Canadian Internet Policy and Public Interest Clinic
Clinique d'intérêt public et de politique d'internet du Canada

Submission to the Office of the Privacy Commissioner of Canada: Rogers' Use of Deep Packet Inspection Equipment

Tamir Israel, Staff Lawyer

December 2, 2009

Samuelson-Glushko Canadian Internet Policy and Public Interest Clinic
University of Ottawa – Faculty of Common Law
57 Louis Pasteur Street, Ottawa, ON., K1N 6N5
Tel: (613) 562-5800 ext. 2553
Fax: (613) 562-5417
cippic@uottawa.ca
www.cippic.ca

Table of Contents

Introduction and Summary of Submissions

Recent regulatory proceedings regarding new traffic management practices adopted by ISPs have created confusion regarding factual issues about the use of DPI equipment. Most comprehensive statements on these factual issues have come from ISPs who, while accurately presenting the evidence, have put an inaccurate spin on it that has the effect of misrepresenting its impact. Accuracy is important with respect to these issues, however. The incorrect understanding currently presented by some ISPs, and endorsed in this office's Report of Investigation on Rogers use of DPI,[1] can have serious and far-reaching impact on a number of areas of law if accepted predominantly.

This submission focuses on establishing the importance of accuracy in this field and on correcting the following mistaken conclusions:

A. that there is such a thing as an application header that is distinct from the payload of a telecommunications packet;
B. that there is a clear-line distinction between layer 7 control information or 'application headers' and user-generated content;
C. that there is a qualitative difference between filtering telecommunications messages to determine types of applications being used and doing so to isolate specific user content; and
D. that software applications, as opposed to users of those applications, can be 'heavy users' of bandwidth; that the impact P2P downstream traffic has the same network impact as P2P upstream traffic; and that congestion on networks is "largely caused" by P2P applications.

A. Application Headers and Telecommunications Packet Payloads

Header information is often seen as 'envelope' information – information contained on the outside of an envelope and attracting lower expectations of privacy then the actual letter (the content or 'payload' of the envelope). Layer 7 data has recently (and inaccurately) been portrayed as 'header' information distinct from the payload or content of a telecommunications message.

The terms 'header' and 'payload' as coined in the OSI classification system are transitive and shift depending on the stage of the data transfer process at which data is being discussed.[2] Each layer encapsulates data passed to it from the layer below with control information, and this control information becomes the 'header' *for that layer*. So, in that respect, it *might* be accurate to speak of a layer 7 application header when discussing data while it is interacting with layer 7 services – that is, while it is sitting on a server or on a user's computer.[3] It does *not* make sense, however, to do so when speaking of

[1] Office of the Privacy Commissioner, *Report of Investigations – Rogers use of DPI: File #6100-03064*, August 26, 2009 ["Report"]

[2] K. Mochalski and H. Schulze, *White Paper: Deep Packet Inspection: Technology, Applications & Net Neutrality*, Ipoque, 2009 available online at: <http://www.ipoque.com/userfiles/file/DPI-Whitepaper.pdf>, ["Ipoque Whitepaper'] at p. 2 'The Header-Payload Confusion'; CISCO, *Interworking Basics*, Interworking Technology Handbook, Ch. 1, available online at: <http://www.cisco.com/en/US/docs/internetworking/technology/handbook/Intro-to-Internet.html> at p. 1-7; CCITT, *Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, International Standards Organization, (1994E) ISO/IED 7498(E)-1, ITU-T Rec. X.200, ["Original OSI Document"] available online at: <http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.200-199407-I!!PDF-E&type=items> [PDF], at s. 5.8.8.1.1, p. 23 and Figure 10, p. 26; and H. Zimmerman, "OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection", (1980) 28(4) *IEEE Transactions on Communications* 425, available online at: <http://www.comsoc.org/livepubs/50_journals/pdf/RightsManagement_eid=136833.pdf> [PDF] at p. 426.

[3] Dr. D.P. Reed, *Demarcation of Headers and Content orPayload in the Internet, and So-Called Application Headers*, August 20, 2009, Attachment 1. Dr. Reed makes this point very poignantly – 'Dear x' and 'Re:' lines in a standard letter could be loosely considered 'address' information. These terms share some characteristics with 'address' information typically found on the outside of an envelope – they are standard fields contained in most letters, they state who the letter is addressed to, they provide general information (Re: xxxx) that can help individuals to decide how to handle the rest of the letter (the 'content'). However,

telecommunications messages or packets and of how such data is managed and routed across a network. At that stage of the telecommunications exchange, data is not interacting with OSI layer 7 infrastructure or services (namely, the end user's computer and applications). It is interacting with layers 1-3 and at times layer 4. When speaking of ISPs and data transport, then, the 'header' of the telecommunications message includes layers 1-4 and the 'payload' or 'content' is anything else.

Any other characterization, such as that contained in the Report, is inaccurate, and has the potential of greatly changing how privacy expectations with respect to such information as it passes through a network are analyzed. Not surprisingly, there is broad consensus among engineers, DPI product developers, and in other submissions of your office that what DPI does, its essence; what makes it 'DPI', is look within the payload or content of a telecommunications message to find the layer 7 control information it requires.

This clarity is important because a great deal of data contained in the layer 7 control information (the 'application header') can only be characterized as 'user-generated'.

B. What's in a Header? 'Application Headers' and User-Generated Data

The reason that layer 7 data often contains a great deal of user generated personal information is that, from the inception of the Internet, the distinction between lower layer data (data from OSI layers 1-4) and all other data has been a fundamental architectural element of traffic management. The lower layer or 'data transport' data is intended for routing, while the upper layer or 'application' data (anything above layer 4) can and often will include user-generated content since that information is to be used and implemented in the end user's computer. Indeed, one central reason for this initial division between upper and lower layers was to allow users to encrypt the *payload* of a telecommunications message, allowing them to protect their own information (contained *inside* the payload) without interfering with the routing activities of carriers.

Much of the data contained in the 'application header' will be highly sensitive. This can include URLs visited, search queries entered into a search engine, source and destination email address, email subject lines and, through cookies, a vast range of information including, potentially, credit card numbers, SINs, etc. Given the lack of specificity and structure imposed on layer 7 control information, it can potentially include any other user data that an application developer wishes to transport in there. It can even include locational data such as longitude or latitude, now that mobile phone applications are using such data.[4] Thus it is inaccurate to distinguish, as some ISPs and the Report attempt to do, between 'application headers' on the one hand and 'user-generated content' on the other.

C. Fishing for Headers v. Fishing for Content

Perhaps it is this false dichotomy between 'application headers' and user-generated information that leads to the mistaken conclusions that Rogers' DPI is incapable of gaining knowledge of or collecting user-

---

this terminology only makes sense in the context of the recipient of the letter, and not in the context of the carrier delivering the letter. When discussing that carrier (Canada Post or Rogers, as the case may be), 'address' information means P.O. Box and postal code. The analogy is especially illuminating when one considers that information such as 'Dear x' and 'Re: xxx' *is* contained in the 'application header' of emails.

[4] S. Perez, "Dear iPhone Users: Your Apps are Spying on You", ReadWriteWeb, August 17, 2009, online at: <http://www.readwriteweb.com/archives/dear_iphone_users_your_apps_are_spying_on_you.php>, accessed November 12, 2009; and Y. Benjamin, "Apple Privacy Score – Snow Leopard – 10, iPhone – 0", SFGate, August 27, 2009, online at: <http://www.sfgate.com/cgi-bin/blogs/ybenjamin/detail?blogid=150&entry_id=46236>, accessed November 12, 2009. Perez points out how iPhone applications are collecting longitude and latitude (even when they do not need them) through simple queries to the iPhone API. Benjamin demonstrates how some iPhone applications transport similar information (phone numbers they do not need, acquired from the API), unencrypted, in the HTTP 'application header'.

generated information. This is incorrect as any DPI equipment has the capacity to gain such information. It is merely a question of configuration – a task that the versatility of DPI equipment makes relatively simple. Gaining knowledge of user generated content contained in the application layer control information could be as simple as adding new signature strings to the existing signature library that is a fundamental element of any DPI box. In addition, DPI equipment has the capacity to intercept entire packet flows so that the full breadth of a telecommunications exchange can be reconstructed and analyzed. While it would be difficult for DPI equipment to conduct such analysis and storage on its own (external storage is required to store all this data and to reconstruct it at more leisurely speeds), it would be extremely difficult if not technically infeasible to do so on a broad scale *without* DPI equipment. This is because DPI, by its nature, offers an unparalleled capacity to analyze packet flows at inline speeds, so that an ISP wishing to intercept entire classes of telecommunications could isolate those specific classes and copy and store those alone. Without this inline analytical capacity, such an ISP would have to capture *all* classes of telecommunications – an immense and unmanageable amount of data.[5]

So, it is inaccurate to conclude (as some ISPs and the Report do) that DPI equipment lacks the capacity to gain knowledge of user generated data. It can do so easily, by filtering telecommunications messages inline for specific bits of user-generated content, including key words or phrases that may be of interest to an ISP. These key words can come from email subject lines, emails, URLs, search queries, and so on. It can include information, for example, identifying the specific file being transferred over a P2P network by an individual. While CIPPIC lacks the specifications of Rogers' DPI, this capacity is inherent in DPI equipment and relies on basic DPI functions (particularly its capacity for signature string analysis). The specifications of Rogers' DPI may change the *extent* to which its DPI can do this, but not its capacity to do so. In addition, given that DPI equipment has the capacity to intercept data and is a necessary element if one wishes to capture entire telecommunications messages on a grander scale, access to such information is certainly not beyond the capacity of the equipment. It is important not to understate, as many ISPs attempt to do, the capacities of this equipment, and this is especially the case when function creep and upcoming lawful access legislation are considered.

D. P2P: What's the Problem?

It is not 'applications' as a class that are 'heavy bandwidth users', but individuals. Peer-to-peer ("P2P") applications offer a good example of this truth. According to Canadian ISPs, P2P applications generate about 36% of all traffic on Canadian networks.[6] This number is likely an over-estimation of the amount of P2P traffic in Canada – Bell tells us that the P2P figure on *its* networks (the largest in Canada) is much lower (27% of all traffic when unthrottled)[7] and, since the filing of this complaint, P2P usage has been steadily decreasing.[8] Regardless, even at 36%, when one considers that Canadian ISPs that roughly 50%

---

[5] J. Tollet, "Myth 7: All IP Traffic Can Be Recorded", in *7 Myths of IP Networking*, dpacket.org,, September 22, 2008, online at: <https://www.dpacket.org/articles/myth-7-all-ip-traffic-can-be-recorded>, accessed October 18, 2009 [Tollet Myth 7]. Tollet is the Chief Technical Officer of Qosmos, a DPI vendor.

[6] CRTC, *Letter to Interested Parties Telecom PN CRTC 2008-19*, [CRTC Letter] February 11, 2009, available online at: <http://www.crtc.gc.ca/eng/archive/2009/lt090211.htm>, at chart (CRTC) 04Dec2008-1 (b), straight arithmetic average of %HTTP/Streaming traffic (40%) and %P2P traffic (36%).

[7] C. Condon, *Bell - Oral Testimony in Telecom PN CRTC 2008-19*, Transcript of Proceeding, July 14, 2009, available online at: <http://www.crtc.gc.ca/eng/transcripts/2009/tt0714.htm>, at line 6042:

> 27 percent of our traffic is peer-to-peer throughout the day, but at peak that number reduces to 14 percent as a result of our shaping of that traffic.

Given that the 36% figure emerges from a straight arithmetic average of anonymized numbers provided by the CRTC, with correction for the weight of individual ISPs (this data is not available), and given that Bell is Canada's largest provider of service, it can be safely said that the 36% figure is an over-estimation and 27% is likely more accurate.

[8] Arbor Networks, "Two Year Study of Global Internet Traffic Will Be Presented at NANOG47", Arbor Networks – News Releases, October 13, 2009, online at: <http://www.arbornetworks.com/en/arbor-networks-the-university-of-michigan-and-merit-network-to-present-two-year-study-of-global-int-2.html>, and Sandvine, *2009 Globel Broadband Phenomena – Executive Summary*, [Sandvine Global 2009] Sandvine, October 2009, available online at:

of their customers use P2P applications,[9] it is evident that P2P applications are not the problem. These figures tell us that 50% of all Canadian users are generating at most something between 27% and 36% of all bandwidth when using P2P applications. That can hardly be characterized as 'heavy usage'. Ignoring applications traffic breakdown, however, Canadian ISPs tell us that about 47% of *all* their bandwidth traffic is generated by 5% of users.[10] From these numbers it is clear that it is these individual users, not a class of applications, that are 'heavy bandwidth users'. It is inaccurate to characterize applications in this manner, as the Report and ISPs attempt to do. Indeed, using Canadian ISPs' own statistics, HTTP applications generate more traffic (at least 40%) than P2P applications.

Further, to the extent that P2P applications do have the capacity to 'consume more bandwidth' than other types of applications, this is only true for upstream traffic and not at all for downstream. P2P is somewhat unique in that it utilizes a client-to-client model of data transfer, as opposed to the traditional client-server model – as P2P traffic comes *from* a client (not a server) it emerges from within the ISP's network and passes over it in the *upstream* direction. In this respect, P2P has the potential to place a greater strain on the *upstream* bandwidth capacity of a network than other applications and can cause unique problems for networks accustomed to asymmetric and insufficient provisioning – on the *upstream*. Again, this is a temporary feature, as more recent traffic management reports demonstrate quite clearly that the Internet has moved towards a client-to-client model. In these more recent and comprehensive reports, we see that whereas before other types of applications (HTTP) exceeded P2P bandwidth consumption in *net* traffic and *downstream* traffic, they now do so in *upstream* traffic as well.[11]

Regardless, there is no evidence that P2P applications (or any other applications, for that matter) are actually causing any 'widespread' congestion (whether upstream or downstream), as a number of ISPs and the Report claim. The evidence is quite to the contrary. Comcast, for example, is an American cable company much like Rogers that, again, much like Rogers, was complaining of the widespread phenomenon of congestion imposed on its networks by P2P. But as Comcast began, in response to an FCC order, to gather more granular data on how much congestion was actually occurring on its network, it turned out that to neutralize this 'widespread phenomenon' on its networks required throttling of less than 1% of users for rarely for more than 15 minutes at a time.[12] This amount of congestion was not, moreover, *caused* by P2P applications, which were merely one contributing factor (and not even the predominant one), among many.[13] Far from being the primary cause of 'the widespread phenomenon of congestion on networks', the evidence indicates that there *is* no such widespread phenomenon, and to the extent that there is, P2P as a class contributes only a small amount (27%-36% at most) to this – far less then other types of applications such as web applications, which contribute at least 40% of all traffic.[14]

Each of these issues will be addressed in turn, below.

<http://www.sandvine.com/downloads/documents/2009%20Global%20Broadband%20Phenomena%20-%20Executive%20Summary.pdf> at p. 2.

[9] J. Daniels, *Bell - Oral Testimony in Telecom PN CRTC 2008-19*, Transcript of Proceeding, July 14, 2009, available online at: <http://www.crtc.gc.ca/eng/transcripts/2009/tt0714.htm>, at line 6829.

[10] CRTC Letter, *supra* note 6, at chart (CRTC) 04Dec2008-2 (c), straight arithmetic average of 'percentage of total traffic attributable to top 5% end-users'.

[11] Sandvine Global 2009, *supra* note 8, at p. 4.

[12] Sandvine, *Initial Comments to Telecom PN CRTC 2008-19*, February 23, 2009, [Sandvine Initial Comments] available online at: <http://www.crtc.gc.ca/public/partvii/2008/8646/c12_200815400/1029527.pdf>, at para. 95, notes that less than 1/3 of 1 percent of customers [22/6016 customers, in one trial] need to be throttled for less than 15 minutes at a time to neutralize the "widespread phenomenon of congestion in networks".

[13] Condon, Bell Testimony, *supra* note 7, at lines 6293-6294: "every packet that's going through a…congested device is a contributor to that congestion. In our case, as we said, 27 percent of our traffic is peer-to-peer…"

[14] CRTC Letter, *supra* note 6, at chart (CRTC) 04Dec2008-1 (b), straight arithmetic average of %HTTP/Streaming traffic, but see Sandvine Initial Comments, *supra* note 12, for far more accurate statistics (because the CRTC numbers are not weighted to account for different sizes in ISPs) on traffic breakdown, which state that net HTTP traffic in North America is 67%. See note 193, below, for more details.

A. Application Headers and Telecommunications Packet Payloads

The Report states as follows:

> DPI has the capability to inspect these many protocol headers. For example, viewing down to the application header of a packet—the layer directly above the payload—allows an ISP to determine the type of software application that is being used to transmit the packet.[15]
>
> […]
>
> All DPI products have the ability to look within the application header of a packet or traffic stream, and make decisions based on that information, without examining the actual content. While it is possible to examine the content of packets, the DPI technology currently deployed by Rogers does not have the ability to do so.[16]

These statements are inaccurate. To begin with, while there is an application layer – a conceptual layer that governs communications between users and user applications at either end of a telecommunication message[17] – whether or not there is any such well-defined thing as an application *header* in traffic management terms remains seriously in doubt.[18] Certainly, suggesting that any type of 'header' is identical to its corresponding 'layer', as in paragraph 6 of the Report, is inaccurate and merely causes confusion. An OSI 'layer' refers to a series of functions that *can*, but need not, attach control information (a header) to data it handles.[19] While there has been some confusion over how to apply the terms 'payload' and 'header', the term 'packet' refers to data as it interacts with network layer functions.[20] Hence, 'packet payload' includes anything and everything from the upper layers of the OSI model.

The terms 'header' and 'payload' are somewhat transitive, and depend on the stage of the data transfer interaction at which a unit of data is being examined:

> Control Information takes one of two forms: headers and trailers…Headers, trailers and data are relative concepts, depending on the layer that analyzes the information unit. At the network layer, for example, an information unit consists of a Layer 3 header and data. At the data link layer, however, all the information passed down by the network layer (the Layer 3 header and the data) is treated as data.[21]

This has led to some confusion as to how to apply this terminology when discussing network management. Under the OSI reference model, which contains 7 conceptual layers of functions, each layer takes data from a lower layer (which becomes the 'payload') and encapsulates it in a header. Determining

---

[15] Report, *supra* note 1, at para. 6, emphasis mine.
[16] Report, *supra* note 1, at para. 25, my emphasis.
[17] CISCO, *supra* note 2, at p. 1-4.
[18] Reed, *supra* note 3, Attachment 1, explains how there are in some cases structures located in the application layer control information portion of the payload that could be referred to loosely as 'headers'. However, while in network management parlance the term 'header' generally refers to a clearly defined and standardized block of control information used by data transport equipment as it transmits its payload across various networks, so-called application headers lack such standardization and do not interact with network layer equipment and functionality at all. Dr. Reed's envelope analogy is perhaps the clearest way of understanding the distinction. Dr. Reed states that a business letter can include structures in the letter itself – a standard way of addressing its recipient. This structure can include elements such as the date, address, salutation, subject line at the top, and signature at the bottom (as, indeed, this letter does). However, to conflate this information with the address on the outside of the envelope in which the letter is carried would make little sense. See also: CAIP et al, *Application to Review and Vary Telecom Decision CRTC 2008-108*, May 21, 2009, available online at:
<http://www.crtc.gc.ca/public/partvii/2009/8662/p8_200907727/1140124.zip>, paras. 33-46.
[19] CISCO, *supra* note 2, at p. 1-7: "An OSI layer is not required to attach a header…to data from upper layers."
[20] CISCO, *supra* note 2, at p. 1-12:

> A *packet* is an information unit whose source and destination are network layer entities. A packet is composed of the network layer header…and upper-layer data. The header and trailer contain control information intended for the network layer entity in the destination system. Data from upper-layer entities is encapsulated in the network layer header and trailer.

[21] CISCO, *supra* note 2, at p. 1-7. Ipoque, a DPI vendor, provides a detailed examination of these issues in a recent whitepaper it published: Ipoque Whitepaper, *supra* note 2, at p. 2, and the Original OSI Document, the original ISO document setting out the OSI system, sets out this transitive terminology as well at *supra* note 2, at s. 5.8.8.1.1, p. 23 and in Figure 10 on p. 26.

what is in the 'header' and what is in the 'payload' of a unit of data depends on what stage of the OSI communication the data is in. The application layer of the OSI model implements its control information within sending and receiving computers.[22] While it *may* be logical to refer to application layer control information as 'header' information when discussing data as it interacts with layer 7 functionality (i.e. with an application in the sending or receiving computer), it is simply inaccurate to do so in reference to a *packet* that is being transported across a *network*. The term 'packet' refers very specifically to data as it exists at the network layer, where the term payload refers to anything that could be referred to as an application header.[23] And any discussion of traffic management such as that contained in the Report is a discussion of packets *as they traverse the network* – that is, as they interact with OSI layers 1-4, not OSI layer 7. It simply does not make sense to refer to layer 7 control information as 'header' information at that point of the OSI transition. An ISP, in its role as a telecommunications carrier, only ever interacts with packets as they exist at OSI layers 1-4 (the 'data transport' layers in figure 1 below).

There is a very logical explanation for this terminology as well. In order to transport data from one computer to another, network equipment *only* requires the control information attached by OSI layers 1-3.[24] As explained by Ipoque, a DPI vendor, the "only information required by Internet nodes…to deliver a packet" is contained in headers attached by OSI layers 1-3.[25] In addition, OSI layer 4 control information is also necessary for data transport as it provides the bridging element between a network and its end users – it communicates with both.[26] This distinction between control information required for data transport (OSI layers 1-4) and *other* data (anything else or 'payload') is maintained in any discussion of network traffic routing, since only OSI layers 1-4 are required to transport telecommunications messages across a network.

Even discussions of the OSI model differentiate between upper (OSI layers 5-7 or 'Application' layers) and lower (OSI layers 1-4 or 'Data Transport' layers) layer data. As CISCO points out (see figure 1), these upper layers include anything that could remotely be referred to as an 'application header'.

---

[22] Original OSI Document, *supra* note 2, at s. 4.2.4, p. 3; CISCO, *supra* note 2:

> Information being transferred form a software application in one computer system to a software application in another must pass through the OSI layers (p. 1-5)…The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the user interact directly with the software application. (at p. 1-11)

See also Zimmerman, *supra* note 2, at p. 430: "This is the highest layer…Protocols of this layer directly serve the end user by providing the distributed information service appropriate to an application, to its management, and to system management."

[23] CISCO, *supra* note 2, at p. 1-12:

> A *packet* is an information unit whose source and destination are network layer entities. A packet is composed of the network layer header…and upper-layer data. The header and trailer contain control information intended for the network layer entity in the destination system. Data from upper-layer entities is encapsulated in the network layer header and trailer.

[24] Ipoque Whitepaper, *supra* note 2, at p. 2:

> For packet forwarding in the Internet, however, the applications sending and receiving the packets are irrelevant. Packets are sent from one host (represented by the sender IP address) to another (represented by the receiver IP address). These two addresses are the only information required by Internet nodes (the sending and receiving hosts and the intermediate routers) to deliver a packet.

[25] *Ibid.*

[26] Zimmerman, *supra* note 2, at p. 430, my underline:

> Control of data transportation from source end system to destination end system (which need not be performed in intermediate nodes) is the last function to be performed in order to provide the totality of the transport service. Thus, the upper layer in the transport-service part of the architecture is the *Transport Layer*, sitting on top of the Network Layer.

See also: ITU-T, *Series X: Data Networks and Open System Communication*, International Standards Organization ITU-T Rec. X.224 (1995 E) [ISO/IEC 8073:1997(E)], available online at: <http://www.itu.int/rec/T-REC-X.223/en/> at p. 14: "The functions in the Transport Layer [OSI Layer 4] are those necessary to bridge the gap between the services available from the Network Layer and those to be offered to the…users."

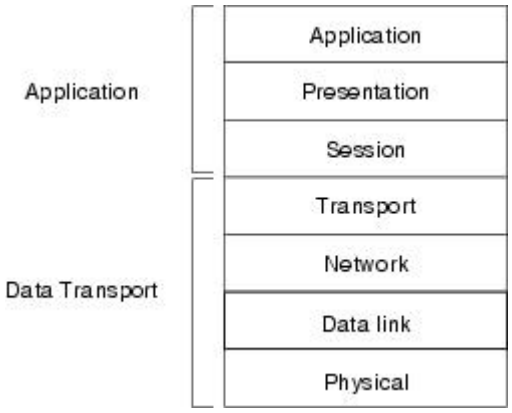

Figure 1: "[I]llustrates the division between upper and lower OSI layers"[27]

Deep Packet Inspection by definition inspects the payload of packets as they pass through an ISP's network. This is what distinguishes it from other forms of packet inspection. Normal network packet inspection examines data attached by layers 1-3, shallow (or stateful) packet inspection examines layer 4 data, while DPI looks within the *payload* or *content* of telecommunications messages as they are carried over a network.[28] As Allot Communications, a DPI vendor, explains, this capacity to inspect the packet payload is precisely the operative difference between *Shallow* and *Deep* Packet Inspection:

> The standard packet inspection process (a.k.a. shallow packet inspection) extracts basic protocol information such as IP addresses (source, destination) and other low-level connection states. This information typically resides in the packet header itself and consequently reveals the principal communication intent…DPI, on the other hand, provides application awareness. This is achieved by analyzing the content in both the packet header and the payload over a series of packet transactions…Analysis by string match involves the search for a sequence of textual characters or numeric values within the contents of the packet.[29]

In addition, Ipoque, another DPI vendor, states quite clearly that:

> DPI systems inspect entire packets travelling the network as part of a communication, looking not only at packet headers like legacy systems, but also at the packet's payload. The central part of this definition is the inspection of packet payload.[30]

Aside from CISCO, Allot, and Ipoque, other objective sources support the fact that the operative element in DPI ('deep') is no more or less than its capacity to look inside the packet payload or content (which includes any so-called application header). An independent report [the "Heavy Reading" report]

---

[27] CISCO, *supra* note 2, at pp.1-4 to 1-5. CISCO further clarifies the difference between upper (application) and lower (data transport) layer control information:

> The *upper layers* of the OSI model deal with application issues and generally are implemented only in software…Both users and application layer processes interact with software applications that contain a communications component…The *lower layers* of the OSI model handle data transport issues. (*Ibid* at p. 1-4)

[28] See Open Internet Coalition, *Initial Comments to Telecom PN CRTC 2008-19*, February 23, 2009, available online at: <http://www.crtc.gc.ca/public/partvii/2008/8646/c12_200815400/1029708.pdf>, [xxx] at paras. 27, 31 (my emphasis):

> DPI involves looking at the content of a communication beyond the header information. DPI devices allow an ISP to inspect the entire content of a communication…ISPs can also engage in the "shallow" inspection of a user's communication, which includes looking beyond the network's IP header into the transport-layer header. Shallow packet inspection allows an ISP to examine the TCP or UDP header and not the payload. This form of inspection can examine information at Layers 2, 3, and 4 of the OSI model…

[29] Allot Communications, *White Paper: Digging Deeper Into Deep Packet Inspection (DPI)*, Allot Communications Inc., April 2007, [Allot Whitepaper] available online at: <http://www.getadvanced.net/learning/whitepapers/networkmanagement/Deep%20Packet%20Inspection_White_Paper.pdf> at pp. 2-3, 5. Also, see Allot figure 1, where Allot defines Shallow Packet Inspection data as "data from packet headers" (*Ibid*. at p. 2).

[30] Ipoque Whitepaper, *supra* note 2, at p. 1.

commissioned by the CRTC in connection with its recent traffic management proceeding stated the following:

> In sum, DPI equipment inspects *the contents* of packets traveling across an IP network. It can more or less accurately identify the application or protocol in use by examining the source and destination IP address, the port number, and packet payload. Port numbers are a basic means for identifying applications; for example, email using the Simple Message Transfer Protocol (SMTP) uses port 25. Packet headers include this information, along with source and destination address and other data including DiffServ class information where relevant. The packet payload itself (eg part of a Web page) may be examined to look for strings in the protocol that identify it (eg "kazaa", which appears in one of the fields used to handle Kazaa requests). Equipment may also look for telltale signs of an application, such as the length of the packet payload.[31]

In addition, the OPC's own contribution to the same CRTC proceeding confirms this point:

> DPI devices have the ability to look at Layer 2 (link layer) through Layer 7 (application layer) of the Open Systems Interconnection (OSI) model. DPI devices can, therefore, examine headers and data protocol structures as well as the actual payload of the message. In other words, DPI technology can look into the content of a message sent over the Internet. To use a real-world example, using DPI is akin to a third party opening an envelope sent by surface mail, and reading its contents before it reaches its intended destination…DPI identifies and classifies traffic based on a signature database that includes information extracted from the data part of a packet, allowing finer control than classification based only on header information (sometimes referred to as shallow or stateful packet inspection).[32]

While Rogers has provided no comprehensive report of how its DPI operates, Bell has provided one that confirms it operates precisely as these two documents describe:

> DPI uses a set of "signatures" or unique fingerprints to identify an application regardless of how it declares itself in the Application, TCP (or UDP) and IP protocol headers that are encapsulated around the content… The "deep packet inspection" or signature detection of the DPI is used to classify a communications exchange between one or more endpoints in order to resolve the exchange to a "flow entry" which is maintained in the hardware of the DPI device.[33]

As pointed in the OPC's submission to Telecom Public Notice CRTC 2008-19 cited above, these 'signatures' or 'unique fingerprints' that DPI inspects are located in *the data part of a packet, allowing finer control than classification based only on header information*. The Heavy Reading report points out above that such signatures or unique fingerprints are located *in the packet payload itself (i.e. part of a Web page)*.

There is a technical corollary to this distinction between packet and header. Lower OSI layer headers have very standardized definitions and precise byte offsets.[34] It is a simple manner for a network data

---

[31] G. Finnie, *ISP Traffic Management Technologies: The State of the Art*, Heavy Reading, Prepared on behalf of the Canadian Radio-Television and Telecommunications Commission, January 2009, available online at: <http://www.crtc.gc.ca/PartVII/eng/2008/8646/isp-fsi.pdf>, at p. 8, my emphasis.

[32] Office of the Privacy Commissioner, *Review of Internet traffic management practices of Internet Service Providers*, submission to Telecom Public Notice CRTC 2008-19, February 18, 2009, available on line at: <http://www.priv.gc.ca/information/pub/sub_crtc_090218_e.cfm#_ednref10>, at paras. 13-14.

[33] Bell, CAIP v. Bell: Submissions of July 11, 2008, [Bell, CAIP submission] available online at: <http://www.crtc.gc.ca/public/partvii/2008/8622/c51_200805153_1/926702.zip>, at paras. 187, 192, my emphasis.

[34] Compare, for example, the network layer packet header specification provided by Dr. Reed (*supra* note 3, Attachment 1). Every piece of data is located at a very specific byte offset – for 'source IP', start reading at byte #96. The TCP protocol specification contains similar imperatives for byte offsets: Information Sciences Institute, *RFC 793 – Transmission Control Protocol*, DARPA, September 1981, available online at: <http://tools.ietf.org/html/rfc793>, at p. 14. The specifications for *application* layer control information are far less precise. The Simple Mail Transfer Protocol (SMTP) used by most Email applications, for example, does provide required 'fields' but contains no guidance with respect to byte offsets: J.B. Postel, *RFC 821 – Simple Mail Transfer Protocol*, Information Sciences Institute, August 1982, available online at: <http://tools.ietf.org/html/rfc821>; the same applies to protocol specifications for the Hypertext Transfer Protocol (HTTP/1.1), such as: Network Working Group, *RFC 2068 – Hypertext Transfer Protocol – HTTP/1.1*, IETF, January 1997, available online at: <http://tools.ietf.org/html/rfc2068>, Section 4.5 General Header Fields, p. 33.

transport device to implement lower layer control information, because it will know at what byte to start reading for each piece of information it needs. So-called application headers do not require the same level of precision as packet headers because they are designed to be read and implemented in destination *computers*.[35] This means that they are not intended to be analyzed at inline speeds, and need not be defined so precisely as packet headers. In addition, one of the original motivations behind the demarcation between *packet* headers and *packet* payload (including upper layer control information) was to enable encryption of the payload in a manner that would not interfere with network equipment routing tasks.[36] This was to protect user-generated data from interception. That is one reason why so-called application headers can and often do contain a great deal of user-generated data, while lower layer headers do not.[37]

For these reasons, statements to the effect that the application header is "the layer directly above the payload" are factually inaccurate on a number of counts:

- they wrongly conflate the terms 'header' and 'layer';
- they create a false dichotomy between an *application* header and a *packet* payload;
- they refer to application headers while the issue of whether there is any such well-defined, standardized term at all remains in question; and
- they lead directly to the erroneous conclusion that in Network management terms there is no difference between information implemented in the network itself and information implemented in the computers of individuals at either end of that network.

The importance of clarity if and when addressing such issues cannot be understated. Any reference to 'headers' and 'payload' that misuses this terminology merely contributes to the web of confusion, perpetuated by some ISPs in recent times, regarding the dichotomy between user/application data on the one hand and network control data on the other. This dichotomy is "the key architectural feature of the Internet and its protocols", and any description of network trafficking techniques that ignores it is deficient and inaccurate.[38] Further, imprecise and inaccurate descriptions of such network elements can have serious regulatory impacts as such terms are instrumental in determining the application of laws such as the *Telecommunications Act*.[39]

Perpetuating and endorsing such mischaracterizations of headers as distinct from payload or content can have serious implications in other areas of law as well. Indeed, the imprecision in question has led to further errors in fact, described below, in the Report itself. Such confusion with respect to email headers can additionally have a serious impact on the amount of protection s.8 of the *Charter* can offer against lawful interception of such information. As Stanley Cohen points out:

> E-mail headers (hidden and exposed) can be likened to the information on the envelope containing a posted letter. The information on the cover carries a lower expectation of privacy than does the message inside. It is therefore conceivable that certain aspects of e-mail transmissions, such as the so-called "gateway information"—headers, names

[35] See *supra* note 22 and accompanying text. As Zimmerman points out (*supra* note 2, at p. 430) the application layer provides information *applications* need in order to provide the information they need to operate. Email applications, for example, require a subject line to parse incoming Emails so the user can be provided with this information without having to open the entire Email.
[36] Dr. D.P. Reed, Email message sent by Dr. Reed to Tamir Israel on August 21, 2009 and on file with the recipient.
[37] See Section B below.
[38] Reed, *supra* note 3, Attachment 1.
[39] *Telecommunications Act*, S.C. 1993, c. 38, T-3.4 (as amended), s. 36 of which places limitations on what a telecommunications carrier can do to interfere with the "telecommunications messages" it carries. Inaccurately referring to 'application headers' as distinct entities from packet payloads changes the determination of what, precisely, a "telecommunications message" is. Characterizing the application header as outside the packet payload is an attempt to take the information contained in that header outside of the term 'telecommunications message', so that it becomes acceptable to interact with information contained in that header and base traffic management decisions thereon. For more details see CAIP, *supra* note 18.

> of senders and recipients and the subject line, may be regarded as of a non-private nature and hence may be accessible to the government…without need of prior authorization.[40]

The Courts have approached the issue in a similar manner, as in *R. v. Weir*, where the Alberta Queen's Bench stated:

> Before moving to the e-mail message I will address the cover. The envelope on first class mail shields the contents of the message. The information on the cover carries a lower expectation of privacy than does the message inside. In the e-mail environment, the headers (hidden and exposed) can be likened to the information on the envelope. The message is directed by its headers. Much repair work to e-mail can be done through headers. Like the outside of the envelope, the headers have a lower expectation of privacy. [41]

Note that in *Weir*, the Court was speaking of privacy vis-à-vis the email service provider, which typically will have access to the application layer control information. One would assume that the same analysis might apply to an ISP, however, with a lower expectation of privacy attaching to anything labelled 'header' or 'control' information with respect to its own operations. This is in spite of the fact that, as will be detailed below, application layer control information can contain a great deal of personal and even sensitive information depending on the context.

An immediate example of where this type of distinction may play a decisive role is in the interpretation of the term 'transmission data' as contained in proposed Bill C-46, which is currently defined as data that:

(a) relates to the telecommunication functions of dialling, routing, addressing or signalling;
(b) is transmitted to identify, activate or configure a device, including a computer program…in order to establish or maintain access to a telecommunication service for the purpose of enabling a communication, or is generated during the creation, transmission or reception of a communication and identifies or purports to identify the type, direction, date, time, duration, size, origin, destination or termination of the communication; and
(c) does not reveal the substance, meaning or purpose of the communication.[42]

The definition of this section may well turn on whether layer 7 control information is defined as 'header' or 'payload'. If classified as part of the 'payload' or 'content' of a telecommunications message, layer 7 control information will not likely relate to the telecommunications functions of dialling, routing, addressing or signalling (a).

However, if classified as merely 'header' or envelope information, much of this data might become transmission data. Take email addresses for example. Traditionally, 'routing' information (a) required to identify the origin or destination (b) of a telecommunications message would refer to IP address alone.[43] These will not, purportedly, betray the purpose of the communication (c). But once layer 7 data is classified as merely 'header' or envelope information – control information merely used to convey a telecommunications message analogous to all other header information – Email addresses (contained in the so-called application layer data) could be deemed as 'relating to the functions of routing' (a) as well as 'identifying the origin and destination' of the telecommunications signal (b) and therefore be included within the definition of transmission data and subject to different disclosure restrictions. The same could apply to other pieces of private and personal data contained in the layer 7 control information.

---

[40] S.A. Cohen, *Privacy, Crime and Terror: Legal Rights and Security in a Time of Peril*, (Markahm, ON.: LexisNexis Canada Inc., 2005) at pp. 494-495.
[41] *R. v. Weir*, [1998] 213 A.R. 285 (Alta. Q.B.), affirmed [2001] 281 A.R. 333 (Alta. C.A.), at paras. 72-73. Note that this case only analyzed how an *email* service provider interacts with email headers, etc., not how an ISP carrier does. Nonetheless, it is the main statement of Canadian law on the issue to date.
[42] Bill C-46, *An Act to Amend the Criminal Code, the Competition Act and the Mutual Assistance in Criminal Matters Act*, 1st reading, available online at: <http://www2.parl.gc.ca/content/hoc/Bills/402/Government/C-46/C-46_1/C-46_1.PDF>, at clause 15, amending s.487.011 of the Criminal Code.
[43] Ipoque Whitepaper, *supra* note 2, at p. 2.

Dr. Reed provides a similar envelope analogy that poignantly explains the misunderstanding and highlights the potential impact for expectations of privacy that can emerge from inaccurate terminology.[44] He points out that there can be some well-defined structures in the layer 7 control information that might resemble other types of headers. Similarly, a letter might have standard 'Dear x' and 'Re: xxxx' lines which could be loosely referred to as 'address' information. These terms share many characteristics with 'address' information typically found on the outside of an envelope – they are standard fields contained in most letters, they state who the letter is addressed to, they provide general information (Re: xxxx) that can help individual recipients (whether the addressee on the envelope, an assistant, etc.) decide how to handle the rest of the letter (the 'content'). It might make sense, from the perspective of the recipient of such a letter, to speak of 'address' information (at the top of the letter) and content (the rest of the letter). However, to maintain this terminology when speaking of the carrier delivering it would make little sense. When discussing that carrier (Canada Post or Rogers, as the case may be), 'address' information means P.O. Box, street number and postal code – whatever the recipient decided to place on the outside of the envelope.

Content means whatever is inside the envelope. Conflating the two types of 'address information' without regard to context would, especially if one were not well versed in the technical intricacies of letter delivery, give one the impression that a 'Re: xxxx' line holds the same expectation of privacy as a postal code. This impression would be highly inaccurate and misleading. The analogy is especially illuminating when one considers that information such as 'Dear x' and 'Re: xxx' *is* contained in the 'application header' of emails. Indeed, this is the reason that Bell adopts precisely this analogy to describe the operation of its DPI: "[t]o use a postal analogy, DPI can look at other identifying characteristics of the envelope, but not inside the envelope".[45]

It is therefore very important to clarify that, from the perspective of a telecommunications carrier, anything that could be referred to as an 'application header' is not 'gateway information', nor is it analogous to an address on the cover of an envelope. It is part of the content or payload of the telecommunications that such a carrier conveys across its network. It is intended only for sending and receiving computers or servers. And any other characterization is false and misleading. As pointed out in the next section, this is especially important from a privacy perspective, as so-called application headers often contain user-generated data, sometimes of a very personal and sensitive nature.

We ask in light of this that the inaccurate portions cited above be either removed or corrected to account for the inaccuracies pointed out here. The application layer is not synonymous with the so-called application header.[46] The application layer is not 'directly above the payload' of the packet, but rather within it.[47] The "ability to look within the application header of a packet" is synonymous with the ability "to examine the content of packets."[48]

B. What's in a Header? ' Application Headers' and User-Generated Data

The Report includes the following statement:

> Information that is transmitted via the Internet is broken down into packets, routed to its destination and reassembled into the original content. The content is the user-generated information (such as email content) that is surrounded by several layers of control information, known as headers, to ensure proper handling and routing.[49]

[44] Reed, *supra* note 3, Attachment 1.
[45] Bell, CAIP submission, *supra* note 33, at para. 184 and at figure 18.
[46] Report, *supra* note 1, at para. 6: "…viewing down to the application header of a packet—the layer directly above the payload…"
[47] *Ibid.*
[48] *Ibid*., at para. 25.
[49] *Ibid*, at para. 5-6, my emphasis.

The misleading impact of conflating 'application header' or 'application control information' with 'packet payload' is also evident in the above-mentioned statement. In this excerpt, 'payload' has now become 'actual' or 'user-generated' content, and this is treated distinctly from 'header' or 'control' information. This distinction, again, is false. Application layer control information, as opposed to control information utilized by lower OSI layers (1-4), can and often will contain 'user-generated' data.

Also contrary to network data transport headers, there is no clear definition of what an application header should contain. This is because application layer control information is added and implemented in sending and receiving *computers* and not designed to interact with network equipment. Since an application developer can often decide what will be done with this control information at both ends when she designs the application, there is often no need for standardization. In contrast, network data transport equipment *requires* certain control information from the sending computer in order to carry its telecommunication message to its destination. But the sending computer has no control over the network data transport equipment. So a great deal of standardization is absolutely necessary if the Network is to work. Some measure of standardization has developed around a number of higher protocols in order to facilitate communication across different applications with similar functions.[50] Email is a good example, where there are various Email applications, but all require some of the same basic information. However there is nothing standard about control information at the application layer *per se,* and any application developer can potentially change all or part of what is contained therein.

More to the point, layer 7 interacts directly with the user and for this reason it will often include information created by one user and intended directly for another, not for network equipment.[51] What this means in practical terms is that any broad distinction between application layer control information and 'user-generated' data cannot be upheld. To begin with, Rogers' DPI already reads layer 7 control information to determine the types of applications a customer is using. Arguably, this information is 'user-generated content'. It is certainly 'content', in that it is located in the telecommunications packet payload and to that effect is akin to 'opening the envelope' in order to read – not the entire letter, but merely enough to determine the nature of the envelope's contents (cheque, business letter, gift, etc.). [52] This amounts to examining the contents of the mail, and is no different from examining the payload of a telecommunications packet flow to identify which application a person has chosen to use. And that information is user-generated, in that the user has *chosen* to use that particular application at that particular time.

But that is only the beginning of the story. Here, for example, is an example of what is found in most email headers:

[50] Reed, *supra* note 3, Attachment 1.
[51] See *supra* note 22 for more information.
[52] Reed, *supra* note 3, Attachment 1. One can imagine a similar technology being employed at the mailbox – one that allows the post, not to read 'the content' of mail, but merely to discern and record the shape of the content to determine 'cheque' or 'letter', or merely to read the 're:' line of letters to see if its business or personal. CIPPIC suggests that this *would* be considered 'examining the content' of the telecommunications packet.

Date: Mon, 25 May 2009 13:38:24 -0400
From: **Chris Donaldson <cdonaldson@cippic.ca>**
User-Agent: **Thunderbird** 2.0.0.21 (X11/20090409)
To: **Tamir Israel <tisrael@cippic.ca>**
Subject: **Dear Tamir: It's Over**
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 7bit

Figure 2 – MIME portion of SMTP Email 'Application' Header[53]

Note that this so-called application header contains not only the sender's and recipient's name and email address, but also the subject line as part of the control information for the email application in question. While the exact layout of the MIME header will change from email application to email application, it will always contain these information fields (sender/receiver, subject line).[54] All of this, and especially the subject line, are 'user-generated' data and can moreover include highly personal information which a carrier does not require in order to transport data across its network.

Due again to the lack of any standardization with respect to so-called application headers, the presence, amount or sensitivity of user-generated data in a header generated by a particular application with a communications component will change from application from application and could potentially be quite extensive. Email applications often include sender, receiver address and subject lines as control information to "ensure proper handling and routing" by the receiving email application.[55] This allows the receiving email application to classify incoming emails differently based on sender and to display just the subject line as opposed to the rest of the message in preview mode. It also enables the email service provider to do maintenance and repair work on email messages once these messages are located on its servers.[56]

Other so-called application headers can include a variety of other personal or sensitive information. BitTorrent application headers, for example, contain an info hash – a unique identifier "that is derived from the semantic description of the file".[57] The info hash is easy to find in an unencrypted BitTorrent header as it is clearly identified by its own header field, and once one has the info hash, a simple google search will identify the file being transferred.[58] As noted in a recent paper on the topic:

> It is also possible that future strategies for identifying file sharers may include deploying in-network hardware to inspect traffic and identify illicit content at an ISP gateway. By inspecting the BitTorrent header, it is possible to identify content by its info_hash header field. Specialized hardware could examine BitTorrent traffic for a set of illicit or copyright-protected info_hash values at line-rate within an ISP.[59]

[53] Email sent from Chris Donaldson to Tamir Israel on Mon, May 25, 2009 using Thunderbird. This is part of the application header for an email application called Thunderbird. Specifically, it is the MIME portion of the Simple Mail Transfer Protocol [SMTP] header for Thunderbird. While all SMTP headers contain an MIME message header, and most look similar to the one above, not all contain the subject line.
[54] RFC 821 - SMTP, *supra* note 34, at p. 4. While different Email applications may insert these fields in a different order, and their byte offsets could change on a message by message basis, these fields are part of the protocol and will generally be included in the 'header'.
[55] Report, *supra* note 1, at para. 5.
[56] *Weir*, *supra* note 41 at para. 73.
[57] K. Bauer, D. Grunwald, and D. Sicker, *The Arms Race in P2P*, 37th Research Conference on Communication, Information and Internet Policy, September 2009, available online at: <http://systems.cs.colorado.edu/~bauerk/papers/bauer_tprc2009.pdf>, at p. 4.
[58] See 'Screenshot' – Attachment 2, taken on October 12, 2009, of a google search for a hash tag. In this case, the sample info hash is: "218c2bb02979399d0149163dd4c74e35c31fff32" and the file would be "CRTC Hearing 2009-07-09 - Internet Traffic Management[tcpubmedia]".
[59] Bauer, *et. al.*, *supra* note 57 at p. 9.

Such 'information' is, again, highly personal and can reveal movie viewing trends, reading preferences, etc. It can also, as noted above, lead to legal liability. It is 'user-generated', in the same way that an email subject line is 'user-generated', in that the user is telling the application which file she wishes to view. Yet it is contained in the 'application header'.

Headers can contain much more. Krishnamurthy and Wills recently demonstrated how, through HTTP headers used by social networking sites such as Facebook and MySpace, third parties are able to get personal information of users and build individual profiles for advertising purposes.[60] Their study found that HTTP headers from SNSs generally contain unique identifiers which are collected by third party aggregators in order to track user activity and create profiles for advertising purposes.[61] The study examined 12 SNSs and concludes that:

> In all, we observed [SNS] id being leaked to a third-party server via one of these ways for 11 of the 12 [SNS]s. Such leakage allows the third-party to merge the OSN id with the profile of tracking information maintained by them.[62]

More troubling, there appears to be consistency across different SNSs, with users enjoying the same internal identifier in Facebook as in MySpace.[63] Such internal unique identifiers are personal information and should be considered 'user-generated' data. Yet this study demonstrates how they are commonly contained in HTTP 'application' headers, either in the Referer header portion of the HTTP header, or through cookies or Request-URI, which are also contained in HTTP headers.

Even more troubling, however, the study also discovered that HTTP headers can include personal information such as user age, gender, email address and zip code – *all* contained in the HTTP application layer control information (the 'application header'):

Figure 3 – Personal Information Contained in HTTP Application Layer Control Data (So-Called Application Header)[64]

---

[60] B. Krishnamurthy and C.E. Wills, *On the Leakage of Personally Identifiable Information Via Online Social Networks*, ACM SIGCOMM 2009 Workshop on Online Social Networks (WOSN '09), August 17, 2009, available online at: <http://conferences.sigcomm.org/sigcomm/2009/workshops/wosn/papers/p7.pdf>.

[61] *Ibid.* at pp. 9-10.

[62] *Ibid.* at p. 10.

[63] *Ibid.* at p. 11:

> Across [SNS]s, once a third-party server is leaked PII information via one [SNS], it may then also be leaked information via another [SNS] to which the same user belongs. For example, the cookie for doubleclick.net shown in the examples of Figure 1 and 2a means that DoubleClick can link the PII from across both MySpace and Facebook. This linkage is important because it not only allows the aggregator to mine PII from more than one [SNS], but join this PII with the viewing behavior of this user.

Note that the cookie that provides DoubleClick with all this information is transported to its third party server over an ISP's network and cookies, with all this identification information, are contained in HTTP control information (so-called application headers).

[64] *Ibid.* at pp. 10-11. Figure 4 from the report is reproduced here for the sake of convenience.

All this information is transported from SNSs to third party servers, over ISP networks, *in the HTTP header*, meaning they are part of the control information used by these applications and are in the application header. Nor are such concerns limited to SNSs. Here is a portion of the HTTP header sent from www.cbc.ca/news to CIPPIC's own computer when we visit:

> at=u=10328408&a=**anonymous_post**&e=**tamir.ii@gmail.com**&t=1255803232&h=a2e97aad86e9e186b5113cbc3111292d&pd=VXNlcm5hbWU9YW5vbnltb3VzX3Bvc3QmRmlyc3QgTmFtZT1UYW1pciZMYXN0IE5hbWU9SXNyYWVsJlByb3ZpbmNlPU9udGFyaW8mQ2l0eT1PdHRhd2EmVXNlclR5cGU9cGx1Y2s%3d; pd=**Username=anonymous_post**&**First Name=Tamir**&**Last Name=Israel**&**Province=Ontario**&**City=Ottawa**&UserType=pluck

Figure 4 – Excerpt from the Cookie portion of the HTTP 'Application' Header sent from CBC.ca[65]

Not that this cookie includes my pseudonym (anonymous_post), my full name, my city and province of residence, as well as my email address. Meanwhile, I comment on CBC news articles thinking only my pseudonym will be exposed. HTTP control information can include more. Here, for example, is a portion of the HTTP control information (HTTP request header) sent from my browser to bing.com when I enter the term 'secret' into its search bar:

> GET: **/search?q=secret**&form=QBLH&filt=all&qs=n HTTP/1.1
> Referer: **http://www.bing.com/**
> Accept-Language: en-us
> Accept-Encoding: gzip, deflate
> User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; InfoPath.1)
> Host: **www.bing.com**

Figure 5 – Excerpt from HTTP Request 'Application' Header sent from bing.com[66]

Here we see that the HTTP header sent to a search engine when making a search query can include the host site to which the query is being made, the Referer site from which the query was sent, as well as, of course, the search query itself ("q=secret").

The contention that user-generated data is not contained in so-called application headers appears difficult to uphold. The OPC website provides a non exclusive list of the type of user-generated data typically contained in cookies, small text files used by websites to track information so as to maintain state:

- the Internet Protocol (IP) address of your computer;
- how many times you have visited the site;
- your preferences, such as a preferred language;
- your user name and password;
- items in your "shopping cart";

---

[65] Taken on October 17, 2009, my emphasis. When logging in, CBC.ca offers users the option of staying logged in for two weeks. This creates a persistent cookie, a portion of which is excerpted above, which contains the Username, Real name (first and last), Province and City of residence, and email address of the user. This is especially troubling as many users believe they are posting comments on news articles anonymously.

[66] Taken on October 19, 2009, my emphasis. For the sake of comparison, here is the 'GET' line from the header sent when googling the word 'revealed', taken the same day:

GET: /search?hl=en&**q=revealed**&aq=f&oq=&aqi=g10 HTTP/1.1
Referer: http://**www.google.com**/search?source=ig&hl=en&rlz=&**q=secret**&aq=f&oq=&aqi=g10
Host: **www.google.com**

Note that in both, the search query itself ('revealed') appears in the HTTP header, as does the host to which the request was sent (www.google.ca). Note as well that the header includes the Referer – the website from which the query was entered. In the Bing example, this is www.bing.com. But the Google query ('revealed') was sent from a page displaying the result of a previously executed search (for 'secret'). This, too, is displayed in the Referer line (http://www.google.com/search...**q=secret**), as the search query is part of the URL of the initial result page from which I entered my second search query ('revealed').

- Web sites you've visited;
- any information such as your name; and
- any unique alphanumeric character string that can be linked to your personal information[67]

But cookies, as well as any data contained within them, are transported from the user's computer to the website's server in the HTTP header and so form part of the application layer control information ('so-called application header').[68] Users generate their own user names, and provide websites with their email addresses, names and places of residence. The terms 'user-generated' and 'actual content' apply as readily to such data as to any other data provided by a user to a website.

Due to the lack of standardization with respect to application layer control data, other applications with a communications component may include even more sensitive and personal user-generated data. There is simply no way to tell, as any application developer can decide for themselves what to put in its header and any web-based application can decide what to put into a cookie. It is important, then, to be clear about what can be contained in a 'header', especially as we enter the world of cloud computing, where more and more applications will have a communications component with accompanying 'application headers'. Lack of such clarity can lead to further and more material misunderstandings as well, as highlighted in the next section.

Therefore, the distinction between user-generated information and control information/headers found above is inaccurate. The "content" is *not* "the user-generated information (such as email content) that is surrounded by several layers of control information, known as headers, to ensure proper handling and routing."[69] Both user-generated information and information that could only be referred to as content will often be found in the control information/headers used by the application layer (though not, of course, that used by *other* OSI layers) to ensure proper handling. We ask again that these inaccurate portions of the Report be removed or corrected to reflect the above.

C. Fishing for Headers v. Fishing for Content

The Report states the following:

> Information that is transmitted via the Internet is broken down into packets, routed to its destination and reassembled into the original content. The content is the user-generated information (such as email content) that is surrounded by several layers of control information, known as headers, to ensure proper handling and routing. DPI has the capability to inspect these many protocol headers…[70]
>
> […]
>
> Rogers confirmed that the DPI technology it is using is not capable of any of the following:
>
> - using any personal information of an individual user;
> - storing or logging any personally identifiable information;
> - gaining knowledge of a user's URL browsing history;
> - gaining knowledge of a user's Internet search activity;
> - gaining knowledge of a user's email topics or content;
> - storing content accessed by a user;

[67] Office of the Privacy Commissioner, *Protecting Your Privacy on the Internet*, www.privcom.gc.ca, last modified July 25, 2004, online at: <http://www.priv.gc.ca/fs-fi/02_05_d_13_e.cfm> (accessed October 12, 2009).
[68] D. Kristol, L. Montulli, *RFC 2109: HTTP State Management Mechanism*, February 1997, available online at: <http://tools.ietf.org/html/rfc2109>. "This document specifies a way to create a stateful session with HTTP requests and responses. It describes two new headers, Cookie and Set-Cookie, which carry state information between participating origin servers and user agents."
[69] Report, *supra* note 1, at para. 5-6.
[70] *Ibid.*, at para. 5-6, my emphasis.

- storing in a cache any content, including user-specific content;[71]

[…]

…Rogers informed that the data is fragmented when transmitted and that the packets are very small–each as little as 8 characters. These characters could be notes in a song, pixels in an image, letters in an email or control characters. The packets would have to be re-assembled at their destinations in order for them to be meaningfully interpreted.

The answer to the question of whether DPI examines specific packet content depends on which product is being used. All DPI products have the ability to look within the application header of a packet or traffic stream, and make decisions based on that information, without examining the actual content. While it is possible to examine the content of packets, the DPI technology currently deployed by Rogers does not have the ability to do so.[72]

Perhaps as a result of the confusion sown with respect to distinctions between user-generated data or content on the one hand and control information or headers on the other, the Report also erroneously creates a false distinction between the capacity of DPI to read one type of information (headers) and the other (user-generated 'content'). This conclusion is wrong on two fronts. First, as demonstrated in Section B above, the application layer control information ("application header") will often contain user-generated data or 'actual content'. Looking within this control information to find 'actual content' is no different than looking into it to find anything else. Second, as demonstrated in Section A above, application layer control data is contained in the payload, alongside the rest of the packet content. Looking within this control information is not qualitatively different from looking into the rest of the payload. Thus, while Rogers' DPI equipment may not currently be *configured* to look at this information, it is nonetheless *capable* of doing so. The capacity of the equipment should be clearly differentiated from its current uses so that potential for function creep and for use of such devices for lawful access purposes is not underestimated. Before analyzing DPI's ability to access 'content', it is helpful to review how DPI works. It is important to keep in mind that, since CIPPIC does not know the specific capabilities and architectural placement of Rogers' DPI, the following discussion focuses on *basic* DPI functions alone. Depending on the equipment in question and on where in the network it is located, DPI can do *much* more than what is described below – it can and has "collect[ed] and use[d] web pages viewed by users, web search terms, the amount of time spent at web sites, response to advertisements, and postal codes."[73]

DPI: Capacities and Limitations

The goal of DPI is to identify specific applications. To do so, DPI network equipment operates inline and analyzes all network traffic as it passes, looking for application signatures (stored in a 'signature library') indicative of the applications it is attempting to identify.[74] Signature analysis once referred exclusively to the process of scanning packets for unique strings of data.[75] Today, signatures are composed of data necessary to identify certain applications through port analysis; analysis of numerical properties such as payload length, the number of packets in a flow or stream, or the numerical offset of a fixed string or byte value within a packet; analysis by heuristics; as well as analysis by string match.[76]

[71] *Ibid.*, at para. 23, my emphasis.
[72] *Ibid.*, at paras. 24-25, my emphasis.
[73] J. Lo, *A "Do Not Track List" for Canada?*, Public Interest Advocacy Centre, 2009, DRAFT copy, to be published, section 3.3 Advertising using information from internet service providers and Deep Packet Inspection, at p. 29. Lo notes that many privacy advocates have described DPI capacities as akin to wiretapping.
[74] Allot Whitepaper, *supra* note 29, at p. 3.
[75] M. Tanase, *The Great IDS Debate: Signature Analysis Versus Protocol Analysis*, Security Focus, February 5, 2003, available online at: <http://www.securityfocus.com/infocus/1663>.
[76] Allot Whitepaper, *supra* note 29, at p. 3.

Port analysis is a shallow packet inspection technique that looks within layer 4 control information to identify the target port that is identified therein.[77] Numerical analysis examines a few features: payload length, declared in the *packet* (or datagram) header, can be determined without reading the application layer control information;[78] the number of packets in a flow or stream can additionally be counted without reading application layer control information. Heuristics analysis involves more sophisticated analysis of features such as whether a given flow changes protocol (UDP to TCP) part way through a communications interchange while maintaining all other flow features, or how packet lengths change over time.[79] Note that, up until this point, none of the techniques mentioned look within the payload (whether at application layer control information or anything else), and so it is difficult to say that any of these techniques involve 'deep' inspection.

Other techniques such as string analysis and numerical string offset require analysis of the application layer control information. These techniques search the payload for specific strings of data associated with a particular application in order to identify it.[80] A common example of this process is Kazaa, a file sharing application which includes the word "kazaa" in the 'user-agent' field of its 'application header'.[81] Other applications such as Thunderbird or Outlook similarly identify themselves in the 'user-agent' field of their SMTP headers.[82] Bell, in describing its own DPI, uses the example of "BitTorrent" to demonstrate how it reads packets to identify signature strings.[83] String identifiers can be far more sophisticated as well, and can include seemingly random combinations of numbers and letters, and even sequences of sub-strings.[84] A DPI patent application describes the process as such:

> Analysis by string match involves searching for a sequence (or string) of textual characters or numeric values within the contents of a packet. Furthermore, string matches may include several strings distributed within a packet or several packets. For example, many applications still declare their names within the protocol itself, as in Kazaa, where the string "Kazaa" can be found in the User-Agent field with a typical HTTP GET request.[85]

DPI boxes contain signature libraries, and these include an often large collection of strings or sequences of strings for which passing packets are scanned.[86]

---

[77] Bell, CAIP submission, *supra* note 33, Figure 19; see also Section A, *supra*, notes 28-29 and accompanying text.
[78] Reed, *supra* note 3, Attachment 1, at p. 2. Transport layer headers (such as for TCP) also announce the header and payload size: RFC 793 – TCP, *supra* note 34, at p. 16: "This pseudo header contains the Source Address, the Destination Address, the Protocol, and TCP length."
[79] Allot Whitepaper, *supra* note 29, at p. 7.
[80] S. Dharmapurikar and J. Lockwood, "Fast and Scalable Pattern Matching for Network Intrusion Detection Systems", (2006) 24(10) *IEEE Journal on Selected Areas in Communications*, available online at: <http://www.arl.wustl.edu/~sarang/jsac_cameraready.pdf>, point out at p. 1.
[81] See Heavy Reading, *supra* note 31, at p. 8. The Heavy Reading report states that DPI examines the "packet payload itself…to look for strings in the protocol that identify it (eg "kazaa, which appears in one of the fields used to handle Kazaa requests). See also, B.C. Park, Y.J. Won, M-S. Kim, and J.W. Hong, "Towards Automated Signature Generation for Traffic Identification", (2008) *Network Operations and Management Symposium, 2008, NOMS 2008, IEEE* 160, available online at: <http://dpnm.postech.ac.kr/papers/NOMS/08/signature.pdf>, at p. 167.
[82] See Figure 2, above, for an example of a portion of a Thunderbird 'application header'. The portion of the header that identifies Outlook as the sending application is located in the Outlook-produced MIME portion of the SMTP Email 'application' header and looks like this:

> **X-Mailer**: Microsoft Office Outlook 11
> **X-MimeOLE**: Produced By Microsoft MimeOLE V6.00.2900.3350

[83] Bell, CAIP submission, *supra* note 33, Figure 19. Park *et. al.*, *supra* note 81 at p. 164, Table I, describe the BitTorrent signature string more precisely as "0x13BitTorrent protocol".
[84] See Park *et. al.*, *supra* note 81. Table III provides a few examples of application signatures of this type.
[85] J.M. Heinz and A.N. Ray, *System, method and apparatus for prioritizing network traffic using deep packet inspection (DPI) and centralized network controller*, Patent Application, 2009, available online at: <http://www.faqs.org/patents/app/20090238071>, at para. 0018.
[86] See, for example, Allot Communications, *Applications and Protocols Supported by Allot DPI*, September 30, 2009, available online at: <http://www.allot.com/index.php?option=com_docman&task=doc_view&gid=25> for a list of the many applications and protocols its DPI can identify through their signatures, most of which will contain unique strings. Some CISCO equipment comes with 1000 signatures built in, and the capacity to add many more: CISCO, "Policies—Signature Definitions", in *Installing*

Many DPI boxes today also have the capacity to analyze previously unknown traffic flows so as to generate *new* signatures and string sequences and expand their libraries, but most signatures strings are added to the library manually.[87] The signature string generation process does not concern us here, only the fact that one of the key functions[88] of DPI equipment is to read packets as they pass in order to identify specific strings of data associated with particular applications and stored in its signature library.

The capacity to read passing packets for strings and match it against a list such as a signature library is not unique to DPI. Most network elements have the capacity to read passing packets for strings of data.[89] Even a basic router must read the IP layer control information for relevant strings of data (such as IP address) and act on them.[90] This method of packet inspection scales poorly and requires a great deal of resources if done inline.[91] However, since *packet* (as opposed to application layer) headers have precise byte offsets, this process is greatly simplified when scanning for *network* information. The task for DPI is more challenging, is it does not know where in the packet to look for its signature strings. Further, while most network equipment simply reads existing information at a particular byte offset (IP address) and acts on it, DPI must scans for a large signature string base – one or more fore each version of each application it is attempting to identify.[92] DPI boxes, as opposed to regular network elements such as routers and switches, must be able to analyze, classify and respond to a large and changing number of application signatures.[93] What enables DPI to do so, what truly sets it apart from other equipment, is its capacity for analysis, its configurability, and its capacity for retaining data.

---

and Using CISCO Intrusion Prevention System Device Manager 6.0, CISCO Systems Inc., 2009, available online at: <http://www.cisco.com/en/US/docs/security/ips/6.0/configuration/guide/idm/dmSigDef.html>. DPI boxes can contain signature libraries with signature stings in the order of a few thousand: Dharmapurikar *et. al., supra* note 80 at p. 1.

[87] N. Anderson, *Deep Packet Inspection Meets 'Net Neutrality', CALEA*, Ars Technica, July 25, 2007, available online at: <http://arstechnica.com/hardware/news/2007/07/Deep-packet-inspection-meets-net-neutrality.ars> notes at p. 2 that, much like virus scanners, "DPI gear needs regular updates to stay on top of new developments." See also, Allot Whitepaper, *supra* note 29, at p. 4 and Park *et. al., supra* note 81 at p. 160, where the authors state that, predominantly, signature strings "have been manually extracted by the network administrators or security experts. It requires preceding protocol semantic analysis or empirical packet payload inspection for pattern recognition. Human decision in such process [sic.] causes a slow response time to deal with new applications."

[88] Dharmapurikar, *et. al., supra* note 80, at p.1 note that "[o]ne of the most frequently performed operation [sic.] in [DPI] applications is searching for predefined patterns in the packet payload."

[89] Bell, CAIP submission, *supra* note 33, at para. 205: In describing the 'theoretical' capacity of its DPI boxes to filter Internet traffic so as to isolate specific IP addresses (that is, to place a specific IP address in its signature library and identify all passing traffic that contains that IP address in its packet header) Bell states the following:

> Additionally, this type of filter could likely be placed on any network element and is therefore not unique to DPI technology. Thus, there is nothing unique about the theoretical capability of DPI technology for this particular task.

[90] F. Hao *et. al.*, "Fast Dynamic Multiset Membership Testing Using Combinational Bloom Filters", (2009) *IEE INFOCOM 2009*, available online at: <http://www.arl.wustl.edu/~hs1/publication/bloomset5.pdf>, at p. 1, describe how Layer-2 routers will typically scan the packet header for its destination address and then query its MAC table to find which port the packet should be sent to help it along its way.

[91] Y. Gong, *Identifying P2P Users Using Traffic Analysis*, Security Focus, July 21, 2005, available online at: <http://www.securityfocus.com/infocus/1843> describes the drawbacks of string analysis as such: "Signature [string]-based identification at the application level (L7) is also highly resource-intensive. The higher bandwidth [sic.] network, the more cost and resources you need to inspect it."

[92] *Ibid*. See also Allot Whitepaper, *supra* note 29, at p. 4.

[93] R.E. Jurga and M.M. Hulbój, "Technical Report: Packet Sampling for Network Monitoring", (2007) CERN – HP Procurve openlab project, 2007, available online at: <http://openlab-mu-internal.web.cern.ch/openlab-mu-internal/Documents/2_Technical_Documents/Technical_Reports/2007/RJ-MM_SamplingReport.pdf>, at p. 1 state this plainly:

> Switches and routers have limited computing power and resources. Their main purpose is definitely not packet monitoring…Thanks to the progress in massively parallel processing within specialized FPGA circuits, the [sic.] Deep Packet Inspection attracted the researchers and hardware manufacturers [as a packet monitoring mechanism].

A key feature of DPI equipment is its internal Central Processing Unit (CPU), which gives it the capacity to analyze greater amounts of traffic inline, without causing significant latency.[94] While some approaches to packet inspection may attempt to rely on hardware such as application-specific integrated circuits (ASICs) in order to facilitate greater inline analysis, even without such dedicated circuits today's network processors (CPUs) can deliver advanced inline packet analysis at multi-gigabit speeds.[95] This means that DPI can analyze more packet information faster and thus has a higher capacity for analysis then other network elements. This analytical capacity is essential to Deep Packet Inspection. While packet header fields (such as those identifying source or destination IP address) have specific byte offsets, application layer control information fields (such as those specifying source or destination email address or URL) do not. This added analytical capacity allows DPI to read packet payload information at inline speeds while other network elements would not be able to do this effectively.[96]

Another key feature of DPI equipment is its configurability. Relying on CPU instead of hardware like ASICs means that updates or changes to what the DPI inspects is a simple matter of upgrading software.[97] Indeed, as noted above, most DPI boxes are configured so that new application signatures (including signature strings) can be added simply by updating the signature library.[98] In addition, the rules that govern what occurs once a packet flow is associated with a given signature are also software, as opposed to hardware, based and so can be changed to a certain extent without the need for expensive and difficult hardware updates. This flexibility in configuration is an essential feature of DPI given the rapid rate at which new applications develop today.

Finally, DPI's capacity for retaining data also separates it from other types of network equipment. Other network equipment will typically have statistics counters, which log each time a particular event occurs.[99] Such events can include the number of packets that use a particular prefix, or the number of packets in each specific TCP stream, for example.[100] DPI equipment will have similar statistics counters, to log each time a particular signature in its signature library passes through it.[101] However, while most network equipment must track hits that can number in the tens of millions per second,[102] DPI signatures would exhibit much fewer hits. This is so for a number of factors. For one, DPI is limited to tracking hits on signature strings in its library (which could number in the thousands, but likely not higher).[103] More importantly, however, DPI typically operates on the principle of flow control, meaning that logged signature string hits would be per *flow* of data, not per *packet*, as is the case with regular routing equipment and, of those flows, only ones that match signature hits are recorded, meaning resources

---

[94] Bell, CAIP submission, *supra* note 33, at para. 193. Other network elements can also have CPUs, but DPI, because of its need for advanced and detailed analysis, will have faster processors capable of a greater degree of analysis.

[95] J. Tollet, "Myth 5: High Throughput Requires Dedicated ASICs", in *7 Myths of IP Networking*, dpacket.org, September 22, 2008, [Tollet Myth 5] online at: <https://www.dpacket.org/articles/myth-5-high-throughput-requires-dedicated-asics>.

[96] Jurga and Hulbój *supra* note 93, at p. 6:

> The most important parts of DPI are regular expression matching and signature based scanning. In this technique the payload of all the packets is checked against the set of known malicious signatures at the wire speed…In case of Deep Packet Inspection, it is often necessary to match the patterns at every byte offset…Header processing is much simpler, as header location within the packet is predefined.

[97] Tollet Myth 5, *supra* note 95.

[98] Allot, *supra* note 86.

[99] D. Shah *et. al.*, "Maintaining Statistics Counters in Router Line Cards", (2002) 22(1) *IEEE Micro* 76, available online at: <http://klamath.stanford.edu/~nickm/papers/IEEE-Micro02.pdf>.

[100] *Ibid*.

[101] Bell, CAIP submission, *supra* note 33, at p. 70, footnote 116:

> The DPI simply stores counters referencing the amount of signatures being hit during hourly intervals. This "signature hit" is stored by hour per DPI, similar to how routers and switches can store information of how much traffic is passing through their interface.

[102] Hao, *et. al.*, *supra* note 90, at p. 1, point out that, at typical line speeds of 10 Gbps, a network element will need to process up to 30 million packets per port per second. Each of these will require the equipment to query tables to so they know how to route it. Shah *et. al.*, *supra* note 99 at p. 77 state that each and every one of these individual packets may also require triggering multiple counters, again at inline speeds.

[103] Dharmapurikar, *supra* note 80, at p. 1.

required will be lower by as much as an order of magnitude.[104] This means that the DPI would need to store considerably fewer statistical hits, as each flow will usually generate only a single hit. This in turn increases the number of signature strings that DPI statistical counters can potentially track, if configured to do so.

It is also important to note the limitations inherent in DPI equipment. There are limits on how much analysis can be performed inline, as such analysis increases latency and scales poorly. In addition, the DPI equipment itself has a relatively limited amount of onboard database space, meaning it can only *store* a finite amount of content. The implications of these two limitations are that, first, a signature string library can be only so large as the more signatures the equipment must look for, the greater the cost in latency. Second, it is impractical for DPI to scan every single packet that passes through an ISP's network for all the signatures in question.[105] This means that, typically, DPI must operate on flow control, where it only scans the first few packets of a flow for the signature strings it needs and then classifies the entire flow.[106] Most of the 'content' of a telecommunication message will often (but not always) be contained in later packets which DPI equipment is not typically configured to scan for traffic management purposes. Finally, while adding new signature strings is a relatively simple process, adding to the collection of rules that govern how these signature strings are applied and what occurs after a match is found may be more complex, depending on the operation in question.

With these capabilities and limitations in mind, then, it is now possible to assess what user-generated data DPI has the capacity to 'gain knowledge' of. Because of terminology adopted by some ISPs, and particularly by Bell, in their descriptions of DPI capacities, it is useful to address capacity for gaining knowledge of content included in the application layer control information distinctly from that contained in the rest of the packet payload. In its submissions, which are the most expansive discussion of the operation and limitations of DPI available from a Canadian ISP, Bell often creates a false dichotomy between "content" or "payload" and packet/protocol headers or control information.[107] For this reason, the next section will examine how and what user-generated content DPI has the capacity to gain knowledge of from application layer control information, and the following section will look at information contained in the rest of the packet payload.

DPI: What it can do

[104] Jurga and Hulbój *supra* note 93, at p. 8 describe the impact that focusing on a limited number of flows can have on sampling requirements. Bell, (CAIP submission, *supra* note 33) describes this flow classification process in detail at paras. 193-195. It refers to the process as 'classify once; switch many', meaning only the first few packets of a flow are searched for signature strings in the signature library, and an entire flow will typically generate only a single 'signature hit' to log in the DPI's statistics counter. Park *et. al.*,(*supra* note 81 at p. 162) describe a packet flow as a "collection of packets that share the identical source IP, destination IP, source port, destination port, and protocol."

[105] Park *et. al., supra* note 81 point out at p. 162 that it would be costly, unnecessary, and "virtually impossible" to conduct the complex string *detection* analysis they advocate in their paper on each and every packet inline. The type of string analysis that we are discussing here, mere string *matching* to a given and well established pre-determined list, is far less resource intensive, yet still has limits.

[106] Bell, CAIP submission, *supra* note 33, at paras. 193-195. To be accurate, DPI will analyze more than the first few packets in a flow, but it will only scan the first few for signature strings. Other DPI functions, such as numerical analysis of packet size or packet length histogram, require a larger number of packets before sufficient data is gathered to classify a flow. Park *et. al., supra* note 33, at p. 162 note that P2P packet size – a telling feature, is only telling after the 'first few packets' initiate the connection, when a file transfer commences and packet sizes increase. These first few packets will, however, contain the information necessary for a signature string match (i.e. user-agent "kazaa"). Also, signature string analysis is more intensive than these other types of analysis, so applying it only to the first few packets is practical (*Ibid*.).

[107] See Bell, CAIP submission, *supra* note 33, at para. 183 for one example of how it employs terms such as 'actual content' and 'protocol' or 'packet' 'headers':

> In short, the Company's use of DPI technology as part of its traffic management practices is such that the actual contents of the communication exchange are not examined. Rather, only the protocol headers are examined, and as described in detail below, the DPI equipment does not retain the information reviewed in the packet headers.

All DPI is capable of scanning for signature strings in the packet payload. This is one of the core functions of 'DPI', and what defines it as such.[108] Rogers' DPI, according to its representations, is currently configured to scan only for signature strings that are indicative of certain applications. Arguably, this is already "examining the actual content" of packet flows in order to "make decisions based on that information".[109] As mentioned in sections A and B, application layer control information is firmly ensconced in the payload of a telecommunications packet and is akin to 'opening the envelope' in order to determine the *nature* of its contents. DPI will attempt to do so, even if the 'payload' (including 'application header') is encrypted.[110] This amounts to *examining the contents* of the mail, and is no different from examining the payload of a telecommunications packet flow to identify which application a person has chosen to use. But what else can DPI do?

We have seen that application layer control information can and often does contain a great deal of information that can only be classified as 'user-generated' content. Is it more difficult for DPI to 'gain knowledge' of such information than it is for it to gain knowledge of information such as application specific signature strings?

In its submissions to Telecom Decision CRTC 2008-108, Bell often states that it would be operationally and technically infeasible for its DPI equipment to find particular strings in the packet payload, relying on the same limitations mentioned above as explanation for this infeasibility.[111] However, when discussing the *capacity* for its equipment to gather such information, Bell makes the following statement:

> While the Company has explained in detail what its DPI is capable of doing, it is helpful to summarize what they do not do. By design, the DPI devices deployed in Bell Canada's network **do not**:
>
> - use any personal identification information of an individual user;
> - store or log any personally identifiable information;
> - have specific knowledge of a user's real identity;
> - have knowledge of a user's content;
> - have knowledge of a user's URL browsing history;
> - have knowledge of a user's Internet search activity;
> - have knowledge of a user's email topics or content;
> - store content accessed by a user;
> - cache any content, including user-specific content, whatsoever;
> - capture and playback any communications exchange; or
> - install or require any specific software on user machines.[112]

CIPPIC notes that this list essentially mirrors that "confirmed" in the Report as activities Rogers' DPI is not capable of.[113] But Bell does not state that its DPI is not *capable* of all these functions. It merely states that it is not currently configured to do these, and that it would be impractical to do so on a large scale. In fact, the OPC's finding in a similar complaint against Bell states that its DPI *does* collect and use personal identification information such as IP addresses.[114] Indeed, Bell plainly points out in its

---

[108] Ipoque Whitepaper, *supra* note 2, at p. 1. Dharmapurikar, *et. al.*, *supra* note 80, at p.1 also point out that "[o]ne of the most frequently performed operation [sic.] in [DPI] applications is searching for predefined patterns in the packet payload."
[109] Report, *supra* note 1, at para. 25.
[110] Bauer, *et. al.*, *supra*, note 57, at p. 10 and following. Ipoque Whitepaper, *supra* note 2, at p. 3 also describes how most DPI will attempt to scan encrypted packets.
[111] Bell, CAIP submission, *supra* note 33, at para. 191.
[112] Bell, CAIP submission, *supra* note 33, at para. 208.
[113] Report, *supra* note 1, at para. 23.
[114] OPC, *Report of Findings: Bell's Use of DPI*, File #6100-03063 [Bell Finding], at para. 68. Bell describes this collection process. DPI equipment creates a unique (and temporary) flow entry for each flow that passes through it so that it can apply one specific action to that entire flow once it has classified it using signature analysis (Bell, CAIP submission, *supra* note 33, at para. 193). A packet flow is defined as a "collection of packets that share the identical source IP, destination IP, source port, destination port, and protocol" (Park *et. al.*, *supra* note 81, at p. 162). Each flow entry, then, collects the source and destination

descriptions of its standard DPI equipment that it, in theory, "a signature could be used to find a particular string in the packet", even without knowing the specific byte offset at which that string is located.[115]

DPI equipment can be configured to gain knowledge of control layer user-generated content without difficulty through basic DPI signature detection/filtering techniques. The statistical counters standard in any DPI (and many other network elements) can be combined with the analytical capacities of DPI to gain a significant amount of knowledge about the telecommunications messages passing through it by setting signature strings such as 'www.bell.ca', and then logging signature hits in association with customer identifiers such as IP addresses. An examination of how this mechanism might operate with respect to browsing history might be instructive.

If one were looking for browsing history, it might be difficult to capture a record of *all* browsing history of *all* users with DPI alone. It would likely require more data storage space than Rogers' DPI has to store all that data.[116] However, URLs are contained in the HTTP (or 'application layer') control information of a packet[117] and since DPI can parse such control information, it should not have difficulty gaining knowledge of destination URLs. Indeed, it is quite standard for DPI equipment to 'collect' both source customer IP addresses[118] *and* destination URL.[119] Collecting the latter at inline speeds is a commonly utilized function of DPI, which is capable of finding such data even without a specific byte offset.[120] Moreover, URL hits are typically stored by the DPI, albeit in aggregate and anonymized format, on the DPI's statistical counters, Bell tells us.[121] Every time the DPI reads a new URL, it sets up a counter for it, which is then triggered every time the same URL passes through the DPI. Is DPI capable of logging these destination URL 'hits' in a manner that is associated with the IP addresses that generated them? It should not be difficult to do so, at least on a limited scale.

Indeed, DPI equipment already logs, in certain instances, information on statistical counters that *is* associated with specific customer identifiers. Bell, for example, states that its DPI purposes are twofold, in that it:

1. allows customer usage data collection functionality for usage billing; and
2. allows for traffic shaping of P2P file sharing applications during peak periods as an additional measure to address congestion.[122]

We see here that standard DPI such as Bell's is capable of logging hits on its standard statistical counters (bandwidth usage hits, in this case) in a manner that is associated with the IP addresses of consumers.[123]

IP address of that packet flow (Bell Finding, at para. 22). While, purportedly, these flow entries are retained only for the duration of the flow, they nonetheless qualify as 'collection' and 'use' of IP addresses.

[115] Bell, CAIP submission, *supra* note 33, at para. 191. Bell goes on to point out the perceived limitations of this capacity.

[116] Although it would *not* be difficult to develop a hybrid approach wherein the DPI collects this information and stores it elsewhere: see Tollet Myth 7, *supra* note 5.

[117] See Figure 5 above for an example of an HTTP/1.1 request header. Such headers MUST include a Host header field, which MUST include the name of the URL (www.bell.ca for any pages on the Bell site): IETF – Network Working Group, *RFC 2616 – Hypertext Transfer Protocol – HTTP/1.1*, IETF, June 1999, available online at: <http://tools.ietf.org/html/rfc2616>, p. 127, section 14.23 Host.

[118] Bell, CAIP submission, *supra* note 33, at para. 195 and in figure 20. See also the Bell Finding, *supra* note 114, at paras. 22-23.

[119] Bell, CAIP submission, *supra* note 33, at para. 200 explains that one of the basic functions of DPI equipment is to collect destination URLs from passing packet flows. Bell's DPI is configured to log each URL hit and to produce aggregate (anonymized) reports for the purpose of determining which URLs are getting the most 'hits'.

[120] Dharmapurik *et. al., supra* note 80, at p. 1 describe how other network elements now utilize DPI to parse HTTP headers at inline speeds and make traffic routing (as opposed to traffic shaping) decisions on that knowledge.

[121] Bell, CAIP submission, *supra* note 33, at footnote 116, p. 70.

[122] S. Morin, "Deep Packet Inspection: Part of Bell's Ongoing Network Management", Presentation in Victoria, B.C., February 2009, available online at:
<http://www.cio.gov.bc.ca/services/security/library/conferences/2009/Presentations/Morin_Suzanne_Bell-DeepPacketInspection.pdf> on slide 13, my emphasis.

In order to log unique URL signature string hits (www.bell.ca) with individual customers, no other capacity is required. Rogers need only enter in the new string into its DPI signature library (www.bell.ca), and apply the same rule it applies to is customer-specific usage based billing functionality.

While it is possible to track customer usage without DPI equipment, since usage (amount of bandwidth) does not require inspection of the packet payload but only a statistical counter associated with each customer, DPI enhances the timeliness and precision of these measurements.[124] But given the versatility of DPI equipment – the fact that it has statistical counters measuring *other* pieces of information (such as signature hits, currently indicating which application or protocol is being used), the capacity to match statistical counter hits to IP addresses and later "resolve" these IP addresses to "unique identifiers"[125] could be used, without too much difficulty, to gain a great deal of knowledge about customers.

In fact, we know that Canadian ISPs (including Bell, and Rogers) who have implemented DPI in their networks already operate their equipment in a manner that allows them to associate signature string hits (indicating applications or protocols used) with specific customers. These ISPs, as opposed to others that do not have DPI, are able to provide statistics demonstrating protocol and application -specific traffic breakdown for specific subsets of customers. Thus, they can tell us that their top 5% bandwidth users are using HTTP traffic 21.7% of the time while using P2P traffic 57.4% of the time in a given month.[126] Their top 10% bandwidth users, on the other hand, allot 25.8% of the traffic they use to HTTP traffic and 54% to P2P traffic.[127] All customers together exchange about 39.5% of bandwidth as HTTP traffic and 36.25% as P2P traffic.[128] This means that, with basic DPI functions, an ISP can determine that x customer used y amount of bandwidth last month, and that her bandwidth distribution was z% HTTP and a% P2P. Since only DPI equipment is capable of distinguishing P2P traffic from regular HTTP traffic,[129] it must have the further capacity to associate this information with specific customers. Without this capacity, Bell and Rogers would not be able to generate these types of statistics.

In summary, DPI equipment is capable of gaining knowledge of signature hits ("kazaa"), of logging these signature hits on statistical counters,[130] of associating statistical counters with specific customers[131] and subsets of customers,[132] of reading and logging destination URLs from passing packet flows at inline speeds,[133] and of adding signature strings to its signature library that will cause it to scan for specific URLs such as www.bell.ca.[134] It should be a simple matter of configuration for Rogers to program its DPI to 'gain knowledge' of browsing activities of its users.

---

[123] Bell describes the process by which it resolves such associations to specific users at [xxx] para. 206. Since Rogers no longer has any Internet service plans without bandwidth usage caps, this means that it must track usage statistics for every individual user, which means its equipment has the capacity to count usage and associate it with a unique account identifier, and to do so at inline speeds: http://www.rogers.com/web/Rogers.portal?_nfpb=true&_pageLabel=INTER_LANDING (accessed November 2, 2009).

[124] Morin, *supra* note 122.

[125] Bell, CAIP submission, *supra* note 33, at para. 206.

[126] CRTC Letter, *supra* note 6, at chart (CRTC) 04Dec2008-2 (d).

[127] *Ibid*.

[128] *Ibid*., at chart (CRTC) 04Dec2008-1 (b).

[129] TELUS, *Response to Interrogatory*, PN 2008-19, TELUS(CRTC)4Dec08-1, ABRIDGED, page 5 of 6, January 13, 2009, available online at: <http://www.crtc.gc.ca/public/partvii/2008/8646/c12_200815400/1005778.zip> (ZIP) states that, do to its lack of DPI, it cannot identify any P2P traffic that is aimed at port 80, the standard HTTP port. This covers the vast majority of P2P traffic.

[130] Bell, CAIP submission, *supra* note 33, at footnote 116, p. 70.

[131] As they must do for customer usage billing.

[132] As they must do with application-specific usage data, seen from their ability to provide customer-specific statistics on this type of usage.

[133] Bell, CAIP submission, *supra* note 33, at para. 200.

[134] Heavy Reading, *supra* note 31, at p. 9; Allot Whitepaper, *supra* note 29, at p. 4, states this update feature as a basic and necessary function of all DPI.

It is important to note that the more signature strings of this type (as opposed to strings aimed at identifying applications) are added to the signature library, the less technically practical this operation becomes. As Bell correctly notes, the "management of such an enormous signature base would be operationally infeasible".[135] Without adding a system to regularly empty the logs to external storage, they would fill up.[136] The *extent* to which Rogers' DPI is capable of recording browser histories of individual customers is certainly dependent on the specific equipment Rogers is employing on its network and where on that network it is deployed. But *any* DPI equipment has the *capacity* to gain knowledge of some browsing history in the manner described above. This is because gaining such knowledge requires little more than a reconfiguration of basic functions common to all DPI equipment. In addition, given that statistical counters are typically designed to handle hits on a far greater order than what is generated by current DPI equipment,[137] there is certainly room for growth in the utilization of DPI to discover URLs, were an ISP inclined to do so. At the least, it would be a fairly simple thing to gain knowledge of which Rogers customers were visiting www.bell.ca.

The same technique could be implemented to 'gain knowledge' of user Internet search activity. Again, it would be infeasible for a DPI box alone to track *all* search queries entered by *all* users without external storage. But say Rogers wanted to know how many of their customers were searching for the terms 'iPhone', 'provider' and 'Canada'. Well, you could add a new signature. This one would register a 'hit' when finding all of the following three data strings:

```
HTTP/1.1
Host: www.google.ca
GET /search?hl=en&q=iphone+provider+Canada
```

Figure 6 – Excerpts from HTTP 'Application' Header sent to Google[138]

With this signature, Rogers could capture roughly 80% of its customer's searches for those terms. If it wanted to be thorough and add another 10% or so, it could prepare a similar signature for Bing.com searches.[139] Using the same logging mechanisms described above for associating other types of statistical counter hits with specific users, Rogers' DPI would effectively 'gain knowledge' of a portion of its customer's Internet search activity. All this would take would be some software and configuration changes.

---

[135] Bell, CAIP submission, *supra* note 33, at para. 191.

[136] Q.(G.) Zhao, J.(J.) Xu, and Z. Liu, "Design of a Novel Statistics Counter Architecture with Optimal Space and Time Efficiency", *SIGMetrics/Performance '06*, (Saint Malo, France: ACM, 2006), available online at: <http://www.cc.gatech.edu/~jx/reprints/Sigm06_counter.pdf>, pp. 1-2. While most routers are able to count transactions in the millions, and Zhao *et. al.* propose easy methods for extending that to the tens of millions, the operation outlined above requires more than just logging a signature hit. It includes logging the IP address as well, which would require additional data storage.

[137] Because, as noted at *supra* note 102 and accompanying text, DPI signature libraries can include thousands of signature strings to log, but typical statistical counters are designed to handle the millions of transactions that occur typically occur in routers and switches at inline speeds. Further, given that only approximately 27% of traffic on Canadian networks is P2P traffic (the type of traffic these devices typically log), and that the equipment only logs one signature hit per flow, this would generate a very small amount of logging activity (Condon, Bell Testimony, *supra* note 7, at line 6042). It would not be overtaxing the equipment to log signature hits along with IP addresses as described above.

[138] Taken on November 3, 2009 from HTTP header sent by my browser to Google after I entered the terms 'iPhone', 'Canada' and 'Provider' into the Google search bar and pressed search.

[139] J. Cheng, "Google Search Share Drops as Bing Gains Momentum", *Ars Technica*, August 3, 2009, available online at: <http://arstechnica.com/microsoft/news/2009/08/bing-continues-to-chip-away-at-googles-search-share.ars>, records Google's search market share at about 80% and Bing.com's at about 10%.

A similar technique could be applied to Email subject lines. Given that the control information used by most Email applications, whether web-based or not, will include a subject line,[140] Rogers can add a signature string to its DPI signature library to read the 'subject' field of the SMTP header in much the same way it currently scans the Host: or GET: fields of the HTTP/1.1 header to gather anonymous URL statistics on its counters.[141] It could minimize the number of hits it had to log by adding specific signature strings: 'I want to change ISPs'. It can then, as above, associate hits on those signature strings with the specific customers that generated them.

Can DPI be used to 'gain knowledge' of the types of files individual customers are transferring? If those files are being transferred using BitTorrent, or another P2P file-sharing application, then the answer is yes. As noted above, all BitTorrent application headers contain an info_hash header field.[142] This is also the case for most other P2P file-sharing applications.[143] This header field contains a data string that identifies the torrent, and hence the file, being transferred.[144] It would be relatively simple for an ISP such as Rogers to add a list of specific identified info_hashes to its signature library and log any data flow that contained these as well as its origin or destination (i.e. the Rogers customer transferring or receiving the file).[145] Ipoque, a DPI vendor, tells us that this can be done using basic DPI functions, in much the manner described above for URLs.[146] A unique string ('218c2bb02979399d0149163dd4c74e35c31fff32' [info_hash] instead of 'www.bell.ca' from above)[147] is first added to the signature library, the DPI then reads packets as they pass for that signature string, then logs signature hits along with a unique identifier (IP address) of the customer who originated or received the flow containing the signature. Rogers would then know how many and which of its customers were downloading the file "CRTC Hearing 2009-07-09 - Internet Traffic Management[tcpubmedia]", as a simple google search would reveal.[148]

Turning an ISP's current DPI network into one geared to identify this type of content would not require a hardware upgrade. It would merely be a matter of adding a few new signatures, and maybe some software or configuration changes. This is because the technique described here to 'gain knowledge' of user content relies on basic signature string detection functions that are at the core of how most DPI equipment operates. As Ipoque notes in describing the signature detection and matching technique noted above, "[t]his measure can be implemented using currently available traffic management systems based on deep packet inspection deployed at network access or peering points."[149] Ipoque also confirms that this same technique can, as noted above, be used to scan application layer control information to gain knowledge of URLs, again using basic DPI functionality.[150] Again, doing so on a comprehensive basis raises challenges surrounding large signature bases and the need to regularly updates these, but the equipment is undoubtedly *capable* of doing this.[151]

---

[140] D.H. Crocker, *RFC 822 – Standard For the Format of ARPA Internet Text Messages*, ARPA, August 13, 1982, available online at: <http://tools.ietf.org/html/rfc822#section-3.2>, at s.3.1.2 Structure of Header Fields, at p. 5.
[141] Bell, CAIP submission, *supra* note 33, at para. 200.
[142] Bauer, *et. al.*, *supra* note 57, at p. 9.
[143] K. Mochalski, H. Schulze, and F. Stummer, "Copyright Protection in the Internet: White Paper", [Ipoque Copyright Paper] (2009) Ipoque, available online at <http://www.foxreno.com/automotive/21537395/detail.html>, at p. 4: "Each file in file sharing networks has a unique ID. In P2P networks, this is the file has, and in file hosting systems, this is the URL."
[144] See *supra* note 58 for a sample BitTorrent info_hash and see "Screenshot", *supra* note 58, Attachment 2, for how the file associated with a given info_hash tag can be identified through a simple google search.
[145] Ipoque Copyright Paper, *supra* note 143, at p. 5-6 describe the technique of "File Hash-Based Identification and Blacklisting". To 'gain knowledge' of the files being exchanged, Rogers would simply need to log every time the signature string associated with it were hit along with the destination or source IP address.
[146] Ipoque Copyright Paper, *supra* note 143, at p. 5.
[147] See *supra* notes 144 and 58 and accompanying texts, as well as 'Screenshot', *supra* note 58, Attachment 2 for a description of info hash tags.
[148] See 'Screenshot', *supra* note 58, Attachment 2.
[149] Ipoque Copyright Paper, *supra* note 143, at p. 5.
[150] *Ibid.*, at p. 4.
[151] *Ibid*., at p. 5.

Signature string detection in the application layer control information of packet flows is a basic function of any DPI equipment. It is "[o]ne of the most frequently performed operation[s] in [DPI] applications".[152] All DPI is capable of performing it. And, since most DPI equipment performs this function to read application layer control information, it is certainly capable of gaining knowledge of the vast array of user-generated content contained therein. It *does* have the capacity to gain knowledge of URL browsing history, of Internet search queries, of Email topics, and of using and logging personal identification information of users.[153]

The *extent* to which DPI can do this (i.e. the number of signature strings it can filter for, the number of hits it can log) will depend on "which product is being used."[154] As noted above, the larger the signature library required, the more resource intensive and 'technically infeasible' the operation becomes.[155] In addition, it is accurate to say that all DPI products have the capacity to look within application layer control information "without examining the actual content" of the messages they are carrying.[156] This is purportedly what Rogers' DPI is currently configured to do.[157] But to say that Rogers' DPI "does not have the ability" to examine the content *or* the user-generated content of packets is inaccurate.[158] All the information described in this section is located in the application layer control information, and Rogers' DPI is fully capable of examining that content. That is, in fact its main function.

DPI: What else it can do

But many have claimed that DPI has the capacity to do more than *just* read the application layer control information to gain knowledge of user content. Some have even said, for example, that it can "peek inside all of these packets and assemble them into a legible record of your e-mails, web browsing, VoIP calls, and passwords".[159]

Rogers confirms Bell claims that its DPI cannot store in a cache actual user content or content accessed by the user, and that it cannot capture and play back any communications exchanges.[160] It adds that the data which passes through its DPI is fragmented, with packets that are often very small and can contain as little as 1 byte of 'content' ['content' now referring to a song fragment or part of the text of an email] per packet. Rogers continues to add that the "packets would have to be re-assembled at their destinations in order for them to be meaningfully interpreted."[161]

It is accurate that telecommunications messages are broken down into often very small fragments for transportation at various layers. When interacting with the network layer (and with DPI), however, this is as true for application layer control information as it is for a song or an Email message.[162] A complete

[152] Dharmapurik, *supra* note 80, at p. 1.
[153] As noted in the Bell Finding, *supra* note 114, at para. 4.
[154] Report, *supra* note 1, at para. 25.
[155] Bell, CAIP submission, *supra* note 33, at para. 191. Certainly, as Bell points out at para. 205, it would be essentially impossible to filter for millions of signatures. But, to perform its basic functions, DPI must, at the least, be able to scan for thousands of signatures, as it needs to be able to account for each version of each application or protocol it seeks to identify: Heavy Reading, *supra* note 31, at p. 9; Allot Whitepaper, *supra* note 29, at p. 4.
[156] Report, *supra* note 1, at para. 25.
[157] That is, it is currently configured to filter only for signatures identifying applications, not for signatures identifying URL or search queries or P2P hashes or Email subject lines. While some might argue that even the type of application one chooses to use it 'content' and 'user-generated' content at that, since it is the user who is deciding which application to use, it would be difficult to argue that the type of content described here does not fit that description.
[158] Report, *supra* note 1, at para. 25.
[159] Anderson, *supra* note 87, at p. 2.
[160] Report, *supra* note 1, at para. 23.
[161] *Ibid*., at para. 24, my emphasis.
[162] Heinz *et. al.*, *supra* note 85, at para. 0018, speaking of DPI string analysis of application layer control information, note:

packet header at the network layer, in contrast, will be attached to *each* packet, so that the network layer equipment will know how to route it. In most cases, a complete iteration of the application layer control information will be contained in the first few packets of a flow, whereas the song or email message may require more packets, depending on their size.[163] The packet header and application layer control information will often include a relatively small amount of data (relatively quick to scan for string matches) whereas a song file will be larger and may be in a different file format.[164] An Email message, it should be noted, will often be *smaller* than the various headers attached to it and would be relatively easy to inspect in whole.[165] While DPI can potentially scan all packets as they pass through it, this would be very resource intensive and highly impractical.[166] It would, moreover, take an immense amount of data storage to log and record of every song, Email and pdf file that passes through Rogers' network.[167]

On the other hand, it is inaccurate to state that data such as songs or emails must reach its destination before it can be re-assembled and meaningfully interpreted. While this would be very difficult to do at inline speeds of even 10 Gbps, it is not difficult to intercept and capture packet flows at such speeds, divert copies of them to auxiliary data storage facilities, and then reassemble and meaningfully interpret them at a more leisurely pace.[168] Due to lawful access requirements in the U.S., virtually all DPI boxes on the market today have the capacity to capture packets and redirect them in this manner.[169] In fact, this packet capture capacity is not unique to DPI, and can be accomplished by other network elements as well.[170] Regardless, it is simply inaccurate to say that packets have to reach their destination before they can be reassembled and meaningfully interpreted.

Further, combining the significant analytical capabilities of DPI and its configurability with its capacity to capture packets and divert them to auxiliary data storage makes it possible for ISPs to collect amounts of information that would be virtually impossible to capture with non-DPI network elements. Capturing all traffic passing through a network element at, say, 10 Gbps, would require a staggering 80,000 terabytes of data storage per year.[171] The UK estimates that it will cost it approximately £380 *per minute* or £200 million per year just to store a small portion of all the Internet traffic of its citizens, as it intends to do.[172] This would be virtually impossible for an ISP to do. However, with DPI, Rogers can use basic DPI functions already discussed above to greatly limit the number of packets that need to be stored for it to get the information it needs. As Jerome Tollet, CTO of Qosmos (another DPI vendor) states, DPI can be configured so that it "identifies the most genuinely useful data, directly extracts the content, and associates it with just the metadata that one needs."[173] In describing basic specifications that all DPI equipment typically has, CERN's HP Procurve openlab project refers to the capacity to group patterns in

---

Analysis by string match involves searching for a sequence (or string) of textual characters or numeric values within the contents of a packet. Furthermore, string matches may include several strings distributed within a packet or several packets.

[163] Park *et. al.*, *supra* note 81, at p. 162.

[164] *Ibid*.

[165] Typical Emails, without sizeable attachments (which, admittedly, anyone can attach), will only on rare occasions exceed 35 kb each, but will often be about 500 bytes in size.

[166] Park *et. al.*, *surpa* note 80, at p. 162.

[167] Tollet Myth 7, *supra* note 5.

[168] *Ibid*. See also, Qosmos, *Light ProtoBook*, Qosmos 2009, available online at: <http://www.qosmos.com/sites/default/files/related_resources/termqosmos_light_protobook_v4_220090226114546.pdf>.

[169] Bell, CAIP submission, *supra* note 33, at para. 201"under certain circumstances, such as legal intercept of traffic pursuant to a court order or for network testing diagnostics, the DPI may be configured to copy specific application or user traffic to an 'auxiliary' interface where an external capture device can be connected."

[170] *Ibid*.

[171] Tollet Myth 7, *supra* note 5, estimates that recording all data passing through a DPI box at 1 Gbps upstream and 1 Gbps downstream would require 8,000 terabytes of data storage space per year.

[172] J. Slack, "Big Brother Britain: £380 a MINUTE spent on tracking your every click", Mail Online, October 21, 2009, available online at: <http://www.dailymail.co.uk/news/article-1221839/State-spying-cost-200m-year-track-online-move.html>.

[173] Tollet Myth 7, *supra* note 5. See also Jurga and Hulbój *supra* note 93 at p. 6 point out how effective this technique can be.

this way as an essential feature of DPI.[174] In this way, DPI turns what would be extremely infeasible into a reality, by allowing ISPs to capture and store *only* the information they want while letting the rest pass.

Assuming the goal was to capture as much information (as opposed to data) as possible about consumers, how would this work? Well, first, DPI can be used to isolate telecommunication flows that are rich in information, but low in data (i.e. bytes). Email message bodies are good examples. While it is possible for Emails to contain very large attachments, depending on the use they are being put to, this will be less frequent. So Rogers could configure its DPI to copy and divert any flow classified as an Email communication to an 'auxiliary' storage interface. It can be even more granular with a progressive scanning technique. It can initially scan application layer control information for signatures that identify packet flows as email exchanges. Then, once these packet flows were identified, it could then proceed to scan the rest of the packet payload for signature strings such as 'Rogers' or 'Bell'. In this way, the DPI will only have to scan entire packet flows if those flows are first identified as email messages. Finally, only those emails that register hits on 'Rogers' or 'Bell' need be diverted to auxiliary storage facilities. This progressive scanning and capture technique, possible only through the flexibility and advanced analytical capacities of DPI, will greatly ease the inline analytical load and allow for capturing a great amount of information at minimal cost. Qosmos advertises this type of progressive scan/capture as a feature of its DPI but, again, it relies on basic DPI equipment capacities – to make any DPI box do this would be a simple manner of software reconfiguration.

The same technique can be applied to other types of information. Entire browsing histories per user *can* be captured, if these are stored in auxiliary data storage. And DPI can be used to ensure only one small piece of data, the URL, is stored per packet flow and stored in association with the user that generated it, so all HTTP traffic need not be stored and later parsed for URL history and discarded. The same can be said for search queries. While the DPI cannot do this on its own, given the vast amounts of information potentially involved, it would be essentially impossible for an ISP to do on a comprehensive basis without DPI.

Even without capturing packets in this manner, DPI is capable of gaining knowledge of some payload content *not* contained in the application layer control information. Again, it would be difficult for DPI to read and analyze an email attachment at inline speeds. It *can*, however, read email texts for particular signature strings (as indicated above). Because DPI does not generally read entire packet flows, Rogers would need to configure its DPI to implement the progressive scan technique described above. It would first classify flows as it normally does. Then, those flows identified as email exchanges would be further analyzed for signature string hits such as 'Bell has better service'. Finally, hits on such signatures would be logged along with source or destination IP address. Doing this for email messages would not overly tax any DPI box, since the text of an email (as opposed to any attachments) is rarely more than a few kilobytes and the Email 'application header' states the boundary between email and attachment. Given the ease with which this reconfiguration can be accomplished and its reliance on basic functions common to all DPI equipment, it is unclear how it can be said that DPI lacks the capacity to read beyond protocol headers or to inspect content.

In conclusion, while it is true that DPI alone is not capable of storing everything that passes through it, it does have the capacity to gain knowledge of a great deal of information, regardless of whether this information is in 'protocol headers' or elsewhere. To say that it *only* has "the capability to inspect [the] many protocol headers"[175] and not any other portion of the packet payload is inaccurate, as inspecting one is not qualitatively different from inspecting the other. All DPI equipment has the additional capacity to use "personal information of individual users" in order to make traffic management decisions, to store/log

[174] Jurga and Hulbój *supra* note 93 at p. 6.
[175] Report, *supra* note 1, at para. 6.

"personally identifiable information", and to gain knowledge of "URL browsing history…Internet search activity…email topics [and] content".[176] It is possible to "meaningfully interpret" information from packets (whether from header or payload) without reassembling them at their destinations.[177] Moreover, it is *also* possible to intercept and "re-assemble" packet flows before they reach their destination,[178] although, without DPI, it is extremely difficult if not impossible to do so in a comprehensive manner.

Whether DPI examines specific packet content does *not* "depend[] on which product is being used."[179] All DPI has the capacity to examine packet content. It merely depends on how the specific DPI boxes in question are configured – what they are configured to look for and where. It is certainly true that DPI has the *capacity* to look within the packet or traffic stream (regardless of where) "without examining actual content."[180] However, this capacity, again, is merely a matter of configuration. Rogers states that its DPI is not currently configured to do so. We argue that the 'content' of a telecommunications packet includes the application layer control information, and that examining for type of application *is*, effectively, examining for content and even user-generated content. Regardless of whether application signature strings are considered to be user-generated content or not, any DPI box, including whatever models Rogers has deployed in its network, is certainly capable of "examining the content of packets."[181] Examining packets for 'content' is no different from examining them for 'application signature strings'. We ask, again, that the Report be corrected to reflect these inaccuracies.

D. P2P: What's the Problem?

The Finding states:

> Some software applications are heavy users of bandwidth and are designed to use available capacity in the network in order to transmit their information. Email is typically a small consumer of bandwidth, while the file sharing and Peer-to-Peer (P2P) applications are large users. Applications that are often used for music and movie file sharing between computers can consumer a great deal of capacity and "slow down" other Internet traffic.[182]
>
> [...]
>
> In the course of our investigation, Rogers stated that the cable network was initially designed for the downstreaming of a television signal—not for the high capacity P2P uploading, which causes disruptions to the network.
>
> Rogers goes on to say the following:
>
> > Accordingly, left unmanaged, peer-to-peer applications can swamp a network's upstream capacity and impair customers' ability to send emails, surf websites or use Internet voice services such as Vonage or Skype, which are time sensitive applications. While Rogers continues to spend tens of millions of dollars on network capacity augmentation, this upstream impairment due to P2P traffic is not a problem that can be cured by increasing capacity. A capacity increase will simply lead to the peer-to-peer applications consuming most of the increased capacity.
>
> This Office's investigation confirmed that the use of P2P applications to transmit content through the Internet does increase network traffic (i.e., consume more bandwidth) than other client-server applications and that the widespread phenomenon of congestion in networks is largely caused by user downloads or computer-to-computer file sharing using P2P applications.[183]

[176] *Ibid*, at para. 23.
[177] *Ibid*., at para. 24.
[178] *Ibid*.
[179] *Ibid*., at para. 25.
[180] *Ibid*.
[181] *Ibid.*
[182] *Ibid*., at para. 6.
[183] *Ibid*., at paras. 10-12.

These statements are inaccurate to the extent that they imply heavy bandwidth consumption is a function of applications as opposed to individuals, that they fail to distinguish between the impact of P2P applications on downstream as opposed to upstream traffic, and that they point to P2P applications as the predominant cause of the widespread phenomenon of congestion in networks.

Bandwidth consumption is a product of individuals, not applications. It is true that file-sharing applications (P2P or otherwise) are generally used to transfer bigger files than Email. But there are file-sharing applications that are server based and not P2P, which transfer the same types of files.[184] Moreover, much of the same type of bandwidth heavy content is transferred across the network by streaming sites.[185] Indeed, a recent Internet traffic study by Arbor Networks, billed as the "largest study of global Internet traffic since the start of the commercial Internet" lists as one of its main conclusions that:

> Applications Migrate to the Web: Historically, Internet applications communicated across a panoply of application specific protocols and communication stacks. Today, the majority of Internet application traffic has migrated to an increasingly small number of web and video protocols, including video over web and Adobe Flash. Other mechanisms for video and application distribution like P2P (peer-to-peer) have declined dramatically in the last two years.[186]

These conclusions are confirmed by another recent study by Sandvine, which has noted "a dramatic shift in consumer behavior towards real-time "experience now" applications and away from bulk download "experience later" behavior.[187] This trend has been confirmed on the record of Telecom Public Notice CRTC 2008-19, where ISPs presented statistics on Canadian traffic usage which demonstrated quite clearly that, even in 2008, HTTP web traffic already outstripped P2P. The traffic breakdown for all customers of these ISPs, including Rogers, demonstrates that about 40% of Canadian bandwidth usage is HTTP (including streaming sites, cloud computing, server-based file-sharing applications, etc.), and only about 36% can be classified as P2P file-sharing traffic.[188] These same usage statistics demonstrate the true source of any congestion problem there might be – the top 5% bandwidth consumers of the same ISPs generate about 47% of *all* traffic on those networks. Of this 47% traffic, a bit more than half (57%) was P2P file-sharing traffic, meaning that P2P traffic generated by top 5% users is responsible for 27% of all traffic.[189] But about 50% of Canadian ISP customers use P2P file-sharing applications.[190] The 45% of Canadian customers that use P2P file-sharing applications and are *not* top 5% users generated only about

---

[184] One example is www.rapidshare.com. Rapidshare.com is rated by Alexa as the 18th most visited site in the world (http://www.alexa.com/siteinfo/rapidshare.com) and facilitates server based sharing of the same types of files typically exchanged through P2P file-sharing sites: Ernesto, "Rapidshares Shares Uploader Info with Rights Holders", TorrentFreak, April 25, 2009, available online at: <http://torrentfreak.com/rapidshare-shares-uploader-info-with-rights-holders-090425/>.

[185] Some estimate that YouTube alone generates 30 million megabits of traffic per second: T. Spangler, "YouTube May Lose $470 Million in 2009: Analyst", Multichannel News, April 3, 2009, available online at: <http://www.multichannel.com/article/191223-YouTube_May_Lose_470_Million_In_2009_Analysts.php>.

[186] Arbor Networks, "Two Year Study of Global Internet Traffic Will Be Presented at NANOG47", Arbor Networks – News Releases, October 13, 2009, online at: < http://www.arbornetworks.com/en/arbor-networks-the-university-of-michigan-and-merit-network-to-present-two-year-study-of-global-int-2.html>.

[187] Sandvine, *2009 Globel Broadband Phenomena – Executive Summary*, [Sandvine Global 2009] Sandvine, October 2009, available online at: <http://www.sandvine.com/downloads/documents/2009%20Global%20Broadband%20Phenomena%20-%20Executive%20Summary.pdf> at p. 2.

[188] CRTC Letter, *supra* note 6, at chart (CRTC) 04Dec2008-1 (b), straight arithmetic average of %HTTP/Streaming traffic and %P2P traffic. This is likely to be an overestimation, as it is a straight arithmetical average and does not account for different ISP sizes. Bell, the largest network in Canada, states (Condon, Bell Testimony, *supra* note 7, at line 6042)that on *its* networks, P2P traffic only accounts for 27% of all traffic when unthrottled:

> 27 percent of our traffic is peer-to-peer throughout the day, but at peak that number reduces to 14 percent as a result of our shaping of that traffic.

[189] CRTC Letter, *supra* note 6, at chart (CRTC) 04Dec2008-2 (c), straight arithmetic average of top 5% usage = 47% of all traffic. Chart (CRTC) 04Dec2008-2 (d), straight arithmetic of Top 5% End-Users %P2P Traffic = 57% of all Top 5% usage is P2P traffic. 57% (percent of top 5% traffic that is P2P traffic) of 47% (percent of all traffic generated by top 5%)= 27% (percent of all traffic generated by top 5% P2P traffic).

[190] Daniels, Bell Testimony, *supra* note 9, at line 6829.

9% of all traffic through their use of P2P. So for the vast majority of P2P users, P2P, as a class of applications, is *not* generating excessive or even disproportional amounts of bandwidth.

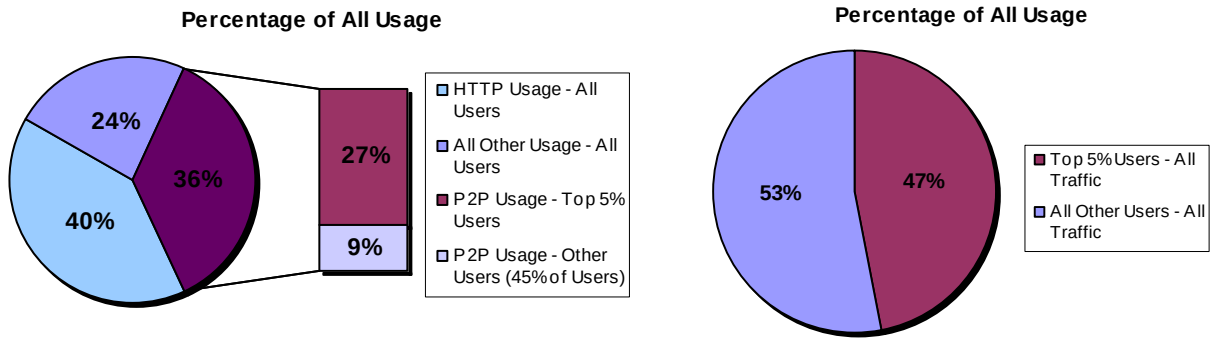

Figure 7 – Breakdown of Bandwidth Usage in Canada[191]

It is not 'software applications' but 'individuals' that are "heavy users of bandwidth".[192] And it is important to make this distinction clear. Based on this misconception, ISPs decide to target P2P applications instead of individual users, stating it is necessary to manage these 'heavy bandwidth users' in order to avoid congestion. While PIPEDA might not require a response that is more targeted to actual causes of congestion, other statutes do, and so accuracy with respect to such issues is necessary.

To the extent that P2P file-sharing applications *can* be characterized as heavier bandwidth consumers, this applies solely to upstream traffic, not to downstream. The distinguishing feature of P2P protocols is, as Rogers states, that such protocols are symmetrical in that, roughly, for every MB downloaded by one peer, another MB is uploaded by another.[193] Non-P2P file transfer models are generally client-server based, meaning that much of the upstream bandwidth load is carried by servers and not by residential lines, often by content hosters located outside of the ISP's networks. This is evident once Internet traffic is broken into its upstream and downstream components.

While overall P2P traffic in North America amounts to approximately 30% of all bandwidth, P2P accounts for only about 18% of all *downstream* bandwidth – far less than web/web streaming [HTTP] (67% of all traffic).[194] Meanwhile, P2P *upstream* bandwidth accounts for about 55% of all upstream traffic – although it must be noted that recently even this number has fallen dramatically to about 31%.[195] This causes problems for North American ISPs, and particularly for cablecos, who have failed to properly

[191] Note again that the percentage of P2P usage reflected here is *greater* than that on most Canadian networks. The 36% figure is derived from a straight mathematical average of anonymized data provided by Canadian ISPs, but the 27% found on Bell's (Canada's largest ISP) networks is likely far more representative (see *supra* note 188 for more details). Better data on Rogers' networks is unavailable to CIPPIC, but it is open to the OPC to request such data.
[192] Report, *supra* note 1, at para. 6.
[193] *Ibid*., at para. 11.
[194] Sandvine, *Initial Comments: Telecom Public Notice CRTC 2008-19*, February 23, 2009, Appendix A, [Sandvine Initial Comments] available online at: <http://www.crtc.gc.ca/public/partvii/2008/8646/c12_200815400/1029527.pdf>. Figure 18 (p. 15 of Appendix A) demonstrates that in North America at peak periods (when there is congestion), P2P applications generate about 18% of all downstream traffic. Meanwhile, HTTP traffic (web [all non-streaming HTTP traffic: 47%] + web media [HTTP streaming traffic such as YouTube and server based file-sharing: 20%]) amounts to about 67% of all downstream traffic. Web, web media and P2P are defined in Table 1, p. 22 of Appendix A. It should be noted that more recent figures put this number at about 15% (Sandvine Global 2009, *supra* notes 8/187 at p. 4).
[195] Sandvine Initial Comments., Appendix A, *supra* notes 12/194, at figure 19 (p. 16 of Appendix A), at peak periods. Although it should be noted that overall upstream bandwidth in general amounts to about a third of downstream traffic (see figure 17 at p. 14 of Appendix A). For more recent figures (31%) see Sandvine Global 2009, *supra* notes 8/187 at p. 4.

provision their upstream network capacity.[196] For this reason, while it is inaccurate to state that P2P applications, by nature, are 'heavy users of bandwidth' and that such applications "consumer more bandwidth…than other client-server applications", it *is* accurate that P2P file-sharing applications, by their nature, will "consume more [*upstream*] bandwidth…than other client-server applications [when transferring the same file]".[197] This distinction, again, is important because some ISPs, notably Bell, ignore it and thereby justify targeting both upstream and downstream P2P file-sharing traffic without proving congestion on the latter, in spite of the fact that downstream and upstream channels are provisioned separately.

Even taking into account the unique impact P2P file-sharing applications have on upstream traffic, it is not true that "the widespread phenomenon of congestion in networks is largely caused by user downloads or computer-to-computer file sharing using P2P applications."[198] To begin with, recent trends in traffic note that usage is moving towards increasingly interactive models, and now HTTP traffic (gaming, streaming, etc.) now outstrips P2P even on the upstream (though to a lesser extent than on the downstream).[199] But regardless, it is difficult to attribute 'congestion' to *any* single application or user. That is just not the nature of the Internet. Bell, in its testimony before the CRTC, expressed it best:

> So, is peer-to-peer a significant cause of congestion? It is a major cause of congestion. Every -- you know, if we just assume some equality in the packets for a second, every packet that's going through a device -- a congested device is a contributor to that congestion. In our case, as we said, 27 percent of our traffic is peer-to-peer, so it's certainly a significant contributor to that.[200]

Further, it is not 'P2P file-sharing applications' that cause congestion, but a small subset of users (5% of the 50% of all customers that use P2P) of those applications. In net, and especially on the downstream, P2P file-sharing applications do not even generate as much traffic as other types of traffic (namely, HTTP), and in total accounted for less than a fifth of all network downstream bandwidth, even at the time this complaint was filed.[201]

On the upstream, P2P applications as a class generated a more substantial proportional amount of traffic and ISPs (particularly cablecos) are ill-equipped to deal with it since they have failed to sufficiently provision their upstream capacity. Nonetheless, when the actual impact of this upstream traffic is measured, it amounts to little. This lack of impact is best demonstrated from traffic measurements produced by Comcast, a major U.S. cableco that purported to face the same challenges (if not more severe) that Rogers and other Canadian ISPs now complain of. Comcast argued that P2P file-sharing applications were causing rampant congestion on its network in order to justify its application-specific traffic management practices.[202] However, when forced to implement an application-agnostic traffic management practice instead, and one that only operates in the presence of *actual* congestion, Comcast found that in order to eliminate *all* "impair[ment of] customers' ability to send emails, surf websites or use Internet voice services", it is only necessary to throttle less than 1% of customers and rarely for more than 15 minutes.[203] This applies to both downstream *and* upstream traffic.

---

[196] Report, *supra* note 1, at paras. 10-11: "Rogers stated that the cable network was initially designed for the downstreaming of a television signal—not the high capacity P2P uploading…"
[197] *Ibid.* at para. 12.
[198] *Ibid*.
[199] Sandvine Global 2009, *supra* notes 8/187, at p. 4.
[200] Cordon, Bell Testimony, *supra* note 7, at lines 6293-6294.
[201] See *supra* note 194 and accompanying text: at the time of this complaint, P2P traffic accounted for 18% of all downstream bandwidth in North America while HTTP traffic generated about 67%. Since that time, P2P usage has only decreased while HTTP has increased: Arbor, *supra* note 8 and Sandvine Global 2009, *supra* notes 8/187 at p. 4.
[202] FCC, *Memorandum Opinion and Order – Comcast*, August 20, 2008, FCC 08-183, available online at: <http://cyberlaw.stanford.edu/system/files/FccComcastOrder.pdf>.
[203] Sandvine Initial Comments, *supra* notes 12/194, at para. 95.

This minimal amount of impact can hardly be described as widespread congestion (broadly defined as "a situation whereby the amount of traffic transiting the network may lead to a deterioration in service for some end-users").[204] Further, even this minimal amount of congestion is not 'caused' by P2P file-sharing traffic. The latter is merely one "contributor" and not ever the most substantial one, at that.[205]

In sum, software applications are not "heavy users of bandwidth", only users can be heavy users of bandwidth.[206] "File sharing and Peer-to-Peer (P2P) applications" are *not*, then, "large users".[207] Further, while applications that are used to share files "between consumers can consume a great deal of capacity and "slow down" other Internet traffic",[208] this is not the end of the story, as *other* types of applications – ones that operate between consumers *and servers*, consume even *more* capacity and slow down other traffic to a greater extent than P2P.[209] Finally, as witnessed above, it is simply false to say that applications using the P2P protocol "consume more bandwidth[] than other client-server applications", as HTTP-based web applications, did at the time of this complaint and do now to a much greater extent, consume *more* bandwidth than P2P file-sharing applications.[210] Finally, the "widespread phenomenon of congestion in networks", if there is any such thing, cannot be attributed largely to "user downloads or computer-to-computer file sharing using P2P applications".[211] P2P traffic, like *all* traffic, contributes to any congestion there may be, but it is not the primary contributor and, at the time of this complaint, did not even contribute a fifth of downstream traffic.

[204] Telecom Public Notice CRTC 2008-19, November 20, 2009, available online at: <http://www.crtc.gc.ca/ENG/archive/2008/pt2008-19.htm>, at endnote 6.
[205] Cordon, Bell Testimony, *supra* note 7, at lines 6293-6294.
[206] Report, *supra* note 1, at para. 6.
[207] *Ibid*.
[208] *Ibid*.
[209] *Ibid*.
[210] *Ibid*., at para. 12.
[211] *Ibid*.