# COMP 535

End-to-End Communication System Design

Simon Foucher (260 223 197)

**McGill University
Monday, Sept 21st 2009**

END TO END

* the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level.
* The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system.
* accomplished--brute force countermeasures, such as doing everything three times, appear uneconomical. The alternate approach might be called end-to-end check and retry.
* that threat may have been eliminated, but the careful file transfer application must still counter the remaining threats; so it should still provide its own retries based on an end-to-end checksum of the file. If it does, the extra effort expended in the communication system to provide a guarantee of reliable data transmission is only reducing the frequency of retries by the file transfer application; it has no effect on inevitability or correctness of the outcome,
* In order to achieve careful file transfer, the application program that performs the transfer must supply a file-transfer-specific, end-to-end reliability guarantee--in this case, a checksum to detect failures and a retry commit plan. For the data communication system to go out of its way to be extraordinarily reliable does not reduce the burden on the application program to ensure reliability.
* Clearly, some effort at the lower levels to improve network reliability can have a significant effect on application performance. But the key idea here is that the lower levels need not provide "perfect" reliability.
* the communication system is too unreliable, the file transfer application performance will suffer because of frequent retries following failures of its end-to-end checksum. If the communcation system is beefed up with internal reliability measures, those measures also have a performance cost, in the form of bandwidth lost to redundant data and added delay from waiting for internal consistency checks to complete before delivering the data.
* **that the end-to-end check of the file transfer application must still be implemented no matter how reliable the communication system becomes**
* is, performing the function at the lower level may cost more--for two reasons. First, since the lower level subsystem is common to many applications, those applications that do not need the function will pay for it anyway. Second, the low-level subsystem may not have as much information as the higher levels, so it cannot do the job as efficiently.
* that knowing for sure that the message was delivered to the target host is not very important. What the application wants to know is whether or not the target host acted on the message;
* First, if the data transmission system perfoms encryption and decryption, it must be trusted to securely manage the required encryption keys. Second, the data will be in the clear and thus vulnerable as they pass into the target node and are fanned out to the target application. Third, the authenticity of the message must still be checked by the application. If the application performs end-to-end encryption, it obtains its required authentication check and can handle key management to its satisfaction, and the data are never exposed outside the application.
* again: If the application level has to have a duplicate suppressing mechanism anyway, that mechanism can also suppress any duplicates generated inside the communication network; therefore, the function can be omitted from that lower level

VOIP

* system try to accomplish bit-perfect communication, they will probably introduce uncontrolled delays in packet delivery is better to accept slightly damaged packets as they are, or even to replace them with silence,
* **will get better performance by implementing exactly the instructions needed from primitive tools; any attempt by the computer designer to anticipate the client's requirements for an esoteric feature will probably miss the target slightly and the client will end up reimplementing that feature anyway**.
* that for any function that can be thought of, at least some applications will find that, of necessity, they must implement the function themselves in order to meet correctly their own requirements.
* the idea of the datagram, or connectionless service, does not receive particular emphasis in the first paper, but has come to be the defining characteristic of the protocol.
* The top level goal for the DARPA Internet Architecture was to develop an ffective technique for multiplexed utilization of existing interconnected networks
* the fundamental structure of the Internet: a packet switched communications facility in which a number of distinguishable networks am connected together using packet communications processors called gateways which implement a store arid forward packet forwarding algorithm.

DARPA GOALS order of importance:

Survivability was put as a first goal, and accountability as a last goal.

An architecture primarily for commercial deployment would clearly place these goals at the opposite end of the list.

1. Internet commuuication must continue despite loss of networks or gateways.

and some failure causes the Internet to be temporarily disrupted and reconfigured

to reconstitute the service, then the entities communicating should be able to continue without

having to reestablish or reset the high level state of their conversation.

It was an assumption in this architecture that synchronization would never be lost unless there was no physical path over which any sort of communication could be achieved. In other words, at the top of transport, there is only one failure, and it is total partition. The architecture was to mask completely any transient failure. To achieve this goal, the state information which describes the on-going conversation must be protected.

case, to protect the information from loss, it must replicated. The alternative, which this architecture chose, is to take this information and gather it at the endpoint of the net, at the entity which is utilizing the service of the network.

important advantages to fate-sharing over replication. First, fate-sharing protects against any

number of intermediate failures, whereas replication can only protect against a certain number (less than the number of replicated copies). Second, fate-sharing is much easier to engineer than replication.

two consequences to the fate-sharing approach to survivability. First. the intermediate packet switching nodes, or gateways, must not have any essential state information about on-going connections. Instead, they are stateless packet switches, a class of network design sometimes called a “datagram” network

2. The Internet must support multiple types of communications service.

differing requirements for such things as speed, latency and reliability.

The initial concept of TCP was that it could be general enough to support any needed type of service. seemed too difficult to build support for all of them into one protocol.

Debigger should not be reliable: asking for reliable communications may prevent any communications at all.

 real time digital speech, the primary requirement is not a reliable service, but a service which

minimizes and smooths the delay in the delivery of packets.

The architecture did not wish to assume that the underlying networks themselves support multiple types of services, because this would violate the goal of using existing networks. Instead, the hope was that multiple types of service could be constructed out of the basic datagram building block using algorithms within the host and the gateway.

3. The Internet architecture must accommodate a variety of networks.

The Internet architecture achieves this flexibility by making a minimum set of assumptions about the function which the net will provide. The basic assumption is that network can transport a packet or datagram. The packet must be of reasonable size, perhaps 100 bytes minimum, and should be delivered with reasonable but not perfect reliability. The network must have some suitable form of addressing if it is more than a point to point link.

4. The Internet architecture must permit distributed management of its resources.

5. The Internet architecture must be cost effective.

6. The Internet architecture must permit host attachment with a low level of effort.

7. The resources used in the iutemet architecture must be accountable.

some of the most significant problems with the Internet today relate to lack of sufficient tools for

distributed management

powerful. The most important change in the Internet architecture over the next few years will probably be the development of a new generation of tools for management of resources in the context of multiple administrations.

Since Internet does not insist that lost packets be recovered at the network level, it may be

necessary to retransmit a lost packet from one end of the Internet to the other.

example of the tradeoff resulting from the decision, discussed above, of providing services from the end-points. The network interface code is much simpler, but the overall efficiency is potentially less.

the Internet architecture contains few tools for accounting for packet flows.

The originaI ARPANET host-to host protocol provided flow control based on both bytes and packets. This seemed overly complex, and the designers of TCP felt that only one form of regulation would he sufficient. The choice was to regulate the delivery of bytes, rather than packets. Flow control and acknowledgment in TCP is thus based on byte number rather than packet number.

was to permit the insertion of control information into the sequence space of the bytes, so that

control as well as data could be acknowledged

byte stream was to permit the TCP packet to be broken up into smaller packets if necessary in order to fit through a net with a small packet size. But this function was moved to the IP layer when IP was split from TCP,

acknowledging bytes rather than packets was to permit a number of small packets to be gathered together into one larger packet in the sending host if retransmission of the data was necessary

Control at the packet level has the effect, however, of providing a severe limit on the throughput if small packets are sent

The architecture tried very hard not to constrain the range of service which the Internet could be engineered to provide.

The designers of the Internet architecture felt very strongly that it was a serious mistake to attend only to logical correctness and ignore the issue of performance. However, they experienced great difficulty in formalizing any aspect of performance constraint within the architecture. These difficulties arose both because the goal of the architecture was not to constrain performance, but to permit variability, and secondly (and perhaps more fundamentally), because there seemed to be no useful formal tools for describing performance.

The fundamental architectural feature of the Internet is the use of datagrams as the entity which is transported across the underlying networks. First, they eliminate the need for connection state within the intermediate switching nodes, which means that the Internet can be reconstituted after a failure without concern about state.

Secondly, the datagram provides a basic building block out of which a variety of types of service can be implemented.

that the role of the datagram in this respect is as a building block, and not as a service in itself.

While the datagram has served veIy well in solving the most important goals of the Internet, it has not served so well when we attempt to address some of the goals which were further down the priority list. For example, the goals of resource management and accountability have proved difficult to achieve in the context of datagrams. As the previous section discussed, most datagrams are a part of some sequence of packets from source to destination, rather than isolated units at the application level. However, the gateway cannot directly see the existence of this sequence, because it is forced to deal with each packet in isolation.