

TCP/IP requires a header of at least 40 bytes for each packet. In connections over Ethernet carrying packets with little data, as in Telnet and similar mostly inactive sessions, this header adds considerable weight to be carried, making slow connections highly undesirable. Often there is only a little information carried within each packet, in the extreme case a single key click. Above all the reaction time becomes intermittently high. Once, when phone modems mainly worked at 2400 baud, Van Jacobson had taken up this problem and offered a technique in RFC 1144 [I].

With the Van Jacobson scheme, the header is compressed to be only 3 to 8 bytes in length. Interesting, and relevant to the implementation, is that the compression requires a memory table for each TCP connection. Only the differences from the last sent packet are transmitted. Frequently occurring special cases are handled in a specific manner and also improve the compression ratio. For example, the transmission of a TCP sequence number is mostly superfluous, because it can be calculated from the number and length of the packet.

TCP retries are generally carried uncompressed, so that the distant end can resynchronize. This is also valid for special, infrequent packet types (such as SYN and RST).

The necessity to store in memory the present status of each TCP connection requires naturally some computing capacity and memory space. The integration into the TCP layer was impossible, since it is a black box, and routers should also be able to handle compression. There is also the question of how many simultaneous connections should be realizable. For user stations 16 TCP connections should be sufficient, for routers it should possibly be more.

If there are more connections than slots available, the slot that is unused for the longest time is reassigned. It can also be a little tight with routers here. It is however not a problem to assign slots to stations using other criteria, for example according to the frequency of your usage.

The entire RFC won't be repeated here, mostly since it is over 40 pages long. The Protocol contains further details, and it is very interesting and well thought out. As far as the author knows, it has not been translated into German.

Implementation

The RFC contains already a practical implementation suggestion in C, which is a good platform upon which to build. Following a suggestion from Thomas Sailer, HB9JNX, I implemented the algorithm in the FlexNet IP coupler. To eliminate the need to implement the "hacks" Jacobson suggested for the transportation over SLIP, two new AX.25 PIDs were found to be needed for proper identification. After a review of all the available information, PIDs 6 and 7 were found to be available. When we informed the ARRL of this selection, they seemed surprised that we assumed them responsible for PID coordination, as was written in the AX.25 protocol definition some 13 years ago. At present, we are awaiting a confirmation of this coordination [2].

PID 6 denotes a compressed TCP/IP packet, with PID 7 an uncompressed packet being transmitted with the Jacobson scheme. That isn't the same as a packet without compression: In an uncompressed packet the slot-information remains, that is used for the compressed packets that follow. Naturally PID 8 used, as is known, for segmented IP packets. These also contain the original PID, and combinations of PID 6 through 8 as well as CC can appear. A segmented packet can contain a compressed header, thus it has PID 8 as well as the PID 6 or 7 packed within.

It is important to understand that the compression runs continuously point to point. The tables are organized as in an AX.25 QSO, and this functions only in Virtual Circuit mode. If a connection is rerouted or re-established, it must not be built on old status tables. One can not coordinate the compressibility of an IP destination, as this belongs always to an AX.25 station or partner.

A disadvantage of the Von Jacobson specification is stated within, that the "partner" cannot provide the slot quantity required, but this should also not be different, since packets can get lost when the receiver has fewer slots than the sender. With the FlexNet implementation this is resolved by allocating a fixed 16 slots to the sender and up to 255 slots dynamically to the receiver, the maximum allowed under the specification. According to the definition, a Link Reset (reconnect) forces both sides of the connection to delete the relevant tables. In this manner, memory can be conserved.

There are no problems with data reliability. Each TCP packet is protected by a checksum. Additionally, all of the measures of an AX.25 connection are used. As long as the L2 connection remains, it is guaranteed that all packets arrive in a defined order. With a Link Reset, all the compression tables are deleted. In this way the defined continuation of the TCP connection is guaranteed, and the inadvertent confusion of a compressed packet from another TCP connection with the same slot number is safely eliminated.

Matthias Welwarshy, DG2FEF, has tested this behavior in WAMPES under LinuX, without the source code being exchanged between us. In this way, we have a high level of confidence that our implementations comply with Jacobson specification. In a multi-week test, run between four stations at different user ports, there were no unmotivated disconnections of the TCP connections. Last but not

least. this shows that the behavior is very robust against sporadic re-establishments of the logical (AX.25) connection.

Conclusion

We can conclude that the Van Jacobson compression scheme provides a very good result **with** minimal difficulties. One must ask himself, why this concept has not (in the author's knowledge) been previously applied to Amateur Radio.

One disadvantage is that, at present, one of the two partners must switch the compression on manually. It must be assured that the other partner controls the system. In a further development, this facility should be handled automatically.

References:

- [1] Jacobson, V., Network Working Group, Request for Comments 1144, February 1990.
- [2] Also awaited for quite some time is the coordination of PID CE for FlexNet.