RFC 872 September 1982
M82-48

TCP-ON-A-LAN

M.A. PADLIPSKY
THE MITRE CORPORATION
Bedford, Massachusetts

Abstract

The sometimes-held position that the DoD Standard Transmission Control Protocol (TCP) and Internet Protocol (IP) are inappropriate for use "on" a Local Area Network (LAN) is shown to be fallacious. The paper is a companion piece to M82-47, M82-49, M82-50, and M82-51.

"TCP-ON-A-LAN"

M. A. Padlipsky

Thesis

It is the thesis of this paper that fearing "TCP-on-a-LAN" is a Woozle which needs slaying. To slay the "TCP-on-a-LAN" Woozle, we need to know three things: What's a Woozle? What's a LAN? What's a TCP?

Woozles

The first is rather straightforward [1]:

> One fine winter's day when Piglet was brushing away the snow in front of his house, he happened to look up, and there was Winnie-the-Pooh. Pooh was walking round and round in a circle, thinking of something else, and when Piglet called to him, he just went on walking.
> "Hallo!" said Piglet, "what are you doing?"
> "Hunting," said Pooh.
> "Hunting what?"
> "Tracking something," said Winnie-the-Pooh very mysteriously.
> "Tracking what?" said Piglet, coming closer.
> "That's just what I ask myself. I ask myself, What?"
> "What do you think you'll answer?"
> "I shall have to wait until I catch up with it," said Winnie-the-Pooh. "Now look there." He pointed to the ground in front of him. "What do you see there?
> "Tracks," said Piglet, "Paw-marks." he gave a little squeak of excitement. "Oh, Pooh! Do you think it's a--a--a Woozle?"

Well, they convince each other that it is a Woozle, keep "tracking," convince each other that it's a herd of Hostile Animals, and get duly terrified before Christopher Robin comes along and points out that they were following their own tracks all the long.

In other words, it is our contention that expressed fears about the consequences of using a particular protocol named "TCP" in a particular environment called a Local Area Net stem from misunderstandings of the protocol and the environment, not from the technical facts of the situation.

LAN's

The second thing we need to know is somewhat less straightforward: A LAN is, properly speaking [2], a communications mechanism (or subnetwork) employing a transmission technology suitable for relatively short distances (typically a few kilometers) at relatively high bit-per-second rates (typically greater than a few hundred kilobits per second) with relatively low error rates, which exists primarily to enable suitably attached computer systems (or "Hosts") to exchange bits, and secondarily, though not necessarily, to allow terminals of the teletypewriter and CRT classes to exchange bits with Hosts. The Hosts are, at least in principle, heterogeneous; that is, they are not merely multiple instances of the same operating system. The Hosts are assumed to communicate by means of layered protocols in order to achieve what the ARPANET tradition calls "resource sharing" and what the newer ISO tradition calls "Open System Interconnection." Addressing typically can be either Host-Host (point-to-point) or "broadcast." (In some environments, e.g., Ethernet, interesting advantage can be taken of broadcast addressing; in other environments, e.g., LAN's which are constituents of ARPA- or ISO-style "internets", broadcast addressing is deemed too expensive to implement throughout the internet as a whole and so may be ignored in the constituent LAN even if available as part of the Host-LAN interface.)

Note that no assumptions are made about the particular transmission medium or the particular topology in play. LAN media can be twisted-pair wires, CATV or other coaxial-type cables, optical fibers, or whatever. However, if the medium is a processor-to-processor bus it is likely that the system in question is going to turn out to "be" a moderately closely coupled distributed processor or a somewhat loosely coupled multiprocessor rather than a LAN, because the processors are unlikely to be using either ARPANET or ISO-style layered protocols. (They'll usually -- either be homogeneous processors interpreting only the protocol necessary to use the transmission medium, or heterogeneous with one emulating the expectations of the other.) Systems like "PDSC" or "NMIC" (the evolutionarily related, bus-oriented, multiple PDP-11 systems in use at the Pacific Data Services Center and the National Military Intelligence Center, respectively), then, aren't LANs.

LAN topologies can be either "bus," "ring," or "star". That is, a digital PBX can be a LAN, in the sense of furnishing a transmission medium/communications subnetwork for Hosts to do resource sharing/Open System Interconnection over, though it might not present attractive speed or failure mode properties. (It might, though.) Topologically, it would probably be a neutron star.

For our purposes, the significant properties of a LAN are the high bit transmission capacity and the good error properties. Intuitively, a medium with these properties in some sense "shouldn’t require a heavy-duty protocol designed for long-haul nets," according to some. (We will not address the issue of "wasted bandwidth" due to header sizes. [2], pp. 1509f, provides ample refutation of that traditional communications notion.) However, it must be borne in mind that for our purposes the assumption of resource-sharing/OSI type protocols between/among the attached Hosts is also extremely significant. That is, if all you’re doing is letting some terminals access some different Hosts, but the Hosts don’t really have any intercomputer networking protocols between them, what you have should be viewed as a Localized Communications Network (LCN), not a LAN in the sense we’re talking about here.

TCP

The third thing we have to know can be either straightforward or subtle, depending largely on how aware we are of the context estabished by ARPANET-style prococols: For the visual-minded, Figure 1 and Figure 2 might be all that need be "said." Their moral is meant to be that in ARPANET-style layering, layers aren’t monoliths. For those who need more explanation, here goes: TCP [3] (we’ll take IP later) is a Host-Host protocol (roughly equivalent to the functionality implied by some of ISO Level 5 and all of ISO Level 4). Its most significant property is that it presents reliable logical connections to protocols above itself. (This point will be returned to subsequently.) Its next most significant property is that it is designed to operate in a "catenet" (also known as the, or an, "internet"); that is, its addressing discipline is such that Hosts attached to communications subnets other than the one a given Host is attached to (the "proximate net") can be communicated with as well as Hosts on the proximate net. Other significant properties are those common to the breed: Host-Host protocols (and Transport protocols) "all" offer mechanisms for flow Control, Out-of-Band Signals, Logical Connection management, and the like.

Because TCP has a catenet-oriented addressing mechanism (that is, it expresses foreign Host addresses as the "two-dimensional" entity Foreign Net/Foreign Host because it cannot assume that the Foreign Host is attached to the proximate net), to be a full Host-Host protocol it needs an adjunct to deal with the proximate net. This adjunct, the Internet Protocol (IP) was designed as a separate protocol from TCP, however, in order to allow it to play the same role it plays for TCP for other Host-Host protocols too.

In order to "deal with the proximate net", IP possess the following significant properties: An IP implementation maps from a virtualization (or common intermediate representation) of generic proximate net qualities (such as precedence, grade of service, security labeling) to the closest equivalent on the proximate net. It determines whether the "Internet Address" of a given transmission is on the proximate net or not; if so, it sends it; if not, it sends it to a "Gateway" (where another IP module resides). That is, IP handles internet routing, whereas TCP (or some other Host-Host protocol) handles only internet addressing. Because some proximate nets will accept smaller transmissions ("packets") than others, IP, qua protocol, also has a discipline for allowing packets to be fragmented while in the catenet and reassembled at their destination. Finally (for our purposes), IP offers a mechanism to allow the particular protocol it was called by (for a given packet) to be identified so that the receiver can demultiplex transmissions based on IP-level information only. (This is in accordance with the Principle of Layering: you don't want to have to look at the data IP is conveying to find out what to do with it.)

Now that all seems rather complex, even though it omits a number of mechanisms. (For a more complete discussion, see Reference [4].) But it should be just about enough to slay the Woozle, especially if just one more protocol's most significant property can be snuck in. An underpublicized member of the ARPANET suite of protocols is called UDP--the "User Datagram Protocol." UDP is designed for speed rather than accuracy. That is, it's not "reliable." All there is to UDP, basically, is a mechanism to allow a given packet to be associated with a given logical connection. Not a TCP logical connection, mind you, but a UDP logical connection. So if all you want is the ability to demultiplex data streams from your Host-Host protocol, you use UDP, not TCP. ("You" is usually supposed to be a Packetized Speech protocol, but doesn't have to be.) (And we'll worry about Flow Control some other time.)

TCP-on-a-LAN

So whether you're a Host proximate to a LAN or not, and even whether your TCP/IP is "inboard" or "outboard" of you, if you're talking to a Host somewhere out there on the catenet, you use IP; and if you're exercising some process-level/applications protocol (roughly equivalent to some of some versions of ISO L5 and all of L6 and L7) that expects TCP/IP as its Host-Host protocol (because it "wants" reliable, flow controlled, ordered delivery [whoops, forgot that "ordered" property earlier--but it doesn't matter all that much for present purposes] over logical connections which allow it to be addressed via a Well-Known Socket), you use TCP "above" IP regardless of whether the other Host is on your proximate net or not. But if your application doesn't require the properties of TCP (say for Packetized Speech), don't use it--regardless of where or what you are. And if you want to make the decision about whether you're talking to a proximate Host explicitly and not even go through IP, you can even arrange to do that (though it might make for messy implementation under some circumstances). That is, if you want to take advantage of the properties of your LAN "in the raw" and have or don't need appropriate applications protocols, the Reference Model to which TCP/IP were designed won't stop you. See Figure 2 if you're visual. A word of caution, though: those applications probably will need protocols of some sort--and they'll probably need some sort of Host-Host protocol under them, so unless you relish maintaining "parallel" suites of protocols.... that is, you really would be better off with TCP most of the time locally anyway, because you've got to have it to talk to the catenet and it's a nuisance to have "something else" to talk over the LAN--when, of course, what you're talking requires a Host-Host protocol.

We'll touch on "performance" issues in a bit more detail later. At this level, though, one point really does need to be made: On the "reliability" front, many (including the author) at first blush take the TCP checksum to be "overkill" for use on a LAN, which does, after all, typically present extremely good error properties. Interestingly enough, however, metering of TCP implementations on several Host types in the research community shows that the processing time expended on the TCP checksum is only around 12% of the per-transmission processing time anyway. So, again, it's not clear that it's worthwhile to bother with an alternate Host-Host protocol for local use (if, that is, you need the rest of the properties of TCP other than "reliability"--and, of course, always assuming you've got a LAN, not an LCN, as distinguished earlier.)

Take that, Woozle!

Other Significant Properties

Oh, by the way, one or two other properties of TCP/IP really do bear mention:

1. Protocol interpreters for TCP/IP exist for a dozen or two different operating systems.

2. TCP/IP work, and have been working (though in less refined versions) for several years.

3. IP levies no constraints on the interface protocol presented by the proximate net (though some protocols at that level are more wasteful than others).

4. IP levies no constraints on its users; in particular, any proximate net that offers alternate routing can be taken advantage of (unlike X.25, which appears to preclude alternate routing).

5. IP-bearing Gateways both exist and present and exploit properties 3 and 4.

6. TCP/IP are Department of Defense Standards.

7. Process (or application) protocols compatible with TCP/IP for Virtual Terminal and File Transfer (including "electronic mail") exist and have been implemented on numerous operating systems.

8. "Vendor-style" specifications of TCP/IP are being prepared under the aegis of the DoD Protocol Standards Technical Panel, for those who find the research-community-provided specs not to their liking.

9. The research community has recently reported speeds in excess of 300 kb/s on an 800 kb/s subnet, 1.2 Mb/s on a 3 Mb/s subnet, and 9.2 kbs on a 9.6 kb/s phone line--all using TCP. (We don't know of any numbers for alternative protocol suites, but it's unlikely they'd be appreciably better if they confer like functionality--and they may well be worse if they represent implementations which haven't been around enough to have been iterated a time or three.)

With the partial exception of property 8, no other resource-sharing protocol suite can make those claims.

Note particularly well that none of the above should be construed as eliminating the need for extremely careful measurement of TCP/IP performance in/on a LAN. (You do, after all, want to know their limitations, to guide you in when to bother ringing in "local" alternatives--but be very careful: 1. they're hard to measure commensurately with alternative protocols; and 2. most conventional Hosts can't take [or give] as many bits per second as you might imagine.) It merely dramatically refocuses the motivation for doing such measurement. (And levies a constraint or two on how you outboard, if you're outboarding.)

Other Contextual Data

Our case could really rest here, but some amplification of the aside above about Host capacities is warranted, if only to suggest that some quantification is available to supplement the a priori argument: Consider the previously mentioned PDSC. Its local terminals operate in a screen-at-a-time mode, each screen-load comprising some 16 kb. How many screens can one of its Hosts handle in a given second? Well, we’re told that each disk fetch requires 17 ms average latency, and each context switch costs around 2 ms, so allowing 1 ms for transmission of the data from the disk and to the "net" (it makes the arithmetic easy), that would add up to 20 ms "processing" time per screen, even if no processing were done to the disk image. Thus, even if the Host were doing nothing else, and even if the native disk I/O software were optimized to do 16 kb reads, it could only present 50 screens to its communications mechanism (processor-processor bus) per second. That’s 800 kb/s. And that’s well within the range of TCP-achievable rates (cf. Other Significant Property 9). So in a realistic sample environment, it would certainly seem that typical Hosts can’t necessarily present so many bits as to overtax the protocols anyway. (The analysis of how many bits typical Hosts can accept is more difficult because it depends more heavily on system internals. However, the point is nearly moot in that even in the intuitively unlikely event that receiving were appreciably faster in principle [unlikely because of typical operating system constraints on address space sizes, the need to do input to a single address space, and the need to share buffers in the address space among several processes], you can’t accept more than you can be given.)

Conclusion

The sometimes-expressed fear that using TCP on a local net is a bad idea is unfounded.

References

[1] Milne, A. A., "Winnie-the-Pooh", various publishers.

[2] The LAN description is based on Clark, D. D. et al., "An Introduction to Local Area Networks," IEEE Proc., V. 66, N. 11, November 1978, pp. 1497-1517, several year’s worth of conversations with Dr. Clark, and the author’s observations of both the open literature and the Oral Tradition (which were sufficiently well-thought of to have prompted The MITRE Corporation/NBS/NSA Local Nets "Brain Picking Panel" to have

solicited his testimony during the year he was in FACC’s employ.*)

[3] The TCP/IP descriptions are based on Postel, J. B., "Internet Protocol Specification," and "Transmission Control Specification" in DARPA Internet Program Protocol Specifications, USC Information Sciences Institute, September, 1981, and on more than 10 years’ worth of conversations with Dr. Postel, Dr. Clark (now the DARPA "Internet Architect") and Dr. Vinton G. Cerf (co-originator of TCP), and on numerous discussions with several other members of the TCP/IP design team, on having edited the referenced documents for the PSTP, and, for that matter, on having been one of the developers of the ARPANET "Reference Model."

[4] Padlipsky, M. A., "A Perspective on the ARPANET Reference Model", M82-47, The MITRE Corporation, September 1982; also available in Proc. INFOCOM ’83.

* In all honesty, as far as I know I started the rumor that TCP might be overkill for a LAN at that meeting. At the next TCP design meeting, however, they separated IP out from TCP, and everything’s been alright for about three years now--except for getting the rumor killed. (I’d worry about Woozles turning into roosting chickens if it weren’t for the facts that: 1. People tend to ignore their local guru; 2. I was trying to encourage the IP separation; and 3. All I ever wanted was some empirical data.)

NOTE: FIGURE 1. ARM in the Abstract, and FIGURE 2. ARMS, Somewhat Particularized, may be obtained by writing to: Mike Padlipsky, MITRE Corporation, P.O. Box 208, Bedford, Massachusetts, 01730, or sending computer mail to Padlipsky@USC-ISIA.