**AMPLE: An Adaptive Traffic Engineering System Based on**

**Virtual Routing Topologies**

**Abstract**

Handling traffic dynamics in order to avoid network congestion and subsequent service disruptions is one of the key tasks performed by contemporary network management systems. Given the simple but rigid routing and forwarding functionalities in IP base environments, efficient resource management and control solutions against dynamic traffic conditions is still yet to be obtained. In this article, we introduce AMPLE — an efficient traffic engineering and management system that performs adaptive traffic control by using multiple virtualized routing topologies. The proposed system consists of two complementary components: *offline link weight* *optimization* that takes as input the physical network topology and tries to produce maximum routing path diversity across multiple virtual routing topologies for long term operation through the optimized setting of link weights. Based on these diverse paths, *adaptive traffic* *control* performs intelligent traffic splitting across individual routing topologies in reaction to the monitored network dynamics at short timescale. According to our evaluation with real network topologies and traffic traces, the proposed system is able to cope almost optimally with unpredicted traffic dynamics and, as such, it constitutes a new proposal for achieving better quality of service and overall network performance in IP networks.

Architecture



**Existing system**

 In Existing System, IGP-based TE mechanisms are only confined to offline operation and hence cannot cope efficiently with significant traffic dynamics. There are well known reasons for this limitation: IGP-based TE only allows for *static* traffic delivery through native IGP paths, without flexible traffic splitting for dynamic load balancing. In addition, changing IGP link weights in reaction to emerging network congestion may cause routing re-convergence problems that potentially disrupt ongoing traffic sessions. In effect, it has been recently argued that dynamic/online route re computation is to be considered harmful even in the case of network failures, let alone for dealing with traffic dynamics.

**Proposed System**

In proposed system consists of two complementary components: *offline link weight optimization* that takes as input the physical network topology and tries to produce maximum routing path diversity across multiple virtual routing topologies for long term operation through the optimized setting of link weights. Based on these diverse paths, *adaptive traffic* *control* performs intelligent traffic splitting across individual routing topologies in reaction to the monitored network dynamics at short timescale.

According to our evaluation with real network topologies and traffic traces, the proposed system is able to cope almost optimally with unpredicted traffic dynamics and, as such, it constitutes a new proposal for achieving better quality of service and overall network performance in IP networks.

**Modules**

1. Virtual traffic allocation
2. Offline Link Weight Optimization
3. Network Monitoring
4. Adaptive Traffic Control

**Virtual Traffic Allocation**

In this Module, the diverse MT-IGP paths according to the link weights computed by OLWO. Monitored network and traffic data such as incoming traffic volume and link utilizations. At each short-time interval, ATC computes a new traffic splitting ratio across individual VRTs for re-assigning traffic in an optimal way to the diverse IGP paths between each S-D pair. This functionality is handled by a centralized TE manager who has complete knowledge of the network topology and periodically gathers the up-to-date monitored traffic conditions of the operating network. These new splitting ratios are then configured by the TE manager to individual source PoP nodes, who use this configuration for remarking the multi-topology identifiers (MTIDs) of their locally originated traffic accordingly.

**Offline Link Weight Optimization**

 In this module, to determine the definition of “path diversity” between PoPs for traffic engineering. Let’s consider the following two scenarios of MT-IGP link weight configuration. In the first case, highly diverse paths (e.g. end-to-end disjoint ones) are available for some Pop-level S-D pairs, while for some other pairs individual paths are completely overlapping with each other across all VRTs. In the second case, none of the S-D pairs have disjoint paths, but none of them are completely overlapping either. Obviously, in the first case if any “critical” link that is shared by all paths becomes congested, its load cannot be alleviated through adjusting traffic splitting ratios at the associated sources, as their traffic will inevitably travel through this link no matter which VRT is used. Hence, our strategy targets the second scenario by achieving “balanced” path diversity across all S-D pairs.

**Network Monitoring**

 In this Module, Network monitoring is responsible for collecting up-to-date traffic conditions in real-time and plays an important role for supporting the ATC operations. AMPLE adopts a hop-by-hop based monitoring mechanism that is similar to the proposal.

The basic idea is that a dedicated monitoring agent deployed at every PoP node is responsible for monitoring:

* The volume of the traffic originated by the local customers toward other PoPs (intra- PoP traffic is ignored).
* The utilization of the directly attached inter-PoP links

**Adaptive Traffic Control**

In this Module, Measure the incoming traffic volume and the network load for the current interval as compute new traffic splitting ratios at individual PoP source nodes based on the splitting ratio configuration in the previous interval, according to the newly measured traffic demand and the network load for dynamic load balancing.

**System Requirements:**

**Hardware Requirements:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Color.
* Mouse : Logitech.
* Ram : 512 Mb.

**Software Requirements:**

* Operating system : - Windows XP.
* Coding Language : C#.Net