Two Ways to Trade Bandwidth for Latency

Joe Touch
USC/Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292, U.S.A.
June 21, 2013

Abstract

This summary reviews the issues in reducing Internet latency. It suggests key ways forward to reduce latency by using smaller packets or anticipation, each of which benefit at the expense of increased bandwidth.

I. Introduction

Latency is *the* fundamental metric of all computing and communication. Many brag about CPU clock rates, gigabits-per-second, and the number of cores, but all are merely means to an end - reducing latency.

The following is a discussion of latency reduction techniques, focusing on communication issues. A framework is presented to organize concepts and issues, as well as ways to mitigate latency effects. Variations of these approaches, as deployed in the Internet, are discussed. Some remaining opportunities for latency reduction are presented.

II. Concepts

Latency issues can be organized as an accounting analysis. A given problem has a latency budget - a deadline for useful response - costs that consume the budget. The goal is to find an approach whose cost is below the specified budget.

Latency budget: A latency budget is based on deadlines, either hard or soft. Hard deadlines describe tasks with no value after the deadline expires; soft deadlines decreases in value gradually (ASAP is zero budget/soft). Budgets are derived from computational or biological requirements. Automated stock trading and real-time signal analysis are computational, *i.e.*, between two computers (broker/exchange) or between a computer and a physical entity (e.g., gravity, light, *etc*.). Biological budgets are driven by eye-hand response (~100ms) or more direct neural stimulus-response (ear, eye, muscle implants).

Latency costs: Costs consume budgets. Except for line-of-sight speed-of-light propagation, all delays are the result of design decisions and can be reduced. A small budget limits potential solutions: stock brokers want to be next to their exchange, real-time games prefer to be intra-continent, and conferencing is always difficult off-planet.

III. Cost Components

Latency is incurred at all layers and hops:

- **Generation**: physical (audio frequency), source format (video frame), storage (RAM, disk)
- **Transmission**: signal propagation, signal encoding (parallel/serial, striping, bit/symbol)
- **Processing**: forward, encap/decap, NAT, encrypt, authenticate, compress, error coding
- **Multiplexing**: shared channel acquisition, output queuing, connection establishment
- **Grouping**: packetization, message aggregation

Generation delays occur between a physical event and data availability. Human audio is limited to 20KHz, so samples are delayed at least 50μs from an event and need another 50μs to affect a listener. Video is delayed 16.7ms at the camera and monitor.

Transmission delays involve the propagation of a signal. An optical fiber signal is 35% slower than free-space RF, even discounting the circuitous route of fiber compared to radio. Signal encoding delays happen when changing width (serial-parallel, striping) or density (binary to multibit symbols).

Processing introduces delays because most algorithms require either translating sets of bits or entire packets, which delays the start of processing even before considering computation time.

Multiplexing introduces delays to share resources, whether to wait to acquire a shared medium (Ethernet, RF) or to emulate that process inside a switch to resolve output port contention. Connection establishment is considered here because it is needed only to establish state for demultiplexing; direct channels don't need it.

Perhaps the least appreciated aspect is grouping. Grouping reduces the frequency of control information and processing. Inside the network. all the bits of a packet are treated together, and at the endpoints one message expresses multiple signals.

Reducing Internet latency requires that each of these areas be addressed. Generation and transmission can be minimized by selecting low-latency components (avoidance) or simply relocating them. Processing can be disabled (avoided) or sped-up (using faster components). Multiplexing costs can be reduced by reusing or pre-placing connection state (anticipation). Grouping costs can be reduced by decreasing the chunk size, reducing the lost wait time (avoidance). These can be summarized as *relocation*, *speedup*, "*wait loss*" (omission), and *anticipation*.

Of the above areas, many are not easily addressed solely in the Internet. Only the last three can be easily addressed within Internet layers.

IV. Cost Containment

Approaches to latency cost containment fall into the following broad categories.

- **Relocation**: moving the endpoints closer
- **Speedup**: increasing operations per unit time
- "**Wait loss**": avoid by omission or substitution
- **Anticipation**: proactive communication

Relocation reduces transmission costs by minimizing signal propagation delays. This is popular for web transactions (distributed processing centers) and stock transactions (automated trading centers). It also includes shifting protocol functions closer to the

network interface within a system, such as with protocol offloading.

Speedup reduces delays resulting from capacity limitations, e.g., bits per second, lookups per second, etc. It can be accomplished by increasing the processing speed (increased CPU clock rate), increasing the symbol rate (transmission BW), or parallelizing to reduce service processing delays [9]. Note that parallelization increases speedup only to the individual transaction limit, *i.e.*, each event incurs the same processing delay, but transactions are not waiting for processing to start.

"Wait loss" reduces the per-event processing cost, either by avoiding processing altogether or by reducing the unit over which processing occurs. For example, the latency of an encrypted link can be reduced by turning off encryption, or by encrypting over smaller units.

Anticipation is the only method that can reduce latency below the speed-of-light cost, approaching - and sometimes achieving - zero delay [10]. It involves predicting either the reuse of previous data (caching) or new data (prefetching, push).

V. Impact on Internet Protocols

Of these approaches, most have been explored in Internet protocols and are under active use:

Relocation:
- Processing offloading (TCP, checksum, ACK)
- 'Zero-copy' network stacks [8]
- Local data centers (stocks trading, clouds)

Speedup:
- Increased link BW (10GE, 100GE, 802.11ac)
- Increased processing capacity

Wait loss:
- MPLS (IP processing)
- TCP Nagle (coalescing delay) [5]
- AQM, RED and other queue "bufferbloat" [3]

Anticipation:
- Content caching (web cache, DNS cache)
- T/TCP, TCP-CT, persist-HTTP [1][7][6]
- 'Prefetching the means' (TCP connection) [2]
- TCP control block sharing (TCP state) [15]

Anticipation that keeps connections open (p-HTTP) or enables faster new connections (TCP-CT, 'prefetching the means') is already under active development and may be more widely used.

Most other approaches are very widely deployed already. Some are well understood, but have been implemented incorrectly (*e.g.*, bufferbloat issues).

VI. Trading BW for Latency

Two opportunities remain for Internet latency reduction - small packets and push-anticipation. Both can decrease latency, but both increase BW.

"Small packets" refers to using packet sizes smaller than the network maximum, *e.g.*, sending 10 150-byte messages rather than one 1,500-byte message. [14]. This reduces store-and-forward delays at every network processing step, but increases the potential for loss and reordering. It reduces latency linearly as the message size decreases, and increases BW linearly as the message number increases. The specific trade-off depends on the link BW, per-packet processing costs, and per-byte processing costs (*e.g.*, encryption, error checking) at each hop.

Push-anticipation reduces cost by predicting future needs. Caching predicts future use based on past use. Prefetching web clients retrieve pages using links on the current page. Push anticipation partially decouples the server and client [10]. This enables server push that can be more effective at reducing latency, useful for local network files (NFS) [11], FTP [12], as well as the web [13]. Its latency reduction is roughly logarithmic compared to the increase in BW; analysis of FTP logs demonstrated that an 8x BW increase could reduce latency by a factor of 3.

Each of these approaches could be implemented in the existing Internet. Small packets has the advantage of being decoupled from application semantics, and so could more easily be implemented in IP and various transport protocols. Anticipation necessarily requires application semantics to drive the push function, either using internal links (web) or cross-application context (email/DNS) [4].

References

[1] Braden, R., "T/TCP -- TCP Extensions for Transactions", RFC 1644, July 1994.

[2] Cohen, E. Kaplan, H., "Prefetching the means for document transfer: A new approach for reducing Web latency", *Computer Networks* 39.4 (2002): 437-455.

[3] Gettys, J., Nichols, K., "Bufferbloat: dark buffers in the internet", Commun. ACM Jan. 2012, pp.57-65.

[4] Hughes, A., Touch, J., "Cross-Domain Cache Cooperation for Small Clients," NetStore 1999.

[5] Nagle, J., "Congestion Control in IP/TCP Internetworks", RFC 896, Jan. 1984.

[6] Padmanabhan, V., Mogul, J., "Improving HTTP latency," 2nd Int. World Wide Web Conf., Oct. 1994.

[7] Simpson, W., "TCP Cookie Transactions", RFC 6013, Jan. 2011.

[8] Sterbenz, J., Parulkar, G., "Axon: A High-Speed Communication Architecture for Distributed Applications", Proc. IEEE Infocom 1990, pp. 415-425.

[9] Sterbenz, J., Touch, J., *High Speed Networking: A Systematic Approach to High-Bandwidth Low-Latency Communication*, Wiley, 2001.

[10] Touch, J., *Mirage: A Model for Latency in Communication*, Ph.D. Dissertation, Univ. of Pennsylvania, Dept. of Computer and Information Science, Jan. 1992.

[11] Touch, J., "Parallel Communication," Proc. IEEE Infocom 1993, pp. 506-512.

[12] Touch, J., Farber, D., "An Experiment in Latency Reduction," Proc. IEEE Infocom 1994, pp. 175-181.

[13] Touch, J., "Defining 'High Speed' Protocols: Five Challenges & an Example That Survives the Challenges," *IEEE Journal on Selected Areas of Communications (JSAC)*, Jun. 1995, pp. 828-835.

[14] Touch, J., "Protocol Parallelization," *Protocols for High Speed Networks IV*, Ed. G. Neufeld and M. Ito, Chapman and Hall, London, 1995, pp. 349-360.

[15] Touch, J., "TCP Control Block Interdependence," RFC 2140, April 1997.