

# Optimizing the MPLS Support for Real Time IPv6-Flows using MPLS-PHS Approach

Imad J. Mohamad  
National Advanced IPv6 (NAV6)  
Universiti Sains Malaysia (USM)  
Penang, Malaysia  
[imad@nav6.org](mailto:imad@nav6.org)

Tat-Chee Wan, Faisal Y. ALZyouid, Putra Sumari  
School of Computer Sciences  
Universiti Sains Malaysia (USM)  
Penang, Malaysia  
[tcwan@cs.usm.my](mailto:tcwan@cs.usm.my) , [faisal@nav6.org](mailto:faisal@nav6.org) , [putras@cs.usm.my](mailto:putras@cs.usm.my)

**Abstract**—The huge coverage space of IPv6 addresses and providing guaranteed support for the ever increasing customer demand, results in the dealing with bigger packet header-size compared to the payload-size especially in some real time video and audio applications, consequently more bandwidth is wasting. Using the advantages of MPLS with IPv6 is still one of the migration challenges at the backbone. Payload Header Suppression (PHS) approach is currently being used in Mobile WiMAX networks for point to point (SS to BS) communications. The proposed approach for this paper (named) MPLS-PHS adapt the concept of the PHS technique mentioned above, modified to be applicable for multihop of Label Switching Path (LSP) of MPLS domain at which multipoint-to-multipoint connection found (Ingresses to Egresses).The results clearly show a dramatic increase in the data throughput for real time IPv6 flows.

**Keywords**-component; MPLS, QoS, IPv6, Real Time Applications and Traffic Engineering

## I. INTRODUCTION

Suppression and Compression are two closely related techniques, which deal with packet header reduction. PHS and Robust Header Compression (RoHC) are two well known approaches used for header reduction in wireless and satellite environment, targeting utilization of their limited and costly bandwidth. This utilization is not free; most available approaches require extra processing time and extra memory space (software and/or hardware solutions) to accomplish this task.

Typically VoIP (as an example) uses the encapsulation *voice/RTP/UDP/IP*. When MPLS labels are added, this becomes *voice/RTP/UDP/IP/MPLS-label*. Also MPLS VPNs use label stacking, and in the simplest case of IPv4 the total packet header at least 48 bytes, while the voice payload is often no more than 30 bytes. When IPv6 used, the relative size of the header in comparison to the payload is even greater [1].

This paper focuses on UDP flows of real time applications. It is organized as follows: Section II covers the essential backgrounds. Related work is discussed in section III. The developed methodology is discussed in section IV. Scenarios and Simulation results explained in section V. Conclusion outlines stated in section VI.

## II. BACKGROUNDS

### A. MPLS fundamental

Ingress and Egress are the input and output doors of the MPLS cloud (Fig. 1), where labels are pushed at the former and popped at the later. The LSR (Label Switching Router) located one hop before the Egress is called Penultimate. The Penultimate pops the label instead of Egress, when this facility is activated. Core LSRs forwards the labeled packets without considering on their layer 3 IP headers, behaving as Transit routers.

The 32 bits MPLS header is located between MAC header and IP header as shown in Fig. 2. At the core of MPLS cloud, routing is done by 20 bits MPLS-label instead of 128 bits (IPv6) of layer 3. Label Distribution Protocol (LDP) was designed for distribution of labels inside MPLS domain. One of the most important services that may be offered using MPLS in general and LDP in particular is the support of constraint-based routing of traffic across the routed network.

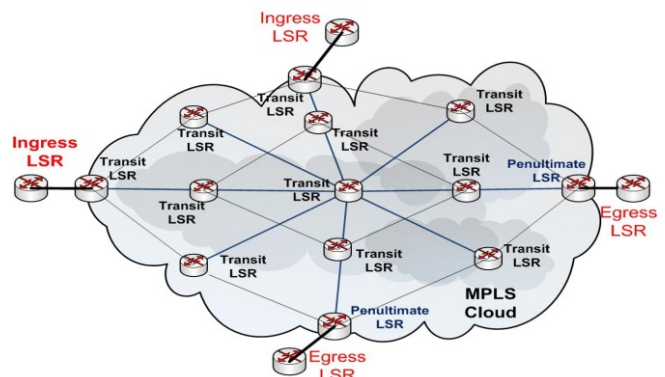


Figure 1. MPLS domain.

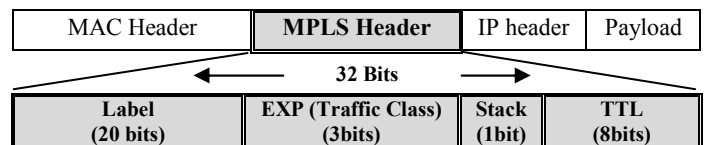


Figure 2. MPLS shim header structure.

Constraint-based routing offers the opportunity to extend the information used to setup paths beyond what is available for the routing protocol. For instance, an LSP (Label Switching Path) can be setup based on explicit route constraints, QoS constraints, and other constraints. Constraint-based routing (CR) is a mechanism used to meet TE requirements [2].

*B. Traffic engineering at MPLS-based backbone*

Pavel’s work shows the operational environment for a sample MPLS-based backbone and the service class assignment at Ingress to support the TE [3]. It explores the classification criteria at the Ingress based on DSCP (Differentiated Service Code Point) as shown in TABLE I.

*C. PHS (Payload Header Suppression)*

The packet header redundancy is considered in WiMAX/IEEE standard. It is an approach developed and implemented as an optional service in 802.16 wireless media; it supports header reduction between a Subscriber Station (SS) and Base Station (BS). In PHS, a repetitive portion of the payload headers of the higher layer is suppressed in the MAC Service Data Unit (SDU) by the sending entity and restored by the receiving entity. When PHS is enabled at MAC connection, each MAC SDU is prefixed with a Payload Header Suppression Index (PHSI), which references the Payload Header Suppression Field (PHSF) [4].

*D. RoHC (Robust Header Compression)*

A highly robust and efficient header compression scheme for RTP/UDP/IP, UDP/IP, and ESP/IP (Encapsulated Security Payload) headers is a standard approach suitable for links with significant error rates and long Round Trip Time (RTT). It has three modes of operation, called Unidirectional, Bidirectional Optimistic, and Bidirectional Reliable mode. Header compression with RoHC can be characterized as an interaction between two state machines, one compressor machine and one decompressor machine [5],[1].

III. RELATED WORK

The class type (6 bits DSCP) of IPv6 header can be activated to support differentiated services and accomplish TE at MPLS backbone domain. Class of Service (CoS) is intended for networks supporting time-sensitive video and audio applications, which are used for flexible DiffServ-over-MPLS

TABLE I. DSCP TO SERVICE CLASS MAPPING [3]

<i>Service class</i>	<i>DSCP</i>	<i>Application example</i>
Telephony	EF	IP telephony bearer: VoIP, CEoIP, virtual wire
multimedia conferencing	AF4x	H.323/V2 video conferencing (adaptive): H.323/v2
real-time interactive	CS4	video conferencing and interactive gaming
multimedia streaming	AF3x	streaming video and audio on demand
broadcast video	CS3	broadcast TV & live events

TE with per-flow traffic policing [6]. It explained the necessity of hardware solutions to solve time complexity problem of packet classification at Ingress.

In [7], PHS method was proposed for WiMAX to reduce the RTP/UDP/IPv6 header from 60 bytes to 15 bytes. Two approaches were compared (PHS and RoHC) in a certain case of VoIP transmission. The work concluded that RoHC is more efficient than PHS, and PHS efficiency may be better than ROHC when a greater number of static headers are found. It is a reasoning point for our proposed work.

Performance analysis of RoHC over Mobile WiMAX for VoIP is introduced in [8]. The OPNET simulator experiments showed that RoHC is able to provide more efficient use of radio resource compared to PHS. Considering the compression efficiency and robustness, in O-mode, especially, the authors observed a better performance as compared to U/R-mode.

[1] RFC4901 defined how to use MPLS to route header compression packets over an MPLS-LSP. It defines some schemes like RoHC, CRTP and Enhanced Compressed RTP (ECRTP). Justification of MPLS-PHS approach and comparison to others is stated in section IV. Our paper produces packet header suppression approach (MPLS-PHS) as an additional reduction choice for backbone domain.

IV. PROPOSED WORK

The MPLS-PHS approach is preferred for the MPLS-based backbone rather than RoHC for the following reasons:

- 1) The high speed and high throughput transmission media at the backbone requires low and fast complexity approach for Header Compression (HC) and Header Decompression (HD).
- 2) Long RTT (LSP) which needs fast HC and HD approach, to avoid unnecessary delay in feedback and signaling.
- 3) The IPv6 first-order difference (static fields) is bigger in size compared to the second-order difference (compression gains are mainly in first order difference).
- 4) The frequent state transitions for RoHC Unidirectional/mode is not efficient for MPLS since MPLS encounters low Bit Error Rates (BERs).

RFC4247 [9] stated the guidelines as requirements for any future development of compression techniques in MPLS domain. The work of this paper (the MPLS-PHS approach) takes these guidelines into consideration during the design and simulation stages. Also the design of our approach considered the symmetry behavior property of both routers where the former (Ingress) is a cloud node at which initial pushing of labels are done for a flow packet and then popped finally at the Penultimate node.

The Suppression stage is a stripping out process of the two IPv6 header fields: SourceIP and DestIP (32 bytes) of the received packet at the Ingress (after a per flow classification stage mixed with per class classification which is already done as a requirement of differentiated services/TE stage), then restored at the Penultimate LSR.

TABLE II. MPLS-PHS CALCULATIONS

At Ingress LSR (Suppression):  
 $\text{NewPktSize} = \text{CurrentPktSize} - \text{SuppressedSize}$   
 $48 = 80 - 32$  (in bytes)  
 At Penultimate LSR (Restoration):  
 $\text{CurrentPktSize} = \text{NewPktSize} + \text{SuppressedSize}$   
 $80 = 48 + 32$  (in bytes)  
 $\text{Reduction \%} = 1 - (\text{NewPktSize}) / (\text{OldPktSize})$   
 $= 1 - (48/80) = 40\%$  (for VoIPv6)

**Example** (see TABLE II):

*voice/RTP/UDP/IPv6 using G729 standard:*

80 bytes Pkt= [40 bytes (IPv6Header)]+[40 bytes (Payload)]

To accomplish our MPLS-PHS approach, the following algorithms are extensions for current MPLS module and are implemented in ns2.33:

*B. Local flow label generator*

Our proposed approach (MPLS-PHS) considered the flow label specification and requirements of RFC3697 [10]. It stated that Flow Label values previously used with a specific pair of source and destination addresses must not be assigned to new flows with the same address pair within the flow state lifetime of 120 seconds.

To keep the uniqueness property of the flow signatures, the one tuple local FlowID is used as a flow signature instead of the original (three tuples) flow signature at the Ingress, then sending the local FlowID instead of the original one to the Penultimate during the suppression stage, finally restoring the original FlowID value at the Penultimate for each served flow packet.

The local flow generator (Algorithm1) is used to distribute the (20 bits) scope value of IPv6 flow label among the no of ingresses in MPLS cloud (in one Autonomous System).

*C. Suppression Process (Ingress)*

The requirements for MPLS-PHS at the Ingress are:

- Activation of (6 bits) ClassType field for the IPv6 packet at the source in addition to the activation of the uniqueness property of Flow Label field at the source.
- Activation to the one of the unused 2-bits of ClassType field of IPv6 packet at the flow source to identify the flow termination to the Ingress (and Penultimate), and to clear the saved (no longer used) flow signatures in PFT (Partial Forwarding Table).

ALGORITHM1. LOCALFLOWID\_GENERATOR

```
NoOfIngresses = k // in cloud
Scope = IPv6_FlowLabelScope = 220-1
IngrLocalFlowIDScope = Scope div NoOfIngresses
//setup LocalFlowIDs at each Ingress
for IngrID = 1 to NoOfIngresses do
    Ingr[IngrID].Min = (IngrID - 1) * IngrLocalFlowIDScope
    Ingr[IngrID].Max = Ingr[IngrID].Min +
        IngrLocalFlowIDScope - 1
end for
end of LocalFlowID_Generator
```

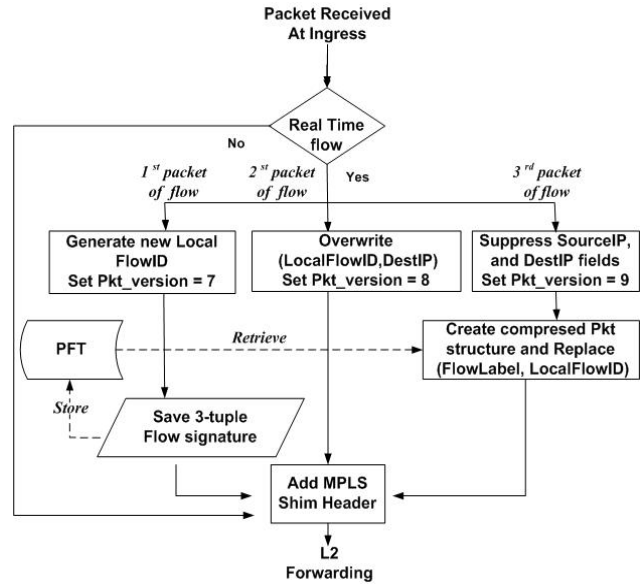


Figure 3. Suppression process at Ingress

- Activation of local FlowID Generator.

The Suppression process (Fig. 3) implemented (added) before the pushing of MPLS label at the Ingress.

The sub tasks of this process are:

- Per flow classification and saving the 3-tuples flow signature of real time flow.
- Management of local FlowID usage.
- Changing the version field value of the IPv6 packet (under MPLS-PHS service) to :
  - Version 7: acts as an identifier to the Penultimate to save original flow signature (FlowID, SourceID and DestID) of the current packet and No change was made in the packet size. This is applied for only the first packet of every flow.
  - Version 8: acts as an identifier to the Penultimate to save local FlowID (which overwrites the DestIP field) of the current packet and No change was made in the packet size. This is applied for only the second packet of every flow.
  - Version 9: from the third packet onwards the version field would change to 9 to identify to the Penultimate that the suppression process of the packets has started.
- Reaction to the Drop Notification message sent by the Penultimate, and to the termination of flow.

*D. Restoration process (Penultimate LSR)*

This process (Fig. 4) activated after the popping of MPLS label at Penultimate LSR.

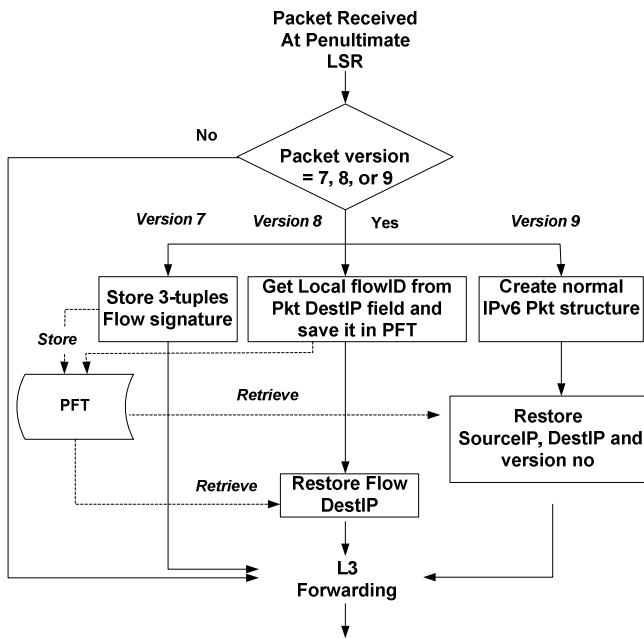


Figure 4. Restoration process at Penultimate LSR

The Sub tasks of restoration process:

- Filter packets of MPLS-PHS approach (Versions 7, 8 and 9) to take suitable action.
- Discovering which Ingress is the source of current received packet (known from the scope of local FlowID).
- Restoration of original packet header information namely, FlowID, SourceID and DestID.
- Sending of Drop Notification to ingress as necessary.

#### E. Extensions to LDP

The following are extensions to the existing MPLS-LDP to support the MPLS-PHS Approach:

- Creation of additional LDP session-peer(s) Ingress-Penultimate PWs (Pseudo Wires), and keeping the compatibility with the RFC4447 [11] specifications of PW signaling between the compressor and decompressor points. These peers are necessary for drop signaling feedback (see Algorithm2).
- The Classifier behavior of MPLS module was extended to take an action when drop of MPLS-PHS packets started. LDP was extended to cover the signaling of drop notification messages between the Penultimate and Ingress peers as follows:

ALGORITHM2. PEERSSETUP

```
// In MPLS cloud
I = NoOfIngresses ;
P = NoOfPenultimates;
for IngressID=1 to I
  for PenultimateID=1 to P
    Setup LDP-Peer(IngressID , PenultimateID);
End PeersSetup
```

ALGORITHM3 NOTIFYINGRESSONDROP

```
Create(DropMessageID)
// Search for TargetIngressID according to
// the received local FlowID of the dropped packet
GetTargetIngrIP( LocalFlowID, TargetIngrIP, PFT)
Create(NewLDP_Pkt)
SourceIP(NewLDP_Pkt) = CurrentPenultimateIP
DestIP(NewLDP_Pkt) = TargetIngrIP
AddTo (NewLDP_Pkt , DropMessageID)
Send(NewLDP_Pkt)
End NotifyIngressOnDrop
```

ALGORITHM4 INGRESSACTIONONDROP (INT LOCALFLOWID)

```
If (DropNotificationMessage received)
// retrieve
ThreeTupleFlowSignature= Lookup(PFT, LocalFlowID);
StopSuppressionOf(Three TupleFlowSignature);
// send the signature again with the next
// coming packet for the notified flow
Update (PFT);
End
```

1) *At Penultimate LSR*: The Algorithm3 (procedure) is activated when the Penultimate detect the packets drop in the supported flows.

2) *At Ingress LSR*: Ingress will react (using Algorithm4) to the received drop message from the Penultimate LSR. It will stop the suppression of the identified flow which suffering from packets drop, sending uncompressed packets with the signature again in normal size, then restarting the suppression process.

## V. SENARIOS AND SIMULATION RESULTS

### A. Topology 1:

It's setup shown in Fig 5. The restoration process at Penultimate router (LSR7) requires at least 1.5 Mb output bandwidth equivalents to the total bandwidth of Ingress's sources, otherwise it starts dropping the packets. Bandwidth requirements are shown in TABLE III. Fig. 6 and Fig. 7 show the simulation results.

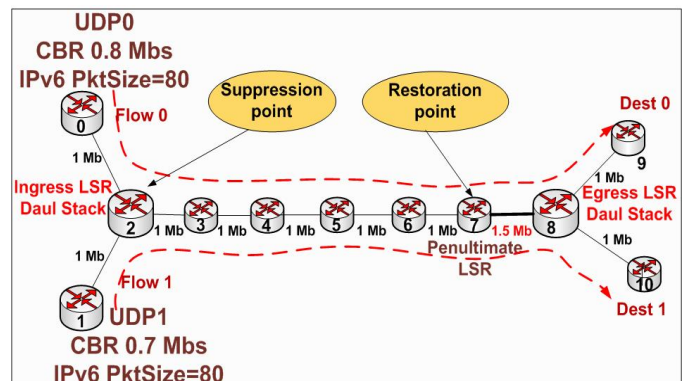


Figure 5. Topology1

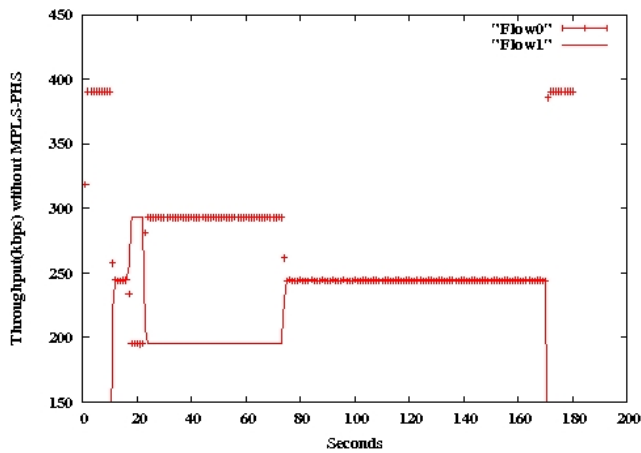


Figure 6. Data throughput (kbps) **without** MPLS-PHS

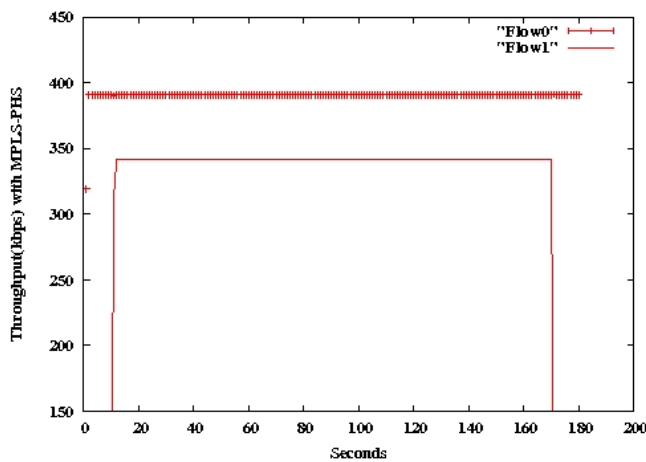


Figure 7. Data throughput (kbps) **with** MPLS-PHS

TABLE III. BANDWIDTH REQUIREMENTS (TOPOLOGY 1)

	Flow0	Flow1	Min. bandwidth required
<i>Without MPLS-PHS</i>	0.8	0.7	1.5 Mb
<i>With MPLS-PHS</i>	0.48	0.42	0.9 Mb
<i>Saved Bandwidth</i>	0.32	0.28	0.6 Mb

### B. Topology 2:

In Fig. 8, eight Voice/UDP/IPv6 flows are used with 32 Kbps-CBR each, using G729 standard (setup parameters shown in TABLE IV).

Fig. 10 shows distinguished improvement in data throughput (over Fig. 9) for all flows with MPLS-PHS. It is expected that the adaptation of this concept for a bigger cloud would save even more bandwidth.

TABLE IV. SETUP PARAMETERS (TOPOLOGY2)

Flows	Ingress(in)	Egress(Out)	Rate (for each)
0,1,2,3,4, and 5	LSR6	LSR12	32Kbps
6 and 7	LSR13	LSR14	32Kbps

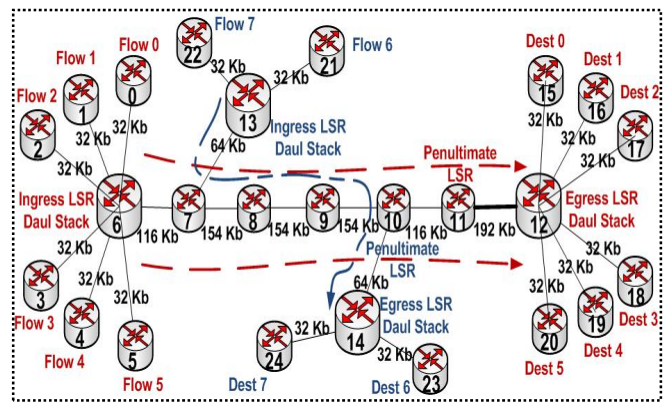


Figure 8. Topology2

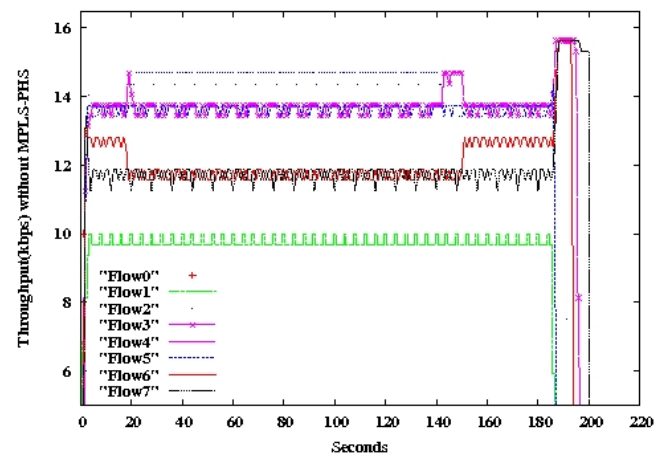


Figure 9. Data Throughput (kbps) **without** MPLS-PHS

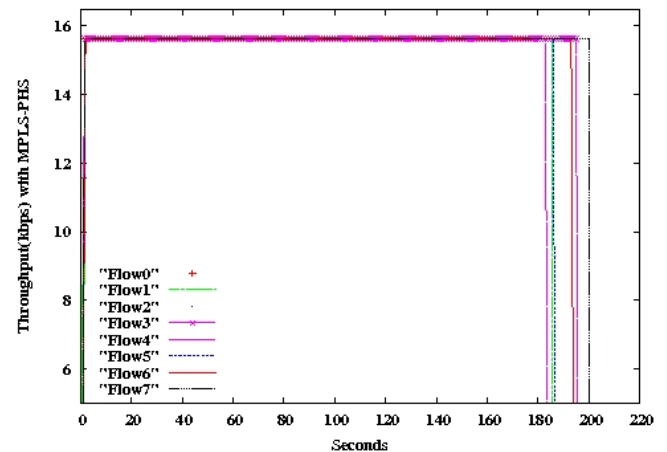


Figure 10. Data Throughput (kbps) **with** MPLS-PHS

The drop statistics and bandwidth requirements (with and without MPLS-PHS) for topology 2 are shown in the following two tables:

TABLE V. DROP STATISTICS (TOPOLOGY 2)

	Sent Packets	Dropped packets
Without MPLS-PHS	75420	28988
With MPLS-PHS	75420	0

TABLE VI. BANDWIDTH REQUIREMENTS (TOPOLOGY 2)

	Min. bandwidth for Flows (0,1,2,3,4,5,6,7)
Without MPLS-PHS	8 x 32 = 256 Kbps
With MPLS-PHS	8 x 19.2 = 154 Kbps
Saved Bandwidth	102 Kbps

VI. CONCLUSION

As a result, 40% reduction of IPv6 packet (assumed 80 bytes packet-size using G729 with one-frame/packet) reduced the congestion in the core of MPLS domain and increased the domain throughput, thus we recommend the adaptation of MPLS-PHS service for the bandwidth utilization of the MPLS-based backbone (Fig. 11).

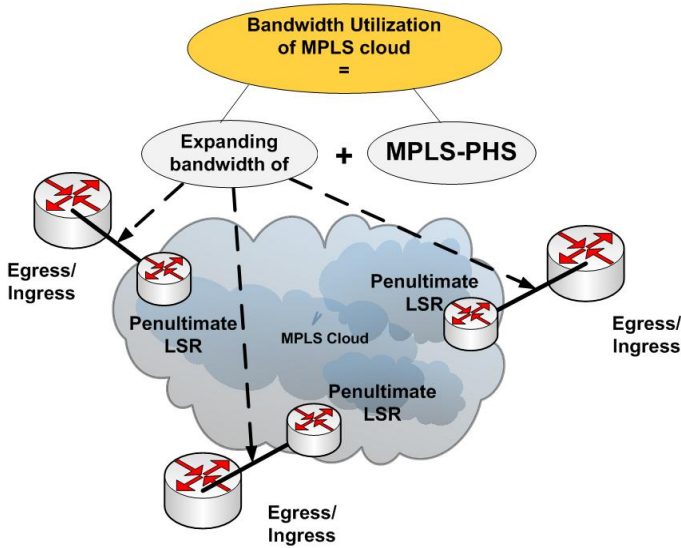


Figure 11. MPLS-PHS Solution for the MPLS-based backbone

REFERENCES

- [1] J. Ash, et al., "Protocol Extensions for Header Compression over MPLS", RFC4901, June 2007.
- [2] B. Jamoussi, et al., "Constraint-Based LSP Setup using LDP", RFC3212, January 2002.
- [3] Pavel Samrah and Josef Verich, "QoS Design and Implementation in the CESNET2 MPLS-based Backbone", CESNET Technical Report 17/2008, CESNET, 2009.
- [4] IEEE Std 802.16™-2004.
- [5] Bormann, et al., "Robust Header Compression (ROHC): Framework and four Profiles RTP, UDP, ESP and uncompressed", RFC3095, July 2001.
- [6] Djahongir Siradjev et al., "Scalable DiffServ-over-MPLS Traffic Engineering with Per-flow Traffic Policing", APNOMS 2006, LNCS 4238, pp. 509 – 512, 2006.
- [7] Loutfi Nuaymi et al., "Headers Overhead Estimation, Header Suppression and Header Compression in WiMAX", WiMob 2007.
- [8] Hyunj Woo, et al., "Performance analysis of Robust Header Compression over mobile WiMAX", ICACT 2008.
- [9] J. Ash, et al., "Requirements for Header Compression over MPLS", RFC4247, November 2005.
- [10] J. Rajahalme, et. Al., "IPv6 Flow Label Specification", RFC3697, March 2004.
- [11] Martini, et al., "Pseudowire Setup and Maintenance using the Label Distribution Protocol (LDP)", RFC4447, April 2006.