DPI Technology from the standpoint of Internet governance studies: An introduction

Dr. Milton Mueller, Syracuse University School of Information Studies

What is DPI?

A study of whether and how DPI technology disrupts Internet governance must have a workable definition of what DPI technology is. This is not as easy as it may sound. Most of the things we call 'a technology' are actually complex systems that combine many different techniques and materials, some new, some old. Writers in Actor-Network Theory (ANT) have called attention to 'punctuation,' the process by which the heterogeneous components of a technological system come to be seen as a single 'thing' (Law 1992). Somewhat confusingly, ANT theorists call these 'things' *networks;* they go on to make the ambitious metaphysical claim that social agency is "an effect produced by a patterned network of heterogeneous materials." (Ibid) When the network performs as intended, the heterogeneous materials meld into a single actor and its constituent parts become invisible to the casual observer (Brey 2005).

Something like that has happened to DPI, at least discursively. DPI has many features of Internet technologies that have been around for a long time, such as firewalls and packet capture or packet sniffing techniques. Yet the combination of these elements into a scalable, widely implemented set of practices is generally seen by industry, technologists and policy critics as a new technology or capability (Weinschenk 2007; Proch and Truesdell 2009; Riley and Scott 2009).

In this essay, we attempt to define and circumscribe DPI in a way that lays the groundwork for analyzing its impact on Internet governance. The discussion is based on reviews of the computer science literature, vendor White Papers and interviews with 15 leading DPI vendors. The paper begins by isolating and describing the generic enabling capabilities of DPI technology. The next section lists and briefly describes the major applications (use cases) that have been developed from those capabilities. The final section assesses the prospects for convergence of those use cases into an integrated network surveillance and control technology.

DPI's key feature

One aspect of the definition of DPI is unequivocal and fundamental. DPI pertains to *information in motion*, not information at rest. There are several applications that scan and detect digital content stored on servers or computers.[1] These may rely on technical methods or scientific principles similar to those used by DPI applications. But we have chosen to reserve the label 'DPI' for applications that detect and

[1] E.g., Google's Content ID, which generates a fingerprint of sound and video files that copyright holders have submitted to Google, and compares all fingerprints to the files uploaded to YouTube. Over 3 million "reference files" have been submitted to Google for this purpose, according to Google's Content ID web site http://www.youtube.com/t/contentid.

shape live traffic on a network. DPI recognizes patterns in TCP/IP packets, and packets are a data format standardized for transmitting information over an electronic network. (Figure 1) The need to recognize and act upon information in real time as it flows through the Internet is a technical problem with its own distinctive features and constraints.[2] At its core, DPI is an attempt to solve that problem.

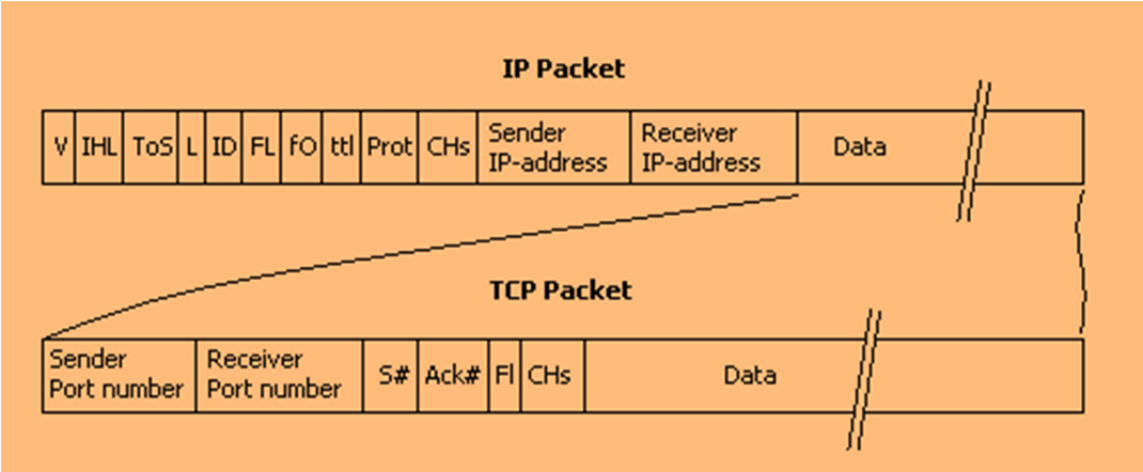

Figure 1: An Internet Protocol (IP) packet encapsulating a Transmission Control Protocol (TCP) packet. "Deep" packet inspection uses any field in either packet to analyze or understand the traffic and to make decisions about how to manage or control traffic.

DPI as Enabling Technology

The best way to make definitional sense of DPI is to separate the basic, enabling capabilities of DPI from the various applications derived from them. In other words, DPI should not be confused with its applications, or with a fixed list of use cases. DPI was characterized by many of the vendors as an "enabling technology." By that they meant that it is a generic capability that supports many different applications or use cases. Each application allows network operators or third parties to regulate or monitor different types of network activity. The distinction between generic enabling capabilities and specific use cases proved to be an important step in understanding DPI.

The primary technical capability underlying DPI is what we call *Recognition;* that capability can trigger two other capabilities: *Manipulation* and/or *Notification*.

Recognition

Recognition involves detection or identification of things in a bit stream as they move through a network. A DPI engine analyzes TCP/IP traffic in real time and considers any and all information in the packets to be subject to inspection, including the payload. The recognition capability can be used to detect:

- Protocols
- Applications
- URLs (web site addresses)
- Media content (specific instances of recorded music, movies, images or books)

[2] Of course, there are gray areas; e.g., when traffic is copied to a DPI appliance to develop forensic analyses for later use.

- Viruses, malware and other cyber-intrusions or exploits
- Text strings
- Data that follows a specific format (e.g., credit card numbers, Social Security Numbers)

Note that the Internet packet header has always been 'inspected' and used as the basis for routing decisions. The differences are: 1) DPI uses *any* part of the packet for detection - including the user-generated information in the packet payload; 2) DPI can look for patterns across multiple packets; and 3) the decisions made involve more than just where to forward the packet.

In addition to the traditional job of header inspection, DPI applications are designed to inspect the payload as well. Often they will correlate information in the headers, such as IP addresses, with that information but the machines, programs and processes that do this could also inspect any other part of the packet. In this respect it might be more accurate to refer to deep packet inspection as *whole-packet inspection*. One interviewee defined DPI as an "act of any equipment not an end point using any field other than the layer 3 header. It could be the port number, payload, length, or any other field for any purpose."

Critically, the recognition process relies on the creation of predefined patterns that must be programmed in advance. As one vendor put it, "Unless [a] security appliance knows what threat signatures or anomalies it is looking for, it is helpless." (eSoft, undated, p. 5) The vocabulary for describing these patterns is not standardized; one hears of 'signatures,' 'fingerprints,' 'rule sets,' and 'regular expressions.' Sometimes those terms are used interchangeably; more often they connote distinct procedures or technologies. The components that perform this function are known as *DPI engines*, and can be implemented in hardware, firmware or software (Lin, Tseng et al. 2007; Bremler-Barr, Harchol et al. 2011).

This is where the real science and mathematics of DPI come into play. Real-time recognition of patterns in high-speed networks demands highly sophisticated algorithms and advanced chipsets. An example of one of the seminal techniques is the Aho-Corasick string matching algorithm (Aho and Corasick 1975).[3] Aho-Corasick "constructs a Deterministic Finite Automaton (DFA) for detecting all occurrences of a given set of patterns by processing the input in a single pass" (Bremler-Barr, Harchol et al. 2011). Scientific work in this area is constantly searching for faster or less resource-intensive pattern-matching techniques (e.g., Ahn, Hong et al. 2010; Huang, Zhang et al. 2010; Kim, Kim et al. 2010).

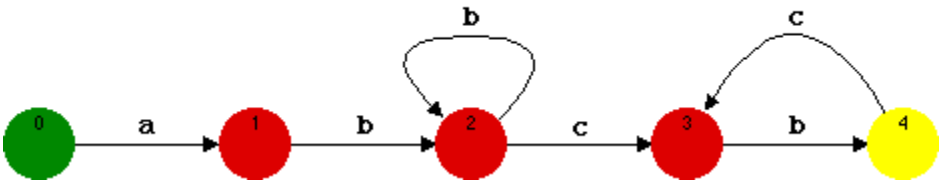

Figure 2: A Deterministic Finite Automaton for matching the string 'abbbc'

[3] Aho-Corasick has been described as "the defacto standard for pattern matching in network intrusion detection systems (NIDS)" (Bremler-Barr et al, 2011)..

Typically, DPI engines start with *regular expressions*. These are minimal distinctive patterns that an application, protocol, malware exploit, etc., will produce in the bit stream. A *signature* will then add more precise information about where to look for that pattern in the traffic it is inspecting, and more specific information that can validate the identification. A *rule set,* which is typically implemented by the network operator rather than the vendor, gives the DPI engine more specific instructions; e.g., which signature(s) to apply to which traffic between specific points. Signatures can thus be considered a complex language for matching strings of information in a traffic flow.

Like any recognition process, DPI engines can suffer from false positives and false negatives. Recognition processes also be consciously evaded or circumvented by clever programmers.

The patterns a DPI engine searches for not only must be *pre-defined,* they must also be constantly *updated* and *expanded* to deal with new or changing phenomena. This means that DPI is a *service,* not just a product, and requires ongoing relationships with signature producers. DPI also imposes heavier management burdens on network operators who implement it. To avoid disastrous results from false positives, operators need to know what kind of traffic is on their network and what kind of events are likely to be triggered by signatures. One vendor noted that "We sometimes have to convince customers that they … cannot rely on a global, generic policy."

The number of these predefined patterns and their complexity in relation to line speeds creates an important scalability constraint on DPI implementations. Moreover, recognition capabilities are limited by culture and by the operating system environment. One vendor noted that the accuracy of their application identification, which might be 90-95% in Europe or North America, will decline to 60% in Japan. Networks in a Unix environment that run signatures written for a Windows environment will get many false positives. Some applications may be regional in scope, and vendors based outside that region may not even have signatures for them.

Thus, DPI does not conform to the naïve, dystopian caricature of a surveillance technology that allows network operators to effortlessly know and manipulate anything and everything their users are doing. One must know in advance what one is looking for, and one must create and distribute (imperfect) instructions for how to recognize it. Likewise, DPI falls well short of the utopian fantasy – a piece of equipment that solves all network problems in a single stroke. One vendor joked that "the majority of our customers just want to push a *secure my network* button," but admitted that DPI implementations do not conform to that simple model.

Manipulation

The second basic technical capability is manipulation. Manipulation is active intervention in a live traffic stream to optimize, control or change it. It is based on rule sets which instruct the network to behave in a certain way contingent upon some form of recognition. The recognition capability is thus primary, and the manipulation capability is dependent upon it. More broadly, manipulation can be programmed to:

- Block the movement of recognized informational objects into or out of the network
- Regulate (rate-limit) packet flow speed
- Change the packet header in some way
- Prioritize (or de-prioritize) some protocol's packets over others
- Prioritize (or de-prioritize) some user's or class of users' packets over others

- Disconnect a session

Typically, a DPI vendor will supply and maintain the signatures that enable recognition, while their customer (the network operator) defines rule sets for manipulating the recognized applications or types of content. An example of a simple rule set would be: *during peak hours, peer to peer protocols have the lowest priority*.

Notification

Notification is a more indirect form of intervention than manipulation. An act of recognition by a DPI appliance might trigger the following actions:

- Generate statistical reports
- Issue alarms or notifications
- Generate a billing incident, or make access to an Internet resource contingent upon pre-payment

As a less direct form of intervention, applications that involve notification need not be positioned directly in the network.

DPI Applications (Use cases)

Using these three basic capabilities, vendors and network operators build DPI applications. The list of DPI use cases is long and the lines between them are blurry. This is to be expected, because various detection and manipulation capabilities can be combined in many different ways to suit different purposes. Any attempt to come up with firm, durable classifications of DPI use cases is bound to fail; use cases are creative responses to needs which change over time and reflect diverse business conditions and technical environments.

Still, it is worthwhile to outline the basic parameters of current DPI uses. As a first cut, DPI applications can be divided into the *passive* and the *active*. Passive applications utilize the *recognition* capability but do no *manipulation,* confining their actions to various forms of *notification*. A passive application, for example, might classify protocols running on a network and generate data that allows operators to quantify the nature of their typical bandwidth usage patterns, but refrain from an attempt to manage traffic.

Another, related distinction is whether DPI is deployed in-line or not. *In-line* installations are positioned directly in the traffic stream; i.e., the live traffic runs through them. *Off-line* deployments divert a copy of the traffic to the DPI appliance. Applications that perform notification do not need to be deployed in-line. Applications that involve more active manipulation generally need to be deployed in-line. In-line deployments could become a point of failure in the network, and must be able to keep up with the network's maximum speeds. The advantage is that they are transparent to the network and to the user, and require no modifications of the network's topology.

Beyond those very generic features of applications, one can discern six 'family' groupings of DPI use cases, with a number of more specific applications under each grouping. They are 1) Network Visibility

and Bandwidth Management; 2) User profiling; 3) Governmental Surveillance; 4) Network security; 5) Copyright policing; and 6) Censorship or content regulation.

1. Network Visibility and Bandwidth Management

Some of the most popular DPI applications are intended to allow commercial network operators to understand and analyze the composition of their traffic, and to intervene more actively and discriminately in the management of that traffic. The basic use case here is to maximize the value of what is perhaps their largest investment and most critical asset – their bandwidth – and to leverage one of their most powerful and strategic information resources – their knowledge of customer usage. Specific applications can be passive and focus exclusively on giving the operator greater visibility into what is happening on their facilities, or they can combine passive and active elements.

In a passive deployment, DPI appliances can recognize and classify applications and protocols running over the network and generate reports about it. This type of application supports capacity planning, investment decisions, or pricing and marketing decisions. When usage visibility (*recognition*) is combined with the *manipulation* capability, DPI can contribute to bandwidth regulation, optimization or congestion response. ISPs can throttle (slow down or impose rate limits upon) the bandwidth allocated to specific protocols or users. They can also prioritize (or de-prioritize) specific protocols, services or users.

2. User profiling/monetization

Subscriber awareness can be considered an extension of the network visibility use case, or it can be seen as a separate use case. When usage visibility is combined with subscriber awareness, it contributes to monetization of services. For example, network operators can use DPI to discriminate between those using the web for ordinary surfing and those using Skype to avoid telephone charges, and require anyone using Skype to pay an additional monthly fee. By relating information about usage volume or types of services consumed to a particular subscriber, the application supports more sophisticated billing and charging structures and functions.

More controversially, it can compile and classify information about what individual users do on the network, creating profiles. These profiles can then be monetized in the same manner as web sites use ‘cookies’ to identify users and adjust what they can see and do. Profiling can allow a network to tailor the advertisements a user sees on a website to their specific behavioral patterns. For example, the controversial “ad injection” technique allowed ISPs to profit from their knowledge of which categories of websites their customers were visiting by making deals with advertising firms to inject ads matching those ‘interests’ into their web traffic.

3. Governmental surveillance (a.k.a. lawful interception)

National laws typically require communication service providers to provide some kind of eavesdropping or surveillance capabilities to government law enforcement or public security agencies. In principle, a DPI application can make anything that happens on a network visible and recordable to governments – as long as someone tells the system who or what to look for in advance.

In liberal democracies these capabilities are usually controlled by regulations and laws (with, alas, many lapses). These laws regulate what kind of persons are subject to surveillance, what information can be collected, how long it can be stored and what is admissible evidence in a prosecution. Ergo, the practice is referred to as “lawful intercept” (LI). LI technologies predate DPI, of course, but DPI applications are

gaining ground in this field because of their ability to efficiently sift through large quantities of information. The system may be told to record all the activities of a specific IP address or user account, or to look for keywords, the use of secure (encrypted) protocols, or other things. (Note that maintaining records of the IP addresses accessed by customers does not require DPI, though this capability can be greatly enhanced by the use of DPI.)

When the government in question is authoritarian or not subject to the rule of law, the same capabilities can be used for arbitrary, unrestricted surveillance of the contents of user communications – subject, of course, to the constraints of recognition techniques and scalability. The exposure of the use of Western DPI products and other network surveillance capabilities by dictatorships in the Middle East has stimulated an international debate about export controls and the role of vendors and governments in ensuring ethical use of the technology.

4. Network Security

Network security was the earliest driver of the development of DPI capabilities. Dating back to the late 1990s, researchers began to combine malware recognition capabilities with packet capture and analysis techniques. This combination spawned early intrusion detection systems (IDS) such as Snort, which still exists. Snort and other DPI engines recognize a growing library of known threat signatures. IDS was first used to detect known exploits and intrusion prevention systems (IPS) were later added functionality. Whereas an IDS identifies threats and then merely logs them or generates alerts, an IPS responds to the detected threats by implementing measures to prevent the attack from succeeding, usually by terminating connections. DPI can be used to detect keystroke loggers, bot infections, abnormal quantities of mail, command and control instructions from bot herders, or communications to servers known to be associated with bot nets. Due to scalability concerns, active IPS is more likely to be used by smaller-scale enterprise (corporate) networks than by carrier-grade public Internet service providers.

This family of applications overlaps with certain forms of content control, discussed below, especially data loss prevention. Data Loss Prevention (DLP) is designed to protect the 'leakage' of sensitive corporate information outside of the network, so the use case is based on a 'security' rationale. DLP software or hardware can be installed at network egress points and analyze outgoing traffic to detect whether sensitive data is being sent in violation of the organization's information security policies. These solutions use multiple methods for packet analysis, ranging from keywords, dictionaries and regular expressions, to partial document matching and fingerprinting.

IDS/IPS also have technical and user affinities with lawful intercept capabilities, discussed previously. The overlap is in the area of network forensics; i.e., capturing and storing information about intrusions, crimes, or anomalous activity, which may be used for analysis and planning, or for prosecution or other forms of law enforcement action. Thus the boundaries between this use case and the other two mentioned are blurry.

5. Copyright policing

The persistence of online file sharing has prompted rights holders around the world to pressure ISPs to "cooperate" in the enforcement of copyrights. In some cases, DPI is put forward as a technical solution.

The most ambitious application of DPI to the copyright problem involves real-time recognition of copyrighted content by Internet service providers as it flows over their network. ISPs are then expected to

automate notification of their users when detection occurs, or possibly even to block it (manipulation). This architecture implies that each piece of copyrighted content – millions of music or sound files, hundreds of thousands of television programs, etc. – has a unique signature that is registered with and can be recognized by the DPI appliance. The vendor Audible Magic has pursued this solution. Its signatures are called 'fingerprints' and involve calculations of the perceptual features of the media. The generation and recognition of a fingerprint is tied to Audible Magic's patented technology. At the time of our interview the company claimed to have 7 million signatures and to be adding 2,000 a day. In operation, the signature of a media content item passing through the network is compared to the registered signatures. Audible Magic works off-line to avoid a single point of failure in the network. Its appliances must separate out and ignore traffic that does not consist of media files.

As a cruder method of copyright policing, DPI can be used to recognize and pre-emptively block peer to peer file sharing protocols. For example, campus network firewalls can use DPI to simply prevent BitTorrent from working, or it can rate-limit it severely. This method relies on in-line architecture and thus must be able to process the entire network load and if it fails the network fails. While less complex in terms of the number of signatures it needs to maintain and recognize, this method does not distinguish between different uses of the P2P protocols; it relies on generalized pre-emption of one method of sharing files, rather than on accurate recognition of copyrighted files.

An alternative to DPI is the use of so-called "over the top" methods. These involve parties outside the ISP network using techniques to identify the IP addresses of users who are sharing files. The IP address must then be matched to the account of a specific user. This method, which is the kind relied upon by France's HADOPI/graduated response regime, is *not* an application of DPI.

6. Content control

DPI can be used to enforce state censorship. This can be done by blocking the URLs of forbidden sites, and by blocking or diverting any content that contains specific keywords or phrases. It can also be used by private networks to restrict access to content based on organizational policies. Governmental implementations of public censorship policies typically need to centralize certain aspects of the country's telecommunications and Internet service provider infrastructures so that the recognition and blocking functions are uniformly applied.

DPI *per se* is not required to block access to URLs; it can also be done through proxy servers. When used for censorship or blocking of URLs, DPI is no more or less accurate than other forms of URL blocking. The controlling factor is still the accuracy and currency of the list of URLs. Only when it comes to recognizing the actual content downloaded or displayed does DPI constitute a potential technical advance. But signature algorithms cannot be created to recognize vague concepts like 'obscenity' or 'illegality;' as noted in the prior discussion of copyright applications, DPI-based content recognition requires unique signatures for each individual item of content and registration of each of those signatures with the DPI appliance at the ISP side. Thus, pure reliance on DPI for content regulation would require registering a fingerprint for every known illegal image, banned book or prohibited movie, and installation of equipment capable of recognizing the fingerprint in every ISP.

The cost and difficulty of true content-based network censorship makes URL blocking the most common form of censorship. Western companies such as WebSense sell a web filter proxy and database containing millions of URLs, which must be continually updated. The URLs are classified, often using automated

means, into categories. Policy-based control allows network operators implementing the system to select categories of web sites to which to restrict access. The restrictions can vary by time of day and by the user groups to which it is applied. Rationales include not only productivity enhancement but also bandwidth conservation and reduction of exposure to legal liability (for, e.g., copyright infringement).

The Web is not the only target of content control. Email content filters based on an organization's compliance regulations or organizational policies can also be implemented. DPI engines can scan all incoming and outgoing emails for organization-defined keywords and phrases in the message, in according with a Data Loss Prevention approach as described above. DPI appliances can also be programmed to examine or restrict various document attachment types, or to block certain file types based on their extension, such as .exe, .p2p, .asp. Here again, content control overlaps with organizational security policies.

Convergence of function/convergence of control?

Will DPI evolve into a more integrated and comprehensive technology of public network surveillance and control? The answer to this question hinges on the way technological factors interact with economic, political, legal and regulatory factors.

As one might expect, network operators would benefit from economies of scope, and thus there is strong demand-side economic pressure for more integration of the functionalities ISPs want. On the other hand, DPI functions are resource-intensive; they require lots of memory and computer cycles. More importantly, some functions are optimally provided in-line, while others are best done off-line. It is difficult to combine all of the disparate functions on the same piece of hardware, and difficult if not impossible to optimize for all of them on an integrated platform. Issues of scalability also mean that functions that might be combined readily on a smaller enterprise network may not be feasibly combined on a larger-scale commercial ISP. And there are some functions that external parties want the ISPs to have, but the ISPs themselves don't want to have.

Nearly all vendors noted that (as of mid-2011) some convergence has been occurring in devices, feature sets, and vendors. Specifically, bandwidth management, service visibility and customer profiling are known to have strong complementarities. "Operators can't manage their network without service visibility," said one vendor. "[They] need to know which services are being used by which user groups before they can offer different pricing and service plans for different subscriber groups." These functions all rely primarily on recognition and notification. Another vendor claimed that bandwidth management, content filtering, and IDS/IPS were a "logical grouping." This implies a convergence of function around an inline recognition and manipulation capability. "Once a network gains visibility of packets it can shape things as an enterprise edge device." However, that vendor noted that only two to four of the six use cases readily integrate; "rarely or never" does one see all six of them together.

Notably, nearly all interviewees agreed that copyright policing was the hardest-to-integrate functionality. ISPs generally do not see it as something that benefits their business. Recognition of millions of specific media files could not be easily deployed together with the other functions. A vendor which aims to "provide a uniform policy control platform," stated flatly that it is not technically feasible to provide copyright policing in line via DPI, much less integrate it with other functions. That conclusion, of course, is contested by the vendors of specialized copyright appliances. But even copyright-oriented vendors

admitted that the functionality benefits the rights holders and not the ISPs, and thus its adoption is more a political/regulatory question than a business case.

Governmental surveillance also poses unique problems that make integration difficult. One vendor saw a basic incompatibility between the bandwidth management/service visibility functions and lawful intercept (LI) functionality. LI applications require buffering, reassembling and decoding the traffic, which introduces latency. Such latency cannot be tolerated in an inline implementation which has to look at and act on packets in real time. Also, the legal requirements surrounding governmental surveillance are so specific in each country that it poses difficulties for more generic solutions.

Conclusion

DPI is an enabling technology that allows network operators to recognize patterns in network traffic and make decisions about how to manipulate traffic or notify operators based on the patterns it is programmed to recognize. The patterns, known as 'signatures,' must be defined in advance and distributed in a timely manner to the DPI engines deployed on a network. Signatures must be constantly updated. The recognition process is subject to false positives, false negatives and to evasion. The database of signatures is constantly expanding. The universe of potential patterns is as enormous as the universe of content, services, and applications running on the Internet. This fact deflates both utopian and dystopian visions of total network surveillance and control. It is very difficult to integrate all DPI applications, though we see growing convergence around use cases related to bandwidth management, monetization and user profiling on the one hand, and use cases related to security, content control and governmental surveillance on the other. On the Internet, network management responsibilities, and thus the scope and scale of DPI deployments, are distributed among many operators and thus the policies and practices applied are highly varied. The most intrusive and controlling applications require either small-scale organizational networks or highly centralized government control and/or operation of the public networks.

References

Ahn, S., H. Hong, et al. (2010). "A Hardware-Efficient Pattern Matching Architecture Using Process Element Tree for Deep Packet Inspection." IEEE Transactions on Communications **E93b**(9): 2440-2442.

Aho, A. V. and M. J. Corasick (1975). "Efficient string matching: An aid to bibliographic search." Communications of the ACM **18**(6): 333-340.

Bremler-Barr, A., Y. Harchol, et al. (2011). Space-time tradeoffs in Software-based Deep Packet Inspection. IEEE International Conference on High Performance Switching and Routing (IEEE HPSR), Belgrade, Serbia, http://www.cs.huji.ac.il/~dhay/publications/BHH11.pdf

Brey, P. (2005). Artifacts as social agents. Inside the Politics of Technology: Agency and Normativity in the Co-Production of Technology and Society. H. Harbers. Amsterdam, Amsterdam University Press**:** 61-84.

Huang, K., D. F. Zhang, et al. (2010). "Accelerating the bit-split string matching algorithm using Bloom filters." Computer Communications **33**(15): 1785-1794.

Kim, H., H. S. Kim, et al. (2010). "A Memory-Efficient Pattern Matching with Hardware-Based Bit-Split String Matchers for Deep Packet Inspection." Ieice Transactions on Communications **E93b**(2): 396-398.

Law, J. (1992). "Notes on the Theory of the Actor-Network: Ordering, Strategy, and Heterogeneity." Systems Practice **5**(4): 379-393.

Lin, Y. D., K. K. Tseng, et al. (2007). "A platform-based SoC design and implementation of scalable automaton matching for deep packet inspection." Journal of Systems Architecture **53**(12): 937-950.

Proch, D. and R. Truesdell (2009). Plumb the depths of deep packet inspection. Electronic Design. **57:** 47-50.

Riley, C. and B. Scott (2009). Deep Packet Inspection: The end of the internet as we know it? Washington D.C., Free Press.

Weinschenk, C. (2007). Deep Packet Inspection and Beyond: Whatcha Got There? Communications Technology.