

A Note on Urgent in TCP
by
John Day

The design of an out-of-band signal or attention for protocols has been a topic of considerable discussion lately both here and in Europe. Several protocols and operating systems have adopted similar models for out-of-band signals. (For example, X.25, CYCLADES TS, EIN VTP, Multics IPC, etc.) Although, these differ in various details and the implementation mechanisms may be different, there is a consistent common model:

Associated with each communications channel, there is another independent channel which is the out-of-band channel. This out-of-band channel is not subject to the flow control constraints of the primary communications channel. The information transferred on this out-of-band channel is generally used to notify the other side of a change of state or to send a command that must be acted upon as soon as possible. The urgent nature of the data requires that the receiver always be able to receive data on this channel. (In most implementations this means that there is very little or no flow control on the out-of-band channel.) Traffic on this channel consists of very short messages (usually less than 100 bits) and sent very infrequently (usually seconds or minutes between messages). Because of the urgency of the information sent on this channel, it is sent at a "higher" priority. Whether or not an actual priority message system is used is not important. It is important that delivery of messages on the out-of-band channel be expedited at every reasonable opportunity to assure delivery at the earliest possible moment.

TCP as it is presently defined supports a different model of an "out-of-band" signal. The logical model used in TCP consists of an out-of-band channel that can convey one bit of information. The meaning associated with this bit is "an urgent condition has been indicated". All information indicating the nature of the urgency is in the primary data stream and is subject to controls on it. The only indication TCP gives of where the urgent data is in the data stream is a pointer. This pointer indicates that the urgent data appears somewhere before this point in the data stream. It is assumed that the user will recognize the urgent data when it is processed. This model is a minor variation of the model used in the Telnet protocol.

There are several problems with this scheme.

- 1) Suppose that there are two or more layers of protocol on top of TCP. An interrupt is sent by the upper layer. This interrupt is mapped into the TCP urgent mechanism. On receipt at the other side the only way the interrupt can be expeditiously handled is

for each layer to read past the urgent pointer as quickly as possible. Since only the upper layer knows whether or not it is safe to discard data, this could lead to a considerable amount of data buffered locally. This would be especially true if each protocol layer on the sending side tried to expedite matters as much as possible and indicated the urgency on the very next packet it was passing down to the next level. The model described above does not have this problem. Since interrupt data is carried in a separate logical channel, the data can be passed independently of the data stream.

2) Consider the following situation. An upper level generates an interrupt with data due to some error condition. This error causes an intermediate level to also generate an interrupt with data. Because of asynchronys in processing it isn't specified what order these two are mapped into urgents in the TCP. Is it possible for one of the lower levels to filter out an urgent such that an upper level is never notified? Is it possible for the interrupt data to be filtered out? (I doubt it.) The specification is not very clear on exactly what happens with successive urgents. It is unwise to assume that they are all generated by the same user.

3) The TCP urgent mechanism cannot support the model described above without the addition of another layer of protocol on top of TCP. There are three major reasons for this:

1) The urgent data cannot be found in the data stream by TCP. The urgent pointer indicates that the urgent data occurs before this point, not that it ends at this point. Also, there is no indication of how much urgent data there is.

2) The TCP user interface does not provide a mechanism for passing the urgent data out-of-band to the user.

3) It is assumed that if the user encounters the urgent data in-band, he will recognize it.

Imposing another layer of protocol or requiring all higher level protocols which need the facility to provide it seems like a heavy price to pay for such a simple facility. If TCP were to adopt such a model the range of protocols and IPC disciplines (such as those found in Multics and other operating systems) could be greatly increased. The urgent model described here can support the current TCP model, but not vice versa.

4) Recently Healy has pointed out that the current TCP can lose urgents. Healy suggests that to fix this problem all urgents be forced to send data. Although, this fix is consistent with TCP philosophy of mixing control and data, it may require TCP or many higher level protocols to define a null data element. This could cause more problems than it solves.

One could also point out that the TCP urgent isn't really out-of-band, but this is merely a trade-off between the efficient and timely delivery of the urgent. I am more concerned with providing the necessary facilities to support the environment TCP is liable to encounter. I am not suggesting that TCP explicitly adopt the out-of-band signalling model described here, but only that it give the upper layers the illusion of supporting it.