Adding Multi-class Routing into the Differentiated Services Architecture

Yin Wang

Supervisor: Professor Raimo Kantola
Instructor: Dr Peng Zhang
Contents

- Introduction
- Background
  - DiffServ
  - OSPF
  - Link-state routing algorithms
- Multi-Class Routing (MCR) based on DiffServ
- OSPF with MCR extensions
- Static MCR Implementation based on Zebra software
- Testing
- Conclusions and future work
Introduction

- Best-effort IP networks need Quality of Service (QoS)
- DiffServ model has been a scalable solution to implement QoS in the current IP networks
  - Big problem is an inter-class effect
- Several hop-by-hop QoS based routing algorithms are tried to alleviate the inter-class effect and reasonably distribute network traffic
- Problem: a single routing table
- Our solution: Adding Multi-class routing into DiffServ
  - multiple routing tables
  - multiple hop-by-hop QoS based routing algorithms
**DiffServ**

- **DS domain**
  - contiguous DiffServ nodes
  - controlled and managed by the same administration
- **DS codepoint**
  - a label
  - classify the network traffic
  - Type of Service (TOS) in the IP header
- **Per-Hop Behaviors (PHBs)**
  - forwarding treatments
  - buffer management and queueing
- **Traffic conditioning**
  - classifier, marker, meter, shaper/dropper and remarke
- **A single routing table**

---

**A DiffServ network**

- Core node or Interior node
- Edge node or boundary node
- Class BE traffic
- Class EF traffic
OSPF

- Open shortest path first
- A link-state routing protocol
- A hierarchical topology
  - Area is a group that consists of one or more contiguous networks and hosts
  - ABR creates Summary-LSAs
  - ASBR redistributes external routing information
- OSPF Sub-protocols
  - Hello protocol
  - Database Exchange protocol
  - Flooding protocol
OSPF (cont.)

- Link state advertisements (LSAs)
  - LSAs represent a router’s routing information
  - Except for Network-LSA, each type of LSAs only contains a metric
  - Different types of LSAs have their own functions
  - LSAs are stored into the link state database

- Routing table calculation
  - A whole routing table calculation includes the intra-area route calculation, inter-area route calculation and external route calculation
  - OSPF protocol supports incremental update calculations, which are related to inter-area route calculation or external route calculation
  - A real routing algorithm is applied to calculate the intra-area routes
  - A distance-vector approach is used to calculate inter-area routes and external routes
OSPF (cont.)

- OSPF convergence
  - A fast, loop-less convergence
  - Two stages:
    - Flood the updated LSAs throughout an area or a whole routing domain
    - Recalculate the routing table
  - The number of LSAs has an impact on the OSPF convergence
- OSPF can be extended to support new functions or mechanisms
Link-state Routing Algorithms

- In OSPF, Dijkstra’s algorithm is used as a routing algorithm.
- Several hop-by-hop QoS based routing algorithms:
  - Shortest Path with the hop count metric (SP)
    - SP computes a shortest path tree with the hop count metric.
  - Bandwidth-inversion Shortest Path (BSP)
    - BSP establishes a shortest path tree with the sum of the inversed bandwidth.
  - Widest-Shortest Path (WSP)
    - WSP calculates a shortest path tree that the path to each destination is the minimum hop count among all feasible paths. If several such paths exist, the one with the maximum bandwidth is used for the destination.
  - Enhanced Bandwidth-inversion Shortest Path (EBSP)
    - It is the extension to the BSP algorithm.
    - EBSP introduces a “penalty” factor $\theta$ into the weight function of the BSP algorithm.
MCR Scheme

- **Multi-Class Routing scheme**
  - In a network, each router can provide different routing tables computed by various routing algorithms for delivering different class traffic.

- **Each router has two new functions:**
  - **Multi-routing management**
    - collect routing information
    - calculate multiple routing tables
    - control routing algorithms
    - update routing tables
  - **Class-based route selection**
    - make the route selection based on the DS codepoint in each packet’s IP header

Functions of the MCR scheme
MCR Approaches

- We can use different MCR approaches for the multi-routing management.
- Static MCR
  - A simple approach
  - Link weights, such as hop count and bandwidth, are assigned by the administrator
  - Produces a minimal impact on the convergence
  - Cannot reflect the real network status
- Dynamic MCR
  - Distributed or centralized
  - Link weights are automatically measured based on time T or a significant event
  - Promptly reflect the network status
  - Result in a frequent convergence process and increasing packet loss
  - Take the time T into consider
MCR DiffServ

- We add Multi-class routing scheme into the DiffServ architecture
- called MCR DiffServ
- In a MCR DiffServ network, a node
  - keeps the same functions as a traditional DiffServ node has.
  - selects the corresponding routing table based on DS codepoint
  - looks up the next hop for each packet by using the selected routing table

An example of the MCR DiffServ network
OSPF with MCR Extensions

Goals of design of OSPF with MCR extensions is

- To implement the multi-routing management function
- To provide two routing tables
- To support static MCR approach and dynamic MCR approaches
- To limit the additions to the standard OSPF version 2 protocol
- To minimize the impact on the original OSPF, such as code and convergence, etc.
OSPFWithMCRExtensions2

- We create MCR optional capability for OSPF
  - M-bit is introduced into the Option field as an indicator of the MCR capability

<table>
<thead>
<tr>
<th>M</th>
<th>unused</th>
<th>DC</th>
<th>EA</th>
<th>N/P</th>
<th>MC</th>
<th>E</th>
<th>unused</th>
</tr>
</thead>
</table>

The modified Options format

- The modified Option field will be used by all Hello packets, Database Description packets and all LSAs
OSPF with MCR Extensions 3

- Encoding Extended TOS
  - Goal: to make extended TOSs distinguish from the standard definition in RFC1349
  - 72 is defined for bandwidth

- Encoding Bandwidth Resource
  - Problem: the metric field in Router-LSAs is only 16 bits long.
  - Assumption: the maximum bandwidth is 10 Gbits/s
  - Solution:
    TOS metric = bandwidth / 159.59 (kbits/s)
OSPF with MCR Extensions 4

- **OSPF packets**
  - None of the formats of OSPF packets needs to be changed

- **Modified LSA formats**
  - Each type of LSAs, except for Network-LSAs, contain two metrics: TOS 0 metric and TOS 72 metric

- **New types of Summary-LSA**
  - Each routing table has its own types of Summary-LSAs
  - For example, type 12 and type 13 Summary-LSAs are used for the second routing table
OSPF with MCR extensions

| extended TOS | 72 | link’s bandwidth |

MCR Router-LSA format

<table>
<thead>
<tr>
<th>LS age</th>
<th>Options</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link State ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertising Router</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LS sequence number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LS checksum</td>
<td>length</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>V</td>
<td>E</td>
</tr>
<tr>
<td>Link ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td># TOS</td>
<td>metric</td>
</tr>
<tr>
<td>TOS</td>
<td>0</td>
<td>TOS metric</td>
</tr>
<tr>
<td>Link ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OSPF with MCR Extensions 6

- Software architecture
  - Major original OSPF modules keep unchanged
  - Some new modules are introduced
    - More than one route table computation module
    - Precomputation trigger module
  - Some modules are extended or modified in order to handle new types of LSAs and new routing tables
    - ABR task manager module
    - Local interface status manager module, etc.
OSPF with MCR Extension 7

- ABR task manager module
  - Be in charge of creating Summary-LSAs and performing the route summarization for one routing table
  - Be extended

- Precomputation trigger module
  - Be responsible for correctly trigger routing table calculations
  - A new module

- Route table computation module
  - Provide the whole routing table calculation and the incremental update calculation
  - Use a specific routing algorithm
  - More than one route table computation module in the software architecture

Part of software architecture
Static MCR Implementation

- Static MCR implementation
  - Major core Zebra OSPF functions remain unchanged
  - Some important data structures are modified, such as ospf
  - Several functions are modified in order to support MCR capability
  - a lot of new functions are created
  - bandwidth of a link can be configured through VTY commands

Zebra system architecture

- Zebra Software
  - a multi-process architecture
  - a collection of several separate daemons
- zebra is the kernel routing table manager
- ospfd is the daemon for OSPF protocol
Testing

- A hierarchical topology
- We used it to do following tests:
  - Basic function test
  - Pre-computation trigger test
  - Route summarization test

- A triangle topology
- We used it to do Multi-path test
  - Routes to the same destination are different in diverse routing tables

Figure 6.1: A basic function testing environment

Figure 6.3: A multi-path test environment
Results of Test

- Basic function test

Router-LSA
LS age: 1016
Options: 130
Flags: 0x2 : ASBR
LS Type: router-LSA
Link State ID: 192.168.1.1
Advertising Router: 192.168.1.1
LS Seq Number: 80000004
Checksum: 0xc4fa
Length: 56
Number of Links: 2

Link connected to: a Transit Network
(Link ID) Designated Router address: 130.233.154.59
(Link Data) Router Interface address: 130.233.154.36
Number of TOS metrics: 1
TOS 0 Metric: 300
TOS [72] Metric: 13

Link connected to: Stub Network
(Link ID) Net: 192.168.1.0
(Link Data) Network Mask: 255.255.255.0
Number of TOS metrics: 1
TOS 0 Metric: 600
TOS [72] Metric: 33

- Multi-path test

debian1_ospfd# show ip ospf route
-------------- OSPF network routing table --------------
N 130.233.154.0/24  [1100] area: 0.0.0.0
    via 192.168.1.1, eth1
N 192.168.0.0/24  [400] area: 0.0.0.0
directly attached to eth0
N 192.168.1.0/24  [800] area: 0.0.0.0
directly attached to eth1
-------------- OSPF router routing table --------------
-------------- OSPF external routing table --------------
-------------- OSPF BSP network routing table --------------
N 130.233.154.0/24  [8318] area: 0.0.0.0
    via 192.168.1.1, eth1
N 192.168.0.0/24  [9578] area: 0.0.0.0
    via 192.168.1.1, eth1
N 192.168.1.0/24  [3277] area: 0.0.0.0
directly attached to eth1
-------------- OSPF BSP router routing table --------------
-------------- OSPF BSP external routing table --------------
Conclusions

- Based on our OSPF with MCR extensions, the static MCR approach is implemented for the MCR DiffServ system
- Distinct routing algorithms can be used to compute the different routing tables. The routes to the same destination could be different in the diverse routing tables
- A new routing table is added into the OSPF code
- To decrease the number of Summary-LSAs, the route summarization mechanism should be implemented for each routing table
- To prevent any unnecessary whole routing table calculation, the incremental update calculation is needed for each routing algorithm
- To avoid any unnecessary routing calculation, the pre-computation trigger should correctly determine which routing algorithm is executed
Future work

- Different dynamic MCR approaches need to be further discussed and designed.
- Approaches to Measure link weights, such as delay and residual bandwidth, need to be determined since the measurement of link weights could have an impact on the convergence.
- The local interface status manager module is implemented by means of the dynamic MCR approaches.
- New methods to encode bandwidth are needed because our encoding method is not suitable for the dynamic MCR approaches.
- One or more QoS routing algorithms are implemented in the OSPF code supporting the MCR capability.
- Performance and scalability of different MCR approaches are tested in different network environments.
Thank you