A Hybrid Asymmetric Traffic Classifier for Deep Packet Inspection Systems with Route Asymmetry

Kasim Oztoprak
Department of Computer Engineering
KTO Karatay University, Konya, Turkey
Email: kasim.oztoprak@karatay.edu.tr

Mehmet Akif Yazici
Informatics Institute
Istanbul Technical University, Istanbul, Turkey
Email: yazicima@itu.edu.tr

***Abstract*—A flow is said to be asymmetrically routed if its packets follow separate paths for forward and reverse directions. Routing asymmetry leads to problems in flow identification, policy enforcement, quota management, traffic shaping etc. in DPI systems. There are two existing approaches to battle routing asymmetry: clustering and state sharing. The latter fails with stateless traffic, while clustering leads to large traffic overhead. We propose the Hybrid Asymmetric Traffic Classifier (HATC) method that merges the best aspects of the two existing methods. HATC is able to handle all types of asymmetric traffic with reduced overhead compared to clustering. Numerical evaluation of HATC using two real traffic traces is also presented.**

I. Introduction

The Internet is designed upon the philosophy of end-to-end communication. The routers and switches used to build the Internet move packets to their destinations without caring about their content; in fact, most devices cannot even access the content of the packets they process. In general, routers only process packets according to their destination addresses.

Technological advances in telecommunication systems enable us to monitor the data flows in real-time to take specific action according to the predefined rules through Deep Packet Inspection (DPI) systems. DPI systems sit in between the Internet and the users, and enable the service providers to inspect the payload of the packets. Processing the payload allows Internet Service Providers (ISP) to discriminate among traffic classes, and "implies the end of the end-to-end principle". This ability can be observed in two different approaches: it may either change the Internets future, or mark the end of end-to-end communication philosophy by bringing "intelligence" into the routers and switches by enabling the payload inspection [1].

Although DPI systems came into existence as security tools to protect enterprises from vulnerabilities with very fast response compared with end user protection systems in the beginning [2], nowadays it has various employment areas summarized as follows: i) traffic classification which is the main requirement of dynamic security systems (email spam, antivirus prevention, intrusion detection/prevention), ii) traffic shaping and quality of service (QoS) management, iii) quota management for billing systems, iv) payload processing that enables the ISPs to filter the content according to predefined (URL or pattern based) rules, and v) content caching. There is also increasing interest in using DPI systems as the base for revenue generation platforms [3], [4].

978-1-5090-5252-3/16/$31.00 © 2016 IEEE

The increasing bandwidth demand by the applications forces the ISPs to use multiple links to serve requested capacity as well as bringing routing redundancy. Routing decisions occur independently for each flow with the ability to follow different physical links even if the end points are the same. Adding the capability of load balancing and fault tolerance to the system also brings a side effect to the networking world, which is called "route asymmetry". Asymmetric routing can be defined as follows: "If packet streams between two end-points follow different physical links (rather than the same set of links) for forward and reverse directions, the routing is called asymmetric" [5]. However, a typical DPI system needs to see both request and response packets for a flow to bring a clear view of network traffic to the operators. Since routing asymmetry may cause the request and response packets associated with a particular flow to pass through distinct DPI devices, it leads to problems. The asymmetry problem is a common problem to all domains where DPI systems can be effectively used, in which the complexity increases while precision and accuracy decreases. For traffic identification, DPI systems should see both request and response packets for a flow to correctly identify the traffic pattern. However, asymmetry removes the visibility of both packets in a flow by the same device.

In this paper, we will address the traffic asymmetry problem of DPI systems by first describing the existing solutions in the currently deployed systems, namely *state sharing* (synchronization of state information) and *clustering*, and then propose a hybrid method that combines the best aspects of the two existing methods. State sharing [6] works only on TCP traffic, resulting in low accuracy in traffic classification due to non-TCP traffic while having low overhead in networks with little to no route fluctuations. On the other hand, clustering is proposed in [7] to solve the asymmetry for all traffic types at the cost of high traffic overhead.

A hybrid model, Hybrid Asymmetric Traffic Classifier (HATC), is proposed in this study by combining the best features of the state of the art solutions, which outperforms current approaches. The proposed method, HATC, brings low overhead for TCP traffic as in state sharing, whereas it works

for all traffic patterns as in clustering with acceptable traffic overhead.

The rest of the paper is organized as follows. In section 2, we give a brief summary of related literature in the routing asymmetry problem in DPI systems. Background definitions and formal description of the proposed solution to the asymmetry problem are given in section 3. Numerical validation and comparison of the proposed method to the existing schemes are presented in section 4. Conclusions are given in section 5.

II. Asymmetry Problem in DPI Systems

The Internet traffic volume is increasing an average of 22% every year. Online video use is increasing very fast and currently, 70% of the traffic is video traffic in the Internet. Globally, mobile data traffic is expected to increase eightfold in the term 2015-2020 [8]. Uncurbed increase in the volume of traffic resulted in the demand to classify the traffic in the Internet, and triggered the development of several classification tools [9] for this purpose.

The traditional network devices performing traffic analysis can only see layer 2 to layer 4 traffic while DPI systems enable full visibility of packet payload on application level [10]. This task was performed using port numbers of the applications to classify the traffic in early traffic classification tools, and lacked accuracy with applications using dynamic port numbers. The ability to read the application layer data allows the DPI systems to "understand" the type of data involved in the communication to gain useful intelligence about the network [11].

Currently, there are several open source traffic classifiers based on DPI techniques [12], [9]. Ref. [12] presents a comparison of the abilities of the tools on a mix of application traffic including streaming video and online games in addition to traditional Internet traffic. In that study, the accuracy of the tools is also investigated. There are also several manufacturers commercially producing traffic analysis and classification products in the DPI market. According to [13], Sandvine Inc. is the most innovative and widely deployed DPI vendor, followed by Allot Communications, Procera Networks and Huawei.

The service providers should be confident with their traffic identification solution, since the accuracy of the solution will affect all the operation on business intelligence, billing, and policy enforcement. In addition, the solution should not be limited to measuring the bytes exchanged, and should be able to compute parameters concerning quality of experience (QoE) [7]. The recent improvements in traffic classification allowed the service providers to gain insight on the quality and the duration of the streams in addition to the identification of their types. Successfully classifying network traffic helps the service providers to apply powerful network policies. According to ref. [7], a typical DPI system should be capable of performing i) traffic identification to detect the type of application, protocol, video provider, etc., ii) traffic measurements such as the duration, and the number of network events occurring throughout the duration, iii) measurement of advanced metrics to give insight like video QoE, iv) billing and charging by counting volume, duration, and events for all alternative scenarios, and v) policy enforcement to manage the network as expected by employing QoS marks, traffic shaping and rate-limiting, and session management.

The ability to identify and measure the traffic accurately intrinsically contradicts with the nature of broadband networks by design [14] due to routing asymmetry. The main problem of traffic identification in the case of asymmetry is the lack of visibility of all packets belonging to a flow at a single point. Traffic asymmetry can take two forms in a network: i) Flow asymmetry, where different packets of a flow can traverse different physical links, and ii) IP asymmetry, where multiple flows occurring from the same IP address traverse different links [7]. The latter is more relevant to a routing problem in the case of IP pairs while the former is more relevant to traffic classification and analysis [5]. Therefore, the focus of this paper is going to be flow asymmetry.

The increase in the use of encapsulation, tunneling and encryption also affects traffic classification. While encrypting traffic hides the content of the traffic, it is still possible to detect and identify the type of traffic by utilizing statistical traffic analysis and behavioral methods. Traffic asymmetry is again a problem in accurately identifying tunneled and encrypted traffic. In addition, asymmetric routing lowers the precision of statistical classifiers by increasing the false positives [15].

The impact of traffic asymmetry is not limited only to DPI systems, all network based security systems and transparent caching systems are also affected. Source routing overcomes the asymmetry problem at the expense of scalability and routing redundancy, therefore, is not considered in this paper. In the next section, existing solutions to the routing asymmetry problem for traffic classification systems will be covered, and the Hybrid Asymmetric Traffic Classifier model is introduced.

III. The Hybrid Asymmetric Traffic Classifier (HATC)

The proposed method, HATC, combines the best aspects of the two existing schemes. Therefore, we start with describing these methods. Then, we describe HATC.

A. Existing Asymmetry Handling Methods

There are several approaches to overcome the asymmetry problem. However, currently the only fully functional solution is the removal of asymmetry. In order to remove the asymmetry, a single box should cover either all the links, or the asymmetric part of the traffic should be transferred to the node that the request has originated from [7]. As it is not feasible and/or possible for a single box to cover all links, there should be other mechanisms to eliminate the asymmetry. Two existing proposals for this mechanism are summarized below.

In Clustering, the proposed method in [7], the asymmetric traffic is transported entirely back to the DPI box the original request associated with the flow has originated from. Clustering method is sketched in Fig. 1 for a very simple set-up, and summarized below:

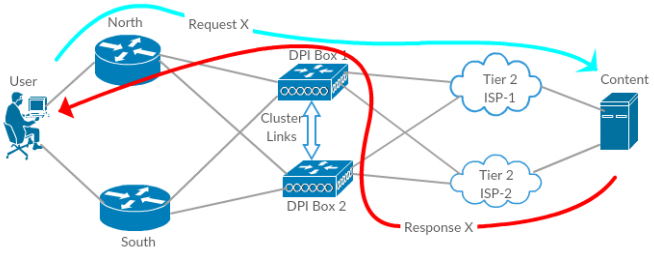

Fig. 1. A typical Clustering Scenario with two DPI Devices

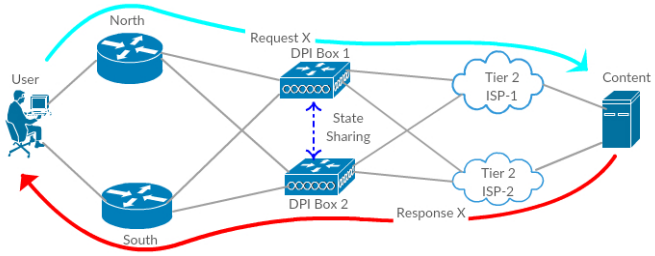

Fig. 2. A typical State Sharing Scenario with two DPI Devices

1) Download request is initiated by the user through *North Router* → *DPI Box 1* → *ISP-1* path with *Request X*. *DPI Box 1* is assigned as the "cluster manager" for the flow, *Flow X*.
2) *Response X* for *Request X* comes back through, say, *ISP-2* → *DPI Box 2* → *South Router* path.
3) No information exists on *DPI Box 2* for *Flow X*.
4) *DPI Box 2* asks to the clustered devices (*DPI Box 1* in this scenario) for the cluster manager for *Flow X*.
5) *DPI Box 1*, which is the cluster manager, responds to *DPI Box 2* about *Flow X* by sharing the 5-tuple: source IP address, destination IP address, source port number, destination port number, and the protocol ID.
6) *DPI Box 2* forwards *Response X* and the rest of the traffic belonging to *Flow X* to *DPI Box 1* and then, *DPI Box 1* delivers the data through *North Router* to the subscriber.

In this approach, the cost of transporting asymmetric traffic within the cluster and the latency incurred by the additional hop are both concerns to be considered. For a typical telecommunication operator, the cost of the links should not be the concern unless the rate of the asymmetry is higher than accepted thresholds. The latency incurred by traversing the response path does not matter much for a user since the additional delay would be in the order of milliseconds, whereas the total latency of the packets are in the order of seconds. In addition to the cost of transporting asymmetric traffic, the cost of cluster management (deciding which flow to reroute through which DPI boxes, as well as policy enforcement and related decisions) should also be considered. The number of DPI devices to be clustered should be below a certain limit as a full mesh between them is required. Moreover, the intended packet flow is altered, i.e. the routes computed by the routing devices are changed by the DPI devices, which may lead to performance degradation in terms of load balancing if it was indeed a parameter in the routing process. The effect of route change becomes more severe with increased asymmetry ratio, unless the DPI devices involved exchange similar amounts of asymmetric traffic reciprocally somehow. On the other hand, the advantage of this approach is that it works for all possible scenarios and traffic types.

According to ref. [5], the asymmetry ratio is much higher at the core compared to the access (edge). It can be seen from [16] that the asymmetry is around 80% in the core of the network. As clustering leads to traffic overhead proportional to the asymmetry level between DPI devices, such systems should be deployed as close to the edge as possible to minimize the asymmetry, and thus the redundant traffic.

The second approach, namely state sharing, is similar to clustering, but, rather than transporting all the response packets, only the first two packets of the session are queried from all DPI devices and then the information is shared by both DPI devices associated with the flow. State sharing, or alternatively termed flow synchronization, has been described in [17] in the context of network intrusion detection and prevention systems, and has been applied to DPI systems in [6]. Kim at al. [6] presents different approaches to overcome asymmetry problem based on traffic arbitration and clustering techniques, and offer their "flow synchronization" model, which significantly reduces the overhead.

In state sharing, all DPI devices engaged by a flow share their knowledge about the traffic. Thus, the method works for TCP, which is a stateful protocol. However, it fails with UDP flows, DNS messages etc., in which there is no state to be shared. On the other hand, the obvious advantage of state sharing is reduced traffic duplication. According to ref. [7], the overhead for state sharing is typically between 2% and 6% of the total traffic, which is much less than the typical asymmetry levels.

The DPI devices involved in the state sharing of a flow build an "alliance". This method is sketched in Fig. 2, and summarized below:

1) Download request is initiated by the user through *North Router* → *DPI Box 1* → *ISP-1* path with *Request X*. *DPI Box 1* is assigned as the "master node" for the flow, *Flow X*, to inform the policy enforcements to further DPI boxes that happen to carry *Flow X*.
2) The response for *Request X*, *Response X*, comes back through, say, *ISP-2* → *DPI Box 2* → *South Router* path.
3) No information exists on *DPI Box 2* for *Flow X*.
4) *DPI Box 2* broadcasts to all devices in the alliance to find out the master node for *Flow X*.
5) *DPI Box 1*, which is the master node, responds to *DPI Box 2* about *Flow X* by sharing the 5-tuple: source IP address, destination IP address, source port number, destination port number, and the protocol ID; along with any policy information regarding *Flow X*.
6) *DPI Box 2* forwards the first two packets of *Flow X* to the master node, *DPI Box 1*. At this point, *DPI Box 1*

TABLE I
STATE SHARING VS. CLUSTERING: ADVANTAGES AND DISADVANTAGES

	Advantages	Disadvantages
State sharing	Very little traffic overhead Keeps computed route intact	Works only with TCP
Clustering	Works for all scenarios	Heavy traffic overhead Alters computed route

and *DPI Box 2* have knowledge of the flow.

7) *DPI Box 2* delivers the data through *South Router* to the subscriber.

In addition to traffic identification, another function of DPI systems is policy enforcement, which becomes much more difficult in the presence of route fluctuations as well as asymmetry. For example, say a subscriber has 10 MB of remaining quota, but starts a download with the size of 20 MB. If a path change occurs after every 5 MB of download, then state sharing mechanism would fail to stop the download session after the 10 MB quota expires unless there is a mechanism to disseminate flow information as needed. Therefore, the first DPI box to handle a flow is assigned as the master node of that flow to undertake this responsibility.

Obviously, the overhead incurred by state sharing is very little compared to the clustering method. The only contributors are the broadcasted query packets which involve 5-tuple that represents the flow, the response packet again with the 5-tuple along with policy information, and the first two packets of the flow that has been kept by the master node. The 5-tuple would comfortably fit in a minimum sized 64 byte-packet, whereas a 128 byte-packet would be sufficient for the master node's response. The first two packets of the flow, which hold application protocol identifiers, are sufficient for flow identification. On the other hand, state sharing only works for TCP traffic. Regardless of the application and data carried, it is not applicable to stateless traffic.

It should be noted that the method described here, which is used by many vendors such as Procera Networks [18], deviates from what is presented in ref. [7] as state sharing. In the method described in ref. [7], the DPI box processing the request broadcasts this information to the entire set of DPI devices, and whenever asymmetric traffic is detected, the DPI box receiving the response knows which DPI box to share state information. In this scenario, if the route changes more than once during the lifetime of the flow, i.e. *fluctuates* at least twice, there is a risk of the DPI box that lies on the third route having no idea which boxes to share the state with, since it will probably forget about the flow after a time-out mechanism in order not bloat its own state memory. Hence, it is claimed in ref. [7] that state sharing can be effectively used only in scenarios in which traffic can only take a maximum of two paths through the network. However, the method we describe does not suffer from such a limitation.

It should also be noted that clustering technique includes some kind of state sharing mechanism to coordinate the DPI boxes in the cluster and to decide to reroute which flow to which DPI box. However, the overhead incurred by cluster management is negligible compared to the overhead caused by asymmetric data rerouting.

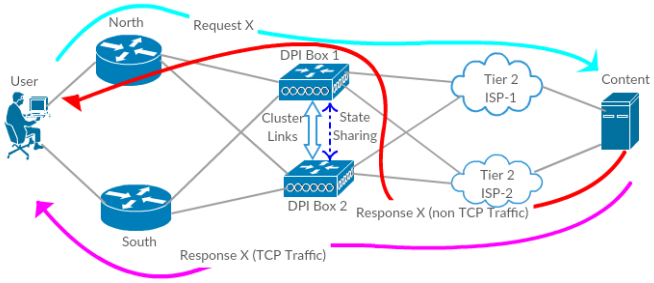

Fig. 3. A typical Hybrid Asymmetric Traffic Classifier (HATC) Scenario with two DPI Devices

B. Hybrid Asymmetric Traffic Classifier (HATC) Algorithm

The advantages and disadvantages of the two existing methods for battling route asymmetry is summarized in Table I. It is clear that clustering achieves maximum traffic identification accuracy limited only by the traffic identification algorithm (see [9] for a number of such algorithms, some of which achieve almost perfect identification under certain scenarios, and others such as [19]) while consuming a huge amount of resources. On the other hand, state sharing dramatically reduces overhead, but fails with stateless non-TCP traffic. Moreover, dynamic routing may give rise to route fluctuations (changes in the traversed links by packets of a flow), in which portions of a single flow follow separate routes, and hence, traverses multiple DPI devices throughout the lifetime of the flow. This situation mitigates the success of the traffic identification.

In light of these, we propose the Hybrid Asymmetric Traffic Classifier (HATC) that follows state sharing whenever possible, i.e. in case of TCP traffic, with no alteration in the computed route of a flow, whereas employs clustering when state sharing is not possible.

The proposed method, HATC, is sketched in Fig. 3, and summarized below:

1) Download request is initiated by the user through *North Router* → *DPI Box 1* → *ISP-1* path with *Request X*. *DPI Box 1* is assigned as the "master node" for the flow, *Flow X*.
2) The response for *Request X*, *Response X*, comes back through, say, *ISP-2* → *DPI Box 2* → *South Router* path.
3) No information exists on *DPI Box 2* for *Flow X*.
   a) If the traffic is maintained by a TCP session, then:
      i) *DPI Box 2* broadcasts to all devices in the alliance to find out the master node for *Flow X*.
      ii) *DPI Box 1*, which is the master node, responds to *DPI Box 2* about *Flow X* by sharing the 5-tuple: source IP address, destination IP address,

TABLE II
DEMOGRAPHIC STRUCTURE OF SUNET TRAFFIC TRACES.

	GigaSUNET 2006			OptoSUNET 2009		
	Total	TCP	non-TCP	Total	TCP	non-TCP
Number of Flows (million)	3.8	1.79	2.01	36	6.55	29.45
Share (percentage)	100	47.04	52.96	100	18.20	81.80
Number of Packets (million)	369.6	348.53	21.07	883.2	717.34	165.86
Share (percentage)	100	94.30	5.70	100	81.22	18.78
Total Bytes (gigabyte)	243.75	237.75	6.00	508.59	446.70	61.90
Share (percentage)	100	97.54	2.46	100	87.83	12.17
Average Packets / Flow	97.26	194.98	10.47	24.53	109.48	5.63
Average Bytes / Flow	68875	146418	3199	15169	83348	2257
Average Bytes / Packet	708	732	306	618	669	401

source port number, destination port number, and the protocol ID; along with any policy information regarding *Flow X*.

iii) *DPI Box 2* forwards the first two packets of *Flow X* to the master node, *DPI Box 1*.

iv) *DPI Box 2* delivers the data through *South Router* to the subscriber.

b) If the traffic is maintained by a non-TCP session, then:

i) *DPI Box 2* asks to the clustered devices for the master node for *Flow X*.

ii) *DPI Box 1*, which is the master node, responds to *DPI Box 2* about *Flow X* by sharing the 5-tuple: source IP address, destination IP address, source port number, destination port number, and the protocol ID.

iii) *DPI Box 2* forwards *Response X* and the rest of the traffic belonging to *Flow X* to *DPI Box 1* and then, *DPI Box 1* delivers the data through *North Router* to the subscriber..

In case of route fluctuation, the DPI box that starts receiving the data broadcasts a query to the other boxes to find out the master node. Then, if the flow is a non-TCP flow, the DPI box receiving data reroutes the flow into the master node. Otherwise, state sharing packets are exchanged between the two DPI boxes. Note that with TCP traffic, the assigned "master node" handles managerial tasks related to the flow such as informing new DPI boxes that starts carrying the flow about it and policy enforcement, whereas cluster management is performed as in clustering with non-TCP flows.

IV. NUMERICAL VALIDATION

In this section, we provide computational results based on real traffic traces to demonstrate the gains of HATC. We studied two sets of traces taken first on Swedish Tier 2 backbone for universities (GigaSUNET) in April 2006, and secondly on one of the links in the OptoSUNET structure, which is a Tier2–Tier1 connection, in January 2009. The traces are provided in [20] and analyzed in terms of symmetry in [16]. The demographic structure of the traces in terms of the number of flows, packets and bytes and the proportions of TCP traffic are provided in Table II. Also, the asymmetry profile of the traces are presented in Table III.

TABLE III
PERCENTAGES OF ASYMMETRIC TRAFFIC FOR SUNET TRACES.

	GigaSUNET 2006			OptoSUNET 2009		
	Flow	Packet	Byte	Flow	Packet	Byte
Whole IP traffic	45.67	32.76	25.94	92.85	73.68	66.19
Whole TCP traffic	45.09	32.77	25.38	90.74	74.01	65.53
TCP trf., data only	31.39	24.76	24.87	89.54	73.79	65.49
Non-TCP traffic	46.19	32.59	48.13	93.32	72.25	70.94

There are a number of observations that can be inferred from Tables II and III. Although these are just two samples, one can argue that they reflect the general trend of non-TCP traffic being on the rise due to the rise of multimedia applications in general. Moreover, the GigaSUNET trace was taken on a Tier 2 link, whereas the OptoSUNET trace was taken on a Tier2–Tier1 connection. This distinction is reflected on the asymmetry profiles, OptoSUNET trace displaying very high asymmetry. Another point to notice is the difference between the asymmetry levels in the GigaSUNET trace between the whole TCP traffic and data-only TCP traffic, from which control packets has been removed. In flow level, there is a dramatic difference between the two (45.09% versus 31.39%), whereas the difference is becomes marginal in byte level (25.38% versus 24.87%). This is due to the fact that a significant amount of the control traffic belongs to failed TCP flows, comprising only control packets, which make up a tiny amount of traffic in terms of bytes compared to TCP data traffic.

After these observations, we will compare the two existing mechanisms and HATC over the traces. First of all, state sharing will obviously flat-out fail with non-TCP traffic. This means that, barring identification failures stemming from the actual flow identification scheme employed, 52.96% of the flows for GigaSUNET, and a staggering 81.80% of the flows for OptoSUNET will go unidentified. However, these failure ratios are 2.46% and 12.17%, respectively, for the two traces, in byte level. The distinction between the two approaches (flow level and byte level) depends on the vantage point. That is to say, number of flows can be considered to be

indicative of client population, and hence the more number of unidentified flows, the more number of either unsatisfied clients (due to possible failure to meet their specific demands), or failures to enforce policy. On the other hand, number of bytes obviously is indicative of the actual volume of the traffic. In this aspect, even though TCP traffic is not in the majority in terms of the number of flows, it is still decisively dominant in terms of actual traffic volume. Therefore, the proportion of the traffic volume that goes unidentified, and thus cannot be shaped/policed, seems to be tolerable, despite the seemingly steep rising trend of UDP traffic.

Regarding clustering, the concern is the overhead traffic among the DPI boxes. Regardless of the application/transport protocol, asymmetry will translate into overhead traffic. In order to find out the overhead, we have to make a set of assumptions about the ratio of the asymmetric traffic that actually goes through DPI boxes that are not the cluster managers of the associated flows. First, we assume that the asymmetry will not be stemming from the client side in general. This means that uplink traffic will not fluctuate away from the cluster manager, and the traffic that needs rerouting comes from downlink side. Notice that GigaSUNET 2006 trace was taken on a Tier 2 link. We do not have a precise ratio as to the uplink/downlink traffic ratio on this trace. However, a similar link on the Turkish ISP, Türk Telekom, has an uplink/downlink traffic ratio of 1/4 [21]. Therefore, we will assume a similar ratio with the GigaSUNET 2006 trace, which means that 80% of the total traffic will be rerouted inside the cluster. OptoSUNET 2009 trace, on the other hand, is taken on a Tier 2–Tier 1 connection, which is closer to the core. Therefore, it would be reasonable to assume a lower uplink/downlink traffic ratio for OptoSUNET 2009, which we assume to be 1/3. That means 75% of the total traffic will be rerouted inside the cluster for the OptoSUNET 2009 trace.

The overhead for clustering should also include the query and cluster management packets. For this purpose, we assume that 64-byte query packets are broadcasted in case either route fluctuation occurs or a non-TCP flow starts. TCP flow initiations would not require this broadcast as the start of a TCP session can be detected from the SYN packet. We also assume a 128-byte packet carrying the necessary flow and policy information is sent to the DPI box handling the asymmetric traffic from the cluster manager.

We compute the overhead traffic in clustering for the two traces as follows. First, 80% (75%) of the traffic will be rerouted inside the cluster for the GigaSUNET 2006 (OptoSUNET 2009) trace, which we will call as the "rerouting overhead", O_R. In addition, for every non-TCP flow, there will be a query broadcast. In case of route fluctuation, we will have an additional query broadcast along with the 128-byte response packet. We can write the overhead due to these management packets as

$$O_M = f \times [64 \times (N-1) \times u + r \times (64 \times (N-1) + 128)]$$

in terms of bytes, where N denotes the number of DPI boxes in the system, u denotes a binary variable which is 1 if the flow is a non-TCP flow and 0 otherwise, r denotes the number of route fluctuations, and f is the number of flows. Then, the overhead in terms of bytes can be expressed as

$$O = O_R + O_M.$$

Consulting Table II, we see that non-TCP flows are generally packet-wise short flows. So, we will assume that only a single route fluctuation occurs in such flows. TCP flows, on the other hand, have in the excess of one hundred packets. However, remembering that TCP is a window protocol and most of its packets are transmitted in bursts, we can assume also that the number of route fluctuations in TCP flows are limited to a few. Considering these, we will assume that 2/3 of the TCP flows that experience route fluctuations will have only a single fluctuation, whereas the remaining will have two fluctuations. Lastly, we assume a system with $N = 10$ DPI boxes. Under these assumptions, the overhead of clustering for the GigaSUNET 2006 trace turns out as

$$\begin{aligned}
O_R &= 243.75 \times 0.8 \times 0.2594 \\
&= 50.583 \text{ GB} \\
O_M &= 2.01 \times 10^6 \times 0.4619 \times \\
&\qquad [64 \times (10-1) + 64 \times (10-1) + 128] \\
&\quad + 1.79 \times 10^6 \times 0.4509 \times \\
&\qquad [64 \times (10-1) + 128] \times (2/3 \times 1 + 1/3 \times 2) \\
&= 1.900 \text{ GB} \\
O &= O_R + O_M = 52.483 \text{ GB}
\end{aligned}$$

which comes to 21.53% of the total traffic. The overhead for the OptoSUNET 2009 trace can be computed in a similar manner, and turns out to be 252.48 GB for rerouting overhead, 39.80 GB for management overhead and thus 292.28 GB for total overhead, which is 57.47% of the total traffic.

Similarly for HATC, there are two sources of overhead traffic: i) rerouted non-TCP traffic, and ii) state sharing and query packets for the TCP traffic. Essentially, the main difference is that the TCP traffic is not rerouted and two TCP packets are forwarded instead of the asymmetric TCP traffic. The contribution of the first item is computed similar to the overhead computation in clustering. To compute the contribution of the second item, we included the following components:

- Query for the master node: A packet of size 64 bytes, broadcasted to all the DPI boxes.
- Response from the master node: A packet of size 128 bytes.
- The first two packets of the flow: Average packet size is used from the traces. These will be forwarded to the master node if they appear at a different DPI box due to asymmetry. Inspection of Table II tells us that TCP flows in GigaSUNET 2006 trace are equivalent to 273.168 maximum segment size units of 536 B, the default value specified in RFC 879 [22], [23]. Therefore, we can assume 9 transmission windows to complete a flow, considering the "slow start" mechanism ignoring

TABLE IV
TRAFFIC OVERHEAD PERCENTAGES FOR BOTH SUNET TRACES IN BYTE LEVEL FOR CLUSTERING AND HATC.

	Traffic overhead (%)	
	GigaSUNET 2006	OptoSUNET 2009
Clustering	21.53	57.47
HATC	1.80	14.59

packet losses. Similarly, TCP flows in OptoSUNET 2009 trace are equivalent to 155.500 maximum segment sizes and would require 7 transmission windows. Therefore, the probability of the first two packets being forwarded due to asymmetry would be 1/9 (1/7) for the GigaSUNET 2006 (OptoSUNET 2009) trace for a single fluctuation, and double that for two fluctuations.

Moreover, if route fluctuation happens during the lifetime of the flow, the management packets will be exchanged once more per each fluctuation. As a comparison, we compute the overhead of HATC in the same scenario as before with 10 DPI boxes.

$$\begin{aligned} O_R &= 6.00 \times 0.8 \times 0.4813 \\ &= 2.310 \text{ GB} \\ O_M &= 2.01 \times 10^6 \times 0.4619 \times \\ &\quad [64 \times (10-1) + 64 \times (10-1) + 128] \\ &\quad + 1.79 \times 10^6 \times 0.4509 \times \\ &\quad [64 \times (10-1) + 128] \times (2/3 \times 1 + 1/3 \times 2) \\ &\quad + 1.79 \times 10^6 \times 0.4509 \times \\ &\quad [732 \times 2] \times (2/3 \times 1/9 + 1/3 \times 2/9) \\ &= 2.071 \text{ GB} \\ O &= O_R + O_M = 4.381 \text{ GB} \end{aligned}$$

This makes 1.80% of the total traffic. Again, the overhead for the OptoSUNET 2009 trace can be computed in a similar manner, and turns out to be 32.93 GB for rerouting overhead, 41.28 GB for management overhead, and 74.21 GB for total overhead, which is 14.59% of the total traffic.

Lastly, we present a simulation-based comparison of clustering and HATC in terms of overhead traffic percentage incurred. We made use of a stand-alone simulation program we wrote in Matlab to simulate both algorithms. We assumed an uplink/downlink traffic ratio of 1/4, 10 DPI boxes and 10^4 total flows. TCP flow lengths in packets are assumed to be geometrically distributed, shifted to have a minimum of 75 packets, and the parameter of the distribution is picked in such a way that the mean number of packets in a TCP flow is 125. Non-TCP flows follow a similar distribution with a minimum of 3 packets per flow and 7 packets on the average. The packet sizes in bytes for both kinds of flows also follow similar distributions where the minimum of a TCP (non-TCP) packet size is 500 (250) bytes whereas the average is 600 (350) bytes. When a flow is known to be asymmetric, the number of fluctuations are computed as follows. For non-TCP flows, each packet has a uniform and independent probability of 0.1 to introduce a new fluctuation. Similarly, with TCP flows, each congestion control window has 0.1 probability of causing an additional fluctuation, where TCP flows are always assumed to stay in the slow start phase. These parameters were selected so as to have a similar setting with the two traffic traces studied earlier.

TABLE V
TRAFFIC OVERHEAD PERCENTAGES IN BYTE LEVEL FOR CLUSTERING AND HATC UNDER THE SIMULATION SCENARIOS AND VARYING TCP TRAFFIC SHARE AND ASYMMETRY RATIOS.

	TCP trf. share %	Asymmetry ratio 0.5	0.6	0.7	0.8	0.9
Clustering	20	44.82	54.39	63.06	72.25	80.63
	40	42.60	51.12	59.57	67.99	76.41
	60	41.72	49.89	58.34	66.91	74.78
	80	41.35	49.55	57.42	65.46	73.92
	90	40.89	49.23	57.05	65.37	73.68
HATC	20	10.92	13.07	15.21	17.26	19.36
	40	5.67	6.75	7.98	9.09	10.21
	60	3.77	4.51	5.23	6.04	6.75
	80	2.76	3.37	3.90	4.41	4.95
	90	2.42	2.92	3.46	3.90	4.40

The overhead traffic percentages for both algorithms are given in Table V for asymmetry ratios varying in the set $\{0.5, 0.6, 0.7, 0.8, 0.9\}$, and TCP traffic share in flows are varied in the set $\{0.2, 0.4, 0.6, 0.8, 0.9\}$. The improvement provided by HATC over clustering is obvious. Not surprisingly, the performance of HATC gets better with increasing TCP traffic share, and both algorithms gets better with decreasing asymmetry. However, even in the most severe scenario, the overhead due to HATC is still below 20%, which is less than half of what clustering causes in the most advantageous scenario in the simulation.

V. CONCLUSIONS AND FUTURE WORK

In this study, we propose the Hybrid Asymmetric Traffic Classifier (HATC) method in order to achieve traffic identification as well as policy enforcement in the presence of routing asymmetry, which is a problem that arises when the packets of a traffic flow follow separate paths for forward and reverse directions. Routing asymmetry leads to problems in flow identification, policy enforcement, quota management, traffic shaping etc. in DPI systems. There are two existing main approaches to battle routing asymmetry: clustering and state sharing.

In the clustering method, asymmetric traffic is physically rerouted between the DPI boxes. Clustering completely solves the routing asymmetry problem, however leads to large volumes of overhead traffic between DPI boxes. ISPs maintaining the DPI system would need to set up a full mesh with high capacity links between the DPI boxes. On the other hand, the state sharing (or alternatively named as flow synchronization) method only forwards packets with information relevant to the flow between DPI box pairs, hence almost eradicating the need of very high capacity on the links between the DPI boxes, and can work on DPI systems with less connectivity than a full

mesh. This advantage, however, comes at the expense of the ability of handling all kinds of traffic as in clustering, since state sharing can not work on stateless flows such as UDP.

The proposed HATC method is therefore a hybrid solution that merges the best aspects of the two existing methods. If the asymmetric flow is a TCP flow, state sharing is employed as it leads to very little overhead traffic. On the other hand, when the asymmetric flow is non-TCP so that state sharing would fail, clustering mechanism is used. In this manner, all types of asymmetric traffic can be handled with reduced overhead compared to clustering.

We also provide numerical evaluation using two real traffic traces taken from Swedish University Backbone Network (SUNET) to demonstrate the gains of HATC. State sharing fails in 52.96% and 81.80% of the flows, and 2.46% and 12.17% of the bytes carried, for the two traces whereas HATC has no such problem. Clustering on the other hand, while working with every scenario, causes 21.53% and 57.47% additional overhead traffic in terms of bytes for the two traces. In comparison, the traffic overhead levels for HATC are 1.80% and 14.59%, respectively, for the same traces, which shows that HATC reduces the traffic overhead drastically.

Future studies will focus on incorporating traffic identification algorithms into the classification algorithm to obtain a holistic solution to the asymmetry problem. Moreover, as telecommunication systems are evolving to become Software Defined Networks (SDN) in general, traffic identification and classification algorithms should be designed to work cooperatively with SDN architectures, which is another aspect future work can focus.

References

[1] R. Bendrath, "Global technology trends and national regulation: Explaining variation in the governance of deep packet inspection," in *International Studies Annual Convention, February*, 2009, pp. 15–18.

[2] F. Yu, "High speed deep packet inspection with hardware support," Ph.D. dissertation, University of California, Berkeley, 2006.

[3] K. Oztoprak, "Subscriber profiling for connection service providers by considering individuals and different timeframes," *IEICE Transactions on Communications*, vol. 99, no. 6, pp. 1353–1361, 2016.

[4] ——, "Profiling subscribers according to their Internet usage characteristics and behaviors," in *Big Data, 2015 IEEE International Conference on*. IEEE, Oct. 2015, pp. 1492–1499.

[5] W. John, M. Dusi, and K. C. Claffy, "Estimating routing symmetry on single links by passive flow measurements," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. ACM, 2010, pp. 473–478.

[6] N. Kim, G. Choi, and J. Choi, "A scalable carrier-grade DPI system architecture using synchronization of flow information," *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 10, pp. 1834–1848, 2014.

[7] "Applying network policy control to asymmetric traffic: Considerations and solutions," Sandvine Incorporated ULC White Paper, online: https://www.sandvine.com/downloads/general/whitepapers/applying-network-policy-control-to-asymmetric-traffic.pdf, 2015, accessed: 8 September 2016.

[8] "Cisco visual networking index: Forecast and methodology, 2015-2020," Cisco White Paper, online: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf, 2016, accessed: 8 September 2016.

[9] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Comparison of deep packet inspection (DPI) tools for traffic classification," Universitat Politècnica de Catalunya, Tech. Rep., Jun. 2013, (UPC-DAC-RR-CBA-2013-3 ed.) Available online: http://vbn.aau.dk/ws/files/78068418/report.pdf. Accessed: 8 September 2016.

[10] S.-K. Park, S.-S. Yoon, and J.-K. Lee, "PCI-based high-speed internet control and analysis platform for application monitoring and control," in *Proceedings on the International Conference on Internet Computing (ICOMP)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012.

[11] T. Margoni and M. Perry, "Deep pockets, packets, and harbors," *Ohio St. LJ*, vol. 74, pp. 1195–1216, 2013.

[12] S. Alcock and R. Nelson, "Measuring the accuracy of open-source payload-based traffic classifiers using popular internet applications," in *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*. IEEE, Oct. 2013, pp. 956–963.

[13] "Next Generation Optimization, DPI, and Policy," online: https://www.abiresearch.com/market-research/product/1014039-next-generation-optimization-dpi-and-polic/, accessed: 8 September 2016.

[14] "Identifying and measuring Internet traffic: Techniques and considerations," Sandvine Incorporated ULC White Paper, online: https://www.sandvine.com/downloads/general/whitepapers/identifying-and-measuring-internet-traffic.pdf, 2015, accessed: 8 September 2016.

[15] M. Crotti, F. Gringoli, and L. Salgarelli, "Impact of asymmetric routing on statistical traffic classification," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–8.

[16] M. Dusi and W. John, "Observing routing asymmetry in Internet traffic," online: http://www.caida.org/research/traffic-analysis/asymmetry/, accessed: 8 September 2016.

[17] C. R. Meiners, J. Patel, E. Norige, E. Torng, and A. X. Liu, "Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems," in *Proceedings of the 19th USENIX conference on Security*. USENIX Association, 2010.

[18] "Procera Networks," online: http://www.proceranetworks.com/.

[19] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "Kiss: Stochastic packet inspection classifier for UDP traffic," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 5, pp. 1505–1515, 2010.

[20] "Internet Traffic Classification," online: http://www.caida.org/research/traffic-analysis/classification-overview/, accessed: 8 September 2016.

[21] "Türk Telekom," http://www.turktelekom.com.tr/tt/portal/About-TT/Company-Profile/About/, Private communication.

[22] J. Postel, "TCP maximum segment size and related topics," Internet Requests for Comments, IETF, RFC 879, November 1983, http://www.rfc-editor.org/rfc/rfc879.txt.

[23] D. Borman, "TCP options and maximum segment size (MSS)," Internet Requests for Comments, IETF, RFC 6691, July 2012, http://www.rfc-editor.org/rfc/rfc6691.txt.