D. Cohen
USC-ISI
11 August 1978

Some Thoughts About the Multiplexing Issue in Networks

For several years we have enjoyed the beauty and the structure offered by the modular approach to protocols. This approach leads to the separation of communication layers (headers, etc) according to their originating and receiving agencies (processes, protocols etc).

It is very unfortunate that an undesired by-product of this approach is the failure to merge communication units (messages) which have a great deal of common communication attributes, even when they have some small differences.

A case in point is the inability to introduce multiplexing arbitrarily into the protocol-tree stucture.

It is intuitive that it is very desirable to be able to merge TELNET messages between the same source/destination hosts pair, especially when a small number of characters are communicated in each message.

Similarly, NCP and TCP messages which are addressed to the same destination host, could be merged even though they are between different protocols.

The same approach applies on ALL levels.

It is important to be able to use this kind of multiplexing in order to share (and therefore save) headers, or parts of headers, and in order to save switching time.

Consider a simple example, voice communication, using NVP through TCP (just for the example), and internetting through the SATNet. The voice data rate is R, and a message is sent every T time.

The amount of voice data in each message is RxT, added to that is the NVP header of length Lv, the TCP header of length Lt, the IN header of length Li, and the HOST/SIMP header of length Ls.

Hence, the communication efficiency is:

$$eff = \frac{R \times T}{R \times T + Lv + Lt + Li + Ls}$$

Assume the following numbers:

R = 2,400 bps
T = 200 milliseconds, for interactive communication
RxT = 480 bits, for a packet of voice data
Lv = 48 bits, including local-extension, time-stamp, parcels count and silence indication.
Lt = 160 bits, including the PORT.
Li = 160 bits, (without the optional OPTIONS field)
Ls = 96 bits, according to PSPWN-100

For these numbers the efficiency is:

$$eff = \frac{480}{480 + 48 + 160 + 160 + 96} = \frac{480}{944} = 50.8\%$$

If N voice communications between different "extensions" in the same hosts pair are multiplexed, then the efficiency (again, at the interface, not in the communication media) is:

$$eff = \frac{R \times T}{R \times T + Lv + (Lt + Li + Ls) / N}$$

Which for N=2 is 65.2% and for N=3 is 72.0%. It is obvious that the efficiency increases with N, and that in this case its limit is 90.9%.

This is, obviously, the efficiency over the HOST/SIMP interface. The more important is the efficiency over the communication media, which is lower than that, due to the SIMP-to-SIMP communication overhead.

Another, more familiar, example is sending a single character over the ARPANET. In this case the payload is 8 bits, which are preceeded by 40 bit NCP header and the 96 bit HOST/IMP header. This results in eff=8/(8+40+96)=5.5% over the interface, and even lower over the 50kbps lines.

Just think what is the efficiency of sending a single character, or even a few, over the SATNet... The numbers cannot be very encouraging...

Next Subject: Multi-address

For certain applications there is a case for multi-address, namely, asking the communication system to deliver the same message-body to several addresses. Mailing-lists and conferences are just two examples.

The multi-address issue is the dual of the multiplexing which was discussed earlier. Multi-address is one message-body with several message-headers, whereas the previous muliplexing is one message-header with several message-bodies.

Multi-address is:

```
<Adr1,Adr2 ; Data1>  = <Adr1 ; Data1> + <Adr2 ; Data1>
```

Multiplexing is:

```
<Adr1 ; Data1,Data2> = <Adr1 ; Data1> + <Adr1 ; Data2>
```

Let's be (sort of) formal

(1) Protocol nesting is:

```
<msg>  ::= <hdr> <body>
<hdr>  ::= "type" "addr"
<body> ::= "data" ! <msg>
```

(2) Multiplexing is:

```
<msg>  ::= <hdr> <body>
<hdr>  ::= "type" "addr"
<body> ::= "data" ! <msgs>
<msgs> ::= <msg> ! <msgs> <msg>
```

(3) Multi-addressing is:

```
<msg>  ::= <hdr> <body>
<hdr>  ::= "type" "addr" ! <hdr> "addr" <body> ::= "data"
```

(4) Obviously the most general system is:

```
<msg>  ::= <hdr> <body>
<hdr>  ::= "type" "addr" ! <hdr> "addr" <body> ::= "data" ! <msgs>
<msgs> ::= <msg> ! <msgs> <msg>
```

Please consider (4) as a proposal.

The point is twofold:

(1) Save overhead to reduce the number of BITs.

(2) Save overhead to reduce the number of PACKETs.

3Mbit/sec at 500 bits/packet is 6,000 packets/sec.

Today's gateways can handle 6 packets/sec, with the hope to double it by next year..... The factor of 1,000 between these two numbers should serve as a warning light.