



---

# Traffic Engineering the Multi-Service Internet

**Prof. George Pavlou**  
**Centre for Communication Systems Research**  
**University of Surrey, UK**  
**G.Pavlou@eim.surrey.ac.uk**

**TEQUILA Consortium**



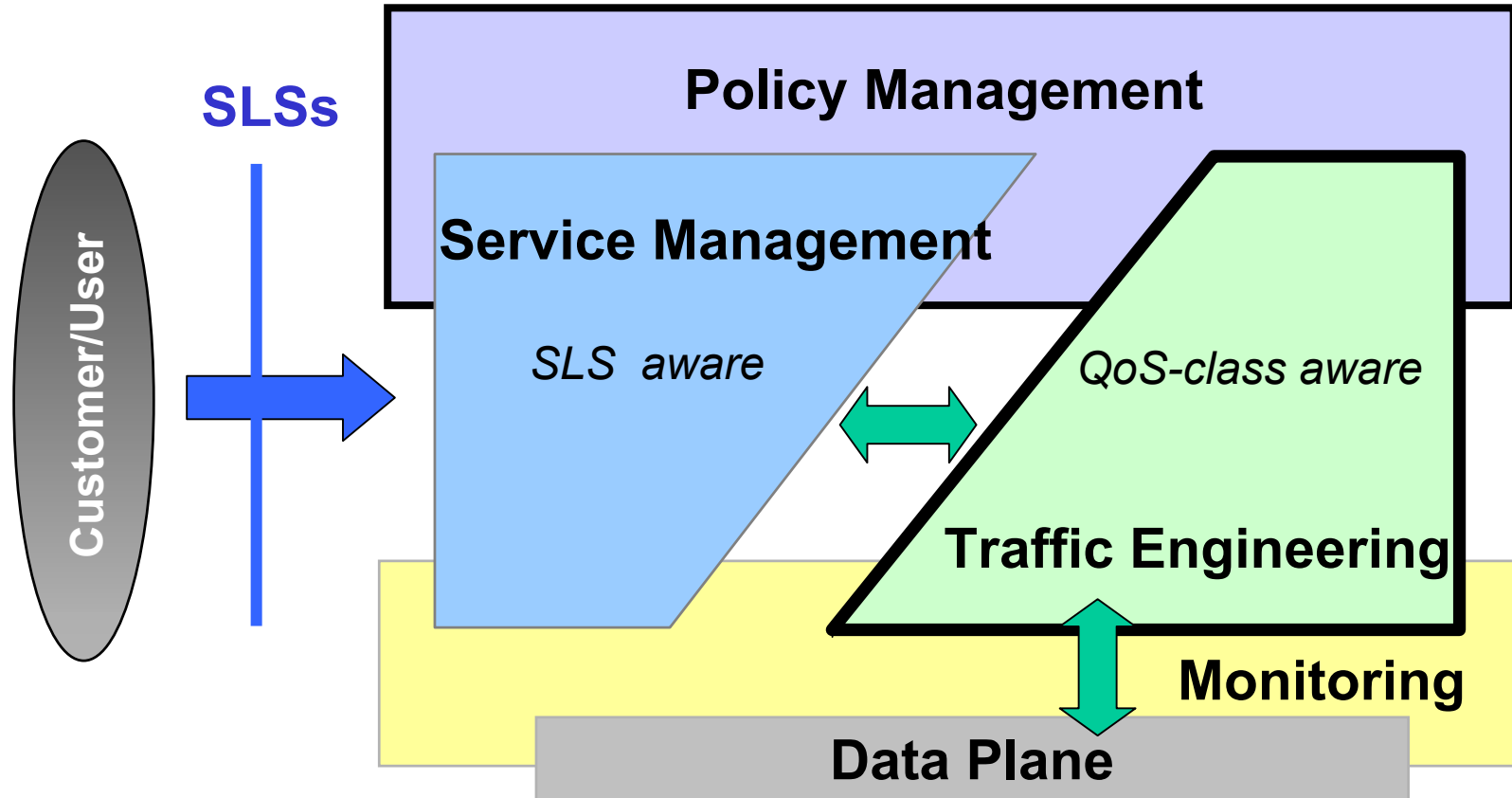
# Introduction

---

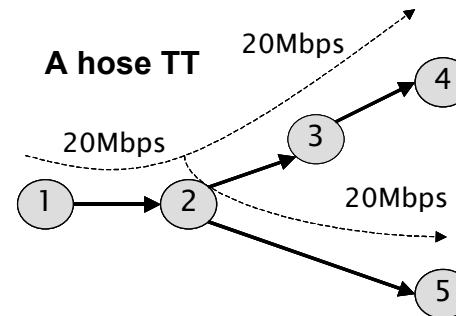
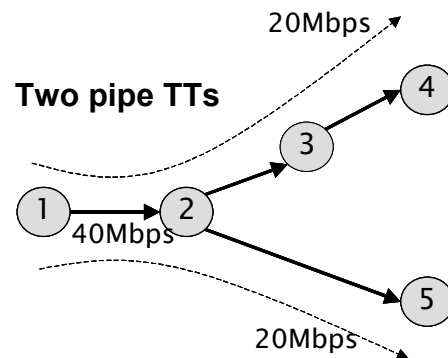
- **Internet: the global multi-service network**
- **Need for scalable Quality of Service (QoS) solutions**
- **Differentiated Services (DiffServ)**
  - **Classify, mark and police at the edges**
  - **Specified “per-hop behaviours” (PHBs) to traffic aggregates**
  - **Scalability compared to per-flow reservation approaches**
- **Traffic Engineering**
  - **Control the manner traffic is mapped to and treated by network to achieve specific performance objectives**
  - **Of paramount importance in DiffServ since there are no explicit resource reservations to flows within the network**
- **Key problem: how to traffic engineer a DiffServ domain to meet edge-to-edge QoS requirements as dictated by contracted SLAs**



# Functional Model for QoS

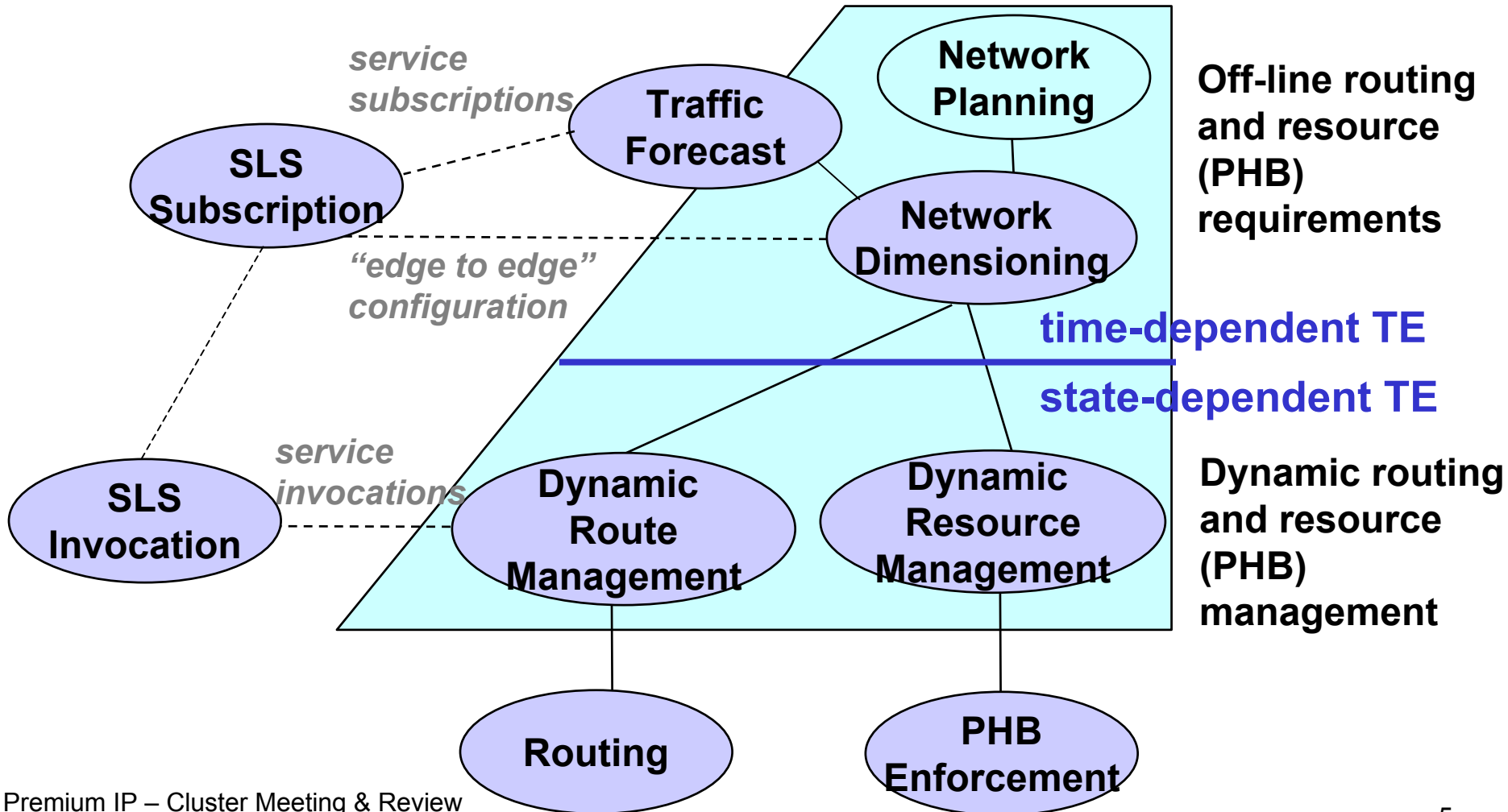


- **Traffic Trunk (TT)**
  - **Ingress, set of egresses**
    - pipe (1:1) or hose (1:N) model
  - **General case the hose model**
    - results in a tree with logically associated “tree bandwidth”





# Traffic Engineering Model





# Traffic Forecast

---

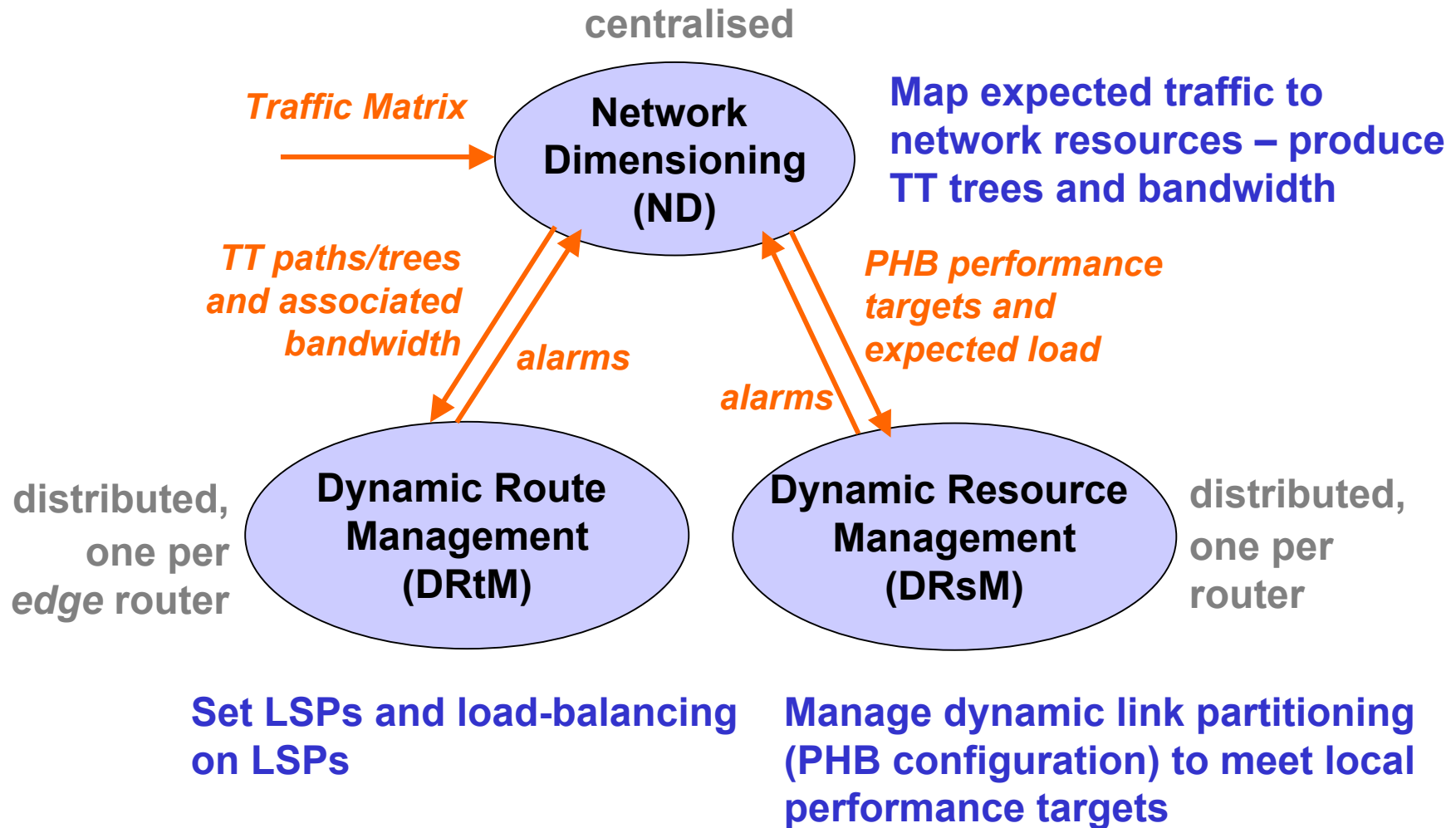
- **Traffic Forecast** is the “glue” between the customer-oriented (SM) and resource-oriented (TE) parts
- Estimates expected traffic demand (**Traffic Matrix**)
  - Derived from service contracts (SLSs) and other information (SLS usage, business policies, forecast/projection)
  - A traffic matrix for every provisioning cycle
- **Aggregation** required for scalability
  - Maximum entries per edge node  $N \cdot 2^{N-1} \cdot Q$  for  $N$  edge nodes and  $Q$  QoS classes



# TE Approaches

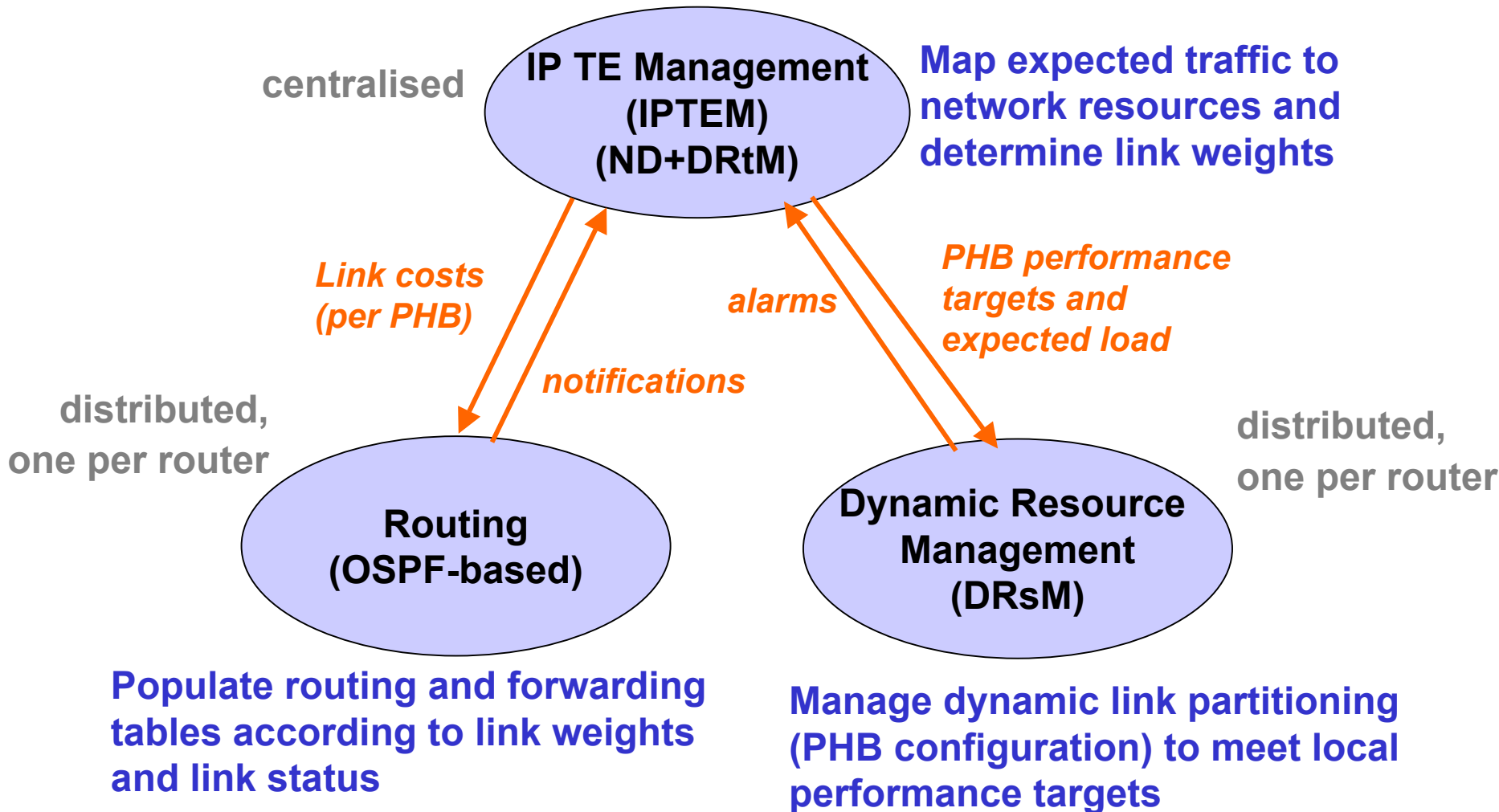
---

- **MPLS-based**
  - Explicit routing through LSPs with alternative LSPs for source-destination combinations for load balancing
  - LSPs are *not* explicitly associated with bandwidth
- **IP-based**
  - Hop-by-hop routing, OSPF-based with Equal Cost Multi Path (ECMP) for load balancing
  - Assignment of link-weights
- **Loss and delay constraints are translated to route hop-count constraints (PHBs are associated with delay and loss bounds)**



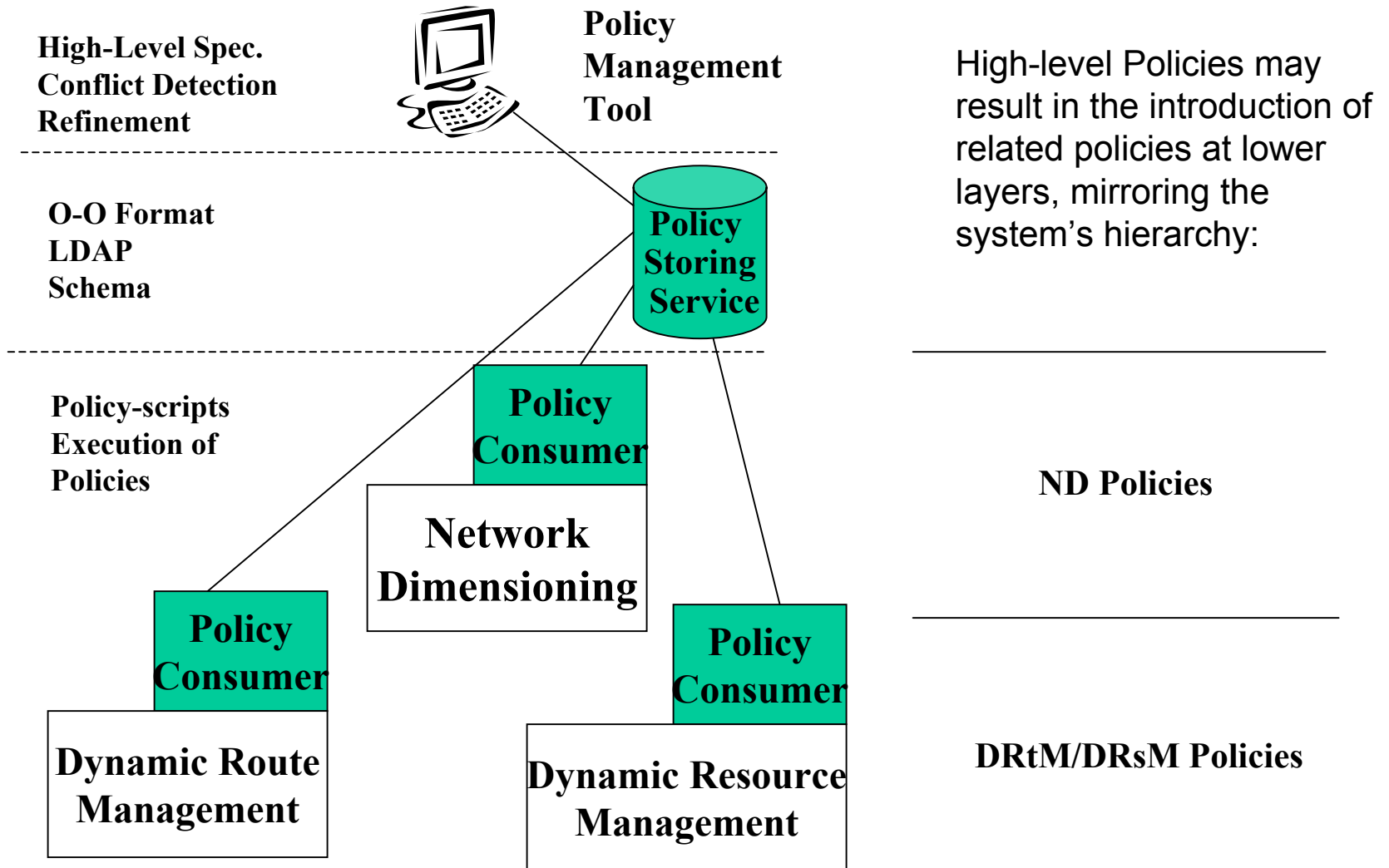


# IP-based TE





# Policy-based TE - Hierarchical Approach





# Policy Consumer Decomposition

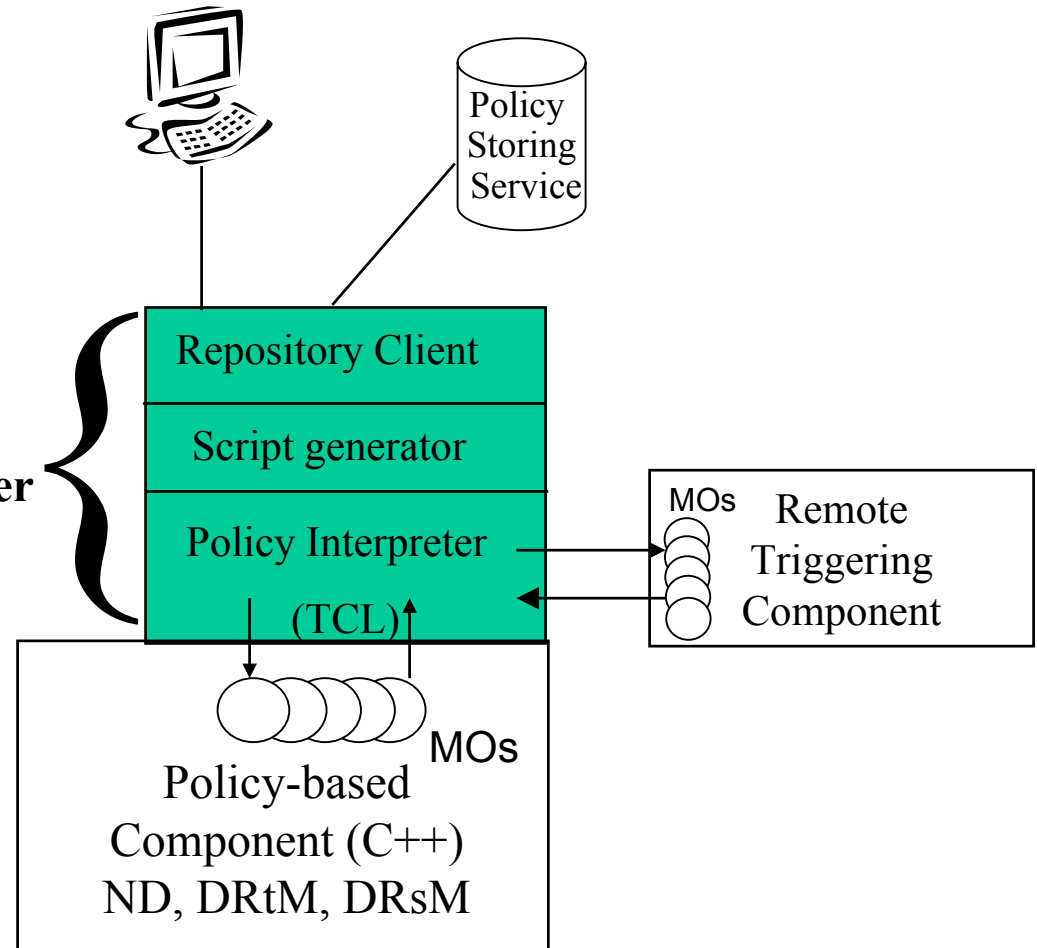
## Policies are seen as:

- a means to achieve programmability in the Tequila System
- logic (scripts) downloaded, interpreted and executed on the fly

Policy Consumer

## Target:

- a TEQUILA system able to sustain requirement changes and evolve through policies without changing its initial “hard-wired” logic





# Network Dimensioning Policies

---

- **Setting initial ND parameters e.g. maximum number of alternative trees, cost function, dimensioning period**
- **Influencing the capacity allocation and LSP creation e.g. allocation of network/link allocated bandwidth per class of service, explicit setup of LSPs, treatment of spare/over-provisioned capacity**



# Network Dimensioning Optimisation Problem (MPLS)

- Satisfy the QoS requirements of traffic trunks
  - Avoid overloading parts of the network (I)
  - Minimise overall network utilisation (II)
- Assuming a cost function  $f(x)$  per PHB depending on the current load and  $F(x)$  the derived cost function per link

- minimise  $(\max_{l \in E} F_l)$  satisfies (I)

- minimise  $\sum_{l \in E} F_l$  satisfies (II)

- Combined objective function

- minimise  $\sum_{l \in E} (F_l)^n$  satisfies both: (II) for  $n=1$  and (I) for  $n \rightarrow \infty$

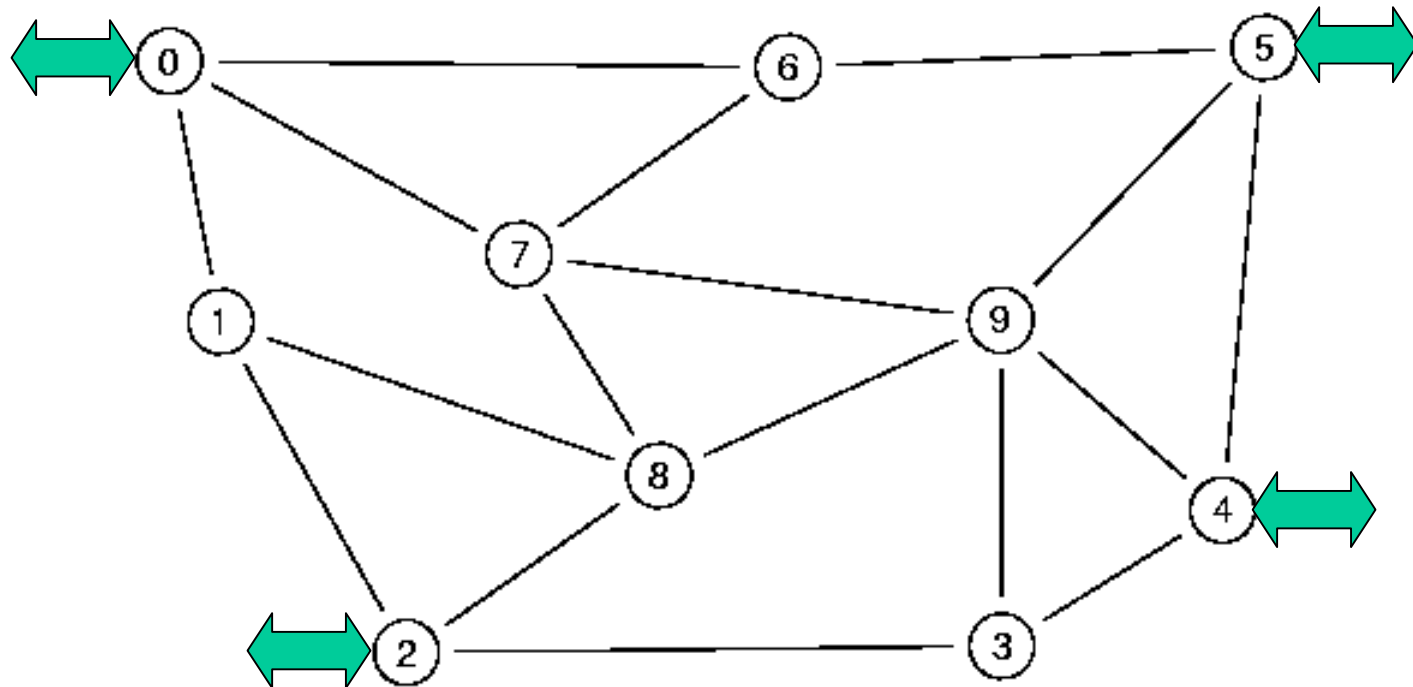


# Optimisation Approach

---

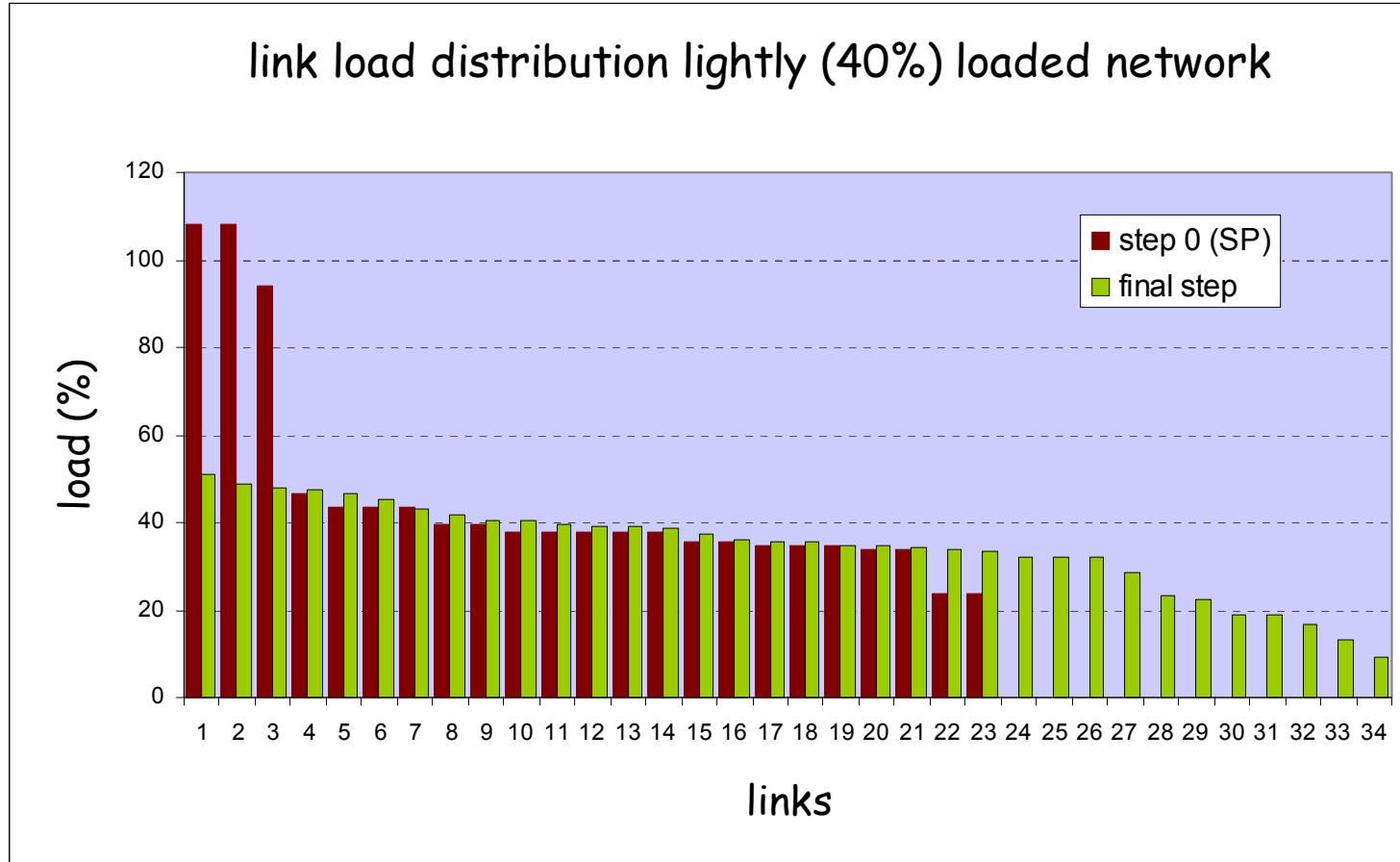
- **Solution-based on an iterative procedure**
  - Start with an initial TT allocation
  - Improve gradually on this solution by moving capacity to “better” paths
- **At each step, a **constrained shortest path** algorithm is used for pipe trunks and a **constrained Steiner tree** algorithm for hose trunks**
  - The algorithms are **constrained due to delay & loss bounds**, translated to a hop count constraint
- **Iterative** algorithm works well and converges quickly for pipe trunks
  - Results are presented next
- **The constrained Steiner tree algorithm also works well, in the next step we will be integrating the two**

- 10 node network – 4 edge nodes { 0, 2, 4, 5 }



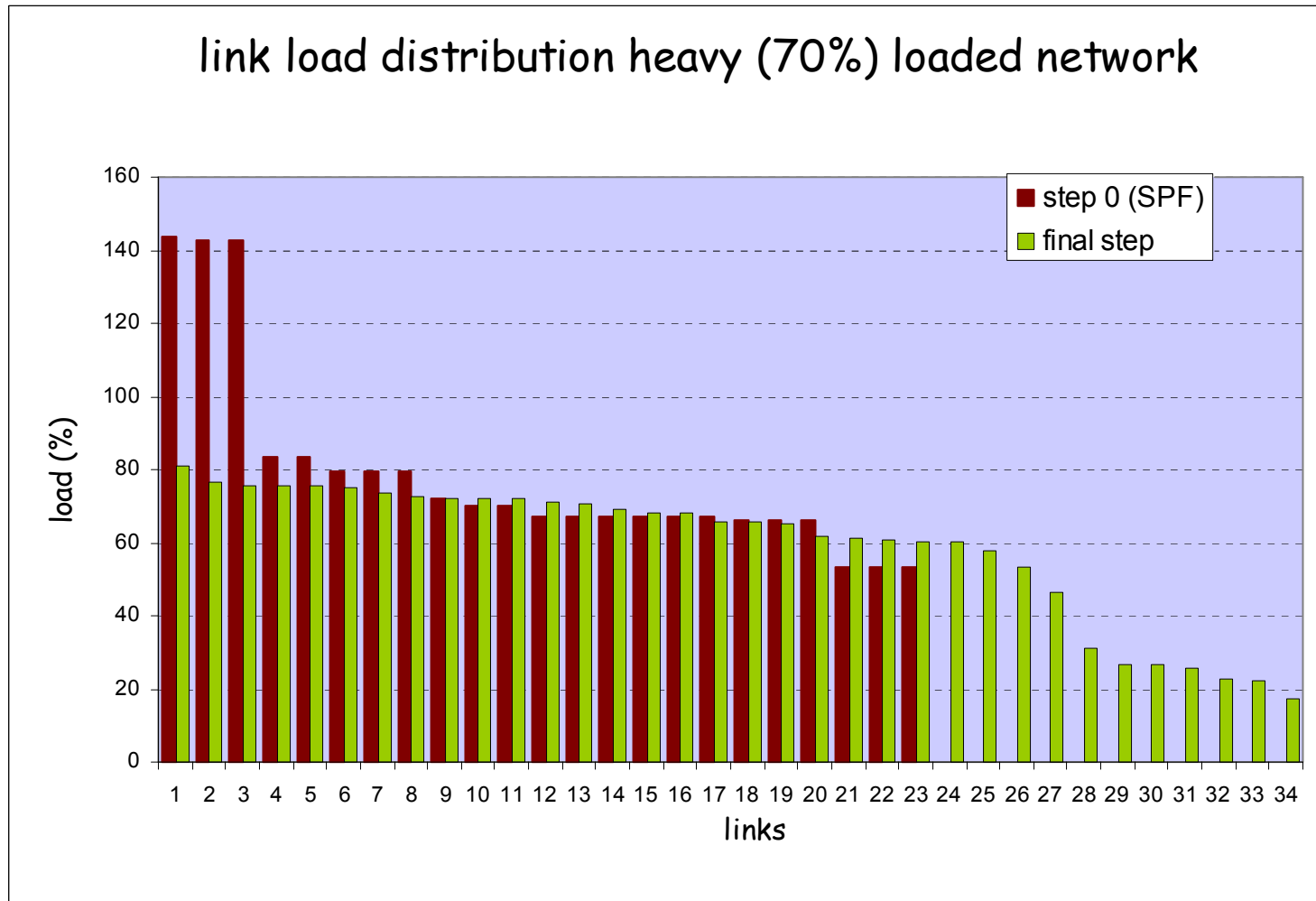


# Link Load Distribution – light load (40%)



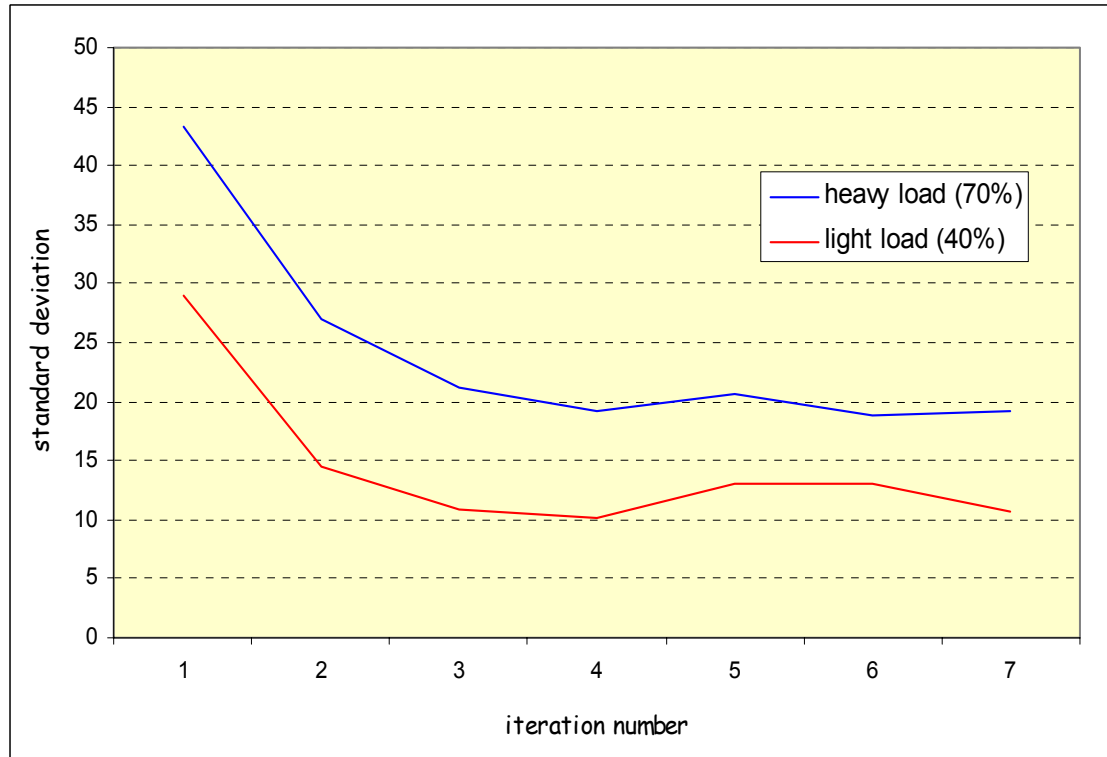


# Link Load Distribution – heavy load (70%)





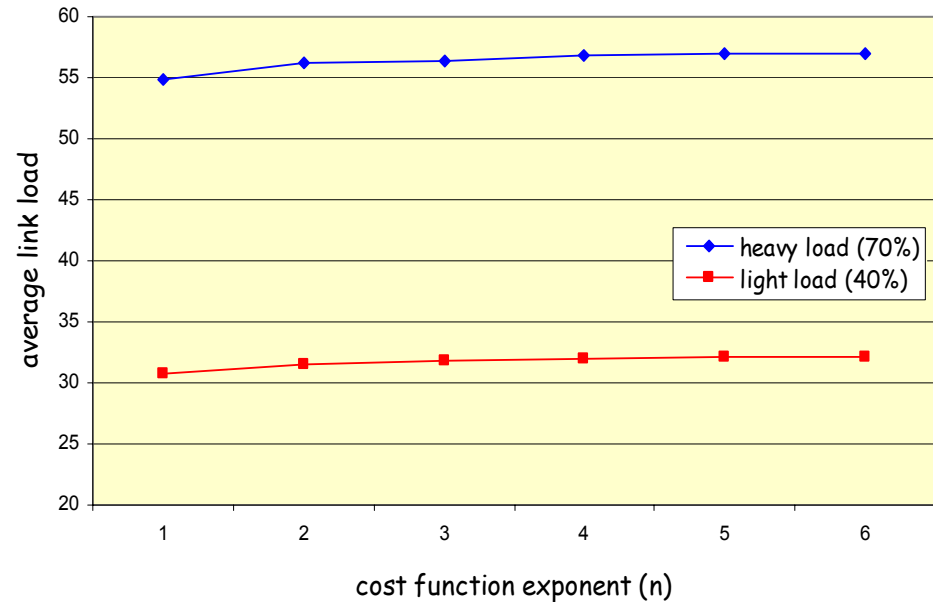
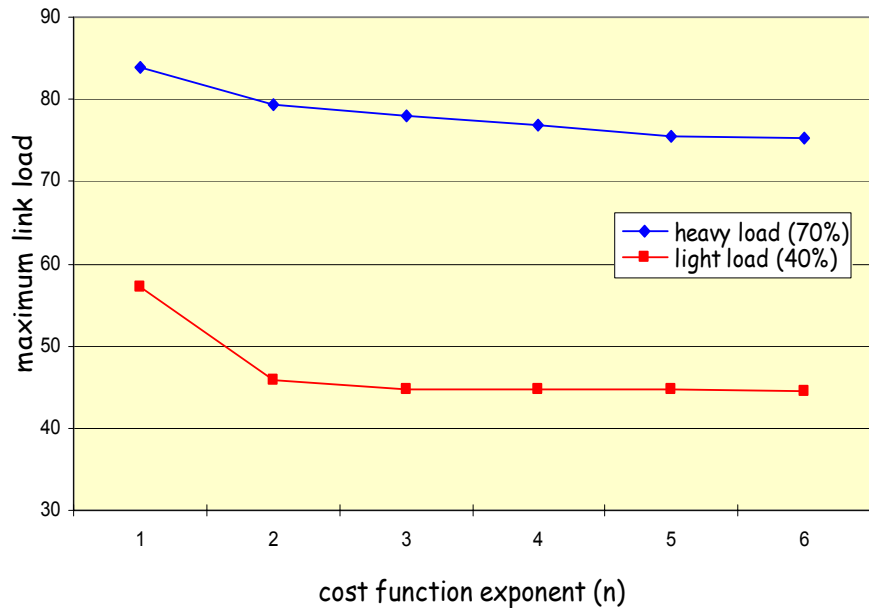
# Stepwise (Per-Iteration) Solution Improvement



**cost function exponent  $n=2$**



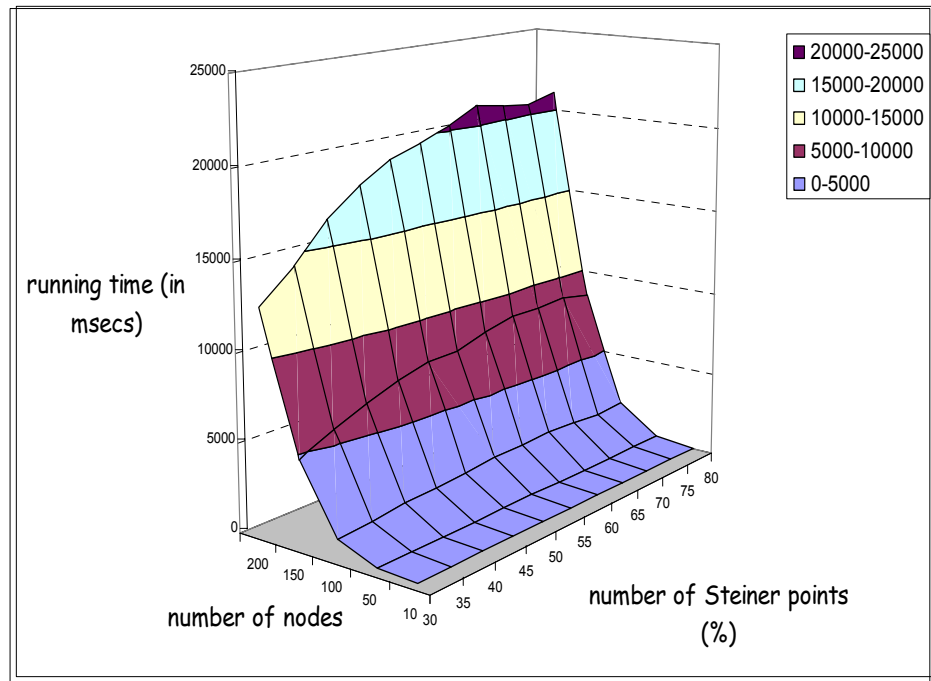
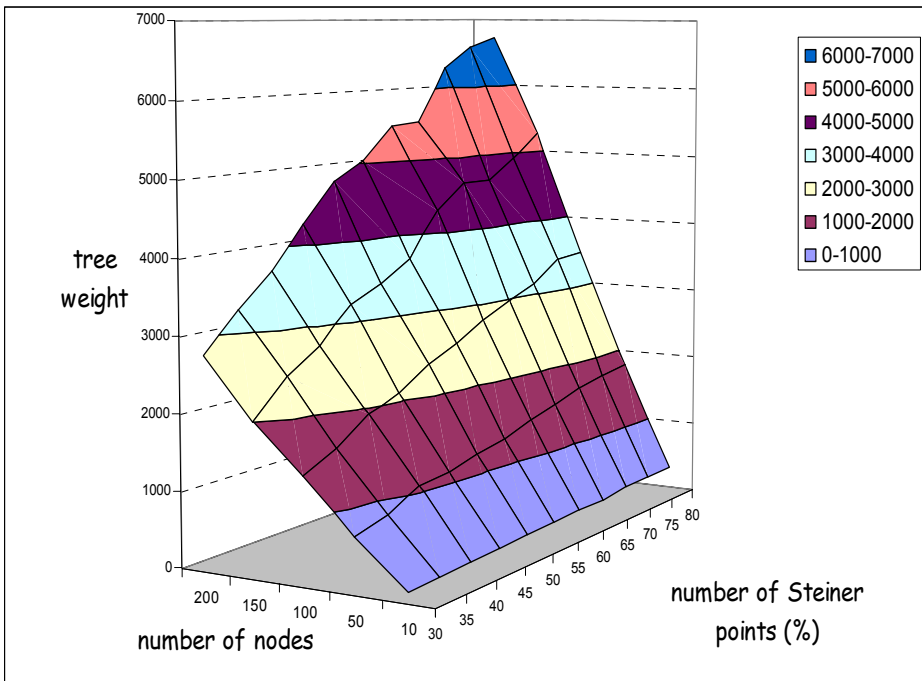
# Effect of Optimisation Criterion (Cost Function)



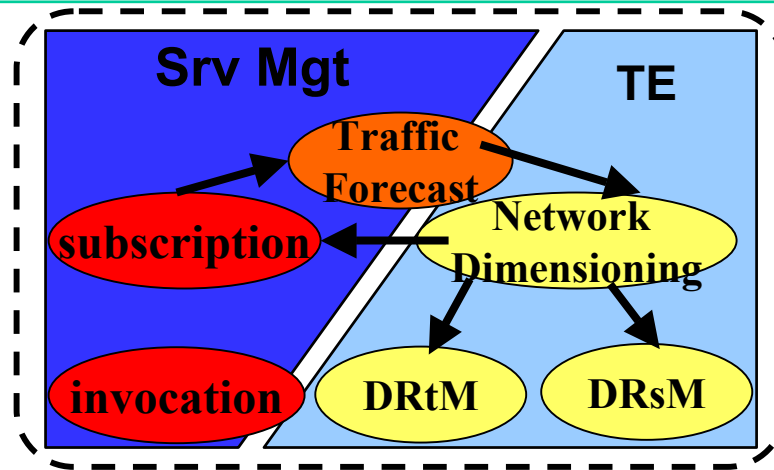
$$\text{minimise } \sum_{l \in E} (F_l)^n$$



# Steiner-tree Algorithm Evaluation



**Since this experimentation, we have improved the above results by approximately 50% (tree weight/cost) and 20% (running time)**



- **Service driven traffic-engineering**
- **Two-level TE approach**
  - long-term --> guidelines for network operation
  - short-term --> handling traffic fluctuations
- **Policy-driven TE operation**
  - graceful evolution to requirements changes
- **Interim results prove the feasibility of the approach**



# TE Functionality

- **Network Dimensioning**
  - Input: network topology, traffic matrix, policies, alarms
  - Objective: optimisation problem
    - Maintain low link cost while satisfying QoS objectives
  - Output in the form of configuration directives:
    - Explicitly routed paths (MPLS-based TE) – via DRtM
    - Values for the link cost metrics (IP-based TE) – directly configuring routers through combined DRtM and ND (IPTeM)
    - Configuration of PHB partitioning per link – via DRsM
- **Dynamic Route Management (DRtM)**
  - Multi-path load distribution at ingress nodes for load balancing
  - Explicit component in MPLS, implicit in OSPF/ECMP
- **Dynamic Resource Management (DRsM)**
  - Re-configuration of PHBs within allowed bounds i.e. dynamic link (re-)partitioning among service classes
  - Same functionality in both MPLS and IP-based TE



# Policy Information Model/Language

- **Information Model based on PCIM and PCIME and corresponding LDAP schema**
- **High-level policy language syntax:**  
[Policy ID] [Group ID] [time period condition] [if {condition [and] [or]}] then {action [and]}
- **Example :**  
**PolicyRule1: From Time=0800 to Time=1800 If (OA == EF) then allocateNwBw > 30%**  
**Description: At least 30% of Network Bw should be available for EF traffic between 08:00 and 18:00 ”**



# Dynamic Route Management (MPLS) – static part

---

- **Explicit component only in MPLS-base TE**
  - One instance in every *edge* Label Switch Router (LSR), uses monitoring facilities within that LSR
- **Static part** gets as input from ND TT trees (and paths) with logically associated bandwidth
- **Creates LSPs to support paths and trees**
  - For trees, one LSP per egress leaf is created (unique path)
  - LSPs are created via LDP from the ingress nose with the full explicit path
  - No bandwidth is associated to LSPs but the edge LSR knows the logically associated bandwidth with each LSP for a path or set of LSPs realising a tree
- **Maps “address spaces” from collected prior statistics (and SLS information) to LSPs according to their bandwidth**
  - Configures LSR forwarding tables according to this assignment



# Dynamic Route Management (MPLS) – dynamic part

---

- **Re-maps address spaces** to alternative LSPs for load balancing according to traffic fluctuations
  - Difficult issue since existing micro-flows should be left to terminate
- **Sends TT over-utilisation alarms to ND**
- **Sends alarms to SLS Invocation that traffic capacity to particular egress nodes is saturated**
  - These act as “Congestion Notifications” to admission control
- **Learns about critical PHB QoS performance** at nodes crossed by its LSPs in order to take *proactive* measures
- **Learns about critical end-to-end LSP QoS performance** in order to take *reactive* measures
  - Knowledge about PHB and LSP performance is obtained from monitoring
  - Critical performance means “persevering conditions” and not instantaneous inability to deliver the specified QoS



# Dynamic Resource Management

- **Common in both MPLS and IP-based TE**
  - one instance in every router, uses monitoring facilities within that router
- **Static part** configures PHB parameters
  - queue type – Drop tail, RED, etc.
  - queue parameters – buffer size, precedence/drop levels
  - scheduling parameters – RR, WRR, PRI, etc.
- **Also configures performance targets set by ND per PHB**
  - bandwidth, max loss probability, max delay
  - allowed bounds for dynamic (state-dependent) operation
- **Dynamic part** manages PHB resources in real-time
  - re-distributes (within allowed bounds) bandwidth, buffer and scheduling resources to meet dynamic traffic fluctuations
  - sends over-utilisation alarms to ND