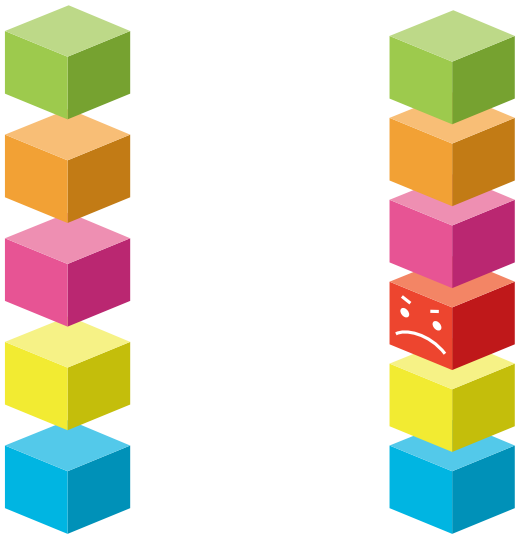

Packet Forgery By ISPs: A Report On The Comcast Affair

By Peter Eckersley, Fred von Lohmann and Seth Schoen

pde@eff.org fred@eff.org schoen@eff.org

ELECTRONIC FRONTIER FOUNDATION
eff.org

Version 1.01 November 28, 2007

Packet Forgery By ISPs: A Report On The Comcast Affair

Comcast is the second largest Internet Service Provider (ISP) in the United States. They run the cable TV and cable Internet networks in many parts of the United States, and many consumers know them as their duopoly or monopoly provider of residential broadband Internet access.

Some time around May 2007, Comcast installed new software or equipment on its networks that began selectively interfering with some of Comcast's customers' TCP/IP connections.[1] The most widely discussed interference was with certain BitTorrent peer-to-peer (P2P) file-sharing communications, but other protocols have also been affected. This white paper is intended to set forth the current state of public knowledge about Comcast's interference activities.

How Do We Know Comcast Is Forging Traffic?

Initial reports of users having trouble with BitTorrent connections began to circulate on discussion forums around May 2007.[2] Those affected appeared to be Comcast subscribers, and observers began speculating began about the causes. A Comcast subscriber named Robb Topolski ran a tool called a packet sniffer[3] while attempting to "seed" (i.e., offer to others for download) files on BitTorrent and discovered unexpected TCP RST packets that were causing inbound connections to his computer to die. Based on his observations, he speculated that Comcast may have been responsible for this interference.

TCP is a standard protocol that computers use to exchange information on the Internet.[4] RST packets, also known as "reset" packets, are a kind of TCP message that is normally sent when a computer receives TCP packets that it believes it should not have received, or when it thinks it has closed a connection but keeps receiving traffic from the other side. When received, RST packets will generally cause ordinary networking software to close its side of the connection in response.

1 Circumstantial evidence suggests that Comcast may be utilizing equipment distributed by a company called Sandvine Incorporated. See <http://tinyurl.com/3bxtff>.

2 There were some complaints prior to the May 2007 reports, but they were not tied to Comcast interference. See <http://tinyurl.com/2amrgo>; <http://tinyurl.com/2fbxkl>; <http://tinyurl.com/28nsjl>.

3 The Internet and most other digital networks operate by sending small parcels of information, called "packets," backwards and forwards between computers. Visiting a single webpage, or sending a single email, typically involves many packets being sent back and forth. A "packet sniffer" is a program that allows a human being to record and retrospectively examine some or all of the packets being sent and received over a network.

4 All of the traffic on the Internet uses a protocol called IP ("Internet Protocol") to arrange for information to get to the right computers. About two-thirds of the traffic on the Internet also uses TCP ("Transmission Control Protocol") along with IP. TCP ensures that computers communicate sensibly over networks even where transmissions may be lost or corrupted or arrive in a different order than they were sent in. Aside from TCP, most of the remaining third of Internet traffic is UDP over IP. See CAIDA Passive Network Monitors <http://tinyurl.com/yqgfxb> (accessed Nov. 2007).

After becoming aware of Topolski's research, EFF contacted Comcast to inquire about these reports of interference with BitTorrent communications. A Comcast representative told us that while Comcast did perform "network management" that might interfere with particular subscribers in rare circumstances, it did not block or target any application or protocol. [5]

In the wake of Comcast's representations to us, we continued to receive reports of protocol-specific interference, leading us to begin performing our own tests. We observed that our attempts to seed a test file (a public domain book) using BitTorrent over a Comcast residential broadband account failed, with connections being disrupted by unexpected TCP RST packets. The Associated Press (AP) was apparently conducting similar experiments, and they subsequently brought the story to widespread public attention.[6]

The EFF tests used a packet sniffer called Wireshark at both ends of a connection: one on Comcast's network, one from elsewhere. Our tests confirmed that the RST packets were being forged and injected somewhere in the network between the two communicating parties. For example, if we call one end of the connection Alice and the other end Bob, Alice receives a number of RST packets (typically 3-5) from Bob, but Bob's packet sniffer has no record of his computer ever having sent them. Bob, in turn, receives a series of RST messages from Alice, but Alice's computer similarly has no record of having sent them. These inconsistencies in the packet logs at each end of the connection demonstrate that some intermediate party was forging traffic in both directions; each side receives forged RST packets that contain a sender IP address and TCP sequence number that falsely indicates that it was sent by the other.

EFF's tests corroborated AP's results — comparisons of packet logs between two communicating parties showed that an intervening computer (almost certainly Comcast's) was injecting forged RST packets into the communications, effectively telling both ends of the connection to stop communicating. We replicated these tests using Comcast residential broadband accounts in California and Oregon. We controlled for the possibility that other intermediary ISPs might have been involved by testing several connections provided by other ISPs (including Sonic, AT&T, and overseas ISPs). In a series of over a dozen tests, we observed only jamming of connections inbound to Comcast subscribers.[7] The only likely explanation of these observations is that Comcast was forging and injecting the RST packets in order to interfere with certain connections.

For readers who are interested in the full technical details of this process, as well as instructions on replicating the experiments, EFF has published a separate, and much more detailed, technical guide.[8]

What Communications Are Affected?

Initial investigations suggest that Comcast is interfering with some subset of protocols, rather than interfering equally with TCP/IP traffic generally. EFF has run tests of Comcast's treatment of BitTorrent, Gnutella, and World Wide Web (i.e., HTTP) protocols. We have seen

5 See Seth Schoen, "Comcast and BitTorrent," *EFF Deeplinks* blog, Sept. 13, 2007, <http://tinyurl.com/27jftt>.

6 Peter Svensson, Comcast Blocks Some Internet Traffic, *SF Chronicle*, Oct. 19, 2007, <http://tinyurl.com/2kq6n4>.

7 The connection being established by the non-Comcast user does not necessarily tell us which of them was going to be downloading, and who uploading, the data, although it is usual for connections to BitTorrrent seeds to be established by the downloader.

8 Seth Schoen, Detecting packet injection: A guide to observing packet spoofing by ISPs, EFF White Paper, <http://www.eff.org/wp/detecting-packet-injection>

definite interference by injection of RST packets into certain classes of BitTorrent and Gnutella TCP sessions (which we explain in more detail below).

There have also been credible reports of TCP RST packet forgery occurring against Lotus Notes communications, a "groupware" suite used by many businesses for email, calendaring and enterprise file sharing[9]. Following public discussion of this issue, Comcast reportedly adjusted its systems so that Lotus Notes works correctly again.[10] One firm also reported that Comcast was jamming their clients' Windows Remote Desktop connections. The report appeared quite credible (the submitter informed us that they had numerous clients, and were experiencing problems only with those using Comcast), but it did not contain concrete evidence in the form of packet logs. The submitter subsequently informed us that the problem had dissipated. Because the resolution coincided with the resolution of Lotus Notes interference, we believe that changes to Comcast's jamming algorithms are the most likely explanation for these changes.

EFF has also received unconfirmed reports that Comcast is interfering with other protocols. In particular, some Comcast users have reported that medium and large-sized FTP and HTTP transfers have been interrupted. The FTP and HTTP reports, however, have not included enough detail for us to be certain that there is a problem attributable to forgery of packets by Comcast. Our attempts to test for interference in large HTTP transfers have occasionally resulted in what seem to be interrupted connections, but these results are not consistently reproducible, and we cannot say at this point that there is any interference or that it is caused by Comcast.[11]

We do not presently have enough data to form complete theories about the details of the algorithm that Comcast has been using to select connections for interdiction. We intend to continue testing, however, and will post an update based on our results or those of others.

What Are The Effects Of Comcast's Packet Forgery?

There has been some confusion about the impact of Comcast's interference, with Comcast characterizing the impact on its customers as "delaying" some network communications. As both a technical and metaphorical description, this characterization is incomplete and misleading.

The consequences of Comcast's spoofing of TCP RST packets are complicated. At a low level, the forged RST packets cause the targeted TCP connections to die as soon as computers try to establish them.[12] But the practical consequences depend on which higher level protocol

9 See Kevin Kanarsi, "Comcast f iltering Lotus Notes (update)" <http://kkanarski.blogspot.com/2007/09/comcast-filtering-lotus-notes-update.html>; Peter Eckersley, "Comcast is also Jamming Gnutella (and Lotus Notes?)", *EFF Deeplinks*, <http://www.eff.org/deeplinks/2007/10/comcast-also-jamming-gnutella-and-lotus-notes>.

10 See Peter Eckersley, "Comcast needs to Come Clean", *EFF Deeplinks*, <http://www.eff.org/deeplinks/2007/10/comcast-needs-come-clean>. Comcast has asserted in direct communications with EFF that Lotus Notes interference was caused by a bug unrelated to their treatment of BitTorrent and Gnutella traffic. However, Lotus Notes communications were being reportedly affected by TCP RST injections similar to the ones we observed jamming other P2P connections, and Comcast has not provided any technical details to corroborate their characterizations of the Lotus Notes problem.

11 In particular, forged TCP RST packets were not the observed cause of difficulties with HTTP transfers. EFF is hoping to write additional software to test for other kinds of modifications to Internet traffic that would be more subtle than outright RST forgery and will report on the results of further investigations.

12 Reset packets are defined in the TCP specification. See RFC 793 / Internet Standard STD 7, <http://tools.ietf.org/html/rfc793>; see also RFC 4614, <http://tools.ietf.org/html/rfc4614> (surveying supplements to RFC 793); W. Richard Stevens, *TCP/IP Illustrated Volume 1: The Protocols* 246-250 (1994) (for an excellent

(Gnutella, BitTorrent, Lotus Notes, etc) was using the TCP/IP connection, and on the particular software that is implementing that protocol, and on the way that the user interacts with that software.

In many cases, however, injection of forged RST packets will cause software to fail in its attempts to do something a user asks of it. For instance, a BitTorrent client elsewhere on the Internet may fail in downloading a rare document that is available as a BitTorrent seed from a Comcast user.[13]

In the case of a typical Gnutella node, RST forgery will impair the node's ability to discover and establish proper communications with other parts of the Gnutella network. Gnutella connections can normally be started in either direction: the Comcast user connects outwards, or other Gnutella nodes connect inwards. So, for example, when Alice's Gnutella client starts up, it runs through a "cache" of nodes that it has communicated with in the past.[14] It attempts to make outbound connections to these nodes, in the hope that some of them are currently online (most of them are not, because Gnutella nodes are usually transient). At the same time, other Gnutella nodes may be connecting inwards, either because they have Alice's IP address in their cache, or because a node Alice has established a connection with tells them that Alice is online.[15] We observed these inwards connections being jammed by Comcast. The practical result is that it takes longer — potentially much longer — for Alice's Gnutella node establish connections with a sufficient number of other healthy Gnutella nodes to ensure reliable data transfers.[16] Because it takes longer to establish these connections, it takes longer for the node to begin obtaining meaningful results for its searches (generally speaking, only after users have meaningful search results, can they initiate downloads). Comcast's interference will also have certain large-scale effects on the structure of the Gnutella network, because there is a large set of nodes (those on Comcast's network) which can only be talked to by outside nodes when the Comcast nodes initiates the connection. So, for instance, Comcast's jamming prevents conversation between Comcast nodes and nodes that are behind firewalls. These limits on interconnection are likely to reduce the effectiveness of the Gnutella network for all of its users.

In our tests, we did not observe Comcast forging RST packets to interfere with Gnutella search, upload or download operations.[17] It was only the initial connection attempts that failed.

exposition). According to RFC 793 / STD 7, RST packets were conceived as a means for a computer to signal that the connection no longer exists at its end (see RFC 793, Section 3.4); normally, this might be caused by a computer rebooting or by a very large number of dropped packets causing a connection to be closed. They may be used to perform an abortive reset when one party wishes to close a connection quickly and signal an error to the other party; see Stevens at 247-8. RST packets are also used in response to a TCP connection attempt to signal that the connection was refused by the destination host; see RFC 793 at 69; Stevens at 247.

13 Note that in this case, the greatest harm is suffered by the non-Comcast user who was trying to download the rare file, although the Comcast user is also frustrated in their attempt to share the file with others.

14 Note that the details of how Gnutella maintains the cache and performs the discovery process vary between different implementations of the Gnutella protocol.

15 This story is a slight simplification, because modern variants of Gnutella use two kinds of nodes: ultra nodes, that connect to many other ultra nodes and leaf nodes, and leaf nodes, which connect to a small number of ultra nodes. If a leaf node connects to another leaf node, it will disconnect automatically, but it may exchange some information about the addresses of ultra nodes first.

16 Some Gnutella nodes are run by spammers and send various types of fake results rather than participating in the network properly; the impact of Comcast's jamming is likely exacerbated by this fact.

17 We have not yet tested the impact on Gnutella "push" downloads, which are a mechanism Gnutella uses to upload files from behind firewalls (Gnutella arranges for the TCP connection to be established by the uploader rather than the downloader). It would not be inconsistent with the pattern of observed jamming for "push" downloads to a Comcast subscriber to be blocked. Our tests continue.

Users whose ability to upload or find and download a rare file on the Gnutella network was dependent on a connection that would have been established from a non-Comcast node to a Comcast node will have lost this functionality because of Comcast's interference. Also, some users will be discouraged enough by Gnutella's reduced performance that they give up. While it is difficult to say how many users are in this category, Comcast's efforts to impair Gnutella's connection establishment will drastically effect how well Gnutella works for this set of users.

So, in many cases, Comcast subscribers will experience problems more severe than a mere "delay" to their traffic. For instance, a user who tries to publish a file by seeding it on BitTorrent (as the Associated Press did with the Bible, and as we did with other copyright-free texts in our tests) will find that others are unable to download the file from them. And, as described above, a user who tries to use Gnutella to find a file but gets no meaningful search results after trying for ten minutes may well give up, concluding that Gnutella is ineffective. In both of these examples, Comcast's packet forgery prevents the transfer of data rather than delaying it.

In fact, the characterization of Comcast's packet forgery as "delaying" certain traffic is only true under special conditions, and is certainly not true in general. We can think of only two examples of such special conditions:

— If Comcast does not jam connections all of the time, and the software that is being jammed keeps reattempting its connections indefinitely, and the user doesn't give up and close the software, then the packet forgery would have had the effect of merely delaying a certain communication.

— If a non-Comcast user named Alice was trying to download a file over BitTorrent, and that file was seeded by a Comcast user named Charlie and another non-Comcast user named Delilah, then even if Alice's connection to Charlie is jammed, she might still be able to download the file from Delilah. Comcast might argue that Alice's download is merely "delayed" (i.e., she was forced to download the file more slowly from non-Comcast customers only) rather than prevented altogether.

In circumstances other than these special cases, Comcast customers will not experience the interference as a "delay"; their software will simply not work.

What Is So Bad About Comcast's Actions?

One objectionable aspect of Comcast's conduct is that they are spoofing packets — that is, impersonating parties to an exchange of data. Comcast is essentially deploying against their own customers techniques more typically used by malicious hackers (this is doubtless how Comcast would characterize other parties that forged traffic to make it appear that it came from Comcast). In this sense Comcast is behaving worse than if they dropped a proportion of packets under congested circumstances in order to throttle bandwidth usage, or even if they blocked certain ports on their network. In other words, Comcast is essentially behaving like a telephone operator that interrupts a phone conversation, impersonating the voice of each party to tell the other that "this call is over, I'm hanging up."

It might be argued that Comcast is primarily deceiving computers, rather than human beings, but humans may be misdirected and forced to cope with Comcast's deception. The failure of packets to convey the meaning specified by the protocol means that human beings will get misleading messages from their software ("remote host closed connection," as opposed to "connect blocked" for instance). It also means that programmers cannot rely on standards to ensure that their software responds in a manner appropriate to the circumstances. If ISPs continue to

forge and inject RST packets, for example, programmers will have to ask themselves "does an RST packet at such and such a moment mean that an old TCP connection is still active, or that the other end doesn't want to talk, or that some ISP is interfering"? In other words, ISPs could become an omnipresent adversary that developers have to constantly worry about when writing their code.

Comcast's conduct also threatens innovation by undermining the end-to-end principle.[18] The Internet has enabled a cascade of innovations precisely because any programmer — whether employed by a huge corporation, a startup, or tinkering at home for fun — has been able to create new protocols and applications that operate over TCP/IP, without having to obtain permission from anyone. Comcast's recent moves threaten to create a situation in which innovators may need to obtain permission and assistance from an ISP in order to guarantee that their protocols will operate correctly. By arbitrarily using RST packets in a manner at odds with TCP/IP standards, Comcast threatens to Balkanize the open standards that are the foundation of the Internet.

Comcast's interference is potentially troubling as well to the extent it may hobble potential competitors deploying next-generation video distribution services. BitTorrent Inc., for example, now distributes films under license from Hollywood movie studios[19] and thus competes with Comcast's cable TV products. Similarly, Vuze, which recently filed a petition with the FCC for rule-making regarding Comcast's interference practices, also sells downloads from a huge library of licensed content, using BitTorrent as a distribution mechanism.[20] Other companies and products, such as Joost and Miro, also rely on P2P protocols that are similar to those that are being impeded by Comcast. Efforts undertaken by Comcast that interfere with the ability of these next-generation competitors in the video distribution marketplace are cause for concern.

What About "Network Management"?

Comcast has asserted, without any details, that its actions are necessary for managing the impact of high-volume users who cause congestion on their cable networks. Based on the information Comcast has disclosed, it does not appear that this presents a compete picture of Comcast's activities, nor does it adequately justify them.

It is true that some broadband users send and receive a lot more traffic than others, and that interfering with their traffic can reduce congestion for an ISP. This does not imply that protocol-specific packet forgery is a necessary or legitimate means of responding to the congestion;

18 The end-to-end principle holds that the Internet should allow users' computers—the end points—to talk to each other without interference. That way, the functionality of the network is not determined by any of the parties that operate the network's core, but by the users at the ends of each link and the software they choose to run. This ensures both that the best information is available to implement features efficiently, and also that network users have autonomy in determining how their software will communicate. The end-to-end principle was originally set out in J.H. Saltzer, D.P. Reed & D.D. Clark, "End-to-End Arguments in System Design", 2 *ACM Transactions on Computer Systems*, 277-288 (1984) <http://tinyurl.com/3ceaux>. RFC 1958, Architectural Principles of the Internet, 1996, <ftp://ftp.isi.edu/in-notes/rfc1958.txt>, argued that the end-to-end principle was an essential and threatened dimension of the Internet's design; more recent developments are discussed in RFC 3724, The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture, 2004 <http://www.ietf.org/rfc/rfc3724.txt>. The argument has propagated from technical to legal and policy circles. See Lawrence Lessig & Mark A. Lemley, "The End of End-to-End: Preserving the Architecture of the Internet in the Broadband Era," 48 *UCLA Law Review* 925 (2001).

19 See <http://www.bittorrent.com/about/partners>.

20 See <http://www.vuze.com/>.

there are more reasonable mechanisms available to ISPs to ensure that low-volume users are not crowded out by high-volume users, which we discuss below.[21]

Furthermore, in our testing, we saw no evidence that Comcast was targeting their jamming efforts at customers based on their individual consumption of bandwidth. For example, an attempt to seed a 500KB file to a single BitTorrent downloader, instigated after the seeding Internet connection had been idle for the preceding day, triggered the injection of forged RST packets. The pattern of interference by Comcast was exactly the same after the user had uploaded 500MB or so of data over the following day. If Comcast had carefully engineered its interventions to prevent certain users from contributing disproportionately to network congestion, we would expect to see jamming only after subscribers consumed large amounts of bandwidth, or when they were participating in large numbers of connections in a short period of time.[22]

There are methods available to Comcast to limit the amount of traffic that P2P software transmits on their network, without preventing any categories of connections, interfering with any protocols, or forging packets. For example, ISPs can implement dynamic per-user traffic shaping. They can set a limit on the amount of data per second that any user can transmit on the network. They can also set these limits on a dynamic basis, so that (1) the limits are gradually relaxed as the network becomes less congested and vice-versa and (2) so that the limits primarily slow the traffic of users who are downloading large to very large files that take minutes to transfer. We have observed Comcast to take most of these steps in managing their cable networks, but in our testing, we have never seen them make the kinds of dynamic adjustments to their rate limits that would be necessary to gracefully avert severe network congestion.[23] This suggests – though it cannot prove – that even if Comcast began forging RST packets in response to problems with network congestion, they did not exhaust the reasonable, user-

21 The FCC has made this point itself in connection with the pending auctions of 700 mhz spectrum: "C Block licensees cannot exclude applications or devices solely on the basis that such applications or devices would unreasonably increase bandwidth demands. We anticipate that demand can be adequately managed through feasible facility improvements or technology-neutral capacity pricing that does not discriminate against subscribers using third-party devices or applications." Service Rules for the 698-746, 747-762, and 777-792 MHz Bands, Second Report & Order, 22 FCC Rcd. 15,289, 15370-71 (August 10, 2007).

22 A handful of engineers have hypothesized that Comcast's activities might be motivated by specific interactions between P2P protocols and the DOCSIS protocols that cable modems use to share the loop of cable that runs around each street. See <http://tinyurl.com/2exmdz>; <http://tinyurl.com/257pwa>; <http://blogs.zdnet.com/Ou/?p=852>. It is true that deployed variants of DOCSIS do suffer from design flaws that make them inherently bad for carrying protocols built on top of TCP, such as HTTP or BitTorrent. See Jim Martin, "The Interaction Between the DOCSIS 1.1/2.0 MAC Protocol and TCP Application Performance," *Proc. Int'l Working Conf. on Performance Modeling & Evaluation of Heterogeneous Networks*, P57/1-10, (2004). But we believe there are serious technical inconsistencies present in all of the public and private conjectures we have seen that purport to explain why wholesale interdiction of connections is "necessary" for preventing DOCSIS-specific problems. Comcast itself has not offered any technical claim or explanation for why RST forgery might be necessary.

23 In our observations, an upload from a machine on Comcast's network will initially be given around 180 KB/s in bandwidth; after a short period, the transfer is throttled back to around 45 KB/s. This has the effect of prioritizing latency-sensitive downloads (like HTTP requests for web pages) over sustained downloads of large files with any protocol and is similar to the static rate limiting that Martin (Id.) concluded was insufficient for TCP over DOCSIS. But if Comcast were encountering severe DOCSIS request-to-send congestion (along the lines discussed by Martin), and Comcast were making full use of traffic shaping to tackle the problem, we would expect to see upload and download rate limits vary over time, to ensure that the request-to-send channel never reached dangerous levels of contention. We therefore believe that even if Comcast were motivated by congestion to introduce their jamming systems, they did not exhaust reasonable and non-discriminatory rate limiting options first.

friendly, and standards-compliant responses before they began taking decidedly less reasonable measures.

Whatever congestion control mechanisms an ISP may choose to deploy, it is critical that it informs consumers of the limits that such mechanisms will impose on their Internet access. Unfortunately, ISPs frequently advertise their services as "unlimited," unmetered Internet connections. Subscribers who purchase "unlimited Internet access" have no reason to expect that particular applications or protocols will fail based on protocol-specific interference by their ISPs. In fact, increased transparency in the market for Internet access may encourage marketplace solutions that encourage customers to sort themselves into high- and low-bandwidth groups.[24]

What Countermeasures Are Available Against Comcast's Interference Activities?

Individual users have few (if any) options to unilaterally defend themselves against Comcast's packet forgery. Collectively, however, the community of users and software developers may be able to develop effective countermeasures against Comcast's current interference activities, although the costs of deploying these may be high.

Individual users cannot do much to protect TCP connections against RST spoofing, because the forged packets are being sent in both directions. Although Alice might be able to configure a firewall to recognize and intercept Comcast's forged packets before they affect the state of her computer's network communications, there is no way she can ensure that Bob has gone to the same lengths. Moreover, Alice acting alone may have difficulty filtering out Comcast's forged RST packets without the risk of also blocking RST packets that were legitimately sent by the parties with whom she is communicating. The use of cryptography offers another possible countermeasure, but it again requires that Alice secure Bob's cooperation before it can be deployed.

Because unilateral RST filtering and encryption are ineffective, the only feasible option for end users is to find protocols, or alternative use-cases for their existing protocols, that are not blocked by Comcast. For example, users intent on sharing large files could opt to do so using email attachments or Lotus Notes, assuming Comcast is not interdicting those protocols. Of course, Comcast could begin interfering with other protocols at any time.

Software developers have more options than individual users to defend traffic against RST spoofing. Their strongest card is cryptography. By modifying the software that both Alice and Bob run, software developers can ensure that both Alice and Bob use the same encryption system. Encrypting traffic theoretically lets them authenticate the authorship of each packet, ensuring that none of them are forged, and prevents ISP intermediaries from telling which protocol a particular connection is using. If ISPs cannot identify the protocol a particular connection is using, they cannot directly discriminate based on protocol.

In practice, achieving this outcome may be difficult and costly for software developers. On top of the engineering required to implement an encrypted variant of existing protocols, there are

24 The Australian broadband market offers an illustration of how this can work in practice. The selection of Australian broadband options can be searched at <http://bc.whirlpool.net.au/bc-plan.cfm>. It includes a wide selection of plans with different peak and off-peak quotas, some with a traffic shaping after a quota has been passed and others with a wide range of per-gigabyte fees. It also includes explicitly "no set limit" plans where the ISP reserves the right to deem certain usage excessive, and more expensive, truly unlimited plans where the user can saturate their link 24/7 if they wish.

numerous other considerations. For example, developers will have to find an adequate public key management system for P2P protocols; they may also have to employ low-level cryptographic protocols (such as IPsec) to effectively disguise the underlying protocol being used, requiring changes to the users operating system. They may also need to design their applications to resist ever more determined "traffic analysis" attacks by ISPs seeking to determine what protocols and kinds of data subscribers may be using.[25] This "arms race" may ultimately force ISPs to rely on dynamic, protocol independent traffic shaping — something Comcast could implement today.

Acknowledgements

Thanks to Robb Topolski for recognizing the anomalies on Comcast's network and for helping assisting with some of our tests.

25 Of course, the increased deployment of secure, encrypted data protocols on the Internet may yield collateral benefits to Internet users in the form of enhanced privacy protections and protection from unauthorized or unlawful surveillance.