Provider-Side VoD Content Delivery Using OpenFlow Multicast

- Master Thesis Defense -

Bernard Niyonteze
Email: bernard@postech.ac.kr

Supervisor: Prof. James Won-Ki Hong
Co-Supervisor: Prof. Jae-Hyoung Yoo

Department of Computer Science and Engineering
POSTECH, Korea

June 30, 2014

Outline

- Introduction
- Problem Statement
- Related Work
- Proposed Method
- Validation
- Conclusion & Future Work

INTRODUCTION

Introduction

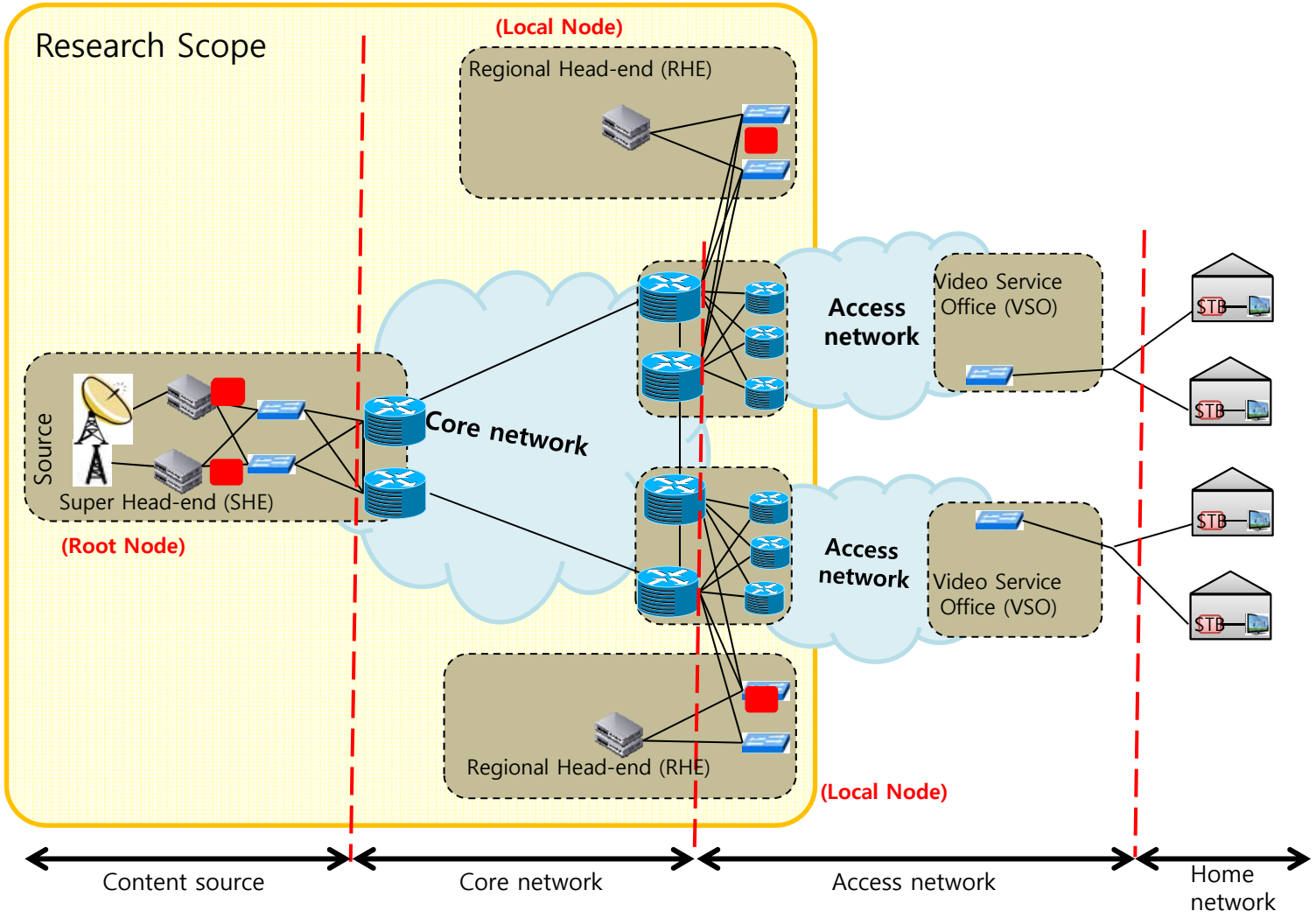
❖ What is IPTV

- Internet Protocol Television (IPTV), technology that delivers video or TV broadcasts over the IP network

❖ IPTV Services

- Live Broadcast
  - Stream content to multiple clients
  - High priority (not delay-tolerable)
- Video on Demand (VoD)
  - Send content upon customer request
  - Lower priority (relatively delay-tolerable)

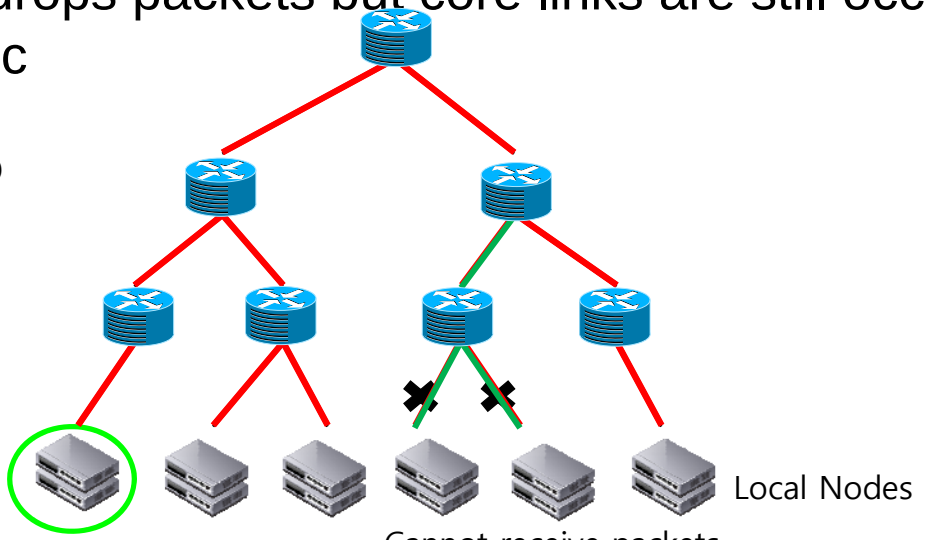
IPTV Network Architecture



Problem Statement

- **IP multicast presents a waste of network resources in VoD content delivery**
  - Inefficient to manage multicast tree
    - Source transmits packets continuously even if edge links are congested
    - The receiver drops packets but core links are still occupied by multicast traffic

Multicast-based VoD content delivery



Research Goals

- **Propose multicast-based VoD delivery service using Software-Defined Networking (SDN)**
  - Provider-side efficient content delivery
  - Dynamically adjust multicast tree while monitoring link utilizations
    - Implement control loop at SDN controller
    - Resolve wasted network resources
- **Validate proposed method**
  - Emulating proposed method on Mininet with 20 nodes and 80 switches
    - Approximately ½ scale of a Korean IPTV service provider with 5 million subscribers
  - Comparing number of active links with IP-based multicast
  - Detecting link status and pruning unnecessary links

Related Work (1/2)

❖ Content Delivery Networks

- Content Anycast & P2P (APAN, 2010)
  - Content can be stored in multiple nodes (server as well as client)
  - Servers redirect requests to clients
- C-flow (ICOIN, 2014)
  - Use OpenFlow to enhance content delivery using dynamic re-routing, parallel transmission and cache management

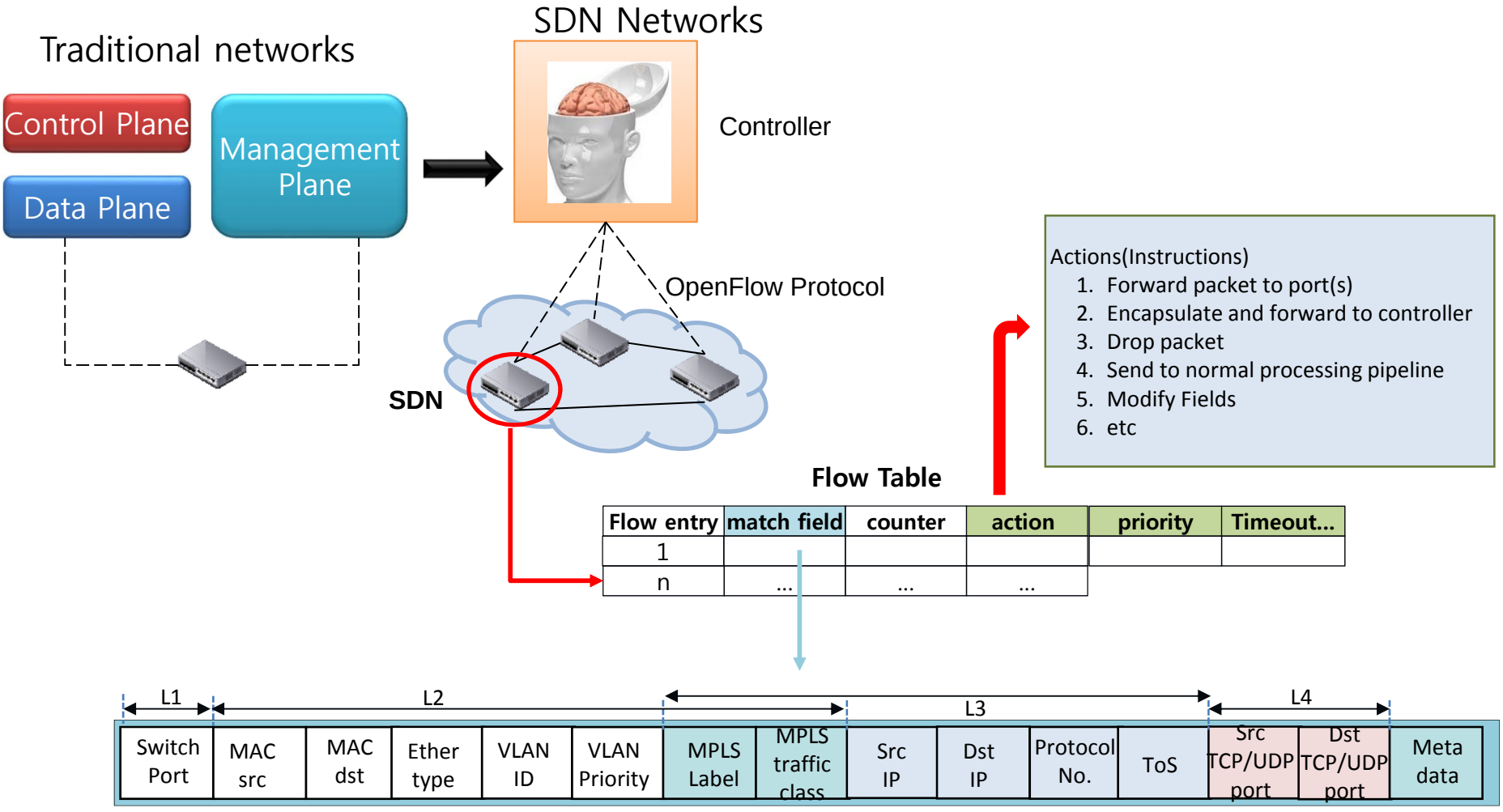
❖ IP Multicast

- IP Multicast Content Delivery System (APSITT, 2005)
  - Extension of IP multicast service model
  - Support multicast addressing and filtering

Related Work (2/2)

❖ Software-Defined Networking (SDN)

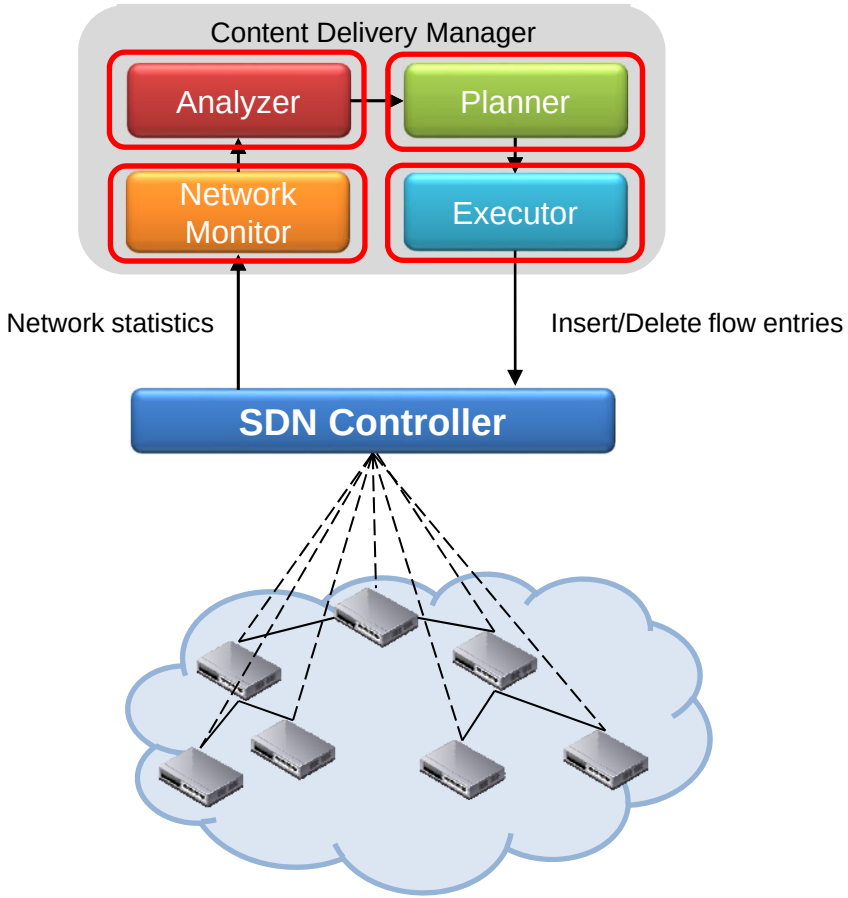
- Separate the control plane from the data plane

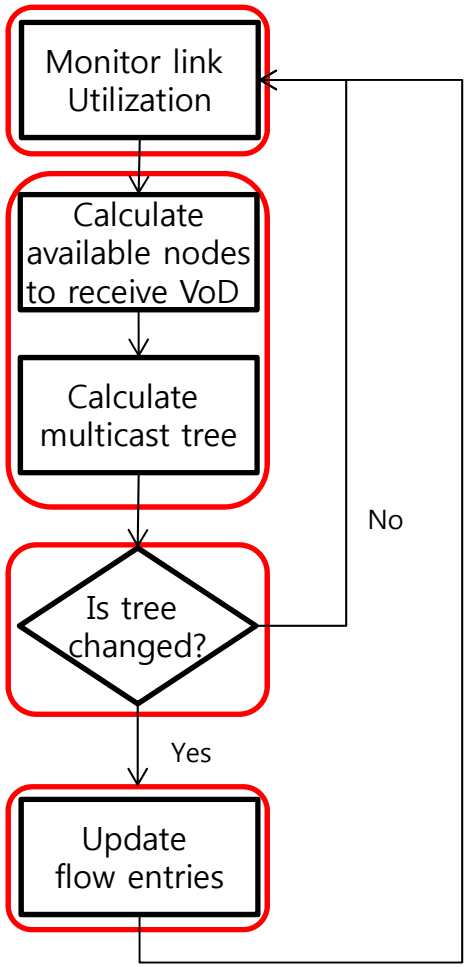


PROPOSED METHOD

Proposed Method

❖ Architecture and control loop





Multicast Tree Calculation

Algorithm 1: OpenFlow based Multicast Tree Building Algorithm

```
input : Current Multicast Group Entries: current_ge
        Available Local Nodes: available_nodes
        Sender Node: sender_node
        Topology: topology
1  Initialize new_ge;
   /* Build a new group table instance                    */
2  for switch ∈ topology do
3      for port ∈ GetAllPorts(switch) do
           /* Insert the uplink flows                     */
4          descendantNodes = GetDescendants(switch, port);
5          if sender_node ∈ descendantNodes then
6              AddEntryToMCTree(new_ge, switch, in_port = port);
           /* Insert the downlink flows                   */
7          else if available_nodes ⋂ descendantNodes ≠ ∅ then
8              AddEntryToMCTree(new_ge, switch, out_port = port);
   /* Different part of current and new group entries     */
9  Δ ← Diff(new_ge, current_ge);
   /* Update group tables of switches                     */
10 UpdateToGroupTable(Δ);
11 current_ge ← new_ge;
```

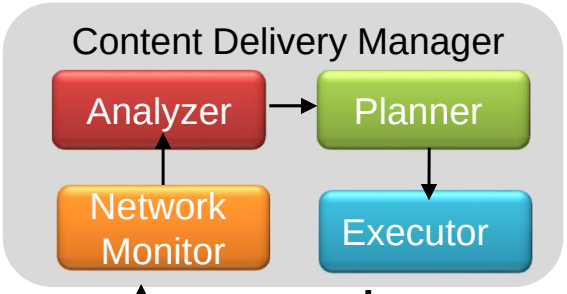
Obtained by link utilization data from network monitor

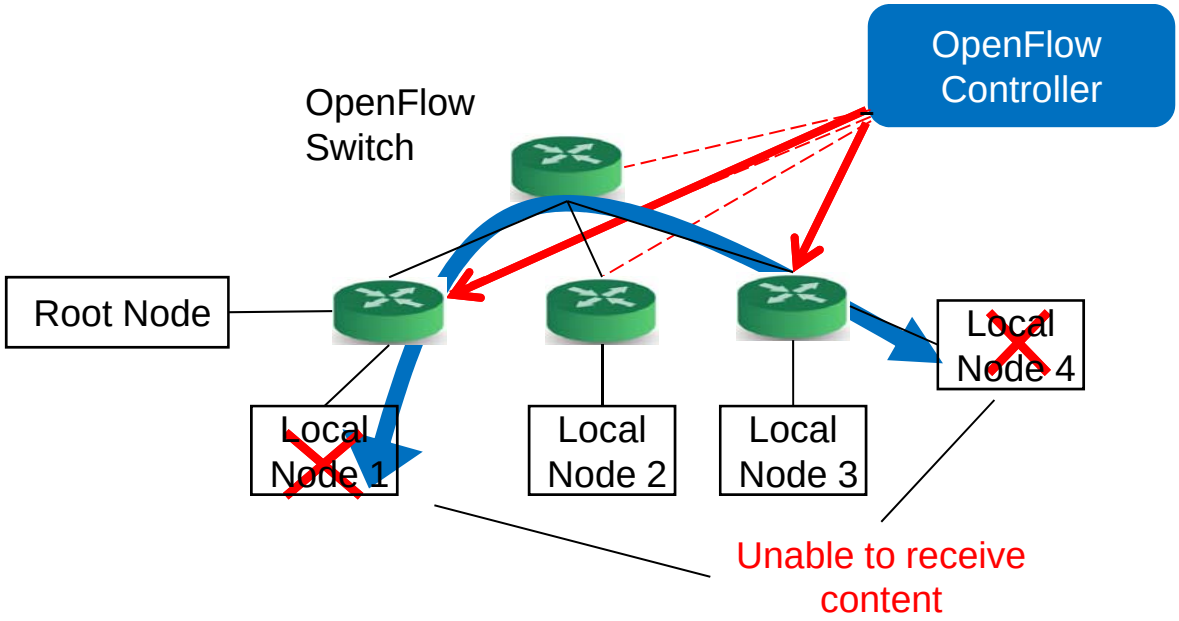
Find nodes filtered by given switch and port

Actually send update messages to switches

Example – Low Throughput Caused by Cross Traffic

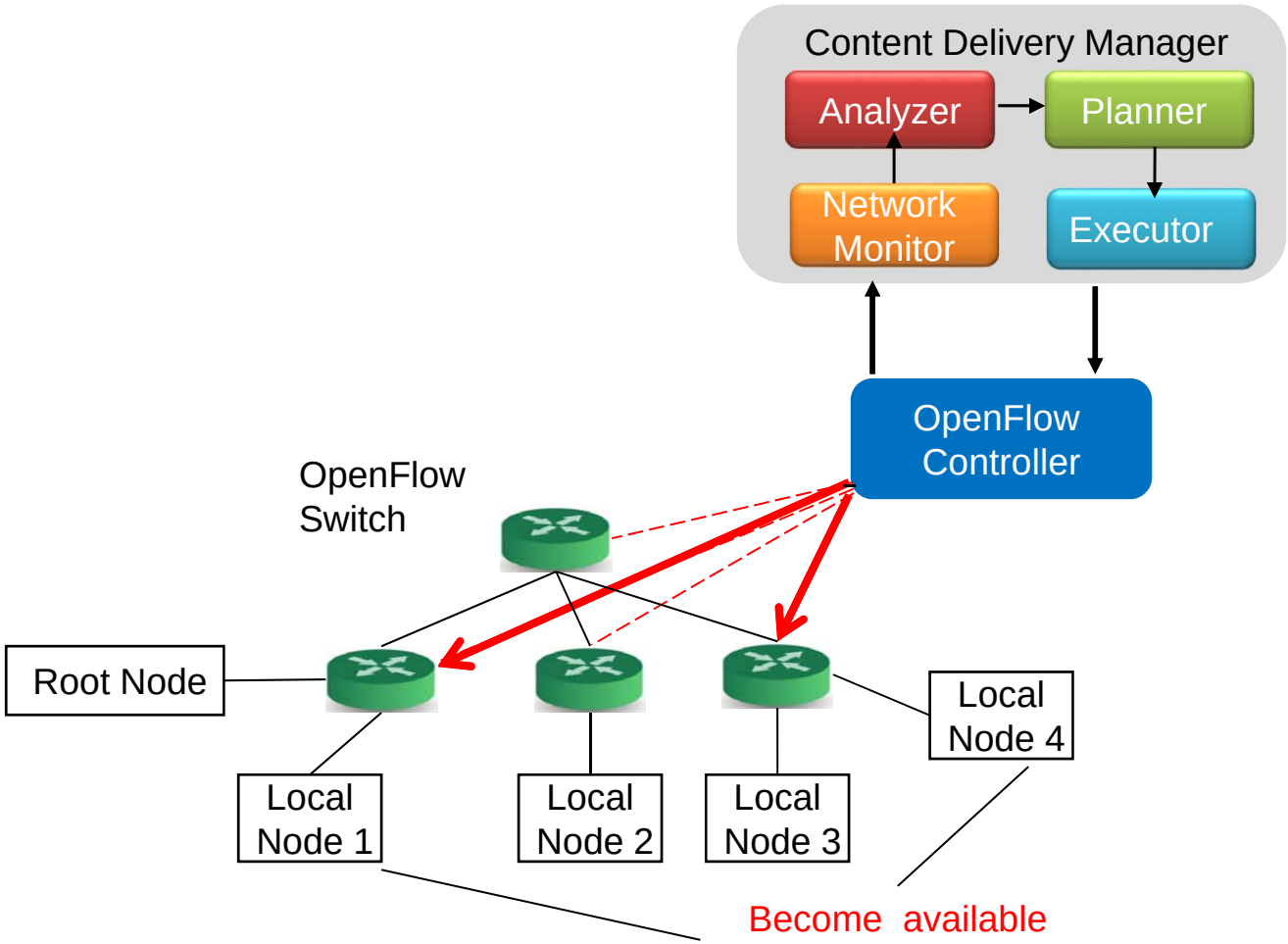
- **Cross traffic**
  - Interfere with VoD traffic
    - Increase packet loss probability
  - To avoid packet loss
    - Serve higher prioritized traffic first
    - VoD distribution is controllable traffic





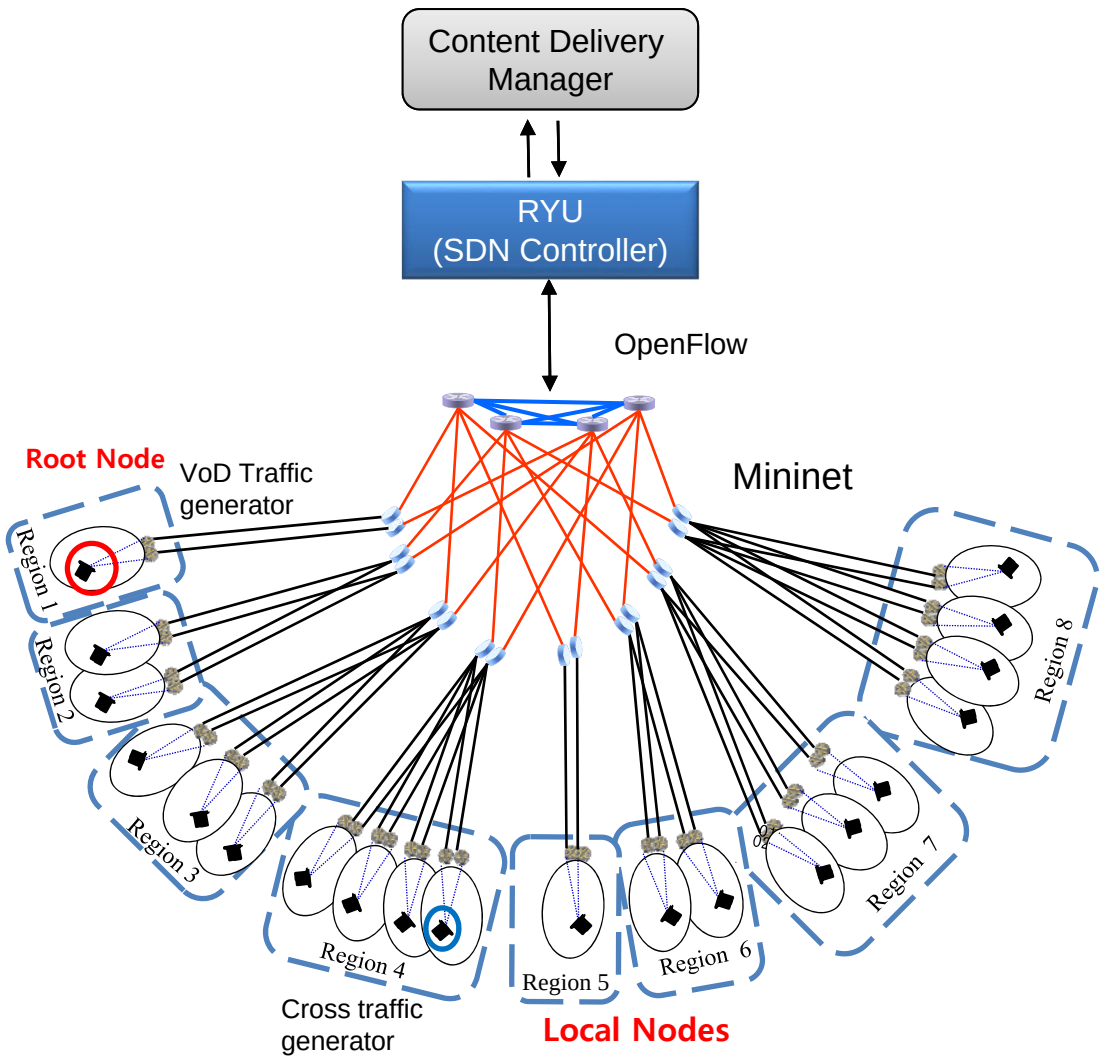
Cross Traffic Occurs

Example – Low Throughput Caused by Cross Traffic



VALIDATION

Experiment Environment



Software

- Controller : Ryu 3.9
- OpenFlow Switch: OVS 2.1.2
- Network Emulator : Mininet 2.1.0
- Traffic generator
  - VoD traffic: implement in Python
  - Cross traffic: Iperf
- OS: Ubuntu 12.10

Topology

- Approx. ½ scale of a Korean IPTV service provider
- 8 regions
- 1 to 4 subnets in each region
- Number of switches: 80
- Root node (sender): 1
- Local nodes (receiver): 19

Simulation Scenario

Simulation parameters:

- Threshold: 35 MB/s
- Generated VoD traffic : 20 MB/s
- Generated cross traffic : 22 MB/s
- Scaling down simulation parameters
  - Difficulties in generating high bitrate VoD and cross traffic

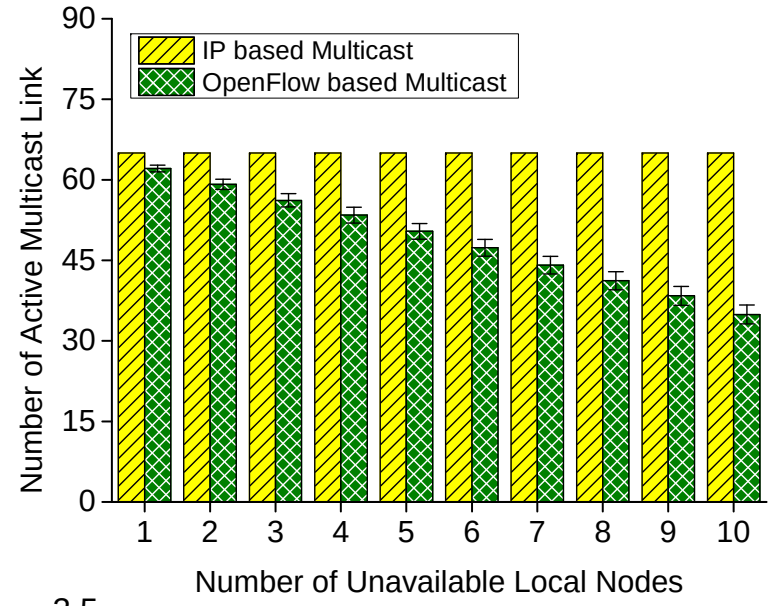
Traffic generation

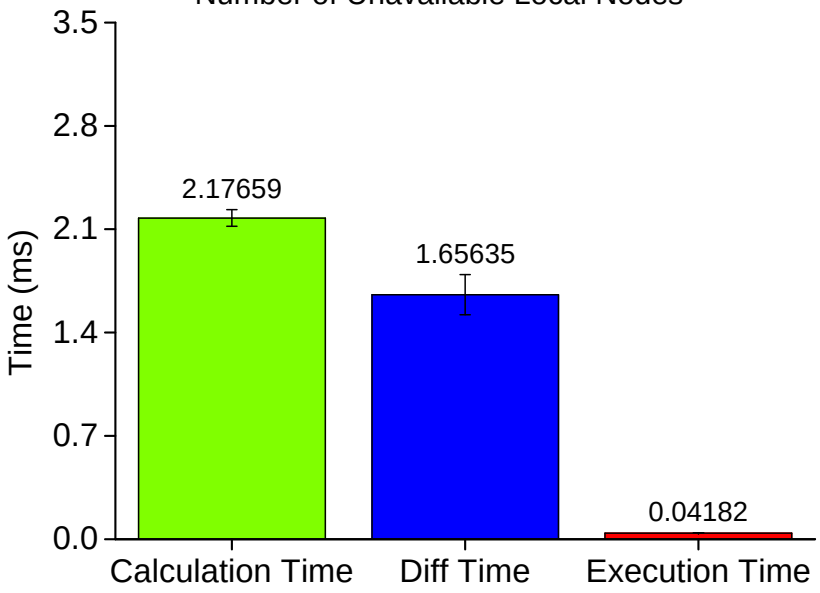
- Sending VoD traffic from root node to all local nodes - multicast
- Sending cross traffic from a local node to another local node - unicast

Monitoring and actions

- Every 1-second, collect byte counts of edge links to find which nodes cannot have enough bandwidth to receive VoD traffic
- Generate VoD traffic and cross traffic
- Calculate multicast tree and compare it with previous tree
- Find added/deleted/modified flow entries (switch id, output ports)
- Send message to update flow/group entries

Simulation Results (1/2)

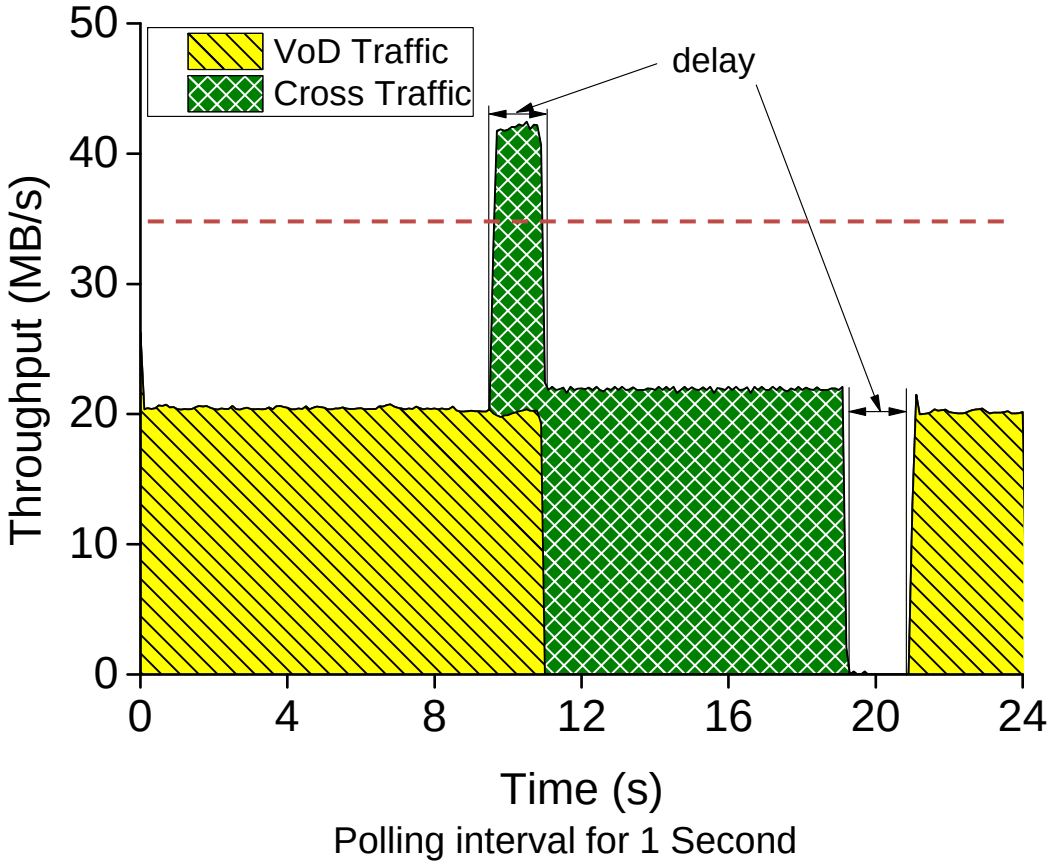




- **Number of active links in multicast**
  - Comparing with IP-based multicast
    - OpenFlow multicast shows less number of active links
    - OpenFlow multicast prunes unnecessary links in multicast tree - benefit
- **Calculation time and execution time**
  - Tree calculation time and comparison time
    - 2.18 ms and 1.66 ms respectively
  - Execution time is 0.0418 ms
    - Controller just sending update messages without ACK

Simulation Results (2/2)

❖ Control loop and detection time



CONCLUSION

Conclusion and Future Work

❖ Contributions

- Design OpenFlow-based content delivery system for service providers
- Dynamically adjust multicast tree
  - Design and implement simple control loop at controller
  - Save wasted link bandwidth
- Validate proposed method using Mininet with real-world-like topology (20 nodes; 80 switches)

❖ Future Work

- Explore other features of OpenFlow to guarantee QoS and QoE
- P2P-based content delivery system on OpenFlow

THANK YOU

References

[1] A. Azgin, G. AlRegib and Y. Altunbasak , "Cooperative On-Demand Delivery for IPTV Networks", Global Communication Conference (GLOBECOM),2012 IEEE

[2] M. Othman Othman and K. Okamura, "Improvement of content server with Content anycasting Using OpenFlow" In Proceedings of the Asia-Pacific Advanced Network, 2010

[3] D. Agrawal, M.S. Beigi, C. Bisdikian, L. Kang-Won, "Planning and Managing the IPTV Service Deployment. In proceeding of 10 the IFIP/IEEE International Symposium on Integrated Network management IM '2007, Munich, Germany May 2007

[4] D. Eager, M. Ferris and M. Vernon, "Optimized Caching for on –Demand Data Delivery." In Proceedings of Multimedia Computing and Network (MMCN 199), San Jose, California January 1999

[5] D. Chang, M. Kwak, N.Choi, T.Kwon and Y.Choi "C-flow: An efficient content delivery framework with OpenFlow", ICOIN, 2014

[6] N. New Khaing , T. Phyu and T. Thu Naing ,"IP multicast Content Delivery System for Large Scale Applications" Information and Telecommunication technologies, 2005. APSITT 2005 Proceeding 6th Asia Pacific Symposium

[7] Y. Yu, Q. Zhen, L. Xin, C. Shanzhi," OFM: A novel Multicast Mechanism Based on OpenFlow", Advances in information Sciences and Service Sciences, May, 2012

[8] G. Pallis, and A. Vakali, "Insight and Perspectives for Content Delivery Networks," Communications of the ACM, Vol. 49, No. 1, ACM Press, NY, USA, pp. 101-106, January 2006

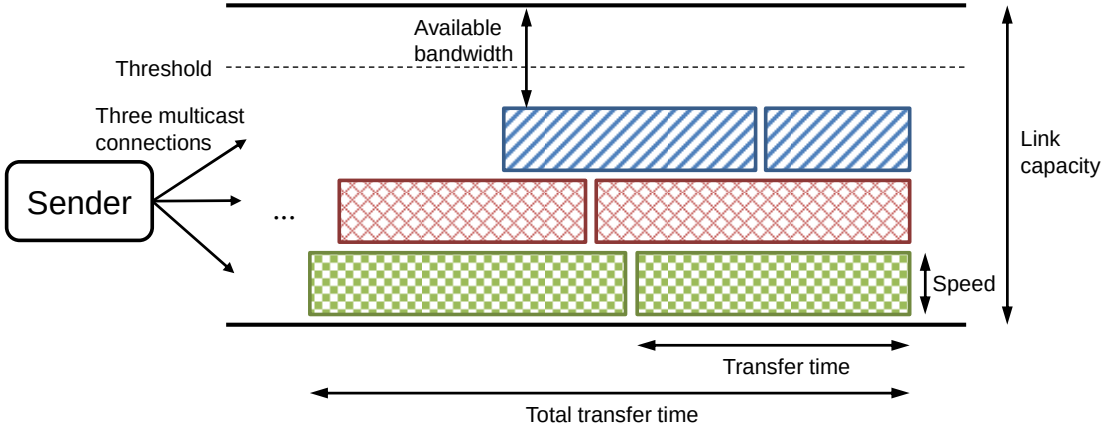
[9] N. McKeown, T. Anderson, H. Balakrishnan, G.Parulkar, L.Petterson, J.Rexford, S.Shenker and J.Turner, "OpenFlow: Enabling Innovation in Campus Networks" ACM SIGCOMM April, 2008

[10] A. Nagata, Y.Tsukiji and M. Tsuru, "Delivering a File by Multipath-Multicast on OpenFlow Networks. INCoS, 2013 5th International Conference.

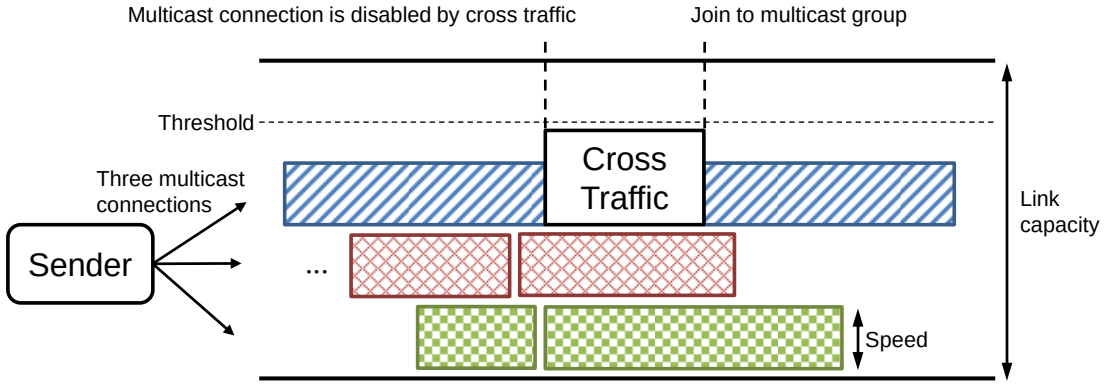
[11] E. Hilmi Egilmez , S. Tahsin Dane, K. Tolga Bagci and A. Murat Tekalp, "OpenQoS: An OpenFlow Controller Design for Multimedia Delivery with End-to-End Quality of Service over Software-Defined Networks"

Traffic Model

VoD Traffic



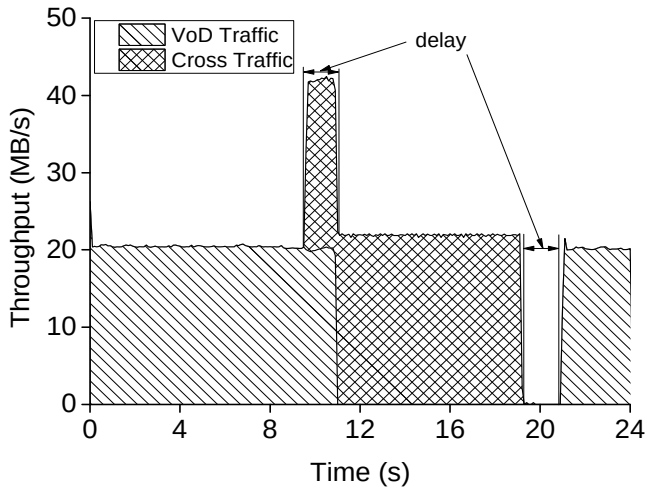
VoD Traffic with Cross Traffic



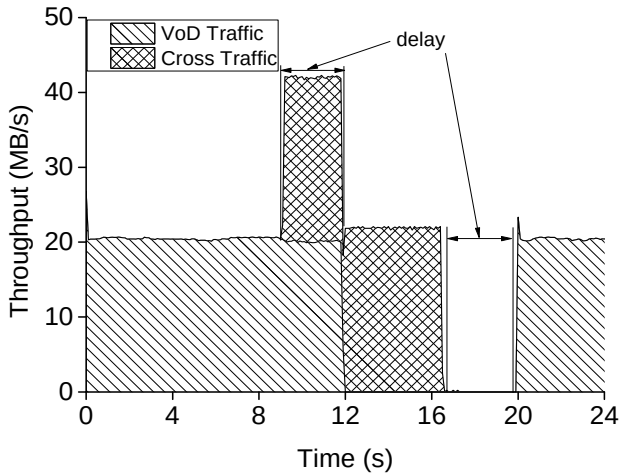
Simulation Results (2/3)

❖ Detection delay

Polling interval for 1 Second



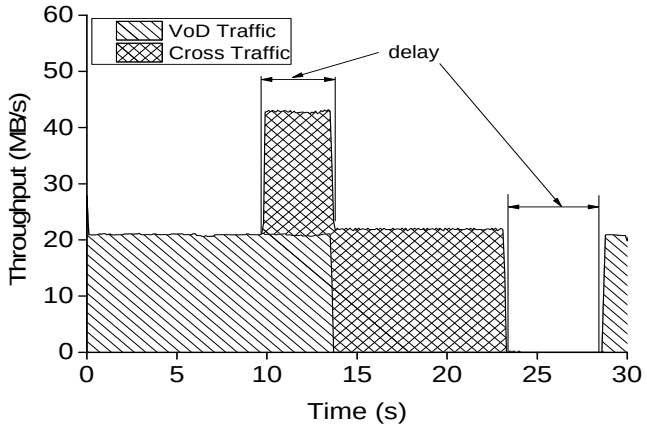
Polling interval for 2 Seconds



Simulation Results (3/3)

❖ Detection delay

Polling interval for 3 Seconds

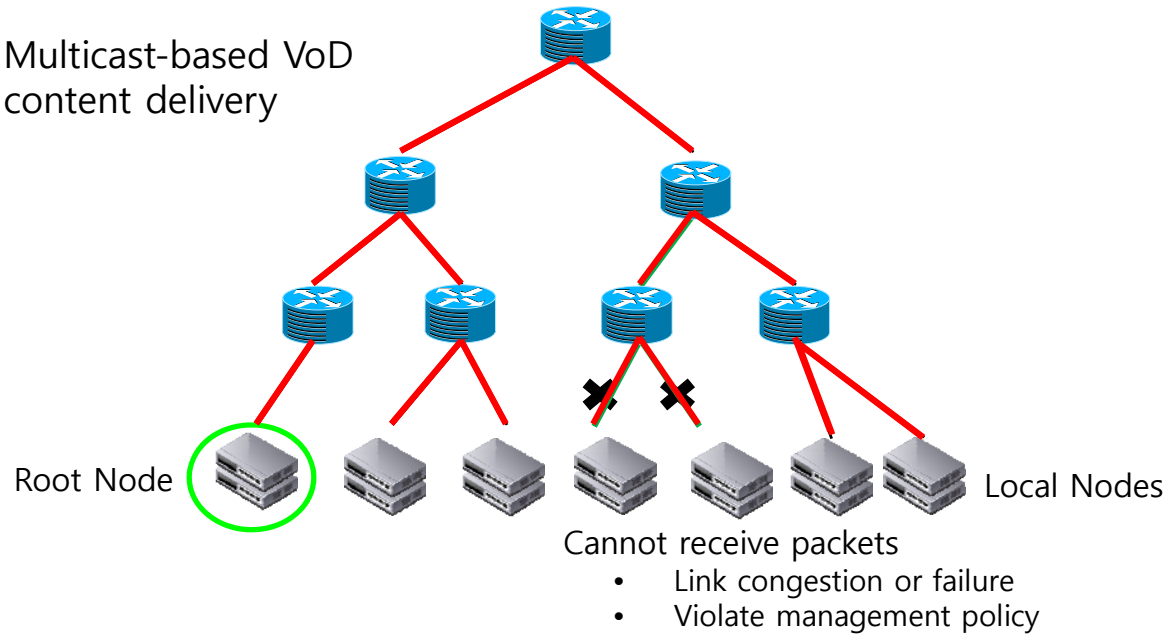


Detection delay

	Polling Interval Time		
	1 second	2 seconds	3 seconds
Lag time during leaving multicast group	1.3	2.8	4.8
Lag time during joining multicast group	1.2	2.7	4.7

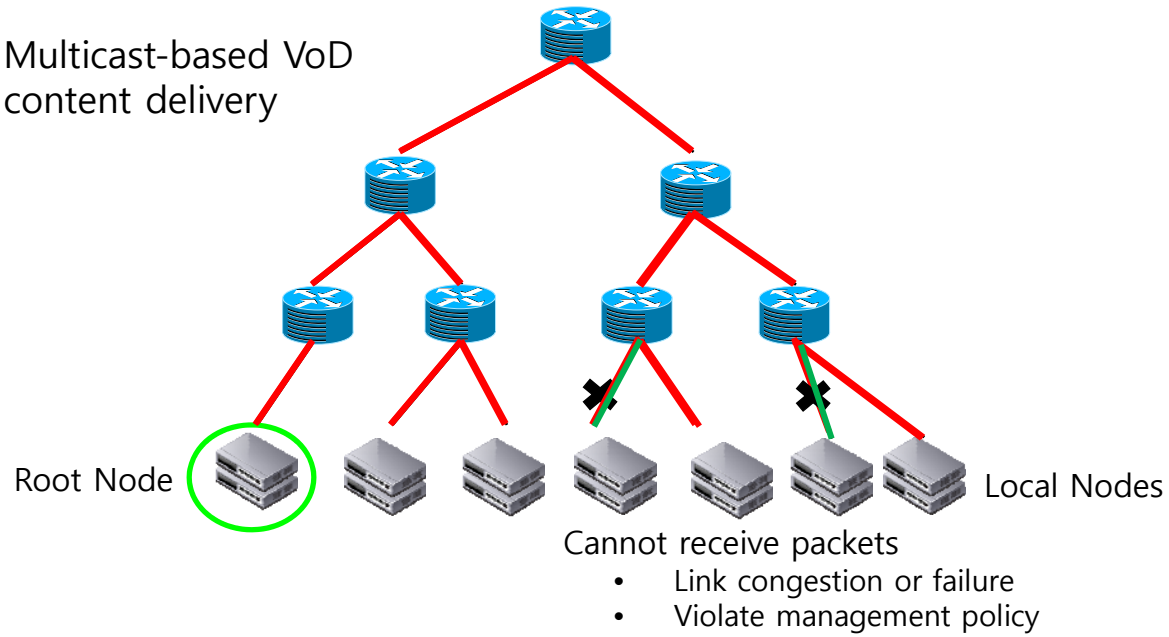
Time to calculate multicast tree was around 6.4 ms

Two Unavailable Nodes (1/2)



Prune three links

Two Unavailable Nodes (2/2)



Prune two links