A Technical Approach to Net Neutrality

Xiaowei Yang Gene Tsudik Xin Liu
Department of Computer Science
University of California, Irvine
[xwy|gts|xinl]@ics.uci.edu

ABSTRACT

A recent statement by AT&T CEO Ed Whitacre sparked considerable fear in the public that the Internet may not be open any more: the ISPs dictate which sites/applications flourish and which flounder. The statement triggered the heated debate on net neutrality and ignited the battle to enact net neutrality legislation. However, by the date of writing, all attempts to pass net neutrality laws have failed.

This paper states our proposition on net neutrality: ISPs should not be able to discriminate against packets based on contents, application types, or packet sources or destinations that are not their own customers; but they are eligible to offer differentiated services to their customers. We present a technical design that aims to achieve this definition of net neutrality. Our design prevents an ISP from deterministically harming an application, a competing service, or singling out an individual innovator for extortion.

1 INTRODUCTION

In November 2005, AT&T CEO (formerly SBC CEO) Ed Whitacre was quoted in **BusinessWeek** as follows [3]:

> "Now what they [Internet upstarts like Google, MSN, Vonage, and others] would like to do is use my pipes free, but I ain't going to let them do that because we have spent this capital and we have to have a return on it," says Whitacre. "So there's going to have to be some mechanism for these people who use these pipes to pay for the portion they're using. Why should they be allowed to use my pipes?"

This statement triggered strong reactions from consumers and Internet companies. A number of net neutrality draft bills [20] were introduced to Congress since March 2006, in attempt to enact net neutrality. Unfortunately, none has succeeded so far. Grassroot coalitions such as "Save The Internet" [18] and "It's Our Net" [13] were created. Hundreds of organizations and companies joined the coalitions, and more than a million signatures were collected to support net neutrality. Competitors such as Google, Yahoo, and Microsoft, grassroot groups from both the left and right (e.g. Moveon.org and Christian Coalition of America), stand on the same side to support net neutrality.

Proponents of net neutrality argue that the openness of the Internet, i.e., the ability to access any content, run any application, or attach any device to the Internet, leads to the very success of the Internet. This openness and freedom drives innovation, promotes free speech, and encourages democratic participation [4, 18]. If ISPs were able to discriminate packets based on content or ownership, innovative ideas would not necessarily be rewarded. Instead, well-funded ideas or ISPs' own services would be more likely to succeed.

Opponents of net neutrality (e.g. telcos) [11] argue that tiered service, or data prioritization, is a legitimate business model. The increasingly popular video and audio applications on the Internet put a high bandwidth demand on their networks. Tiered service can provide desired quality of service to different applications and recoup the capital investment used to upgrade their networks. Some opponents dislike the idea of regulation in principle, arguing that market forces are sufficient to regulate what broadband ISPs would do. If one ISP blocks contents or applications that consumers desire, consumers would switch to a different ISP.

Both sides have their points, which makes the debate over net neutrality a murky matter. On the one hand, data prioritization can improve quality of service and is a legitimate business model. A strict neutrality law that does not allow fee-based data prioritization, e.g. the Markey Amendment "Network Neutrality Act of 2006" [16], will prohibit ISPs from selling differentiated services, and prevent customers from purchasing improved quality of service based on their willingness to pay. On the other hand, ISPs may abuse data prioritization to discriminate packets to their favor. For instance, telcos may give a high priority service to their own VoIP service and intentionally slow down a competitor's service.

It is difficult to conclude whether any net neutrality law should or will be passed in the near future, partly because of the difficulty of line-drawing and the suspicion that there is sufficient market competition. A few pundits have advocated the wait-and-see approach to avoid potentially harmful or toothless regulations [10]. However,

the status quo without regulation has its own risk: the openness of the Internet may gradually erode and innovations may be stifled. It's true that there is some level of competition (i.e. with cable competing with DSL) in the broadband market, but practically speaking, users tend to stay with their existing service providers despite of mild service dissatisfaction for a number of reasons, e.g., costs of switching, bundling deals, or cancellation hassles [8]. A broadband ISP may take advantage of this inertia to gradually migrate to a closed Internet. As an example, a broadband ISP may intentionally degrade the VoIP service offered by Vonage, but give a high priority service to its own VoIP offerings. A user that experiences a low-quality VoIP service from Vonage but could use a substitute service offered by his provider might not bother to switch. Using this tactic, gradually, a broadband service provider may drive Vonage out of business and make its own VoIP service thrive. Then it can start to degrade another competitor's service and so on.

Alternatively, individual innovators that can afford to pay (say Google) might choose to pay every access provider to avoid appearing slow to users. However, it's unclear whether there is sufficient market force to regulate the price Google needs to pay, because once a user has chosen his access provider, that access provider becomes a monopoly to Google. There is no way for Google to bypass the access provider to reach the user.

In this paper, we propose a definition of net neutrality as follows: ISPs should not be able to discriminate against packets based on contents, application types, or packet sources or destinations that are not their own customers. We call this type of discrimination *non-neutral* discrimination. But ISPs are eligible to offer differentiated services to their customers. Our hypothesis is that the present market structure may not have sufficient competition to prevent an access ISP from degrading the service of a particular application or a site, but might be sufficient to keep them from intentionally ill-treating their own customers. For instance, if AT&T slows down a customer's VoIP traffic from Vonage, the customer may not care to switch to a different provider. But if AT&T slows down all traffic the customer sends or receives, or charges a higher price for the same quality of service than what the customer can get at a different provider, the customer may decide to switch. We rely on the existing market competition to regulate how ISPs treat their own customers or peers' traffic. If there turns out to be sufficient market competition, then ISPs would not overcharge their customers for the desired quality of service, and would strive to meet the service requirements of their customers. In this situation, a customer's traffic will not be intentionally harmed regardless of how ISPs prioritize their own services or other customers' high-quality services. If there is no sufficient competition, then consumers as a whole would suffer. Hopefully they as a whole are a stronger voice than individual innovators and this situation can make it clear what net neutrality regulation is needed.

This paper presents a technical design that aims to realize our definition of net neutrality, a solution that prevents non-neutral discrimination yet allows tiered services. A key design challenge is to prevent an ISP from discriminating against a source or a destination that is not its customer (or peer). Standard end-to-end encryption techniques (e.g., IPsec) can be used to prevent content-based or application-based discrimination, but the source or destination address of a packet may still reveal the identity of a non-customer. To address this challenge, we design an efficient and stateless neutralizer service that allows an ISP to "blur" packets to or from its customers. Thus, other ISPs cannot target an individual customer of the ISP and double charge the customer.

The rest of the paper is organized as follows. We describe our design assumptions and design details in § 2 and 3. § 4 provides a preliminary performance analysis. § 5 discusses related work, and we conclude in § 6.

2 ASSUMPTIONS

We assume that there are ISPs that support net neutrality, perhaps due to the competitive pressure in the backbone market or sharing the belief that an open Internet is the key to foster innovation. This assumption is not unrealistic. For instance, Cogent has made a public statement in support of net neutrality [6]. We assume such ISPs are willing to offer services to their customers to protect them from being double-charged by broadband access ISPs. We also assume that host software can be modified to support our design.

We refer to an ISP that intends to discriminate packets in a non-neutral manner as a discriminatory ISP. We assume that a discriminatory ISP, despite its eagerness to make money, will not launch active attacks at its customers or peers. Those attacks include modifying packet contents, man-in-the-middle attacks, and denial of service (DoS) attacks. The ISP may eavesdrop on all traffic, perform traffic analysis, delay or drop packets within its network, but it cannot eavesdrop on traffic outside its network. We believe this assumption is realistic because malicious behavior, once detected, may severely damage an ISP's reputation.

For simplicity, our current design does not consider traffic analysis attacks that infer application types or packet ownships using packet size and timing information. If in the practical deployment ISPs can use traffic analysis to successfully discriminate, we will consider incorporating mechanisms such as adaptive traffic masking [19] to defeat such attacks.

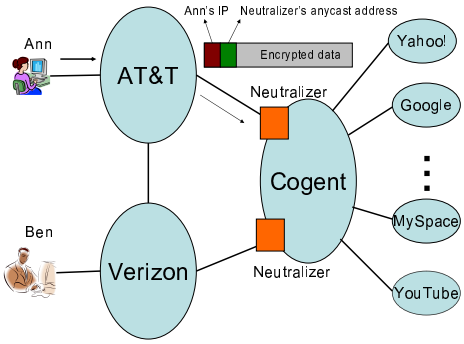


Figure 1: **The neutralizer boxes mix packets sent to and from Cogent's customers. Other ISPs such as AT&T and Verizon cannot differentiate its customers' packets.**

We assume each packet carries a standard IP header, and additional fields needed by our design are carried in a shim layer between IP and an upper layer. The protocol field in an IP header is set to a fixed and known value. The source and destination address fields refer to the fields in a standard IP header.

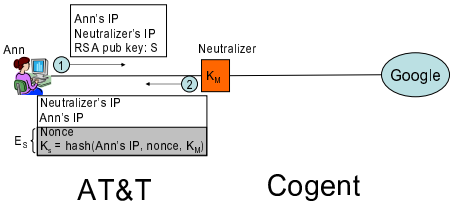
3 DESIGN

The goal of our design is to prevent an ISP from discriminating packets in a non-neutral manner. We use two techniques to accomplish this goal. One is the existing end-to-end encryption techniques. The other is a neutralizer service as described below.

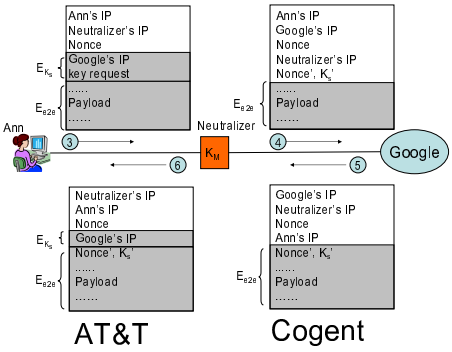
Figure 1 shows a high-level view of our design. A non-discriminatory ISP, Cogent in this example, places neutralizers at the boundary of its domain. These neutralizers can either be inline boxes or part of a border router's functionality. A neutralizer helps an ISP's customers to hide their addresses from other ISPs. In Figure 1, all packets sent to (or from) the customers of Cogent, e.g. Google, Yahoo!, MySpace, or YouTube , will have the neutralizer's IP address as their destination (or source) addresses. We use an anycast address to represent the neutralizer service of an ISP. All customers of an ISP use the same neutralizer address, regardless of where they are located. Further, end-to-end encryption is used to protect packet payload. With this design, a discriminatory ISP, e.g. AT&T in this example, can only discriminate against Cogent's packets as a whole, and cannot deterministically harm an individual customer of Cogent.

3.1 Bootstrapping

To bootstrap a connection, a source inside a discriminatory ISP needs to obtain a destination's IP address, the destination's neutralizers' addresses, and the destination's public key for end-to-end encryption. This bootstrapping information can be stored at a destination's DNS records, and a source may obtain this information via DNS queries.



(a) A user Ann in a discriminatory ISP sets up a symmetric key with a neutralizer.



(b) Ann sends and receives packets with encrypted addresses via the neutralizer.

Figure 2: **The shaded areas represent encrypted data, and non-shaded areas represent plain text. The letter E denotes the encryption function, and the subscript denotes the encryption key. K_M is the master key of a neutralizer. The first two fields in a packet diagram are the source and destination address fields. The rest of the fields are either in a shim header or payload. As can be seen, the address of Cogent's customer, Google's IP, and packet contents are not visible inside AT&T.**

A discriminatory ISP may eavesdrop on its customer's DNS queries and discriminate DNS queries based on the query destination. For instance, AT&T may delay queries for `www.google.com` if Google does not pay AT&T to use its pipes. To address this problem, a source needs to encrypt its DNS queries and send the queries to DNS resolvers that are not controlled by the discriminatory ISP. We assume a third party, such as a non-discriminatory ISP, a large overlay network, e.g. the PlanetLab [17], or Google itself, can provide DNS resolvers to clients inside discriminatory ISPs. Those clients will be configured with the IP addresses, the public keys, and the neutralizers' addresses (if there is any) of those DNS resolvers.

End-to-end encryption can use standard techniques such as IPsec. In this paper, we use end-to-end encryption as a black box, and do not discuss the details.

3.2 Efficient and Stateless Neutralizer

After a source obtains the bootstrapping information, it can send packets to a destination via the neutralizer. The source and the neutralizer must keep the destination address as a secret to avoid discrimination. This is sim-

ilar to anonymous routing [7]. However, a neutralizer in our context must be highly efficient and scalable, because it may receive all traffic sent to or from a large ISP at a peering point. Therefore we cannot use existing anonymous routing techniques [7, 14], that require per-flow state and expensive public key operations.

Our design uses a combination of light-weight public key encryption and symmetric key encryption to improve efficiency, and a technique that allows a neutralizer to compute a symmetric key from a packet header to avoid state. We cannot completely avoid public key encryption (or any technique that's equivalent) because protecting the secrecy of the destination address requires the establishment of a shared secret between a source and a neutralizer. It is a well-known result in theoretical cryptography that the establishment of a shared secret in the presence of an eavesdropper is impossible using only the techniques of symmetric-key cryptography [12].

Figure 2 illustrates how a source communicates with a destination via a neutralizer. The source first generates a short (e.g. 512-bit) one-time RSA public key S, and sends the key to the neutralizer. The neutralizer keeps a long term master key K_M. When it receives a public key from a source, it chooses a random nonce, and computes a keyed hash (K_s) from the nonce, its master key, and the source address: $K_s = hash(K_M, nonce, srcIP)$. K_s will be used as the symmetric key between the source and the neutralizer. The neutralizer encrypts the nonce and the symmetric key using the source's public key, and returns the encryption to the source. The source decrypts the packet, and retrieves the nonce and the symmetric key. It can then encrypt a destination address with the symmetric key and send the packet to the neutralizer. The source sends the nonce in clear text for the neutralizer to recover the shared key K_s. When the neutralizer receives the packet, it recomputes the symmetric key as $K_s = hash(K_M, nonce, srcIP)$ and decrypts the destination address.

This key setup process has the advantage that the neutralizer is stateless and performs the more efficient RSA encryption operation, while the source executes the slower RSA decryption operation. (An RSA encryption may involve as few as two multiplications, if the exponent in the public key is 3.) It also maintains the stateless and fault-tolerant feature of IP routing. As long as the neutralizers of a domain share the master key K_M, any neutralizer can decrypt the destination address and forward the packet.

A short RSA key represents a tradeoff between efficiency and security. A 512-bit RSA key is only as secure as a 56-bit symmetric key. To improve security, we let a source use a short RSA key only once, and expire the symmetric K_s key (encrypted with the RSA key) quickly. As shown in Figure 2, when a source sends the first packet to a destination using the the symmetric key K_s, it also sends a key request. When the neutralizer forwards the packet with a key request, it stamps a new nonce, and a new key K_s' into the packet. When a destination receives this packet, it uses strong end-to-end encryption, e.g. 1024-bit RSA encryption, to encrypt this new (nonce, key) pair together with its packet payload, and sends them to the source. A source can encrypt the destination address in its subsequent packets with this new key until it obtains a newer one. As long as a discriminatory ISP does not factor the short RSA key before K_s' is returned to the source (which takes two round trip times), the discriminatory ISP cannot decrypt the destination address, and cannot discriminate packets based on the destination address.

Another advantage of this design is that if a neutralizer cannot support RSA encryption at line speed, it can offload the encryption operation to any customer in its domain that is willing to help. The neutralizer inserts the nonce and the symmetric key K_s in the source's key request packet and forwards the packet to the customer to encrypt using the public key in the request packet. A customer (e.g. Google) would have incentive to help because the source may intend to communicate with it.

We have also considered a more obvious design choice that lets a source encrypt a destination address using a neutralizer's public key when it sends the first packet to a destination. This alternative has the advantages of saving one round trip time for key setup and preventing the man-in-the-middle attack, as the public key of a neutralizer can be certified. However, it places a higher burden on a neutralizer: the neutralizer must perform a public key decryption operation that cannot be offloaded to any customer.

We chose the first key setup approach because a higher processing overhead makes a neutralizer more vulnerable to DoS attacks and temporary traffic overloading, while an extra round trip time for key set up can be amortized over multiple packets. A source can use the same symmetric key to send any packet destined to any customer in the neutralizer's domain until the neutralizer's master key expires. We also assume that a discriminatory ISP will not risk its reputation to launch man-in-the-middle attacks. Thus, the second advantage of the alternative approach is not essential.

A return packet from a customer in a neutralizer's domain must have its source address anonymized, yet the recipient must be able to retrieve the necessary key to decrypt the payload of the packet. In our design, when a destination returns a packet to the source, the return packet will include the destination's address in the source IP address field, the neutralizer's address in the destination IP field, the initiator's address and the nonce included in the forward packet in a shim header, as shown

in Figure 2. When a neutralizer receives a return packet from a customer (we assume the neutralizer can tell this from the source address field), it encrypts the source address (Google's IP in Figure 2) using the symmetric key indicated by the nonce, replaces the source address with its own anycast address, and sets the destination address to be the initiator's address. When the initiator receives this packet, it can use the nonce and the neutralizer's address to locate the key K_s it shares with the neutralizer, and decrypt the original source address (Google's IP). It can then use this address to identify the communication session, and decrypt the packet payload.

3.3 Reverse-direction Communication

Communication initiated by a customer (e.g. Google in Figure 2) inside the neutralizer's domain to an outside destination can happen in a similar fashion but with less overhead. In the initial key set up phase, the customer may simply request a nonce and a symmetric key from a neutralizer without encryption. After the customer obtains a shared key with a neutralizer, the rest of the process is similar to what we have described above. The customer encrypts the shared key with its intended destination's public key and sends the encrypted key. When the destination, e.g. Ann's computer in Figure 2, receives a packet, if it cannot locate a symmetric key corresponding to the nonce and the neutralizer's address, it will attempt to use its public key to decrypt the packet. If successful, the destination obtains the shared key and can use it to encrypt the initiator's address and send packets via the neutralizer.

3.4 Quality of Service

In our design, a discriminatory ISP can still offer differentiated services [1] to its customers, as a neutralizer will not modify the Differentiated Services Code Point (DSCP) in a standard IP header. The discriminatory ISP may provide differentiated services according to the DSCPs in packet headers.

However, a discriminatory ISP can no longer keep per flow state (a flow refers to a source and a destination pair) to provide guaranteed services [2] to anonymized traffic. There are at least two remedies to this problem. The first is for a neutralizer to assign a dynamic address to a customer that initiates a QoS session, e.g. an RSVP session. This dynamic address allows the discriminatory ISP to identify a flow, but does not allow it to map the flow to a specific customer. Another possibility is that the customer may request not to be anonymized if it has purchased guaranteed service from the discriminating ISP. The neutralizer service is optional, and a customer does not have to use it.

3.5 Multi-homed Sites

A site may connect to multiple ISPs that offer the neutralizer service. In this situation, the site may publish multiple neutralizers' addresses in its DNS records, each address corresponding to one ISP. The ISP-level path of the site's incoming and outgoing traffic is then controlled by how other sources pick the neutralizers, and is no longer controlled by the site's BGP routers. The path chosen by other sources may interfere with a site's traffic engineering effort. For instance, if one provider is congested, the site may want all traffic comes from a different provider, but other sources may choose the congested provider. A similar situation occurs in IPv6 [9], in which a multi-homed site obtains multiple addresses, one from each provider. The path of the incoming traffic to a site is determined by which address a source chooses to contact the site. We can borrow any technique that can balance traffic load in that context to balance traffic between different neutralizers. In general, two hosts may always use trial-and-error to find a path that's working for them.

3.6 What Can Still Go Wrong?

Denial of Service Attacks: A neutralizer box may be subject to DoS attacks. Although our design places the more efficient RSA encryption operation at a neutralizer, a public key operation is still expensive. If attackers flood key setup packets at line speed, a neutralizer may be overloaded. It is outside the scope of this paper to come up with a complete DoS defense mechanism. But a neutralizer can invoke DoS defense mechanisms such as pushback [15] to get rid of attack traffic.

One complication is that if attackers are inside a neutralizer's domain, the neutralizer's anonymization function may hide attack sources. This problem is similar to source address spoofing. Pushback can be used to defend this type of attack, as it is designed to function well with source address spoofing and does not rely on source addresses to filter attack traffic.

Packet discrimination: Our design does not completely "blur" all packets. A discriminatory ISP can still discriminate packets in at least three ways: 1) discriminate based on its customers' or neutralizers' addresses; 2) discriminate against encrypted traffic; 3) discriminate against key setup packets. (The third discrimination is possible because an ISP may infer a key setup packet from the nonce field, or from the packet length, or from inter-packet timing.) We are not concerned with these types of discriminations because none of them allows an ISP to deterministically harm an application, a competitor's service, or a non-customer/peer. If an ISP can only deterministically discriminate against its own customers (or its direct peers), then we rely on market forces to discipline what they would do (§ 1).

Good-intentioned discrimination: If packets are not encrypted or neutralized, an ISP may inspect packet contents and prevent unwanted traffic (e.g. viruses) from reaching an end user. Unfortunately, our design prevents such good-intentioned discrimination. Nonetheless, we believe it is a worthy tradeoff, because we cannot afford to lose the openness of the Internet.

4 PRELIMINARY EVALUATION

This section presents a preliminary evaluation on the performance of the neutralizer. We implemented the packet processing logic of a neutralizer using a modified Click Router [5] on Linux 2.6.16.13. The neutralizer in our testbed has an AMD Opteron 2.6GHz dual core CPU and an Intel pro/1000 GT quad-port server adapter.

A neutralizer does an RSA encryption when processing a key setup packet. In our experiments, the neutralizer can output response packets at 24.4kpps. If we assume a neutralizer's master key lasts for an hour, a source outside a neutralizer's domain at most needs to send a key request once an hour. Thus, a commodity PC can simultaneously serve 88 million sources for key setup.

A neutralizer does a hash computation and a symmetric key encryption or decryption when it receives a normal data packet. Our implementation uses 128-bit AES for both hashing and encryption/decryption. In our experiments, a client machine sends neutralized UDP packets with 64 bytes payload to the neutralizer. The total packet size is 112 bytes after adding headers, nonce, encrypted destination IP address, and alignment padding. The neutralizer is able to output packets with decrypted destination IP addresses at 422kpps. This throughput is limited by the neutralizer's hardware architecture, as the neutralizer can only forward vanilla IP packets of the same size at 600kpps. Our openssl speed tests show that the CPU of the neutralizer can perform the cryptographic operations at 2.35 million per second. We expect that special hardware that is optimized for packet forwarding can achieve a much higher packet throughput.

5 RELATED WORK

The most related work is anonymous routing [7, 14]. Anonymous routing aims to anonymize both the source and destination addresses of a packet, while our design only aims to anonymize the non-customer address from a discriminatory ISP. As a result, our design is considerably more efficient and scalable in terms of resource consumption. In our design, routers don't keep per-flow state, and perform much fewer public key encryption/decryption operations.

6 CONCLUSION

The debate over net neutrality has caught much public attention recently. Despite much effort, no essential net neutrality legislation is passed. This paper presents a technical approach to net neutrality. We describe a design that prevents ISPs from discriminating packets based on contents, application types, or non-customer/peer addresses. Our design prevents an ISP from deterministically discriminating against a competitor's service, a novel application, or singling out an individual innovator for extortion. At the core of our design is a neutralizer service that anonymizes traffic sent to or from a discriminatory ISP. The neutralizer is stateless and uses highly efficient cryptographic operations. We believe it can scale to support the traffic load of a large ISP. It's our future work to implement the neutralizer service and evaluate its performance.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. Internet RFC 2475, 1998.

[2] R. Braden, D. Clark, and S. Shenker. *Integrated Services in the Internet Architecture: an Overview*. IETF, June 1994. RFC 1633.

[3] BusinessWeek. Online Extra: At SBC, It's All About "Scale and Scope". `http://www.businessweek.com/magazine/content/05_45/b3958092.htm`.

[4] Center for Democracy & Technology. Preserving the Essential Internet. `http://www.cdt.org/speech/20060620neutrality.pdf`, June 2006.

[5] Click router. `http://www.read.cs.ucla.edu/click/`.

[6] Cogent Supports Net Neutrality. `http://www.cogentco.com/htdocs/neutrality.php`.

[7] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of USENIX Security Symposium*, 2004.

[8] B. Dipert. Net neutrality: Permissible restrictions. `http://www.edn.com/blog/400000040/post/1720005172.html`, Nov. 2006.

[9] R. Draves. *Default Address Selection for Internet Protocol version 6 (IPv6)*, 2003. RFC 3484.

[10] E. W. Felten. Nuts and Bolts of Network Neutrality. `http://itpolicy.princeton.edu/pub/neutrality.pdf`.

[11] Hands off the Internet. `http://www.handsoff.org/`.

[12] R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *Proc. of 21st Annual ACM Symposium on the Theory of Computing*, pages 44–61, 1989.

[13] It'sOurNet. `http://www.itsournet.org/`.

[14] S. Katti, D. Katabi, and K. Puchala. Slicing the Onion: Anonymous Routing without PKI. In *ACM HotNets*, College Park, MD, November 2005.

[15] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling High Bandwidth Aggregates in the Network. *Computer Communications Review*, 32(3), July 2002.

[16] E. Markey. Network Neutrality Act of 2006. `http://markey.house.gov/docs/telecomm/Markey%20Net%20Neutrality%20Act%20of%202006.pdf`, May 2006.

[17] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A Blueprint for Introducing Disruptive Technology into the Internet. In *Proc. of HotNets-I*, Oct. 2002.

[18] Save the Internet. `http://www.savetheinternet.com`.

[19] B. Timmerman. A security model for dynamic adaptive traffic masking. In *Procs of the 1997 workshop on New security paradigms*, 1997.

[20] Network Neutrality. `http://en.wikipedia.org/wiki/Net_neutrality`.