10 Networking Papers: Readings for Protocol Design

David Wetherall
University of Washington
djw@cs.washington.edu

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

algorithms, design, performance, reliability, security

Keywords

protocol design, readings

The last two issues of *ACM Computer Communication Review* included short pieces on reading list recommendations. I enjoyed them enough that I started to write my own. But I quickly found an overall list to be a daunting task, as I tended toward well-known, albeit classic, papers. Instead, to make progress, I have narrowed the focus to an interest of mine: protocol design.

The end-to-end argument [16], the robustness principle [14], soft-state [5] and application level framing [6] are well-known strategies that can strengthen the design of protocols as part of your research. They are an excellent starting point for readings if you are not familiar with them already. But, having read them, what else can help? The readings that follow are part of my answer to this question.

Each paper below has something to say about protocol design. That makes this list different than the earlier ones: I intend the papers to be read with an external context in mind as well as for their intrinsic value. Some provide *examples of strategies* that can be adapted and re-used elsewhere, as I've tried to point out. Others present *experiences with designs* that serve as food for thought. I have chosen papers that surprised me when I first encountered them. And I've tried to include at least some papers that you are unlikely to have read; I'd appreciate hearing of other papers I might not have read that would fit on this list too. Enjoy! And, when you are done, consider submitting your own list to *CCR*.

- J. Byers, M. Luby, M. Mitzenmacher and A. Rege, *"A Digital Fountain Approach to Reliable Distribution of Bulk Data,"* SIGCOMM '98, 1998.

There is a lovely synergy between broadcast transmission and the use of forward error correction codes. That is because different receivers can obtain different information and yet still make progress. This paper explains the basic synergy and shows it is powerful enough for efficient reliable transfer to many receivers without any classic retransmissions whatsoever! Many other designs exploit this synergy; the recent interest in network coding [7] can be considered a more general application of it with coding in the network rather than end-to-end.

- M. Harchol-Balter and A. Downey *"Exploiting Process Lifetime Distributions for Dynamic Load Balancing,"* SIGMETRICS '96, 1996.

There are many papers that characterize properties of the Internet in terms of heavy tails. This paper will tell you why it matters. It examines the problem of balancing load over a cluster and explains why very different strategies – whether to migrate running jobs or not – make sense if the job lifetime distribution is heavy-tailed and not otherwise. Heavy-tails are counter-intuitive for design. The same distinction benefits flow switching [13] and limits the value of caching [17].

- T. Rodeheffer and M. Schroeder *"Automatic reconfiguration in Autonet,"* 13th SOSP, 1991.

This paper describes a plug-and-play local switched network with automated mechanisms to mask faults. It is interesting for the notion of a "skeptic" that damps repeated faults for exponentially increasing intervals to minimize disruption, and for its experience tackling real-world faults. Both damping and online parameter tuning via exponential backoff are generally useful techniques; here you can see exponential backoff outside of Ethernet collisions.

- P. Danzig, K. Obraczka and A. Kumar, *"An analysis of wide-area nameserver traffic,"* SIGCOMM '92, 1992.

This paper will make you think about what it means for large and heterogeneous systems to be robust. It used traces to study the melting pot of the DNS. It found that the vast majority of traffic was the result of interactions between poor implementations rather than functionally necessary. This seems shocking. Yet even more shocking is the fact that few had noticed since it matters little for day-to-day operation — the design of the DNS [12] is highly robust. This thread continues with a more recent study, almost a decade on, that reveals a similar mess [3].

- S. Savage, *"Sting: A TCP-based Network Measurement Tool,"* 2nd USITS, 1999.

You can often extend protocols without changing their interfaces, simply by using them in unexpected ways. This paper provides an example. To paraphrase Savage: "Don't

think of TCP as a protocol, think of it as an opportunity." Sting co-opts TCP running on public web servers to measure loss in each direction. It will stretch your notion of backwards-compatibility. Many Internet measurement and mapping tools now extract their results by twisting the deployed protocol base in unanticipated ways.

- D. Katabi and C. Blake *"Inferring Congestion Sharing and Path Characteristics from Packet Interarrival Times,"* MIT LCS Technical Report 828, June 2001.

Packet timing is a rich source of information. This is apparent from the many tools and techniques for bandwidth estimation [15]. Yet I am always surprised at how much information remains to be used (or abused). As an example, this paper shows how to infer which connections are bottlenecked at the same resource. It does this with only receiver timing, as viewed through the lens of entropy, and with no support from senders or the network whatsoever. Since its publication, network timing has been used in completely different ways too, e.g., for fingerprinting devices [9] and extracting private keys [4].

- A. Shieh, A. Myers and E. Sirer, *"Trickles: A Stateless Network Stack for Improved Scalability, Resilience and Flexibility,"* NSDI '05, 2005.

A classic trick is to carry state along with messages for later use instead of keeping it with the party that generated it. This is analogous to the idea of continuations in programming languages. In a network context, it can shift a burden from one party to another to provide scalability. This paper uses the trick to the hilt to design a TCP-like transport without server state. Once you are used to this trick, you will recognize it in less extreme forms in common use, e.g., Web cookies [10] for sessions that store state with clients, and TCP SYN cookies [2] that push setup state to clients.

- N. Borisov, I. Goldberg and D. Wagner, *"Intercepting Mobile Communications: The Insecurity of 802.11,"* MobiCom '01, 2001.

This is a cautionary tale. It tells of weaknesses in the 802.11 security mechanisms that render them wholly ineffective in practice, not merely in theory. (Other work shows that 802.11 is vulnerable to denial-of-service attacks [1] too!) How could 802.11 be so broken? The perils of "rolling your own" security are well-known. Examples such as this remind me to seek the solid ground of proven security mechanisms.

- D. Thaler and C. Ravishankar, *"Using name-based mappings to increase hit rates,"* IEEE/ACM Transactions on Networking, 6(1):1–14, 1998.

This is a neat technique that might be described as "hashing for networks" because it is suited to distributed settings with some inconsistency. It is used in the paper for distributed load balancing that is simple, provides good locality for caching, and handles changes in the server set gracefully. (Note that consistent hashing [8] was a contemporaneous method.) It demonstrates the value of algorithms that are well-suited to the problem at hand.

- R. Perlman, *"Protocol Design Folklore,"* Ch. 19 of *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols,* 2nd ed., 1999. (Available online as draft-iab-perlman-folklore-00.txt.)

Finally, this chapter provides tips and examples over a broad set of issues. There are relatively few such readings, and the author has more experience than most of us will ever have in designing protocols that are used in the real world. It is weighted more towards protocol issues, whereas the classic "Hints" paper [11] is weighted more towards interface issues.

Acknowledgments

My thanks to Ratul Mahajan, Tom Anderson and Jim Kurose for feedback on a draft.

References

[1] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *12th USENIX Security Symposium*, 2003.

[2] D. Bernstein. Syn cookies, 1997.

[3] N. Brownlee, k claffy, and E. Nemeth. DNS Measurements at a Root Server. In *Globecom 2001*, 2001.

[4] D. Brumley and D. Boneh. Remote Timing Attacks are Practical. In *12th USENIX Security Symposium*, 2003.

[5] D. D. Clark. The design philosophy of the DARPA internet protocols. In *SIGCOMM '88*, Aug. 1988.

[6] D. D. Clark and D. L. Tennenhouse. Architectural considerations for a new generation of protocols. In *SIGCOMM '90*, 1990.

[7] C. Frangouli, J. Le Boudec, and J. Widmer. Network coding: An instant primer. *CCR*, 26(1), Jan 2006.

[8] D. Karger et al. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *STOC*, May 1997.

[9] T. Kohno, A. Broido, and k.c. Claffy. Remote physical device fingerprinting. In *IEEE Symposium on Security and Privacy*, 2005.

[10] D. Kristol and L. Montulli. HTTP State Management Mechanism, Request For Comments 2965, 2000.

[11] B. W. Lampson. Hints for computer system design. In *9th SOSP*, 1983.

[12] P. V. Mockapetris and K. J. Dunlap. Development of the domain name system. In *SIGCOMM*, 1988.

[13] P. Newman, G. Minshall, and T. L. Lyon. IP switching — ATM under IP. *IEEE/ACM Transactions on Networking*, 6(2):117–129, 1998.

[14] J. Postel. Transmission control protocol. Request for Comments 793, Sept. 1981.

[15] R. Prasad, M. Murray, C. Dovrolis, and K. Claffy. Bandwidth estimation: Metrics, measurement techniques and tools. *IEEE Network*, Nov/Dec 2003.

[16] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, Nov. 1984.

[17] A. Wolman et al. On the scale and performance of cooperative web proxy caching. In *17th SOSP*, 1999.