The Internet's Not a Big Truck: Toward Quantifying Network Neutrality[⋆][†]

Robert Beverly[1], Steven Bauer[1] and Arthur Berger[2]

[1] MIT CSAIL, Cambridge MA 02139, USA
[2] Akamai/MIT CSAIL, Cambridge MA 02139, USA
{rbeverly,bauer,awberger@csail.mit.edu}

Abstract. We present a novel measurement-based effort to quantify the prevalence of Internet "port blocking." Port blocking is a form of policy control that relies on the coupling between applications and their assigned transport port. Networks block traffic on specific ports, and the coincident applications, for technical, economic or regulatory reasons. Quantifying port blocking is technically interesting and highly relevant to current *network neutrality* debates. Our scheme induces a large number of widely distributed hosts into sending packets to an IP address and port of our choice. By intelligently selecting these "referrals," our infrastructure enables us to construct a per-BGP prefix map of the extent of discriminatory blocking, with emphasis on contentious ports, i.e. VPNs, email, file sharing, etc. Our results represent some of the first measurements of network neutrality and aversion.

1 Introduction

As the Internet has matured, its success has spurred not only technical innovation, but also social, economic and regulatory responses [1]. One initially unanticipated response is a form of policy control employed by network operators known as "port blocking." Port blocking relies on the close coupling between particular applications and their assigned TCP or UDP port. Since many applications use well-known port numbers [2, 3], port blocking is one technique to stop traffic belonging to a particular application or class of application.

This research seeks to quantify the extent of port blocking on the Internet. We present a hybrid active/passive measurement-based approach that is capable of rapidly testing large parts of the Internet topology. Our scheme induces peer-to-peer (P2P) clients in the Gnutella network to probe for port blocking as part of their natural overlay formation process. Our technique does not degrade or disrupt the performance of the P2P network.

Our objective is to provide unbiased information about port blocking on the Internet. We therefore do not attempt to argue which network operational practices are "legitimate" or "justifiable." Such judgments are not purely technical, but rather must be made in the context of a well-informed larger discussion.

[⋆] Flippant title adopted from Senator Ted Stevens' remarks to the United States Senate Commerce Committee vis-à-vis network neutrality.

[†] This work supported in part by Cisco Systems and NSF Award CCF-0122419

The prevalence and type of port blocking is of technical interest to application developers and academics but, perhaps more importantly, has prominently arisen in regulatory and policy debates – in particular debates over **network neutrality** [4,5]. For example, the United States FCC recently ordered a provider to cease port blocking a competing telephony service [6].

> *"Allowing broadband carriers to control what people see and do online would fundamentally undermine the principles that have made the Internet such a success."* – Vint Cerf [7]

Unfortunately, many underlying arguments that guide neutrality discussions are based on assumptions rather than careful measurement. While many definitions of neutrality exist, port blocking is an important and well-defined dimension of the debate. Port blocking is a simple and cheap mechanism for operators to control the type of traffic on their network. Indeed blocking can be employed for altruistic reasons, for instance to staunch the spread of Internet worms [8], or as a security measure to protect potentially vulnerable applications [9]. However, providers also leverage blocking for anti-competitive or economic purposes, for example blocking high-bandwidth file sharing applications or forcing subscribers to use their provider's email gateway [10–13].

Our primary contribution is the design and implementation of a novel methodology for measuring Internet port blocking. Based on initial measurements collected to date, the methodology seems to hold significant promise for systematic large-scale measurement of the port blocking dimension of network neutrality.

2 Measuring Port Blocking

In designing a methodology for measuring the extent and nature of Internet port blocking, we first examine what such measurements should include:

- ***Generality:*** Test any arbitrary port number in the 16-bit range allocated to the TCP and UDP protocols, i.e. $(0, 2^{16}-1]$. Several ports bear special notice such as HTTP (port 80), SMTP (port 25), P2P file sharing, virtual private networks and games. These applications are some of the most contentious.
- ***Range:*** Test a wide range of networks across the entire Internet.
- ***Quantity:*** Test a large number of hosts across the entire Internet.
- ***Minimal Participation:*** Assume no active, coordinated or cooperative participation from remote hosts.

Active client participation, such as that used in the Spoofer Project [14] or the IPPM metrics [15], would enable us to comprehensively test a wide range of ports and even quality of service properties. However, per our last requirement above, we cannot assume active participation since it is at odds with testing a large quantity and range of networks. The seemingly difficult problem is to induce hosts, randomly distributed on the Internet, to send packets to a destination and port of our choice. Our approach uses clients participating in the Gnutella P2P file sharing overlay in a novel manner to accomplish the aforementioned goals.

2.1 Functional Overview

Unstructured overlays such as Gnutella allow nodes to interconnect with minimal constraints. To scale, they rely on a two-level hierarchy of leaves and “SuperPeers” [16]. The Gnutella overlay is formed organically with SuperPeers actively managing the number of connections they maintain both to other SuperPeers and to leaves [17]. A peer can turn away connection requests via a `busy` message. The busy response also includes other peers to try so that new nodes can bootstrap. Nodes successively attempt connections to peers until they find a stable set of links. Our system crucially relies on the fact that this busy “referral” includes both the IP address and port number of other peers to contact.

Figure 1 depicts the high-level architecture of our system (the complete system is described in §2.4). We manage two separate machines, a Rogue SuperPeer (RSP) and a measurement host. The RSP joins the Gnutella SuperPeer mesh and routes queries and responses according to the normal Gnutella protocol. Once connected, the presence of our RSP is advertised by other SuperPeers. When new leaf node clients attempt to connect to our RSP (step 1), it sends a busy message and deterministically advises the client to try connecting to our measurement host (step 2). In this fashion, we have effectively tricked the client into sending a packet to the IP and port number of our choosing (step 3).

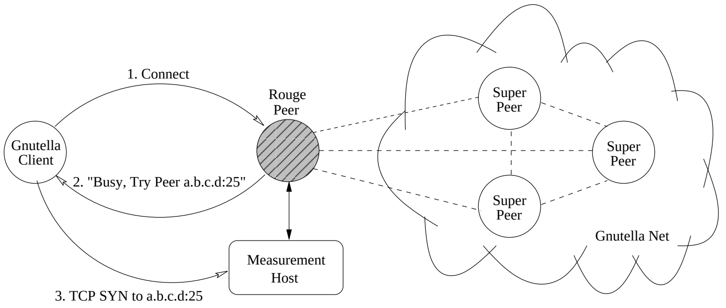

Fig. 1. Methodology: The Rogue SuperPeer joins the Gnutella network. (1) Client attempts to connect; (2) RSP rejects, referring the client to a measurement host under our control; (3) By correlating connections, we map Internet port blocking.

Any distributed system which allows arbitrary redirection messages is suitable for our task, for instance Bittorrent, HTTP links, etc. However, we use Gnutella as it easily facilitates global advertisement of the presence of our RSP. Additionally, the size and scope of the Gnutella network, approximately 2 million users [18], allows our method to elicit a high number of connections and thus redirect them for measurement purposes. Note that the initial SYN sent by clients does not contain data and hence is not affected by middle boxes that might drop packets based upon deep packet inspection.

2.2 A Map of Internet Port Blocking

Consider a client c residing on network W, i.e. c's IP address belongs to W. If c follows the busy referral from our RSP and connects on port p, we conclude that W does not block p (and thus is neutral to applications that use port p). However, the client c may not follow the referral or attempt the connection. Our measurement host must disambiguate whether the absence of a connection from c implies that W blocks p, or c never attempted to connect.

By intelligently selecting p in the busy redirect message of step 2 in Figure 1 on the basis of the client's network W, we overcome this ambiguity. We use a BGP routing table [19] to associate the client's IP address with BGP prefix b in the set of advertised prefixes B (i.e. a map of $client's\ IP \mapsto b \in B$). Once our measurement host receives a successful attempt from a (p, b) pair, the RSP does not attempt to test p for any future clients connecting from b. Next, we detail the system's port selection process in the face of uncertainty.

2.3 Probabilistic Inference

If a particular client does not heed the busy referral message, probabilistically the system will encounter another client that does. In the limit, our measurements can construct an accurate picture of the extent of discriminatory network port blocking. To formalize the conditions under which there is a high probability that a given network is blocking traffic, we first give the definitions in Table 1.

Table 1. Formal Definitions for Blocking Inference

$f(IP) = b \in B$	A function $f()$ on an IP address IP that gives an identifier b in the set of all BGP prefixes B
$P = \{p \mid p \in \mathbb{N}, 0 < p < 2^{16}\}$	The set P of all possible TCP or UDP ports
$n(p, b) \in \{0, 1\}$	A binary indicator variable. Is "1" (respectively "0") if IP traffic with destination port p is allowed (respectively blocked) on the path from originating BGP prefix b to the measurement host.
$H(p, b, i) \in \{0, 1\}$	A binary indicator variable. Is "1" (respectively "0") if the measurement host observed (respectively did not observe) a packet destined to port p from any of i clients with IP addresses in prefix b.

Given that the measurement host observes a packet, $H(p, b, i) = 1$, we trivially conclude that traffic to port p is allowed: $P(n(p, b) = 1 \mid H(p, b, i) = 1) = 1$. Not as trivial is the probability that traffic to port p is blocked, given that no packet was observed, $P(n(p, b) = 0 \mid H(p, b, i) = 0)$. By Bayes' Theorem:

$$P(n(p,b) = 0 \mid H(p,b,i) = 0) = \frac{P(H(p,b,i) = 0 \mid n(p,b) = 0) P(n(p,b) = 0)}{\sum\limits_{j=\{0,1\}} \left(P\left(H(p,b,j) = 0 \mid n(p,b) = j\right) P\left(n(p,b) = j\right) \right)} \tag{1}$$

Since no packet will be observed if indeed traffic to the port is blocked, we have that $P(H(p,b,i)=0|n(p,b)=0)=1$. Empirically (see §3.1), we find that the probability that a Gnutella client does not use the p reference the RSP passes along is approximately 0.8, which we conservatively estimate as 0.9. Assuming independence across the i clients, the probability no packet is observed if indeed traffic to the port is allowed is 0.9^i, i.e. $P(H(p,b,i)=0|n(p,b)=1)=0.9^i$. Prior to our observations, we assume no information as to whether the port is blocked, and equal prior probabilities: $P(n(p,b)=1)=P(n(p,b)=0)=\frac{1}{2}$. [3] Substituting into (1), we obtain the probability that traffic to port p is blocked given that no packet was observed:

$$P(n(p,b)=0|H(p,b,i)=0)=\frac{1}{1+0.9^i} \tag{2}$$

$$\approx 1-0.9^i \quad \text{for } 0.9^i \quad \text{small} \tag{3}$$

We wish to set i such that if our measurement host does not receive a packet pair (p,b) after i redirect messages to hosts residing in prefix b, then the probability is suitably large that port p is indeed blocked. Choosing i such that

$$P(n(p,b)=0|H(p,b,i)=0)=0.995 \Rightarrow i=\log_{0.9}(0.005)\simeq 50 \tag{4}$$

Thus, we must send ≈ 50 referrals to b for p to conclude, with probability 0.995, that port p is blocked on the path from b to the measurement host $\square$.

2.4 Full Methodology Design

Based on the prior discussion, we present the full system methodology in Figure 2, an augmented version of Figure 1. All state is maintained in a database. The RSP and measurement hosts asynchronously read and write to the database to update the current state. The database also facilitates later off-line analysis.

Both the RSP and measurement host interface with a BGP database, built from a routeviews [19] table, that provides a mapping between an IP address and the longest matching prefix to which that address belongs. Each unique prefix is assigned a unique identifier in the database.

The "NextPort updater" is a process which runs every five minutes. The updater implements the logic in (§2.3) to intelligently update the database's notion of which port the RSP gives out in the next referral for a particular prefix in order to glean the most information. The updater orders the choice of p according to those most likely to be blocked, e.g. VPNs, file sharing, etc. Appendix A gives a complete description of the ports we explicitly test.

Lastly, the measurement host implements a front-end multiplexer which transparently redirects traffic from any incoming port to the port on which the SuperPeer is listening. In this fashion, clients connect to an actual SuperPeer irrespective of the port in the RSP's referral messages.

[3] One could choose to assume prior information. Suppose $P(n(p,b)=1)$ is set equal to δ, and $P(n(p,b)=0)=1-\delta$. Then equation (2) becomes $P(n(p,b)=0|H(p,b,i)=0)=\frac{1}{1+0.9^i*\delta/(1-\delta)}\approx 1-0.9^i*\delta/(1-\delta)$. And (4) becomes $i=\log_{0.9}(0.005*(1-\delta)/\delta)$.

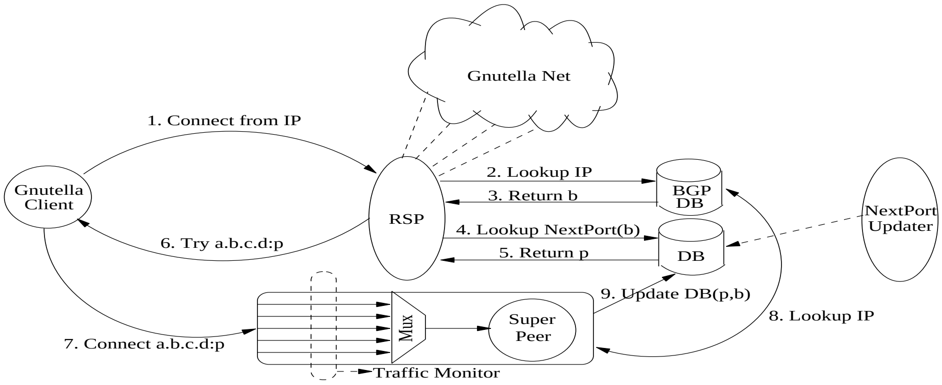

Fig. 2. Full Port Blocking Measurement Methodology

3 Results

We collected data using our infrastructure over two months in order to validate the methodology, refine testing and gather results. While our experiment is ongoing, these initial results are very promising. The anonymized data from this study is publicly available from: `http://ana.csail.mit.edu/rsp`.

3.1 Efficacy of Methodology

The efficacy of our methodology depends firstly on issuing referrals to many Gnutella clients distributed across many networks. As seen in Table 2, over two months our RSP sent approximately 150k referrals to 72k unique Gnutella clients. These clients represent some 31k different global BGP prefixes, a non-trivial fraction of the Internet.

Table 2. Collection Statistics, Period: 02-Oct-2006 to 02-Dec-2006

	Count	**Rate**
Unique BGP Prefixes	31,219	0.7/Minute
SYN Packets Received	973,865	21.0/Minute
Unique IP Sources	328,437	7.4/Minute
Unique Gnutella Peers	72,544	1.6/Minute
Referrals Sent	147,581	3.3/Minute

Second, Gnutella clients which receive the specially crafted referrals from the RSP must follow the referral, i.e. attempt a connection on the basis of the referral, a non-negligible fraction of the time. Since a Gnutella client attempts only to find a stable set of connections into the network, it is unsurprising that not all potential SuperPeers are explored. We observe variability in the fraction of referrals that clients follow. Figure 3 depicts the fraction of followed referrals versus the cumulative fraction of clients.

Fully 78% of the clients to which which our RSP sends referrals never result in a connection attempt. However, approximately 5% of the clients take half of all referrals and another 10% follow all referrals. Manual inspection of the clients which follow all referrals suggests that these clients are actually Gnutella network spiders. Because the spiders attempt to search and index the network, they follow all possible links in the overlay.

Thus, our referral methodology operates exactly as anticipated and allows us to build a map of port blocking given a sufficiently large collection window. The measured 78% non-connection attempt rate corresponds directly to the conditional probability of a Gnutella client not following an RSP reference from equation (2): $p(H(p,b,1)=0|n(p,b)=1)=0.78$.

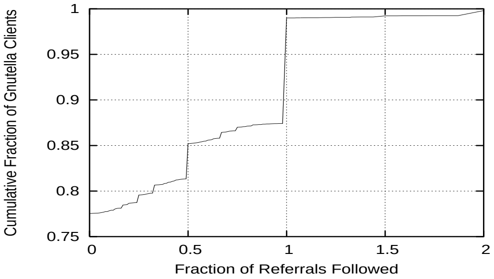

Fig. 3. RSP Referral Efficacy: proportion of referrals followed vs. cumulative fraction of clients

Note that we record incoming connection attempts from clients the RSP has not interacted with and ports for which the RSP has not handed out referrals. On inspection, these connections appear to be from malicious hosts and malware performing random scanning. As this connection information is in some sense additional data for free, we include it in our analysis.

3.2 Observed Port Blocking

Given the approximately 1M incoming SYN packets observed by our measurement SuperPeer and induced by our RSP, we can begin to make per-BGP prefix inferences of port blocking. In this initial analysis we restrict our definition of blocking to blocking at any point along the path from the client to our servers; in future work we plan to use additional techniques to understand individual autonomous system behavior. Of the 31,000 prefixes, we find 256 prefixes which exhibit blocking for one or more ports as determined by Equation (4). Let α_p be the ratio of number of inferred prefixes blocking p to the total number of prefixes for which our measurement host has classified. Let $\#\left\{A\right\}$ denote the number of elements in set A. Then formally:

$$\alpha_p = \frac{\#\left\{b \text{ such that } n(p,b)=0\right\}}{\#\left\{b \text{ such that } n(p,b)=0\right\} + \#\left\{b \text{ such that } n(p,b)=1\right\}} \tag{5}$$

Figure 4 shows the relative incidence of port blocking by giving p versus α_p for $\alpha_p > 0 \; \forall p$. We use $p = 6969$ as a control group as this port is unassociated with any applications or vulnerabilities and is typically unblocked.

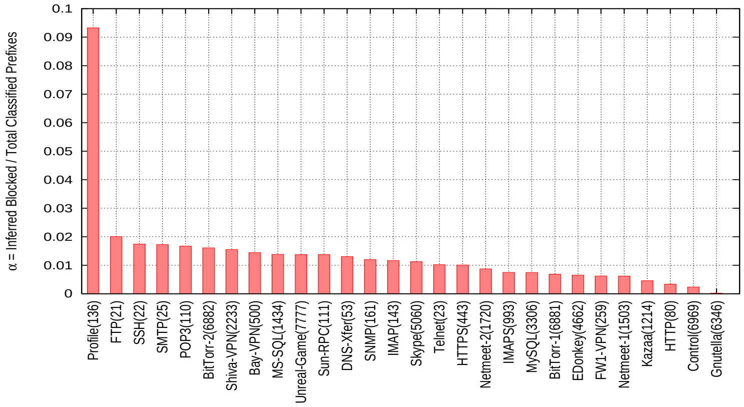

Fig. 4. Per-port α: blocked versus total inferences over observed BGP prefixes.

We highlight only the most interesting results due to space constraints. The most frequently blocked port is 136, the collateral damage port which we discuss in (§4). The three lowest α_p in descending order are HTTP, Control and Gnutella which matches our intuitive notion of commonly open ports and serves as a methodology litmus test. Email ports (25, 110, 161, 143) are more than twice as likely to be blocked as our control port. Port 1434 was widely blocked due to a worm [8] and shows prominently three years after the initial outbreak. FTP, SSH, Bittorrent and VPNs round out the remaining top blocked ports.

Manual inspection of the BGP prefixes to which blocking is attributed reveals several ISPs and universities blocking outgoing P2P ports (1214, 4662, 6346, 6881). We find Canadian and US ISPs as well as a Polish VoIP prefix blocking Skype. Email, especially outbound port 25 (SMTP) is blocked by several large cable and DSL Internet providers as well as large hosting services.

3.3 Measurement Bias

We obtain unbiased measurements from a non-trivial portion of the Internet ($\approx 31k$ BGP prefixes, cf. Table 2). However, our methodology cannot obtain measurements from networks which use content filtering to disallow Gnutella (the RSP listens on the non-default port 30494 to avoid port filtering). Thus, any extrapolation of our results to a characterization of the larger Internet is potentially biased. Networks that we have yet to measure could block more or fewer ports or different ports than those seen in existing results.

Since we wish to measure service provider discriminatory blocking, we analyze our data on the basis of BGP prefix aggregates. We reason, but do not prove, that while an individual customer of an ISP, say a corporation or university, may block Gnutella, it is unlikely that of the ISP's customers ISP block Gnutella. A single reachable node facilitates inference for that ISP. The breadth and scope of the BGP prefixes for which we have data suggest that the qualitative characteristics of blocking in our sample is likely representative of a significant fraction of the Internet. Our ongoing work seeks to further substantiate this characterization.

4 Discussion, Future Research and Conclusion

Understanding common operational practices on the Internet is particularly important as these practices are under close scrutiny in the network neutrality debates. While our data cannot answer which practices should be acceptable, the distribution of practices across different types of providers (c.f. academic and commercial) may provide insights into provider intentions.

For instance, the MIT network drops traffic destined for TCP ports 135 and 137-139, ports associated with Microsoft file sharing. With the same intent, but slightly different effect, Comcast residential broadband blocks the entire 135-139 port range [11]. Interestingly, Comcast's policy results in the *collateral blocking* of port 136, assigned to the innocuous Profile naming service [2]. The fact that MIT and other non-profit organizations block the Windows file sharing ports potentially provides justifiable evidence that Comcast's intentions in blocking the same ports are not abuses of market power. Indeed, here the motivation for blocking is based upon operators' concerns for end-user security and privacy.

Given the infancy of our scheme and the broader evolution of network neutrality, we expect this work to pose as many questions as it answers. By continuing to collect data, we can form a more complete picture of blocking, not only in terms of ports but also networks, autonomous systems and addresses.

Beyond the methodology in this paper there are several interesting and hard data analysis problems we plan to investigate. First, port-specific traceroutes to clients in our study could reveal ingress properties, filtering asymmetry and yield useful path information. By finding partially coincident AS paths with opposite blocking policies, we can infer where in the network blocking occurs. Finally, our data can shed light on the evolution of blocking over time.

Our results represent some of the first measurements in the space of neutrality and discrimination. We hope our findings will better inform the network neutrality debate by providing data on which to make informed decisions.

Acknowledgments

We thank David Clark, Neil Gershenfeld, Sachin Katti, Enoch Peserico, Karen Sollins and our reviewers for support, discussions and invaluable feedback.

References

1. kc Claffy: Top problems of the Internet and what can be done to help. In: AusCERT. (2005)

2. IANA: Well-known port numbers (2006) http://www.iana.org/assignments/port-numbers.
3. Clark, D.: Name, addresses, ports, and routes. RFC 814 (1982)
4. Wu, T.: Network neutrality, broadband discrimination. Telecommunications and High Technology Law **2** (2005)
5. Schewick, B.V.: Towards an economic framework for network neutrality regulation. In: Proceedings of the Telecommunications Policy Research Conference. (2005)
6. FCC: In the Matter of Madison River Communications Companies (2005) File No. EB-05-IH-0110.
7. Cerf, V.: U.S. Senate Committee on Commerce, Science, and Transportation Hearing on Network Neutrality (2006)
8. CERT: Advisory CA-2003-04 MS-SQL Worm (2003) http://www.cert.org/advisories/CA-2003-04.html.
9. Ballani, H., Chawathe, Y., Ratnasamy, S., Roscoe, T., Shenker, S.: Off by default! In: Proc. 4th ACM Workshop on Hot Topics in Networks (Hotnets-IV). (2005)
10. Masiello, E.: Service identification in TCP/IP: Well-Known versus random port numbers. Master's thesis, MIT (2005)
11. Comcast: Terms of service (2006) http://www.comcast.net/terms/use.jsp.
12. America On-Line: AOL Port 25 FAQ (2006) http://postmaster.aol.com/faq/port25faq.html.
13. Schmidt, J.E.: Dynamic port 25 blocking to control spam zombies. In: Third Conference on Email and Anti-Spam. (2006)
14. Beverly, R., Bauer, S.: The spoofer project: Inferring the extent of source address filtering on the Internet. In: Proceedings of USENIX SRUTI Workshop. (2005)
15. Mahdavi, J., Paxson, V.: IPPM Metrics for Measuring Connectivity. RFC 2678 (Proposed Standard) (1999)
16. Yang, B., Garcia-Molina, H.: Designing a super-peer network. IEEE Conference on Data Engineering (2003)
17. Ripeanu, M., Foster, I., Iamnitchi, A.: Mapping the gnutella network. IEEE Internet Computing Journal **6**(1) (2002)
18. Slyck: Slyck's P2P Network Stats (2006) http://www.slyck.com/stats.php.
19. Meyer, D.: University of Oregon RouteViews (2006) http://www.routeviews.org.

Appendix A: Ports of Interest

Port	**Description**
4662, 6346, 1214	Popular Peer-to-Peer
6881-6889	BitTorrent
25, 110, 143, 993	Email
27015, 27660, 7777, 7778, 28910	Popular Games
5060	Skype
2233, 500, 1494, 259, 5631	Popular VPN
80, 8080, 443	HTTP
194, 1503, 1720, 5190	Chat
20-23	Popular User Applications
53, 111, 119, 161, 179, 3306	Popular Server Applications
136	Collateral Damage
1434, 4444	Worms