Design Principles for a Robust Network Infrastructure

David Wetherall and Tom Anderson
Department of Computer Science and Engineering
University of Washington
February 15, 2002.

The Internet is now central to a huge amount of our economy, and yet we are unable to depend on its continued and smooth operation. For some kinds of failures, the Internet has proven to be extremely robust. It has survived hurricanes, earthquakes, tunnel fires, and terrorist attacks with only temporary and partial loss of end-to-end connectivity. At the same time, other, seemingly trivial failures are able to disrupt the Internet largely without bound. This is because a fundamental assumption underlying the design of many Internet protocols is that systems are *fail-stop* – they completely and detectably stop working when they fail. Not all failures are so clean. Implementation bugs, configuration mistakes and malicious attacks may cause systems to obey the syntax of a protocol but in fact be behaving incorrectly, in ways that undermine the Internet's operation. These *arbitrary* failures occur with a surprising regularly and, unlike simple failures, can cascade through the system with serious consequences to parties not at fault. In one recent example, widespread outages were triggered because one vendor's BGP implementation ignores but propagates incorrect route announcements, while another vendor's routers terminate any BGP session propagating an obviously incorrect announcement. Over the past few years, operator errors in the form of router configuration mistakes have led to several spectacular disruptions to Internet connectivity. As a further example, the CodeRed and Nimda worms triggered previously unknown BGP instabilities. To put these incidents in perspective, consider that each of them arguably caused a more serious Internet outage than the September 11 terrorist attacks in which a vast amount of connectivity through New York City was severed by the collapse of the World Trade Center.

We argue that a key research challenge is to design network protocols and distributed systems that are as robust against arbitrary failures as today's Internet is against fail-stop failures. Only by meeting this challenge can we make the Internet significantly more reliable. Like simple failures, arbitrary failures will most likely continue to occur as they have done in the past; a recent study of the DNS produced the staggering fact that the majority of queries at a root nameserver were the result of implementation bugs. Yet unlike the case for simple failures, there is an almost complete lack of principles and techniques to help design protocols that can withstand arbitrary failures. Example principles that have proven their worth over time for fail-stop failures include: end hosts are responsible for error recovery, failures should be assumed to be the common case, all critical state should be "soft" or refreshed periodically, and bad news should be propagated quickly while good news should propagate slowly. Compared with this litany of principles, we are aware of only a single widely known principle that is even relevant to arbitrary failures, due to Postel: "Be liberal in what you accept and conservative in what you send." Even this principle is a double-edged sword, promoting interoperability between differing implementations at the cost of robustness. Indeed, arbitrary failures are typically ad-

dressed in an ad hoc manner, as isolated events, and have received little systematic study; there are even few example designs of truly robust Internet protocols.

Providing robustness against arbitrary failures will require new research. Established bodies of work such as cryptographic security and fault tolerance via consensus are certainly useful tools to help achieve robustness. But they are not a solution in their own right for the arbitrary failures that remain in practice. Consider encryption-based authentication. It is a powerful technique for reducing the scope of problems, as passwords and encryption can be used to validate that only authorized users or machines are allowed to participate in a protocol. Even strong authentication, however, leaves protocols open to a class of errors. Authentication can demonstrate that the speaker is who they say they are, but it cannot answer whether the speaker has been compromised or is simply behaving incorrectly. In the examples given above, problems occurred between properly authenticated hosts and not masquerading intruders. Because of the success of Internet designers in addressing fail-stop errors and the widespread use of authentication for critical services, the remainder – authorized hosts making syntactically correct but factually wrong statements – are a major source of unreliability in the Internet today. Similarly, Byzantine consensus algorithms are not a solution to our problem. Often it is infeasible to replicate a computation (e.g., packet loss at a router) to obtain consensus, and in any case we seek mechanisms that are efficient and proportionate the to danger involved, rather than those that require multiple messages to be sent along multiple paths.

Our thesis is that it is possible to design protocols that are robust against these unforeseen, arbitrary failures, and that this will require design principles that differ from those in use today. Identifying these principles will require the synthesis of knowledge from across the domains of networking and systems, software engineering, formal methods, cryptography, algorithms, and human factors. The resulting principles must be easy to understand, efficient, and scalable, or they will never be used. In the talk that accompanies this position paper, we attempt to set the stage for those principles based on case studies of implementation bugs and Internet router configuration mistakes. We give examples of problems, examples of solutions, and some early lessons learned. Our case studies are encouraging. We find that some of the many "correct" designs for accomplishing a given function are less prone to arbitrary failures than others, and that the failure modes of the more fragile designs are obvious enough in hindsight that they could be anticipated. But there will be no magic bullet. Principles to provide robustness in the face of simple failures have been formed over the past three decades. Our goal is to develop principles for robustness against arbitrary failures that are just as effective.

Biographical Sketches

David Wetherall (Presenter)

David Wetherall is an Assistant Professor in the Department of Computer Science and Engineering at the University of Washington. He joined the faculty in 1999 after receiving his Ph.D. in computer science from MIT. Prior to that he received a B.E. in electrical engineering from the University of Western Australia in 1989, and an M.S. and E.E. in computer science from MIT in 1994 and 1995. David's research interests span the area of computer systems with a focus on networking. His thesis research pioneered the active network approach by investigating the tradeoffs between functionality, performance and security when mobile code technologies are used to ease the introduction of new network services. Before MIT, David worked at QPSX Communications, a high speed networking company that led the development of the IEEE802.6 (DQDB) switching technology. Since arriving at the University of Washington, David has delved into the role of trust in network protocols. He published the first work on the "traceback" problem of finding the approximate source of spoofed packets used in denial-of-service attacks. He developed a version of congestion control signaling that is robust to implementation errors. This work is now being standardized for deployment in the IETF Transport Area Working Group. David also co-founded Asta Networks in 2000 and acted as CTO and Chief Architect. Asta Networks is a Seattle-based startup that makes innovative software products for managing large networks.

Tom Anderson

Thomas Anderson is a Professor in the Department of Computer Science and Engineering at the University of Washington. He received an A.B. from Harvard University in philosophy in 1983, and an M.S. and a Ph.D. in Computer Science from the University of Washington in 1989 and 1991, respectively. He was an Assistant Professor and then an Associate Professor at the University of California at Berkeley from 1991 to 1997, joining the faculty at the University of Washington in 1997. Anderson's research interests include networking and distributed systems. Anderson has won an NSF Presidential Faculty Fellowship Award and an Alfred Sloan Research Fellowship Award. Of the fifty-odd papers he has authored or co-authored, eleven have received awards at prestigious conferences, including the 1989 SIGMETRICS Conference, the 1989, 1991, 1995, and 1997 Symposia on Operating Systems Principles, the 1992 ASPLOS Conference, the 1993 Winter and Summer USENIX Conferences, the 1994 and 1998 HotInterconnects Conferences, and the 1996 HotChips Conference. He has been principal investigator or co-principal investigator on four major projects, starting with the Roboline scalable storage project starting in 1992, the Berkeley NOW project starting in 1994, the Berkeley IRAM project starting in 1996, and Detour/Active Names starting in 1998. In 2000, he co-founded Asta Networks, a Seattle-based startup that makes innovative software products for managing large networks. On the teaching side, Anderson developed Nachos, arguably the most popular software project for teaching undergraduate operating systems; he is currently developing related projects for teaching undergraduate networking and distributed systems.