EE6412 tutorial problems: MPLS

1. Compare the overall scalability of the following three cases:

Solutions follow questions:
(a) Layer 3 forwarding only: each router performs a longest-prefix match to determine the next forwarding hop; Longest-prefix matching can take much processing time if the routing table is very large. This can limit the rate at which packets can be routed.

(b) Layer 3 forwarding and some Layer 2 MPLS forwarding; The layer 2 forwarding helps to reduce the load of the layer 3 forwarding. This method is very scalable if the majority of packets use the layer 2 forwarding.

(c) Layer 2 MPLS forwarding only. Layer 2 MPLS forwarding provides fast packet forwarding capabilities independent of the routing table size. The packets will experience minimal delay on each hop. However the node cannot handle IP packets that do not have MPLS labels.

2. Discuss what factors determine the level of computational load for label assignment and distribution for the following three cases: (a) topology-driven label assignment, (b) request-driven label assignment, (c) traffic-driven label assignment.

Solution:
(a) Topology-driven label assignment – The label assignment for this approach basically follows the shortest-path algorithms used in the IP network. As a result, label assignment is based on the shortest path found by the IP layer and the computational load is minimal.

(b) Request-driven label assignment – The ingress router requests a traffic flow within an MPLS domain. This requires each LSR to have the current status of the link-state network topology. The computational load for label assignment and distribution is higher than for the topology-driven approach.

(c) Traffic-driven label assignment – The router must monitor the traffic flow/pattern on each of its I/O ports before assigning any label to the link. Computational load is the highest among the three approaches because traffic monitoring requires extra processing overhead in this approach.

3. Compare the following two approaches to traffic engineering:

Solutions follow questions:
(a) Offline (centralized) computation: A central site knows the demand (total bandwidth requirement) between each pair of LSRs and computes the paths in the network. The centralized database contains an updated and accurate network state topology. As a result, it provides the optimal path between any source and destination pair within the domain. However, scalability is a problem. As the network becomes large, the volume of requests becomes difficult to handle.

(b) Online (distributed) computation: An ingress LSR independently computes a path to a specified egress LSR as new demands arrive at the ingress LSR. The processing load is distributed to individual ingress nodes. However, the ingress node must have the updated link-state network topology and this topology may be less accurate compared with the centralized database as multiple LSRs may assign paths at the same time.
4. Why does the label change at each LSR?

The label has only local significance. There are two advantages for this:
- Distributed label assignment makes the implementation scalable
- The same label can be reused for other LSRs somewhere else and hence we can use the label space more efficiently.