

Best Efforts Networking

Geoff Huston
July 2001

IP networks are often described as "best effort" networks. This refers to the approach to service quality where the network itself does not actively differentiate in its treatment of services that transit the network. In a best effort IP network all IP packets are treated in the same fashion. The network undertakes its "best effort" to deliver every packet as quickly as it can, but makes no undertaking to treat any class of packets preferentially to any other. While this sounds like a perfectly impartial and fair approach, it is often said of Internet networks that, for certain applications, "best effort" simply isn't good enough. It is claimed that what IP networks need is some way to provide a superior response to support certain classes of applications in some special manner. "Better than best effort" is one way of describing this form of quality of service.

Its not clear that this fits comfortably within the overall architecture of IP, so lets look at this claim in a bit more detail, to see whether best effort is good enough or not.

The conventional approach to communications, as developed with the public switched telephone network, is that of a functionally rich network supporting very simple peripheral devices. After all, a telephone handset is just a speaker and a microphone, so all the functionality required to form a real time connection between one handset and another is found within the network, not within the handset. There's no doubt that the global telephone network is an amazing piece of engineering. At the edge of the network are analogue to digital converters, translating the analogue signal to an equivalent 64Kbps bit stream. These individual bit streams are fed into digital circuit switches that support on-demand connections between one circuit and another. The multiplexing technique used to support the backbone trunk network is that of time division multiplexing, where a number of low rate bit streams are precisely interleaved into higher rate bit streams. The outcome is indeed quite remarkable. This network is capable of supporting precisely clocked digital bit streams between two locations, with negligible jitter and error levels. Remarkable from an engineering perspective, but, from the perspective of the Internet architecture, somewhat irrelevant.

The revolutionary approach adopted by the Internet is to take a position which is the exact opposite of this traditional approach. The Internet makes almost no assumptions about the functionality of the underlying network. The revolutionary approach adopted by the Internet is that all of the functionality used to create services is not found within the network, but instead is found in the software of the equivalent of the handset, the host computer. These edge systems make very few assumptions about the capabilities of the Internet network. They assume that the network is capable of delivering packets to an address destination. But they do not assume that all packets will be successfully delivered, and they don't even assume that they will be informed if a packet is not delivered. If a sequence of packets is sent to the same destination, the host computer must not assume that packet order will be maintained by the network, and also cannot assume that the relative timing of the packets will be maintained by the network. The host computer cannot assume that the network has any particular throughput rate, or bandwidth, nor any fixed end-to-end delay, or indeed any particular level of quality. The Internet is constructed to work across any form of underlying network. Even more challenging is the requirement that the Internet has to work across a sequence of such networks, where each network may have different characteristics. The approach adopted by the Internet is that of assuming as little as possible about the characteristics of the underlying transmission system, and instead placing all the functionality into the host computers at the edge of the network. An approach that is often characterized as a that of "dumb network, smart devices".

Most of this rich edge functionality can be found within the TCP protocol, the protocol used to support reliable end-to-end data transfer. This protocol is the workhorse of most Internet networks, supporting the operation of the World Wide Web, electronic mail, filter transfer. The basic approach used by TCP is one of positive acknowledgement, where a sender can only consider a packet to have been successfully delivered when it receives an acknowledgement from the remote end that signals reception of the previously sent packet. Also, TCP is a rate adaptive protocol, rather than a constant rate protocol. This implies that a TCP transfer will attempt to increase its data transfer rate while the network is able to successfully deliver all packets, assuming that both the sender and receiver can support the increased rate. TCP interprets packet loss as a sign of network congestion, and reacts to network congestion by reducing its sending rate. Rate adaptive protocols can make very efficient use of the network, allowing end systems to increase their data rates while there is available network capacity, and sharply reducing data rates in the face of observed network congestion. This works whether the protocol is working across a high bandwidth low latency LAN, or whether the protocol is working across a low bandwidth high latency satellite circuit. The TCP protocol stack will attempt to optimize its use of the underlying communications system irrespective of the particular characteristics of that system.

From the network's perspective, rate adaptive protocols are the ultimate scavenger of network capacity, and this makes for very efficient low cost networks. If available network capacity is considered as a business cost, and Internet traffic as revenue, rate adaptive protocols such as TCP can maximize the amount of data, or revenue, for a given level of network capacity, or cost. Its little wonder that Internet-based services are often the cheapest form of network service . Not only is the network built from simple packet switches and basic communications services, the network also maximizes the effective data throughput of the network

But, while Internet networks are highly efficient, what forms of service outcomes can you expect? The answer is like a trip in your car. When there is no traffic on the road, your trip is quick. When the road is congested with traffic the trip will take longer. Like your car, the service outcomes of an Internet network are variable. When there is no other network traffic, you can expect high throughput with low levels of jitter. When the levels of traffic increase, you can expect lower throughput, and higher levels of jitter and packet loss. The network does not police sharing, but instead the interaction between multiple TCP streams produces an outcome where the network is shared fairly across all concurrent requests.

Here's where the "better than best effort" arguments enter the scene. Is it possible to alter the Internet such that certain packets are treated preferentially? Can you create an Internet where some packets receive a constant service response, while other see a 'normal' variable best effort network? In short, can you add functionality back into the network itself that is visible to the protocols that operate across the network?

The answer is unclear. In the general Internet model, where the end systems make an absolutely minimal set of assumptions about the capabilities of the intervening packet switching and communications systems of the network, then the answer is a clear 'no'. However, if you are prepared to make more restrictive assumptions about the network, then the outlook is more positive. It is possible to introduce some form of differentiated network response to certain classes of traffic, and it is possible to allow the network to enforce some form of resource control over the users of the network. But the larger question is whether this effort is useful.

A large part of the problem with application service quality on the Internet is not the behavior of the network, but the behavior of the application. Within the Internet application architecture there is a strong component of feedback adaptation, where the sender modifies its behavior based on the stream of acknowledgement signals coming back from the receiver. Applications that ignore such feedback, and blindly push data into the network without regard to the current state of the network will invariably perform poorly. They will perform poorly simply because they are incapable of modifying their behavior to match the prevailing network conditions. To make a

best effort network perform well its necessary that applications understand the dynamic variability of the network and understand that they should adapt to the network, not that the network should adapt to them.

In an Internet architecture, its not the network that needs to get smarter, it's the applications that need the injection of clue.
