Malicious MPLS Policy Engine Reconnaissance

Abdulrahman Al-Mutairi, Stephen Wolthusen

Malicious MPLS Policy Engine Reconnaissance

Abdulrahman Al-Mutairi[2] and Stephen Wolthusen[1,2]

[1] Norwegian Information Security Laboratory,
Department of Computer Science,
Gjøvik University College, Norway
[2] Information Security Group,
Department of Mathematics,
Royal Holloway, University of London, UK
Email: ⟨Abdulrahman.Almutairi.2009, stephen.wolthusen⟩@rhul.ac.uk*

Abstract. *Multi-Protocol Label Switching* (MPLS) is widely used on telecommunications carrier and service provider backbone networks, complex network infrastructures, and also for the interconnection of distributed sites requiring guaranteed quality of service (QoS) and service levels such as the financial services sector, government and public safety, or control networks such as the electric power grid.
MPLS is a policy-based system wherein router behaviour is determined not only by the base protocols, but also by a set of further policies that network operators will typically wish not to reveal. However, sophisticated adversaries are known to conduct network reconnaissance years before executing actual attacks, and may also wish to conduct *deniable* attacks that may not be visible as such that appear as service degradation or which will cause re-configuration of paths in the interest of the attacker. In this paper we therefore describe a *probing algorithm* and a model of MPLS state space allowing an adversary to learn about the policies and policy state of an MPLS speaker. In spite of the restrictions on the adversary, our probing algorithm revealed the policy states of non-directly connected routers. Also, we analyse the confirmed information using a Bayesian network and provide simulative validation of our findings.

Keywords: Multi-protocol Label Switching, Real-Time Networks, Quality of Service, Reconnaissance, Bayesian networks

1 Introduction

The *Multi-Protocol Label Switching* (MPLS) protocol provides a highly efficient mechanism for packet forwarding based on a *label switching* approach that seeks to reduce the need for explicit per-packet routing in wide-area networks by pre-identifying optimum paths to the final destination, thereby allowing intermediate routers to forward information traffic based on a once-applied label rather than an explicit lookup at each intervening router. This is of interest not only for network operators seeking to improve the effective throughput of routers and

overhead required, but also particularly for applications where so-called *flows* can be identified. Consequently flows that share common characteristics such as source and destination as well as other service characteristics could be treated in the same way. By analysing flow requirements and characteristics, it is thus possible to also provide a defined *quality of service* (QoS) for a flow, typically through a process of resource reservation. This is crucial for network operators seeking to accommodate traffic on consolidated IP networks that may also be sensitive, e.g., to real-time characteristics.

Where adversaries seek to analyse and ultimately disrupt service availability such as by disabling and impeding links or routers, a first step will need to be the analysis of network behaviour, which is determined not only by the basic MPLS protocol and real-time or QoS extensions, but also by a number of policies. The revelation of the used policies may not be considered as a vulnerability, but that would make the attack more easy for an adversary and assist or enable the adversary to launch accurate attacks against the policy engines as this type of attacks is referred to as foot-printing attacks [1]. For example, such information would assist the attacker to estimate to what extent the manipulation of labels would propagate or the sensitivity of MPLS networks to sudden changes in specific MPLS nodes. A main purpose of this paper is therefore to study the ability of attackers to learn about the configured policies on MPLS routers whilst having access to limited resources using a limited and legitimate probing within the MPLS network.

The remainder of this paper is structured as follows: we review related work on MPLS security analyses and policy reverse engineering in section 2, followed by a description of the MPLS policy engine in general. A simplified policy model employed for subsequent analysis is introduced in section 3. We then provide a description of the policy state analysis framework in section 4 and study the validity and mapping of our model onto a simulated instantiation in section 5. Then, we introduce a probability model for the confirmed traces left by each of the MPLS policies as well as the relationships among MPLS policies themselves in section 6. We conclude with a brief discussion and summary of our findings as well as an outlook on on-going research in section 7.

2 Related Work

In policy-based protocols among peer networks, the policy under which a network operates must be considered sensitive as this may, e.g., reveal commercial information for operators or can have security implications as it allows adversaries to deliberately target policy behaviour. Research in this area has been largely limited to the exterior Border Gateway Protocol (BGP) [2] where a more state information is revealed in a larger body of security analysis [3–5].

However, few research studies have been conducted on MPLS, generally in the field of integrity and availability. The MPLS label distribution protocol (LDP) was analysed by Guernsey *et al.* [6]. Guernsey *et al.* demonstrated several exploits that may cause route modification, traffic injection and Denial-of-Service

(DoS) mainly by BGP update messages poisoning or directly injecting malicious traffic into Label Switched Paths (LSPs). Grayson *et al.* [7] provided a further analysis of MPLS security with special emphasis on the use of MPLS to realise Virtual Private Networks (VPNs). Mainly, the authors focused on route injection and traffic injection attacks and paid some attention to DoS-type attacks, but placed less emphasis on the reconnaissance and targeted quality of service (QoS) degradation resulting in policy-driven attacks that we are considering in this paper. It should be noted that DoS or integrity violations might not be the main objectives of attacks where the adversary aims to affect the QoS of the routed traffic. The failure to realise such facts in networks operation may have long-lasting impacts on QoS and the direction of flows that go far beyond transitive faults [8].

The main alternative for the MPLS control plane to LDP is the extension of existing protocols for signalling; this is realised both in the form of Traffic Engineering extension of Resource Reservation protocol (RSVP-TE) and Multi-Protocol Extension for BGP (MP-BGP). The security properties of RSVP-TE were studied by Spainhower *et al.* [11]. The authors demonstrated some reconnaissance and DoS attacks. The introduced reconnaissance attacks aim to reveal the record route object (RRO) in the reservation message that contains some topology information, e.g., core addresses as well as the identification of MPLS ingress. However, in our work we aim to reveal the MPLS nodes' states rather than the network topology.

The security properties of MP-BGP, on the other hand, were studied by Liorens and Serhouchni [12]. The authors introduced the notion of using Bayesian networks for defining an approach to penetrate VPNs in order to rank the VPNs perimeter and deciding the probability of the best VPN perimeter to ensure VPNs isolation and integrity in MP-BGP protocol. We are going to use the Bayesian network in slightly different way to demonstrate the probability of different MPLS policies and the relationships amongst them.

The analysis and reverse-engineering of inter-domain routing policies has, e.g., been studied by Machiraju and Katz [2] who proposed a technique for BGP routing policies' reverse engineering by examining the BGP updates in order to reveal local preferences used by Autonomous Systems (ASs). Similarly, Wang and Gao [13] introduced a method to characterise routing policies used in the Internet. Wang and Gao could infer the route preference that influences route selection in import policies by associating local preference values to the inferred relationships among ASs. In addition, the author could infer the export policies that are used for controlling the inbound traffic. Furthermore, Liang *et al.* [14] developed a model to infer routing policy for individual ISPs. Basically, Liang *et al.* aimed to abstract the policy patterns from BGP routing tables and then group the collected data for translation into the high-level policy objectives as a method of routing policy reverse engineering. Liang *et al.* claimed that the developed model achieves over 78.94% average accuracy in routing policy inference.

Siganos and Faloutsos [3] developed a tool (Nemesis) to infer business relationships of ASs by parsing and restoring the information found in Internet

Routing Registries (IRRs) in an easy relational database. Basically, the authors' methodology was to convert the simple text polices into equivalent link-level policies, infer the business relations of ASs (customers, peers and providers), then validate the results against the BGP routing updates to check the consistency of IRRs. Alternatively, Ming *et al.* [15] applied reverse engineering techniques in order to reveal the actions taken by certain ASs in response to false announcements in false Multiple Origin AS (MOAS) events using BGP updates. Ming *et al.* concluded that the bad announcements are not only arising from the originating AS, but other ASs took early actions to withdraw such bad announcements.

To the best of our knowledge, all of the existing studies on routing policy inference are based on BGP updates and mostly aim to reveal the import and export routing policies rather than the other policies that might affect the routing operation such as the MPLS policies, thereby making a direct application of these results difficult. Going beyond this, our aim is to reveal the more limited MPLS state information by analysing the actual effects on signalling behaviour.

3 MPLS Policy Engine

Network operators and service providers employ routing policies not only for the sake of efficiency, e.g., load balancing, but also for business relationships or other operational and political factors that are hard to consider in the classic shortest path routing. Unfortunately, there are many routing policies to be considered and hard to be defined in addition to the complexity of the policies implementation which is well known as an error prone process [16, 17].

In addition, MPLS networks are associated with other mechanisms such as Differentiated Services (DiffServ) or Traffic Engineering (TE) in order to deliver QoS [18] which would result in more complicated policies other than those found in IP based routing networks. However, there is a certain number of policies in the pure implementation of MPLS and included in the MPLS architecture design [19] as well as in LDP specification [20].

Mainly, MPLS networks treat packets based on common classes which are known as Flow Equivalent Classes (FECs) where each FEC presents a group of packets to be treated in the same manner. Furthermore, each FEC is bound to a unique label before each MPLS enabled router or what is known as Label Switch Routers (LSR) could treat them differently as configured. For that reason, there are certain policies used to govern the way of binding labels to FECs and exchanging of the bindings among LSRs as well as the way of treating packets differently.

Policies in MPLS could be divided into two main classes. The first class is *traffic policy* class which governs the operation carried by LSRs on traffic as per packet by packet. Generally, once each LSR receives a packet, it would carry one of the label operations (push, swap, pop or drop) on it based on the configured policies. It should be noted that the only label operations could be done on unlabeled packets, usually by the MPLS ingress LSRs, are push and drop operation. Then, each packet is scheduled and buffered according to the

experiment field (EXP) of MPLS label which has 3 bits (8 values) that could be sorted as different classes of services to be delivered in each LSR separately which is defined by Per-Hop-Behaviour (PHB). Moreover, each LSR could readjust that field depending on the configured policies. The other type is the label policies that are related to the management of labels inside the MPLS domain.

The other class of policies is the *label management* class. The label bindings could be distributed to other LSRs that have not explicitly requested them when *Unsolicited Downstream* (UD) label distribution policy is configured. Alternatively, the upstream LSR has to explicitly request the label bindings from the next LSR when *Downstream on Demand* (DD) label distribution policy is used. In addition, there are two policies govern label allocation in each LSR. The first policy is called *Independent Label Allocation* (ILA) where each LSR assigns a label to the recognised FEC whether or not it received the label assignment from the next hop. However, LSRs need to receive a label assignment for specific FEC in order to create and propagate their own label bindings in the *Ordered Control* (OC) label allocation policy. Also, there are two policies control labels retention strategy as LSRs may receive multiple labels but only use one of them. The *Liberal Retention* (LR) policy keeps the received labels even if they are unused. Alternatively, the *Conservative Retention* (CR) policy leads the LSR to only keep the labels those are used previously and discard the unused ones.

State Space Reduction:

As the two MPLS policy classes mentioned above have essential differences in functionality, setting a restriction on the MPLS policy engine state space by focusing on one of the policy classes would unify and increase the accuracy of the analysis in later sections. While, traffic policies could be generalised by how the LSPs are managed as the routing in MPLS is based on per-flow basis rather than per-packet basis and influences certain flows rather than the MPLS environment, analysis of such policies is beyond the scope of this work and would be investigated in future work. Instead, we concentrate on the analysis of the label management policies that concern with label distribution, allocation and retention strategies.

In addition, the label management policy state space could be reduced due to the limitation of our simulation tool as well as the dynamical nature of certain policies which leave a unified trace. According to Andersson et al. [20], when implementing DD policy with ILA policy which we refer to as ID policy, LSR would answer the requested label binding immediately without waiting for label binding from next hop. On the other hand, LSR would advertise label bindings to its LSR peers whenever it is prepared to label switch those FECs when it is operating in ILA policy with UD policy which we refer to as IU. However, a LSR that is operating in OC policy must only issue a label mapping after receiving a label mapping from the egress LSR.

The label retention policy is going to be addressed only in section 6 due to the limitation of our simulation tool where only CR policy is applicable.

Knowledge of retention policy is critical for our analysis because it represents one of the three main operation policies in MPLS network. Also, there are some dependency could be drawn among these MPLS operation policies. For example, CR policy is typically implemented with DoD policy unlike the case with UD which may implement one of the retention policies fairly [20].

Therefore, the state space we are interested in is restricted in our simulation to a set of three policy states which we denote by S. The set of policy states are Independent Unsolicited (IU), Independent Downstream on Demand (ID) and Ordered Control (OC). Formally, each policy state s is an element of the policy state set S as $s \in S : IU, ID, OC$. All of the three policy states mentioned above are mutually exclusive. Moreover, two of the policy states which are (IU & ID) represent four policies combined together as the IU policy represents ILA policy and UD policy, also the ID policy represents ILA policy and DD policy. However, the third policy state (OC) represents only one policy for two reasons. The first reason is due to the limitation of our simulation tool which only implements OC policy with UD policy. The second reason that the allocation policy OC was taken as an independent state is because the implementation of OC policy dominates other policies, particularly the label distribution policies, i.e., UD & DD. In other words, if any label request message was sent to the egress LSR, each LSR in MPLS domain receive that message would forward it towards the egress LSR as well as forwarding the response, i.e., mapping message from the egress towards the ingress LSR.

4 Policy Engine State Analysis Design

In this part of the paper, we would like to introduce the analysis framework which includes the assumptions and facts that our analysis of MPLS policy engine states is based on. We used NS-2 [21] network simulator in our analysis study. NS-2 is a discrete events simulator that has an extension model for MPLS. Our analysis design consists of the network model, adversary model, probing elements and simulation scenario as follows:

4.1 Network Model

Our network is based on pure MPLS implementation for the sake of simplicity and generality. Network topology is assumed to be stable and unchanged throughout the analysis process, e.g., no new addition or removal of nodes). Each LSR is trusted to process and response accurately to the received LDP signals, also the possibility of signals loss is excluded as well as all cases of channel errors, e.g., channel loose). Even though, the instability, connectivity or changing of nodes states could benefit our adversary to observe most of the needed information passively, the same assumption could affect the accuracy of our probing process.

There are two sources of traffic represented by node-0 and node-1 to two destinations presented by node-14 and node-15 respectively. Also there are twelve

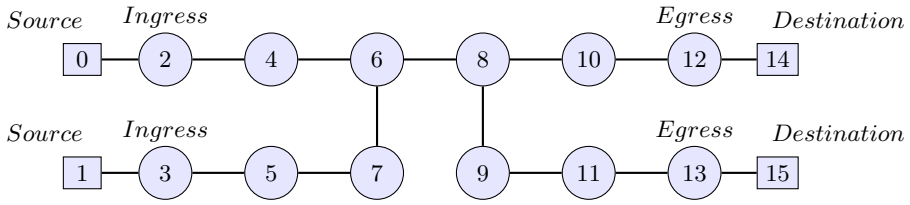

Fig. 1. MPLS Network Topology

LSRs represented by LSR-2,...,LSR-13 where the network ingress and egress edges are LSR-2&3 and LSR-12&13 respectively as shown in figure 1. Each two adjacent LSRs are connected by at most one link. There are two flows which are assigned to FEC-14&15 and pass through the path on node-0→LSR-2→LSR-4→LSR-6→LSR-8→LSR-10→LSR-12→node-14 and the path on node-1→LSR-3→LSR-5→LSR-7→LSR-6→LSR-8→LSR-9→LSR-11→LSR-13→node-15 respectively. It should be noted that this simple network has been chosen for illustration purposes, also to distribute flows throughout the MPLS domain without using traffic engineering which would add a complex routing decision possibility according to the configured policies.

For example, in case of a resource release scenario, different back-up LSPs could be used for forwarding the affected flows, the ingress LSR could communicate with the LSRs alongside the torn-down LSP immediately or different actions could be taken by each LSR that receives the affected flows according to the configured policies on it. Also, the added restrictions on our adversary, as we will see later, limit the ability of probing non-directly connected node. However, there is a possibility that our adversary could discover and manipulate the non-directly connected LSRs by using different exist signalling mechanisms which is beyond the scope of this work and have been avoided by our network model. For example, the adversary could use the LDP extended discovery mechanism which is a mechanism that allows LSRs to establish sessions with potential LDP peers [20], otherwise the adversary could just trick the non-directly connected LSR to exchange fake labels for malicious intents.

4.2 Adversary Model

Most of the service providers and network operators make sure that their network edges and core nodes are well configured and physically secured which reduces the chances of the compromised node scenario [22], hence the compromised node scenario is excluded from our adversary model. We are going to extract a restricted adversary model which we refer to as a probing adversary following the same method that was introduced by Al-Mutairi and Wolthusen [23] to extract MPLS adversary models. Basically, the method was to extract a specific adversary model for a specific analysis purposes from an abstracted framework for the adversarial properties of any adversary that could emerge in MPLS networks.

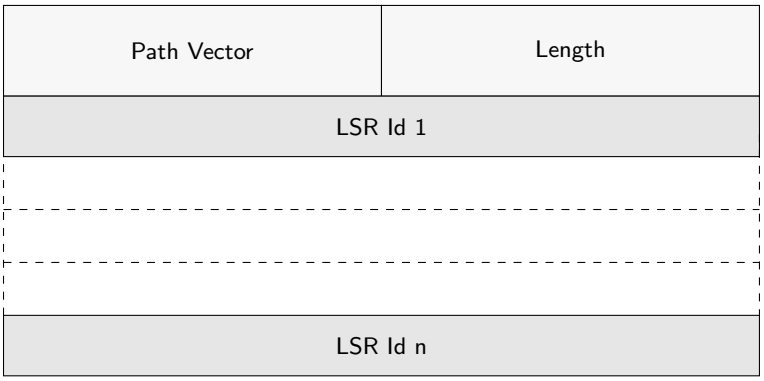

Fig. 2. Path Vector TLV

Therefore, we assume the adversary to have knowledge about the physical information of the network topology, e.g., topological address information. Also, the adversary has access to control information regarding the labels of related flows and can identify them. Moreover, we assume the probing adversary to have access to at most one arbitrary chosen core link which is the link between LSR-6 and LSR-8 with a write/read operation. Also, the probing adversary is capable of fabricating and sending LDP signalling messages to the LSRs that are attached to the compromised link.

4.3 Probe Elements:

The main task of LDP is the establishment of adjacency relationships among peer LSRs and mapping the FECs into the established LSPs [20]. Therefore, we are going to use LDP messages, particularly, label withdraw and release messages in our probing processes to stimulate LSRs to communicate among each other for adversarial analysis. It should be noted that the label withdraw message is sent towards the ingress (imposing) LSR of the withdrawn label and the label release message is sent towards the egress (deposing) LSR of the released label. Also, LDP signalling messages have a common structure that uses type length value (TLV) encoding scheme which would typically include path vector TLV. Path vector TLV records the path of LSRs that label request and mapping messages have traversed [20]. Basically, the path that the message has traversed is presented as a list of router-Ids as shown in figure **??**. Each LSR Id is the first four octets of the LDP Identifier for the corresponding LSR for the sake of uniqueness within the MPLS network.

4.4 Simulation Scenarios:

We configured all of the LSRs with the policy engine states (IU, ID, OC) one by one in order to analyse the traces left by each state and the ability of our

adversary to reveal the LSRs policy engine state. In each one of the above scenarios, the adversary sent release messages for the label related to FEC-14&15 towards LSR-6 as well as withdraw messages for the label related to the same FECs towards LSR-8 and waits for replies from the affected LSRs for analysis purposes in order to reveal the LSRs policy engine states as every policy engine state has a different allocation process.

5 Analysis Results

In this part of the paper we are going to introduce a description of the validation of the probing process and the affect that was noticed on LSRs. Then, we are going to show the ability of our adversary to reveal LSRs policy engine states.

5.1 Probing Process Validation

The probing messages that were sent by our adversary propagated differently through LSRs according to the method that was used to allocate the related labels, i.e., upstream or downstream allocation. While, label withdraw messages were successfully propagated to the ingress LSRs in all cases and the label entries were removed from the upstream nodes (LSR-2,3,4,5,6,7), released messages only propagated in case the released label was upstream allocated and the label entries were removed from downstream nodes (LSR-8,9,10,11,12,13). However, label release messages failed to propagate in case the label was downstream allocated. This problem could be mitigated by our adversary by sending a downstream label mapping or request message depending on the configured policy for the downstream node which is LSR-8. Consequently, after the label entries were removed, the affected LSRs responded differently according to the configured policy as following:

- **Independent Unsolicited (IU):** Label mappings for the withdrawn and released labels were sent independently from LSR-2,3,4,5,6,7,8,9,10,11,12,13.
- **Independent Downstream on Demand (ID):** Label requests for the withdrawn and released label were sent from LSR-2,3,4,5,6,7,8,9,10,11,12,13 and independent label mappings were sent by the peer LSRs in response to the request messages.
- **Ordered Control (OC):** Label requests for the withdrawn labels were sent from the ingress LSR-2&3 to LSR-8. It should be noted that LSR-8 did not forward the request messages for the withdrawn labels because it already has received the label binding from the egress LSRs, hence LSR-8 answered the request messages immediately and sent the label bindings for FEC-14&15. However, a label request for the released label were sent only from LSR-10&11 for the penultimate hop popping mechanism [19]. Clearly, sending a request message to LSR-8 would mitigate this problem by stimulating the downstream LSRs to intervene in the label allocation processes.

5.2 Policy Reveal:

Our simulation has showed different responses to the used probes which we used to reveal the policy states for directly connected LSRs. For the non-directly connected peers we analysed the TLV path vector that is included in the mapping or request messages to discover the LSRs that the messages propagated through. The following policy reveal algorithm 1 was used to analyse the response by the direct LSRs and try to reveal the policy states of other LSRs in the MPLS domain.

Given the LDP signals, the algorithm outputs the policy states for specific LSRs. The algorithm takes the LDP message LDP_m related to the withdrawn/released label l as an input and checks if it is a request for the label REQ_l where the request message is processed to check if the TLV entry includes the ingress LSR to assign all of the LSRs found in TLV entry to the OC state otherwise the LSRs in the TLV entry are set to ID state. However, if it was a mapping message MAP_l, all of LSRs found in TLV entry are set to IU state.

Algorithm 1 Policy Reveal Algorithm

Require: LDP messages LDP_m on the compromised link
Ensure: The policy states S of LSRs
$S[n]$ where n is the number of LSRs
if $LDP_m = REQ_l$ **then**
  if $TLV[1] = 1$ **then**
    for all $i \in TLV$ **do**
      $x = TLV[i]$;
      $S[x] = OC$
    end for
  else
    for all $i \in TLV$ **do**
      $x = TLV[i]$;
      $S[x] = ID$;
    end for
  end if
else if $LDP_m = MAP_l$ **then**
  for all $i \in TLV$ **do**
    $x = TLV[i]$;
    $S[x] = IU$;
  end for
end if
return S

The results that our adversary gained from the reconnaissance probing using the policy reveal algorithm 1 to reveal the policy state of LSRs in MPLS domain are listed below for each one of the configured policy states with a brief description of the results:

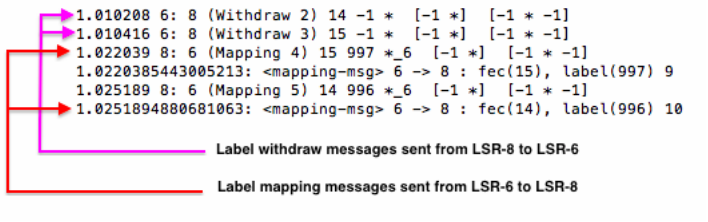

Fig. 3. Independent Unsolicited state response to the probing withdraw message

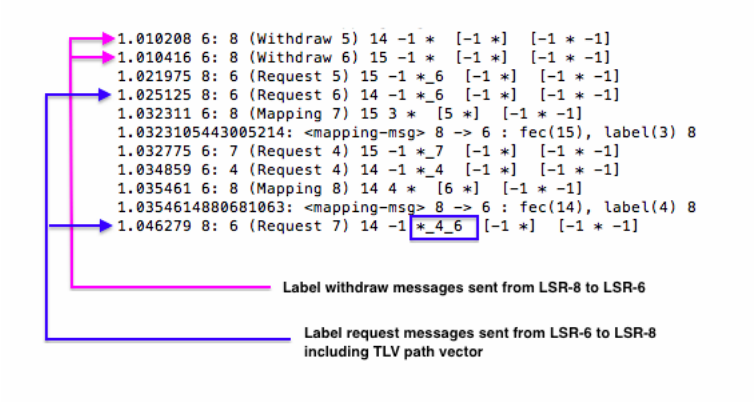

Fig. 4. Independent Downstream on Demand state response to the probing withdraw message

- **Independent Unsolicited (IU):** Only LSR-6 state has been confirmed as shown in figure 3. Theoretically, at least LSR-8 state should be confirmed too in case it sends a mapping message to LSR-6 [1].
- **Independent Downstream on Demand (ID):** The upstream LSRs (LSR-6&4) states have been confirmed as shown in figure 4. Theoretically, even the downstream LSR (LSR-8) state should be revealed by sending a request message to LSR-6.
- **Ordered Control (OC):** The upstream LSRs (LSR-2,3,4,5,6,7) states have been confirmed as shown in figure 5.

Obviously, the reported results have been captured by a restricted adversary with a limited ability and a very simple and stable environment, i.e., network model where some relaxation of restriction on both models (adversary or network model) would reveal more information about other LSRs in MPLS domain. For example a slight change on the network model such as assuming there is another

[1] All or some of the upstream LSRs states could be revealed depending on the time that LSR-6 takes to send the mapping messages for the withdrawn labels

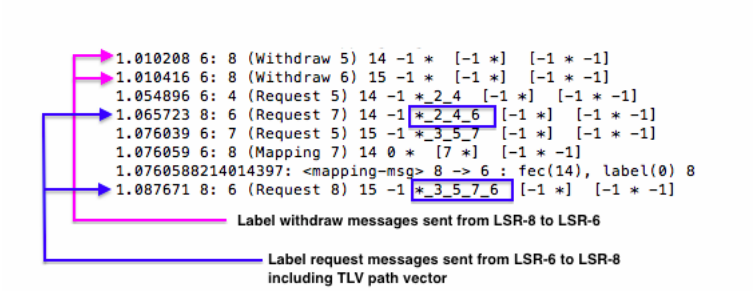

Fig. 5. Order Control state response to the probing withdraw message

source of flow that is routed in the opposite direction would reveal at least the policy state of LSR-8 in case it was running on IU policy state, the policy states of LSR-8,9,10 in case they were running on ID policy state and LSR-8,9,10,11,12,13 in case they were running on OC policy state. Alternatively, making a relaxation of restrictions on adversary model by giving the adversary a read access on more links (in the worst case n/2 links where n denotes the number of LSRs) would reveal the policy state of all LSRs in all cases with no need to analyse the TLV entry that is included in each LDP messages.

6 MPLS Policy States Probability:

The results we gained from the simulation in addition to the knowledge we have about different policy states in MPLS network could be represented in Bayesian Network (BN). Our main aim here is to be able to give approximate estimation about how much to reveal about the policy state by getting some information related to them and to what extent in order to demonstrate the probability of revealing MPLS policies with zero or less prior information.

The BN could answer some useful questions about the probability of the policy states, for example, if a label allocation for the origin LSR's FEC was observed what is the probability that the origin LSR is on independent unsolicited mode. Therefore, we need to define the random variables those playing the roles in MPLS policies after describing the scenario we are interested to model.

Problem Domain:

There are three mutually exclusive states that we suppose each LSR in MPLS domain to have, which are: Independent Unsolicited (IU), Independent Downstream on Demand (ID) or Ordered Control (OC). By having the first state implemented on any LSR, the label allocation of a known FEC will highly be sent to the directly connected peer independently, however a request for label mapping of that FEC will never be sent. On the other hand, a label allocation will not be sent from a node with ID or OC states (except as an answer for a

request), however a label request will be sent for the recognised FEC. The other involved concept is whether the node implements the liberal or conservative retention mode because as we mentioned in section 3 that typically ID will include conservative retention mode other than the liberal mode.

Consequently, the LSR policy state could be presented in various methods, but, the simplest method is to use the graph structure of Bayesian Network to represent policy states (IU, ID, OC) as well as the retention policy and traces found on the simulation where the root node is State (S) and the leaf nodes under the root node are Label Allocation (L) and Label Retention as shown in figure 6. The theoritcial foundation of BN is the Bayes rule:

$$p(h|e) = \frac{p(e|h).p(h)}{p(e)} \tag{1}$$

As $p(h)$ is the prior probability of hypothesis h, $p(e)$ is the prior probability of evidence e, $p(h|e)$ is the probability of h given e and $p(e|h)$ is the probability of e given h. Our BN has a root node S that has three values (IU, ID or OC). The probability of a node having an explicit state is represented by $p(S = IU)$, $p(S = ID)$ and $p(S = OC)$ respectively. Unfortunately, the prior probability for the our root nodes is not available. Therefore, we are going to chose an equi-probable condition for each node. It should be noted that when we reduced the state space for MPLS policies, we specified the three policies based on the label allocation policies, i.e., ILA & OC policy. Which means that the prior probability for each one of the label allocation policy is set to 0.5. Hence, the prior probability of ILA policy should be equally divided between the other two policies, i.e, IU & ID and set to 0.25 for each policy. The prior probability of each root node is calculated as per the following equation:

$$p(S) = p(S = IU) + p(S = ID) + p(S = OC) = 1 \tag{2}$$

The leaf nodes under the root node represent the other policy (retention mode) that would be associated with the MPLS state and the traces observed on MPLS simulation (label allocation). Each leaf node is associated with a conditional probability table (CPT). The retention mode node, denoted by R, includes two values as "Conservative" and "Liberal". The label allocation node, denoted by L, includes two values as "Label Assignment" and "Request". The CPTs correspond to both nodes are shown in Table: 1 and Table: 2 respectively. Each column follows one constraint, which corresponds to one value of the root node. The sum of values of each column is equal to 1. $p(R = \text{"}Conservative\text{"}|S = IU)$ is the conditional probability with the condition that the state is independent

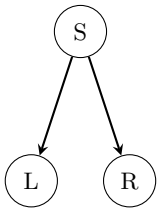

Fig. 6. Bayesian network model

Table 1. The CPT for node R

State	S=IU	S=ID	S=OC
Conservative	0.5	0.9	0.5
Liberal	0.5	0.1	0.5

unsolicited which is 0.5 in the first entry of Table: 1. It measures the probability that the MPLS node is implementing conservative retention mode, given the state as independent unsolicited and so on with the other entries in both CPTs.

By filling the entries in CPTs of MPLS node states BN, the probability of the MPLS node's state could be computed in different aspects by using the Bayes rules. For example, $p(S = IU|R =$ "$Conservative$"$)$ gives the probability that the MPLS node's state is IU by knowing that it is in conservative mode, $p(S = IU|R =$ "$Liberal$"$)$ gives the probability that the MPLS node's state is IU by knowing that it is in liberal mode, while, $p(S = IU|R =$ "$Conservative$", $L =$ "$LabelAssignment$"$)$ gives the probability that the MPLS node's state is independent unsolicited by knowing that it is in conservative mode and a label assignment has been observed.

Therefore, we could now fully specify the joint distribution for MPLS policy states using the following general equation:

$$p(S, R, L) = p(S)p(R|S)p(L|S) \tag{3}$$

Using equation 6 we could calculate the possible twelve entries for the joint distribution over the three relevant variables S, R and L as shown in Table 3.

Table 2. The CPT for node L

State	S=IU	S=ID	S=OC
Label Assignment	1	0	0
Request	0	1	1

Table 3. The probability table for MPLS policy states

State	Retention Mode	Label Assignment	Probability
IU	Conservative	Label Allocation	$0.25 \times 0.5 \times 1 = 0.125$
IU	Conservative	Request	$0.25 \times 0.5 \times 0 = 0$
IU	Liberal	Label Allocation	$0.25 \times 0.5 \times 1 = 0.125$
IU	Liberal	Request	$0.25 \times 0.5 \times 0 = 0$
ID	Conservative	Label Allocation	$0.25 \times 0.9 \times 0 = 0$
ID	Conservative	Request	$0.25 \times 0.9 \times 1 = 0.225$
ID	Liberal	Label Allocation	$0.25 \times 0.1 \times 0 = 0$
ID	Liberal	Request	$0.25 \times 0.1 \times 1 = 0.025$
OC	Conservative	Label Allocation	$0.5 \times 0.5 \times 0 = 0$
OC	Conservative	Request	$0.5 \times 0.5 \times 1 = 0.25$
OC	Liberal	Label Allocation	$0.5 \times 0.5 \times 0 = 0$
OC	Liberal	Request	$0.5 \times 0.5 \times 1 = 0.25$

7 Conclusions

In this paper we analysed the problem of revealing the internal MPLS policy engine state. We have, particularly, paid attention to policy parameters that are based on a pure MPLS implementation. We analysed the ability of an adversary with a limited capability to reveal MPLS policy engine states with simulation. Also, based on our simulation findings as well as knowledge of MPLS specification, we modelled a Bayesian network to illustrate to what extent we could gain information about some policies by getting information about other policies or about the traces founds on MPLS networks.

Future work will seek to extend the policy model and states which could be captured on one hand, but also will investigate different adversary models and capabilities to understand how policy state information can best be kept private. Building on this we are also developing novel attacks aiming to degrade and disrupt MPLS flows both overtly and in deniable form, also focusing on performance parameters relevant for quality of service.

References

1. McClure, Stuart and Scambray, Joel and Kurtz, George and Kurtz: Hacking exposed: network security secrets and solutions. McGraw-Hill (2009)
2. Machiraju, S., Katz, R.H.: Leveraging BGP Dynamics to Reverse-Engineer Routing Policies. Technical report (May 2006)
3. Siganos, G., Faloutsos, M.: Analyzing BGP Policies: Methodology and Tool. In: Proceedings of the Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004). (March 2004) 1640–1651
4. Battista, G.D., Erlebach, T., Hall, A., Patrignani, M.: Computing the Types of the Relationships Between Autonomous Systems. IEEE/ACM Transactions on Networking **15**(2) (April 2007) 267–280

5. Cittadini, L., Battista, G.D., Rimondini, M., Vissicchio, S.: Wheel + Ring = Reel: The Impact of Route Filtering on the Stability of Policy Routing . IEEE/ACM Transactions on Networking **19**(8) (August 2011) 1085–1096
6. Guernsey, D., Engel, A., Butts, J., Shenoi, S.: Security Analysis of the MPLS Label Distribution Protocol, Washington D.C., USA, Springer-Verlag (2010)
7. Grayson, D., Guernsey, D., Butts, J., Spainhower, M., Shenoi, S.: Analysis of Security Threats to MPLS Virtual Private Networks. International Journal of Critical Infrastructure Protection **2**(4) (December 2009) 146–153
8. Bilski, T.: Fluctuations and Lasting Trends of QoS on Intercontinental Links. In: Quality of Service in Heterogeneous Networks. (November 2009) 251–264
9. Alkahtani, A.M., Woodward, M.E., Al-Begain, E.: An Overview of Quality of Service (QoS) and QoS Routing in Communication Networks. In: 4th PGNET2003 Symposium. (2003) 236–242
10. Braden, B., Clark, D., Shenker, S.: Integrated service in the internet architecture: an overview. Program on Internet and Telecoms Convergence (1994)
11. Spainhower, M., Butts, J., Guernsey, D., Shenoi, S.: Security Analysis of RSVP-TE Signaling in MPLS Networks. International Journal of Critical Infrastructure Protection **1**(1) (December 2008) 68–74
12. Llorens, C., Serhrouchni, A.: Security Verification of a Virtual Private Network over MPLS. In: Network Control and Engineering for QoS, Security, and Mobility IV. (November 2007) 339–353
13. Wang, F., Gao, L.: On inferring and characterizing internet routing policies. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ACM (October 2003) 15–26
14. Liang, W., Bi, J., Xia, Y., Hu, C.: RPIM: Inferring BGP Routing Policies in ISP Networks. In: Global Telecommunications Conference (GLOBECOM 2011). (December 2011) 1–6
15. Ming, S., Wu, S., Zhao, X., Zhang, K.: On reverse engineering the management actions from observed BGP data. In: INFOCOM Workshops 2008. (April 2008) 1–6
16. Caesar, M., Rexford, J.: BGP routing policies in ISP networks. Network, IEEE **19**(6) (November 2005) 5–11
17. Mahajan, R., , Anderson, T.: Understanding BGP misconfiguration. ACM SIGCOMM Computer Communication Review **32**(4) (October 2002) 3–16
18. Awduchea, D.O., Jabbarib, B.: Internet traffic engineering using multi-protocol label switching (MPLS). Computer Networks **40**(1) (September 2002) 111–129
19. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol label switching architecture. IETF, RFC 3031 (January 2001)
20. Andersson, L., Doolan, P., Feldman, N., Fredette, A., Thomas, B.: LDP specification (October 2007)
21. Isi.edu: The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/ Accessed 17 Jun. 2014.
22. Palmieri, F., Fiore, U.: Securing the MPLS control plane. High Performance Computing and Communications (2005) 511–523
23. Al-Mutairi, A.A., Wolthusen, S.D.: A Security Analysis of MPLS Service Degradation Attacks Based on Restricted Adversary Models. In: Information Security in Diverse Computing Environments, IGI Global (2014) Print