Notes on P2P Blocking and Evasion

Eric Rescorla
ekr@rtfm.com

1 Introduction

In mid-2007 it was revealed [4] that Comcast was blocking peer-to-peer traffic (most famously BitTorrent) on their network by injecting RST packets to terminate TCP [7] connections. The BitTorrent community almost immediately discovered carrying BitTorrent over an encrypted tunnel (VPN or SSH) was not subject to blocking, thus completing another cycle of the ongoing arms race between peer-to-peer implementors and network operators. This paper explores some predictable next moves in the game and their consequences for the network.

2 Blocking Techniques

It's important to realize what the ISP wants is to *selectively* block the offending traffic. Since they control the customer's link, they can easily cut them off entirely, but that loses them a customer. Similarly, the ISP could just rate limit aggregate customer traffic, but this can produce an unsatisfactory user experience for the protocols that they favor. Thus, the technical challenge is to suppress or throttle only certain traffic classes.

There are two basic blocking strategies available to network operators, depending on the location of the *enforcement point* (EP):

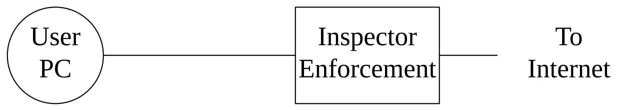

Figure 1: Inline inspection/enforcement

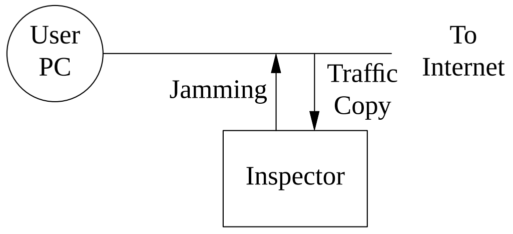

Figure 2: Non-inline inspection and enforcement

In the simplest case, shown in Figure 1, the EP is placed in between the user and the Internet. The EP sees all traffic that goes by, determines whether it should be permitted, and if not drops the relevant packets. This strategy is simple, effective, and flexible, since each packet can be individually handled, but because the EP is in the path for customer data flow, the EP must conform to very high performance and stability requirements.

Figure 2 shows the major alternative design, which is similar to that used by Comcast. The EP is placed so that it can passively inspect the traffic, but out of the main traffic path so that EP failures do not lead to connectivity loss. When the EP detects traffic it wishes to block, it jams the relevant connections; in the specific case of Comcast, this means sending TCP RSTs with source addresses corresponding to the other host, causing the TCP connection to be torn down. This is also the technique reportedly used in the Great Firewall of China [2]. The major disadvantage of this technique is that it is far less flexible. An inline EP can exert very fine control over customer flows, including dropping or delaying individual packets. On out-of-line device can mostly just terminate flows.

In both cases, the ISP needs to be able to distinguish classes of traffic they want to discourage/block from classes of traffic they wish to pass. Because modern P2P systems tend not to run on well-known ports, this cannot be done with simple filters. Instead, the ISP needs to examine the traffic and attempt to identify it. The two major available techniques are *deep packet inspection* (DPI), also known as content analysis, and traffic characteristics analysis (see, for instance, Karagiannis et al. [5]).

3 Evasion Techniques

Broadly speaking, evasion techniques fall into two categories:

- Make your protocol hard to jam.
- Use a protocol that's hard to distinguish from "good" traffic.

We discuss these strategies in sequence.

3.1 Resilient Protocols

It's relatively easy to defeat a non-inline strategy like that shown in Figure 2. As Clayton et al. observe, blocking by RST can be countered simply by modifying both TCP stacks to ignore RSTs.[1] However, this countermeasure is itself easily countered by having the EP send FIN rather than RST packets. RSTs are typically used for jamming used because they are subject to less stringent checking, but with sufficently fine information about the TCP connection, appropriate FINs can still be constructed.

Cryptographic Authentication A more robust defense is to use cryptographic endpoint authentication for communication been the peers. This prevents the the ISP from inserting their own traffic into the connection. This authentication can't be provided at the appplication layer above TCP (e.g., SSL/TLS [3]). The attacker is attacking the TCP layer and security protocols like SSL/TLS depend on correct functioning of the lower layer protocol to function correctly. In fact, SSL/TLS makes the problem worse, since any interference with the ciphertext causes integrity failures and connection failure.

In order to be resistent to this kind of injection, you need to have message integrity at a layer below TCP. The traditional solution would be to use IPsec [6], but software architecture and NAT/firewall concerns make this unattractive. A more likely approach would be to use UDP packets and run security above them as with Datagram TLS. [8]. None of the usual communications security concerns about active attack (e.g., man-in-the-middle) really apply here, since they would require the ISP to proxy the cryptographic handshake for every communicating pair, which is prohibitively computationally expensive, and could be made even moreso with well-known techniques.[2]

This type of cryptographic technique provides an extremely high level of resistance to jamming type countermeasures. If peer-to-peer protocols move to jamming-resistant cryptographic protocols, then ISPs who wish to block P2P traffic will need to use inline techniques.

Partial Blocking Inline blocking techniques introduce their own possibilities for protocol arms races. For example, ISPs may choose to only block some packets on a connection in order to throttle it but not block it entirely. The IETF standard rate control techniques used in TCP and DCCP rely on packet loss as a signal of congestion. Thus, blocking only a small number of packets (order 10-20%) would be enough to force the sender to dramatically reduce his sending rate. However, P2P users and implementors can counter by deploying protocols with more aggessive sending behavior and slower response to congestion, with the result that the amount of offered traffic increases, potentially crowding out other traffic even more, resulting in a need for yet higher levels of blocking.

[1] Note that this has some drawbacks since RSTs are commonly used for a number of purposes. [1]

[2] For instance, forcing each side to solve a computational puzzle during connection initiation.

3.2 Hiding Traffic

The other major strategy is to prevent detection. If the ISP wants to selectively discriminate against certain traffic classes, then the user can respond by making traffic from those classes indistinguishable from traffic that the ISP does not want to block (with Web and business VPN traffic being two very important categories here).

Content Analysis Content analysis/DPI is fairly easy to defeat with standard encryption techniques, which can completely hide the content from any passive inspector. Note that the lower down in the protocol stack encryption is done the better this works. For instance, there have been reports [4], that BitTorrent encryption is not effective against Comcast's blocking techniques, whereas SSH is, which may be the result of the protocol being recognizable as BitTorrent, even if the data being sent is inaccessible to the DPI engine.

Traffic Pattern Analysis The major defense against traffic pattern analysis is to make the traffic the user wishes to protect appear to be as similar as possible to traffic which the ISP does not desire to block. For instance, Karagiannis et al. [5] use two major features to detect P2P traffic: simultaneous use of TCP/UDP between a single source/destination IP address pair and whether source and destination ports are 1-1 between a single pair of IP addresses. Both of these features can be easily changed. For instance, the P2P protocol could be run entirely over UDP and deliberately scatter messages over a number of ports. That these metrics are easy to evade is unsurprising; current traffic identification work seems to focus on fast classification of existing traffic rather than being robust against traffic that is deliberately attempting to evade classification. It is not clear whether there are metrics which cannot be evaded.

If performed at the right layer of the stack, encryption may also be effective against traffic characteristic analysis techniques by hiding the features which are being used for detection. For instance both of the characteristics used by Karigiannis et al. [5] are easily hidden by setting up an IPsec tunnel between the peers prior to sending application traffic. Since IPsec is a popular VPN protocol, this makes it very hard to discriminate against any IPsec traffic. Of course, another way to look at this technique is as having P2P traffic imitate the characteristics of enterprise VPNs.

One set of features deserves special consideration because they are to some extent inherent in the P2P model rather than artifacts of particular design decisions, and are harder to hide with encryption: flow rates and packet size. In neither case do P2P flows exactly match those of other applications. For example, consider the case of rates. Most client/server applications involve either downloads (Web, YouTube, ...), or balanced flows (interactive voice and video); this suggests a strategy of blocking large net outflows might be symmetric. But of course this is easily stopped by sending simultaneous cover traffic in the reverse direction, thus making the flow appear more like a voice/video flow. For greater verisimilitude, one might send the packets out at constant rates without any congestion control. Alternatively, one could upload large amounts of data during idle periods to simulate the bursty up/down flows seen in enterprise VPNs. Note that in both cases, the result is that the ISP cannot effectively block user data, but that the net load on the network has increased dramatically, leaving them worse off than before.

4 Summary

The history of peer-to-peer use on the Internet has already undergone several cycles of blocking and anti-blocking countermeasures. Countermeasures exist for the current generation of blocking technologies and it seems likely that the next several generations of blocking technology also can be countered. It also seems likely that as blocking becomes more aggressive, P2P implementors and users will deploy systems that are more robust but make increasingly inefficient use of the network with the result that the net offered load on the provider networks increases rather than decreases.

References

[1] M. Arlitt and C. Williamson. An Analysis of TCP Reset Behaviour on the Internet. *ACM SIGCOMM Computer Communication Review*, January 2005. http://pages.cpsc.ucalgary.ca/%7Ecarey/papers/2005/TCP-Resets.pdf.

[2] R. Clayton, S. J. Murdoch, and R. N. Watson. Ignoring the Great Firewall of China. Workshop on Privacy Enhancing Technologies, June 2006.

[3] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346 (Proposed Standard), Apr. 2006. Updated by RFCs 4366, 4680, 4681.

[4] Ernesto. Comcast Throttles BitTorrent Traffic, Seeding Impossible. http://torrentfreak.com/comcast-throttles-bittorrent-traffic-seeding-impossible/, August 2007.

[5] T. Karagiannis, A. Broido, M. Faloutsos, and K. claffy. Transport layer identification of p2p traffic. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 121–134, New York, NY, USA, 2004. ACM.

[6] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), Dec. 2005.

[7] J. Postel. Transmission Control Protocol. RFC 793 (Standard), Sept. 1981. Updated by RFC 3168.

[8] E. Rescorla and N. Modadugu. Datagram Transport Layer Security. RFC 4347 (Proposed Standard), Apr. 2006.