Traffic Engineering and Protection

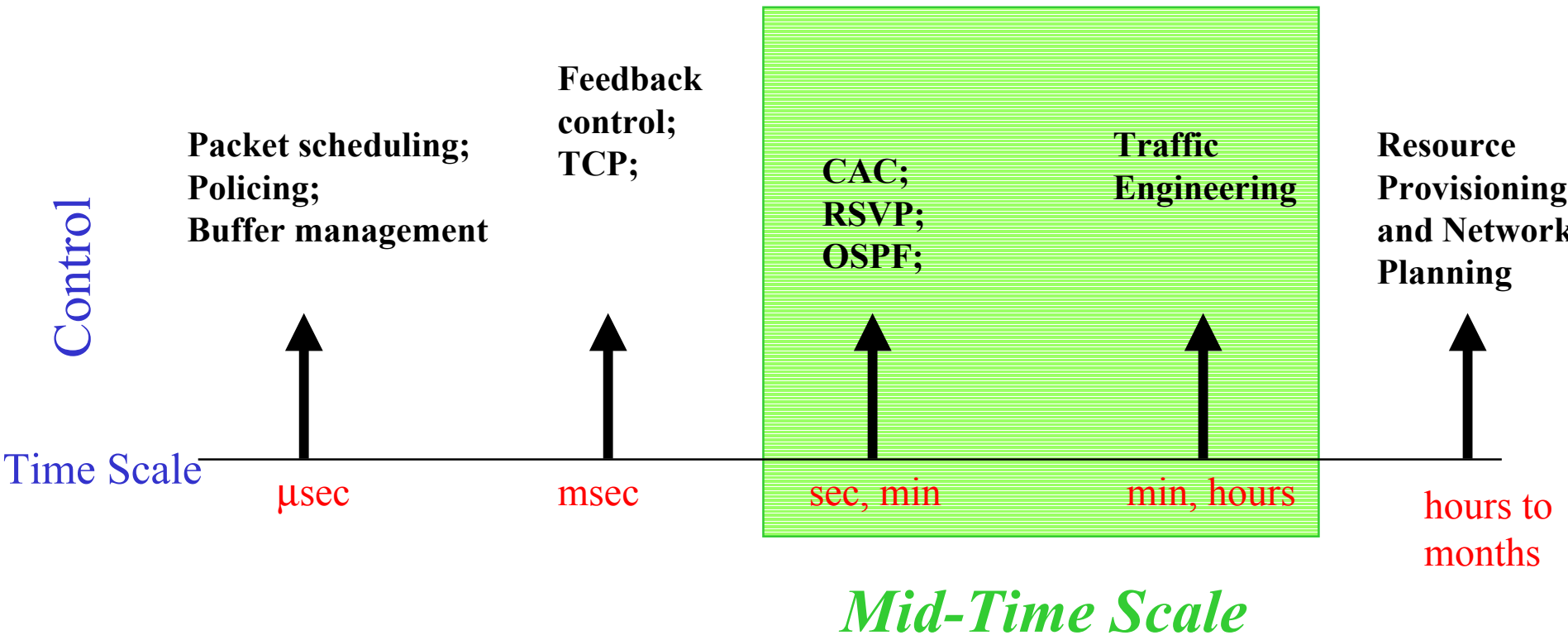
Anwar Walid
with: Debasis Mitra, Nachi Nithi

Lucent Technologies



Lucent Technologies
Bell Labs Innovations

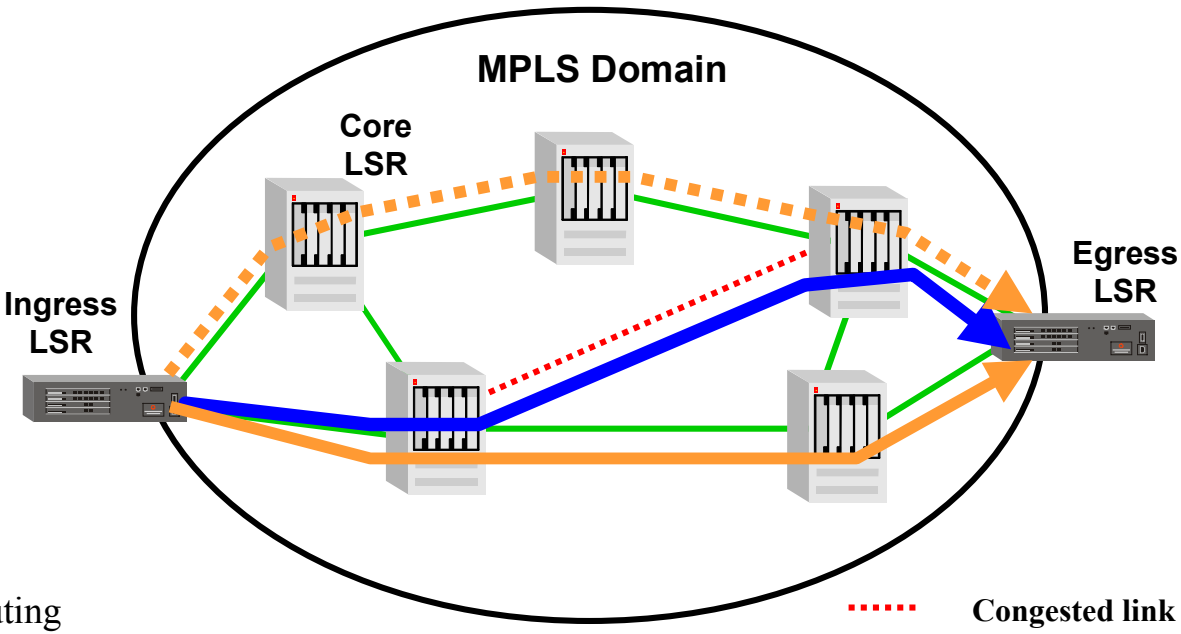
Time-Scale Classification of Traffic Engineering & QoS Management



Lucent Technologies
Bell Labs Innovations

Traffic Engineering (TE)

- **Optimal selection of primary and protection routes based on network topology and resources to maximize carried traffic and meet QoS**
- **Adaptive load balancing among available LSPs to minimize congestion**



LSR: Label-Switched Router
LSP: Label-Switched Path

LSP using shortest-path routing
LSP using optimized routing
Protection LSP

Congested link
Uncongested link

Lucent Technologies
Bell Labs Innovations

Summary of Research Plans

- **Path design and protection :**
  - **Algorithms and methodologies**:
    - Efficient design of many LSPs with various QoS and protection requirements.
  - **Research prototype development and demonstration**

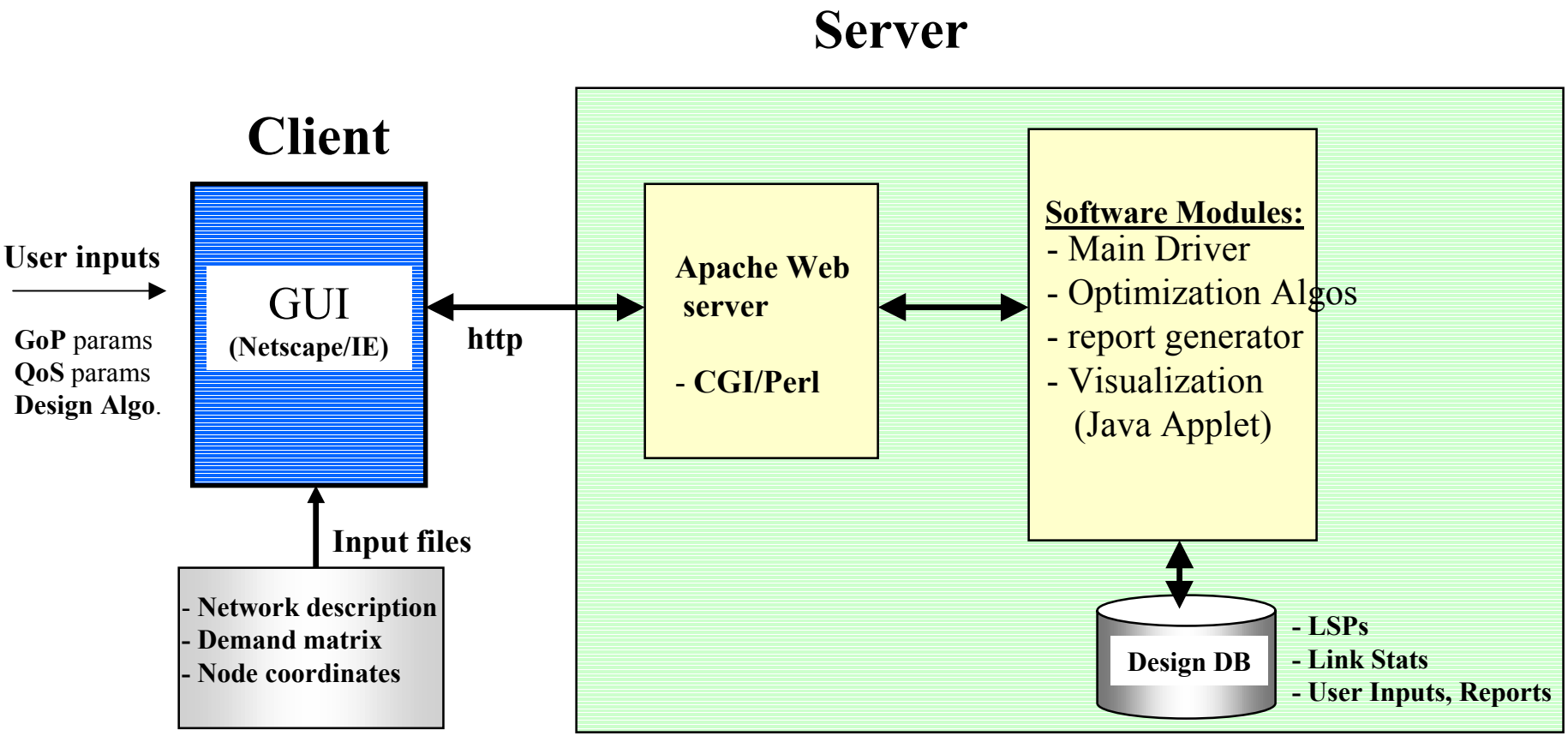
- **Adaptive Load balancing :**
  - **Algorithms and design methodology:**
    - Support multi-QoS classes and path measurements based on router alerts.
  - **Simulation Tool**:
    - Support any network of general topology and link speeds.
    - Explore speed-up techniques, fluid simulation
  - **Software Implementation of Measurement Protocols:**
    - Compare router-alert mechanisms with probing methods in terms of bandwidth consumption, accuracy, response time and inter-operability.
    - Implement hashing functions to filter and distribute packets at the flow level
    - Use real traffic traces to feed the simulation program.
- **Integration of network load measurements with path optimization**

Challenges and Requirements

- **Path Design and Protection:**
  - Involves solving large Optimization problems
  - Design for multi-class QoS and various grades of protection adds to complexity
  - Availability and accuracy of traffic demand matrices
  - SLA conversion: translation of service requirements to network resource requirements

- **Adaptive Load Balancing:**
  - Stability of asynchronous adaptive load balancing
  - Speed of conversion
  - Knowledge of *one-way* path statistics
  - Minimum additional requirements on routers

MPLS LabelStar: Software Architecture



Lucent Technologies
Bell Labs Innovations

MPLS LabelStar: Software Modules

- **GUI (Perl, Java):**
  - •Allows Users to input
    - -network topology and resources description (files)
    - -demand matrix, GoP parameters, QoS parameters, design algo. selection
  - •Visualization of results/network
    - -LPSs, Link statistics, summary statistics and user inputs
    - -Network topology, load on each links, routes of LSPs
- **Plug-able Algorithms (implemented in C, CPLEX)**
  - •Unprotected design (SP and CSP)
  - •1+1 protection design
  - •Shared protection design
- **Pre-processing of input data for cleaning and validation (Perl, C)**
- **Post-processing to interpret and generate various statistics and reports (Perl,C)**
- **Potential for integration with other time-scale modules**



MPLS LabelStar: Outputs

Reports generated:

- LSP table: set of all routed (primary and backup) LSPs are listed for each source-destination node pairs.

LSP	Source - Dest.	Bandwidth	Explicit Path		
Primary	WashingtonDC - SanDiago	20GB	WashingtonDC, SanDiago		
Protection	WashingtonDC - SanDiago	20GB	WashingtonDC,Houston,KansasCity,Denver,SanDiago		

- Link Usage table: gives a summary of LSPs along each link in the network.

Link	Tot. Bandwidth	Used	Primary LSP	Protection LSP
WashingtonDC - NewYork	20GB	10GB	10	5

- Summary statistics : aggregate bandwidth utilization, hotspots, etc.

Online versus Offline TE

Online Provisioning:

- Connection requests arrive one-by-one
- Traffic demand matrix usually not used
- Each path is computed individually
- Performance is sub-optimal and dependent on the order of requests

Offline Provisioning:

- All connection requests are known a priori
- Traffic demand matrix known precisely
- Paths are computed in a batch
- Performance can be optimal

Lucent Technologies
Bell Labs Innovations

Can Online TE be improved?

Provisioning with CSPF:

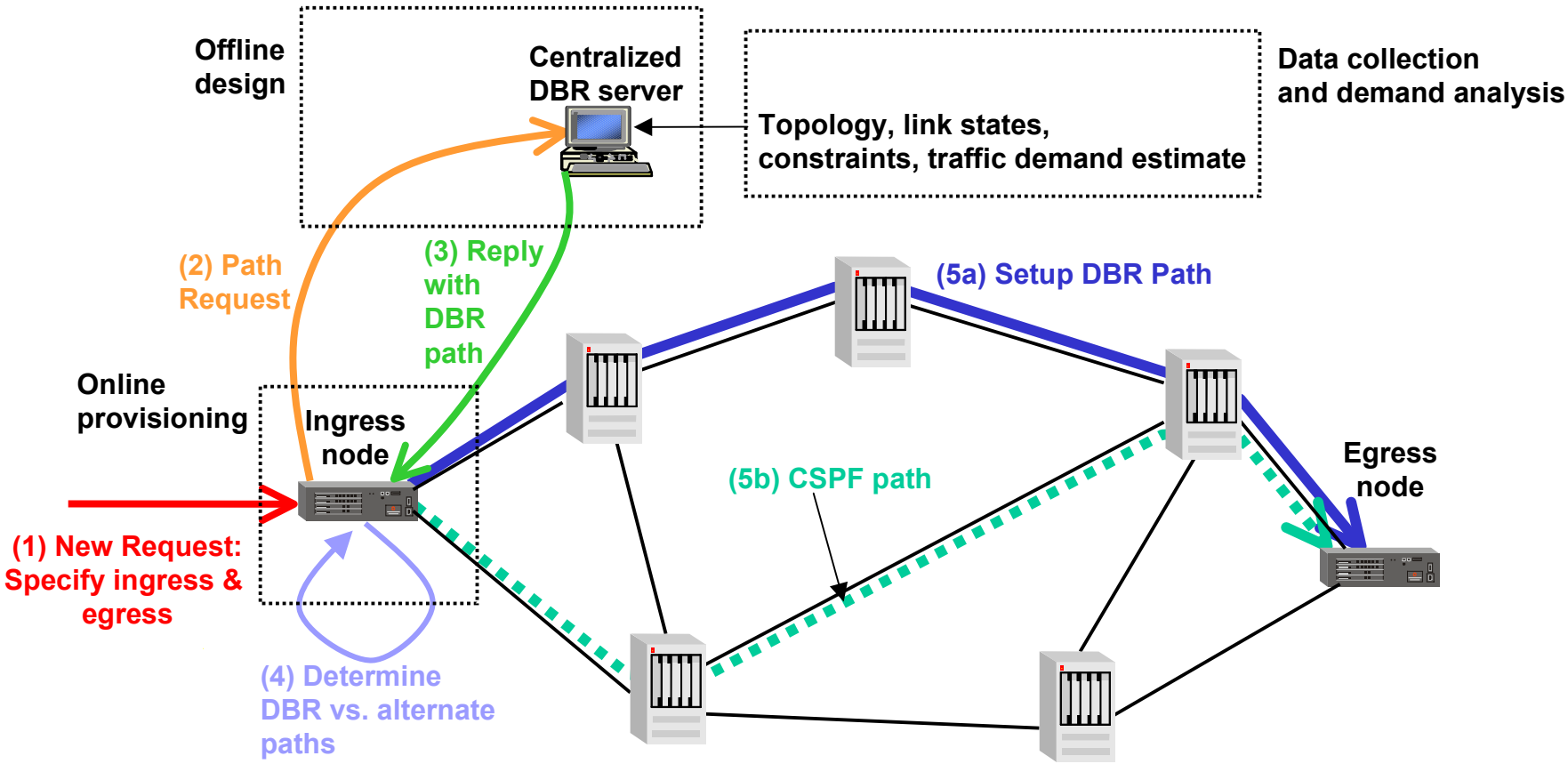
- Connection requests arrive one-by-one
- Traffic demand matrix not used
- Each path is individually computed online
- Performance is sub-optimal

Provisioning with DBR:

- Connection requests arrive one-by-one
- *Approximate* traffic demand matrix assumed
- Paths are computed offline and assisted by online computation
- Performance can be significantly better
- Reference: “Online traffic engineering with design-based routing (DBR)” ITC Specialist Seminar 2002, Widjaja, Saniee, Elwalid, Mitra



Path Provisioning - A Centralized DBR Implementation



- *It is possible to have the path computation process distributed*

Lucent Technologies
Bell Labs Innovations

Path Design

Stage 1:

P1(λ * length)

$$\min \sum_{e \in E'} \lambda_e l_e \quad (1)$$

subject to

$$\sum_{p \in P'_k} x_p \geq d_k, \quad \forall k \in K \quad (2)$$

$$\sum_{p \in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \quad (3)$$

$$\lambda_e \leq C_e, \quad \forall e \in E \quad (4)$$

$$x_p \geq 0, \quad \forall p \in P \quad (5)$$

Stage 2:

Q1(λ * length)

$$\min \sum_{e \in E} \lambda_e l_e \quad (6)$$

subject to

$$\sum_{p \in P_k} x_p \geq d_k, \quad \forall k \in K \quad (7)$$

$$\sum_{p \in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \quad (8)$$

$$\lambda_e \leq C_e, \quad \forall e \in E \quad (9)$$

$$x_p \geq 0, \quad \forall p \in P \quad (10)$$

Notations:

- Network topology *G(V, E)*
- *Each edge e has capacity C_e and length l_e*
- *E' = E U dummy links with high costs*
- *Demand d_k for each node pair k*
- *P_k = set of all paths for node pair k*
- *x_p = amount of flows assigned to path p*
- *λ_e = amount of flows used on edge e*
- *Q_e = set of all paths on edge e*

Lucent Technologies
Bell Labs Innovations

Uncertain Static Connections

Q2(λ * length)

$$\min \sum_{e \in E} \lambda_e l_e \tag{11}$$

subject to

$$Pr\{\sum_{p \in P_k} x_p \geq d_k\} \geq 1 - \epsilon_k, \quad \forall k \in K \tag{12}$$

$$\sum_{p \in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \tag{13}$$

$$\lambda_e \leq C_e, \quad \forall e \in E \tag{14}$$

$$x_p \geq 0, \quad \forall p \in P \tag{15}$$

As in $Q1$, the above stochastic optimization problem provides x_p, the flow assignments to each path p. There are several ways to solve $Q2$, of which the most direct solution is as follows. Define η_k such that

$$erf(\eta_k) = 1 - \epsilon_k \tag{16}$$

where

$$erf(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-y^2}{2}} dy \tag{17}$$

Then the constraint set (12) can be rewritten as

$$\sum_{p \in P_k} x_p \geq \bar{d}_k + \eta_k \sigma_k \tag{18}$$



Lucent Technologies
Bell Labs Innovations

Dynamic Connections

Q3($\lambda * \mathbf{length}$)

$$min \sum_{e \in E} \lambda_e l_e \tag{20}$$

subject to

$$B(\delta_k, N_k) \leq \beta, \quad \forall k \in K \tag{21}$$

$$\sum_{p \in P_k} x_p \geq N_k, \quad \forall k \in K \tag{22}$$

$$\sum_{p \in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \tag{23}$$

$$\lambda_e \leq C_e, \quad \forall e \in E \tag{24}$$

$$x_p \geq 0, \quad \forall p \in P \tag{25}$$

Notice that the non-linear constraint (22) can be eliminated by first solving for N_k. In this case, the optimization problem can be simplified by using the procedure as follows:

Q4($\lambda * \mathbf{length}$)

1. Compute $N_k^* = \text{argmin}_{N_k} \{B(\delta_k, N_k) \leq \beta\}, \forall k \in K$.
2. Solve Q1 with $d_k \leftarrow N_k^*, \forall k \in K$ (using preprocessing P1 if necessary).

Lucent Technologies
Bell Labs Innovations

Performance Investigation

- Investigate how DBR performs with respect to other online path computation approaches
- Experiment with two cases:
  - Static: established connections stay permanently
  - Dynamic: connections arrive and depart at random
- Experiment with different path protection modes:
  - No protection
  - Dedicated protection
  - Shared protection

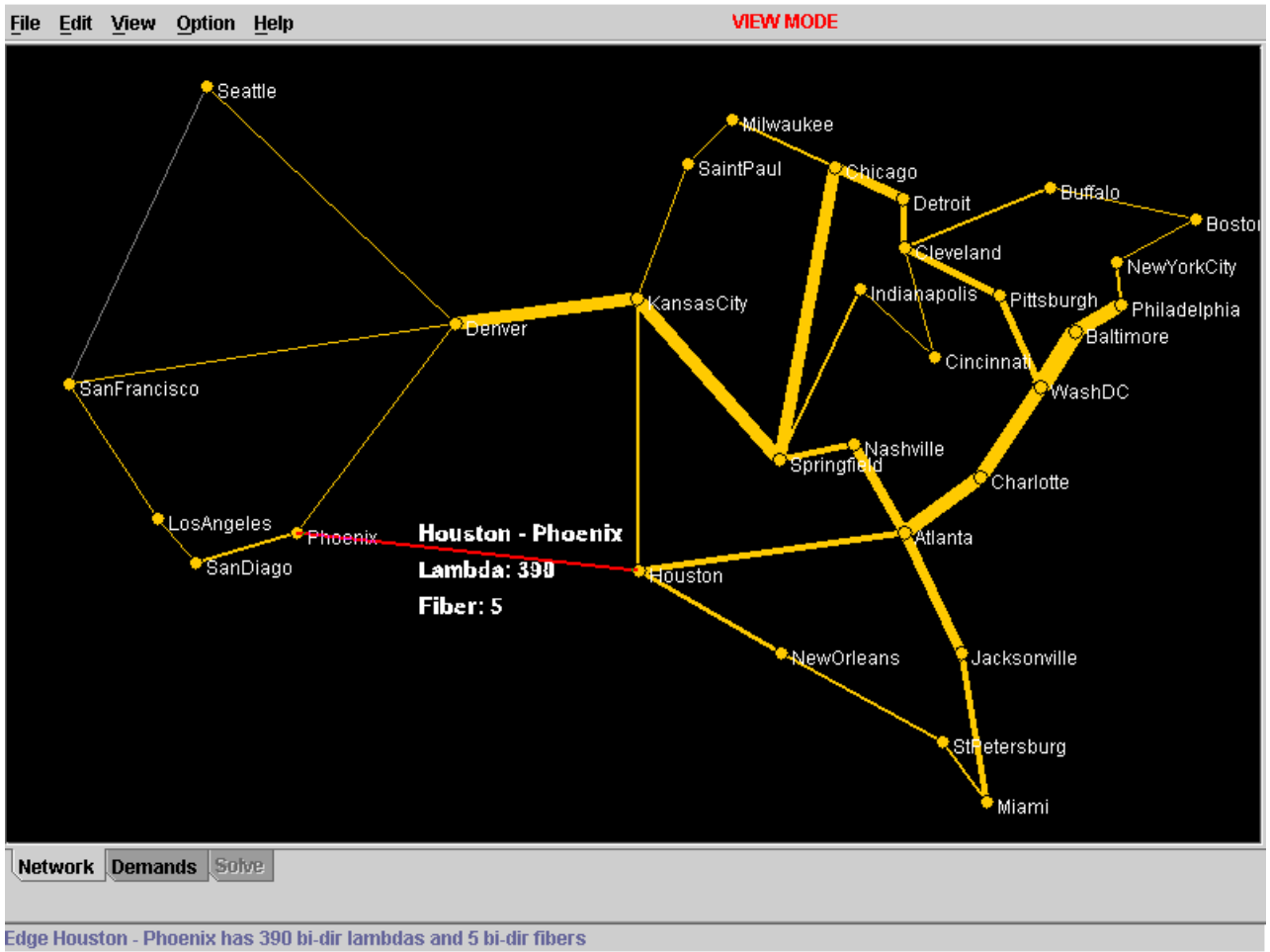
Competitive Analysis

- Performance of scheme X is compared with DBR via the *Competitive Ratio*:

$$CR(x) = \frac{E[\text{\# of routed connections via DBR}]}{E[\text{\# of routed connections via scheme X}]}$$

In the examples, X is Shortest Path First (SPF) or Constraint Shortest Path First (CSPF)

A Generic US Network



30 Nodes

38 Links

435 Demands

each 1-79 λs

average ~8 λs

3495 λs in total

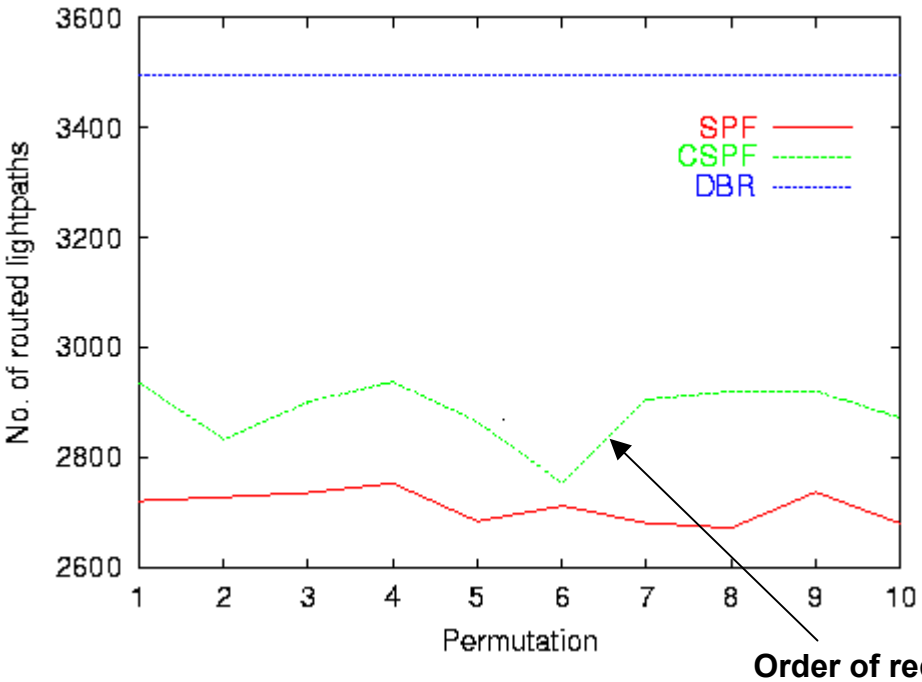
4.3 Average hops per primary path

11.9 Average per primary + protection path

Lucent Technologies
Bell Labs Innovations

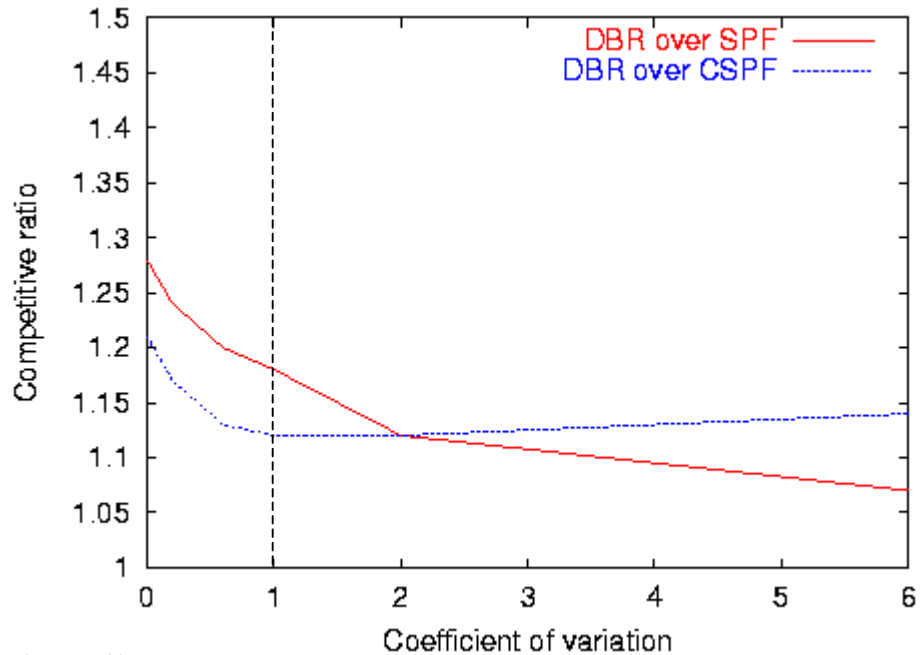
Static Case (No Protection)

Accurate demand forecast:



Order of requests matters

Inaccurate demand forecast:



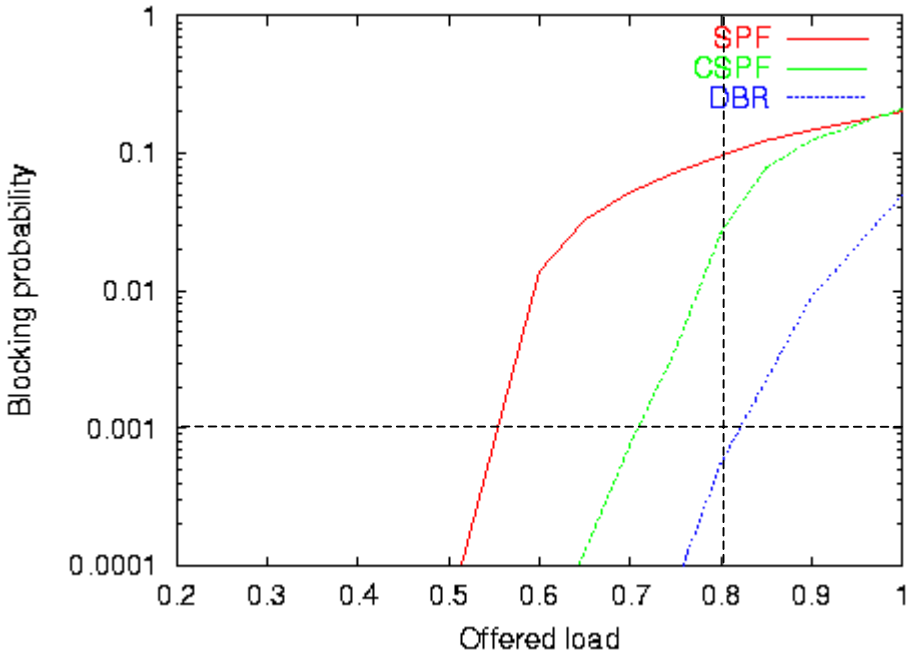
- DBR can route 10-25 percent more traffic than SPF and CSPF

- CSPF deteriorates (cf. SPF) as the network becomes congested

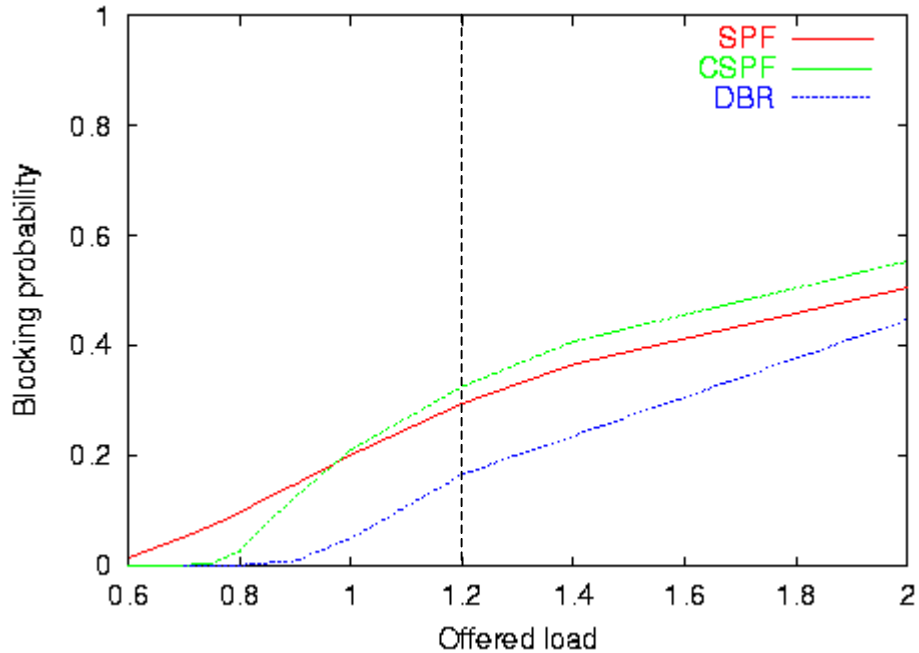
Lucent Technologies
Bell Labs Innovations

Dynamic Case (No Protection)

Moderate load:



Heavy load:

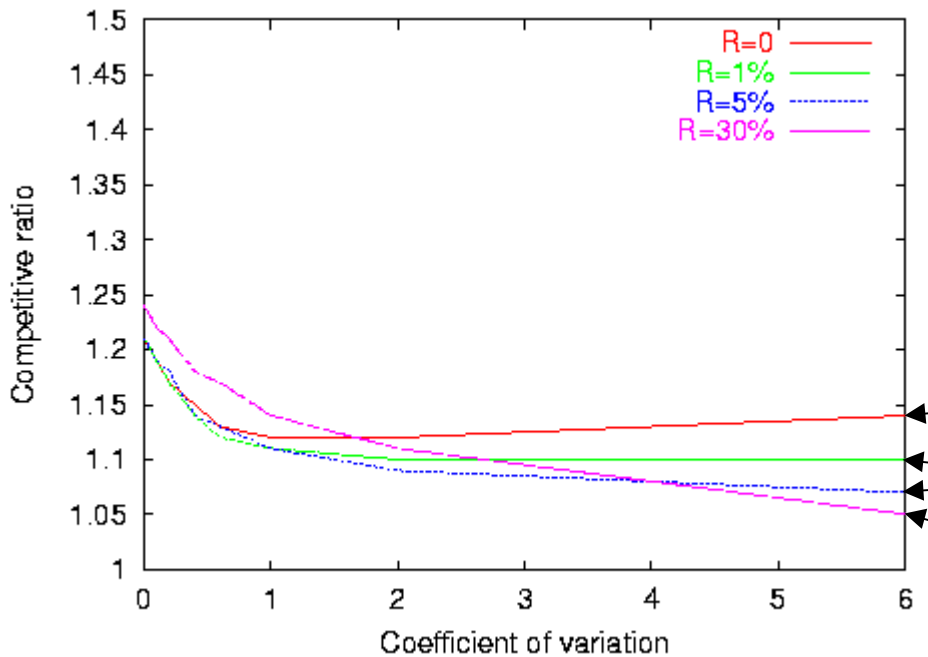


- Under moderate load, DBR reduces blocking probability by two orders of magnitude

- Under heavy load, SPF outperforms CSPF

Lucent Technologies
Bell Labs Innovations

Trunk Reservation for CSPF



Trunk reservation test:

If the available bandwidth on a link is less than *R%* of the link capacity, then reject connection request

Too aggressive

Middle ground

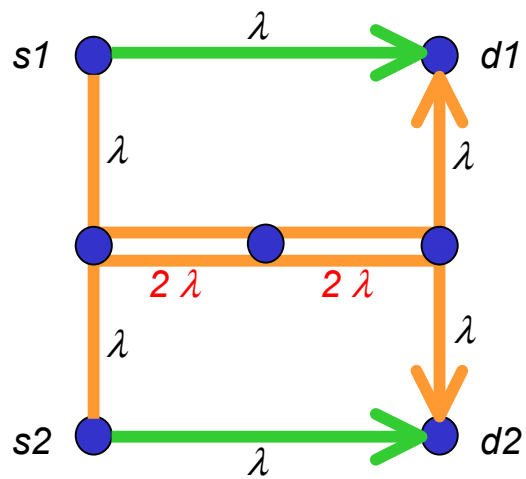
Too defensive

- Trunk reservation of around 1-5% is effective to reduce greed in CSPF

Lucent Technologies
Bell Labs Innovations

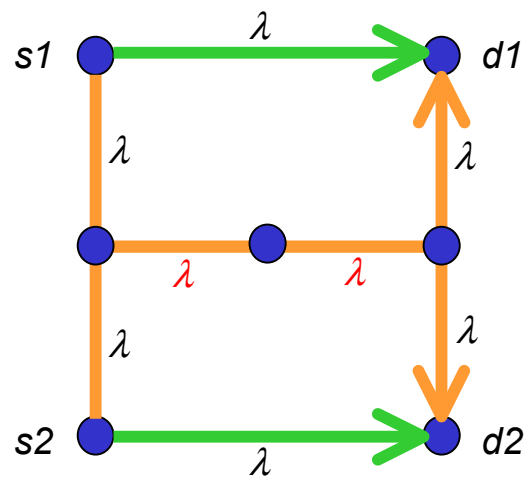
Path Protection

1+1 or 1:1 dedicated protection:



Each protection path is dedicated with its own protection bandwidth

1:1 shared protection:



W_{ij} = working bandwidth that uses links *i* and *j*
Q_{ij} = protection bandwidth to link *i* when link *j* fails

Protection bandwidth needed on link *i* is:
min {0, max$_j$ {Q_{ij}- W_{ij}}}

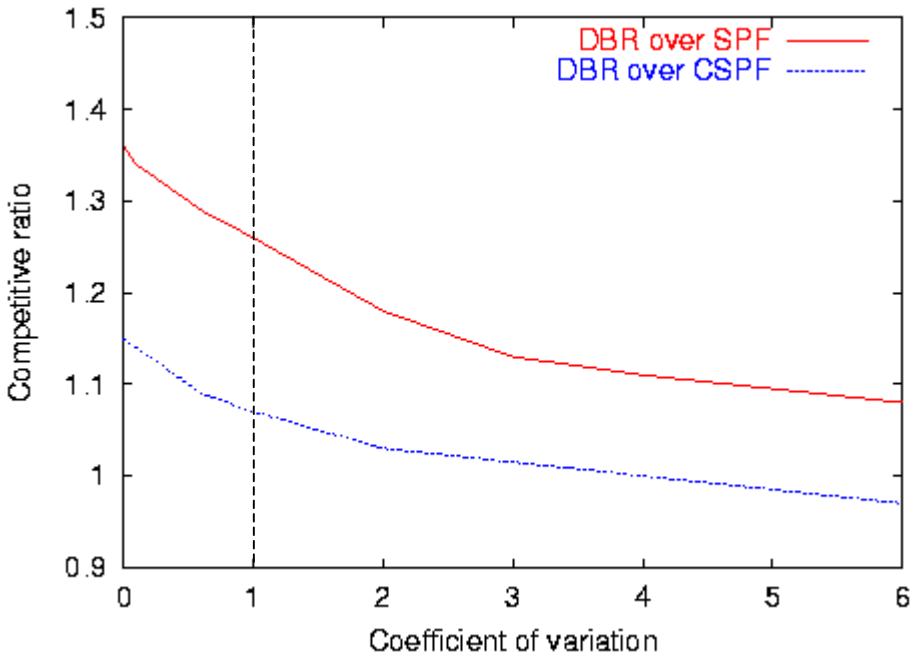
Working path

Protection path

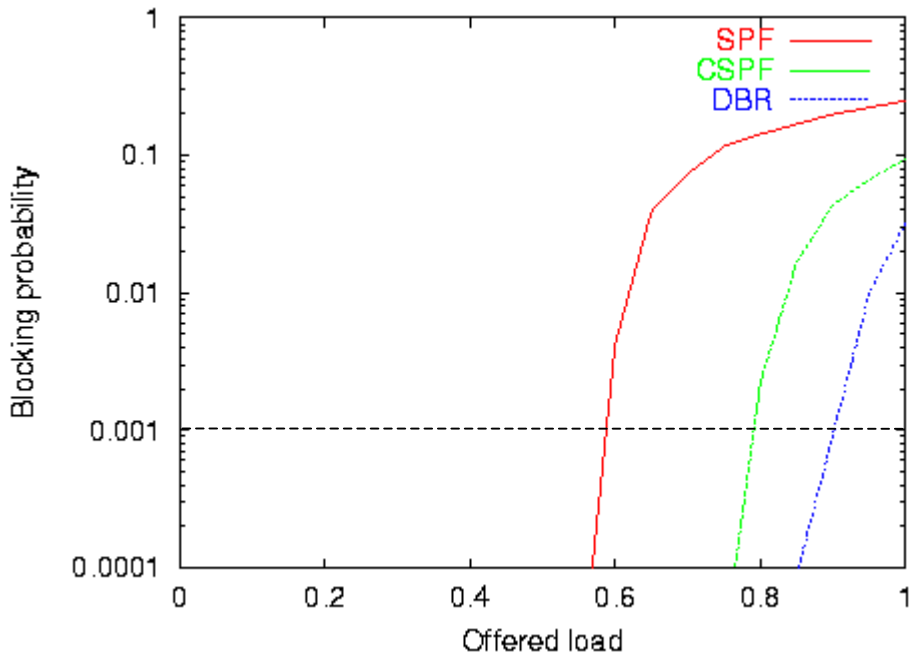
Lucent Technologies
Bell Labs Innovations

Dedicated Protection

Static case:



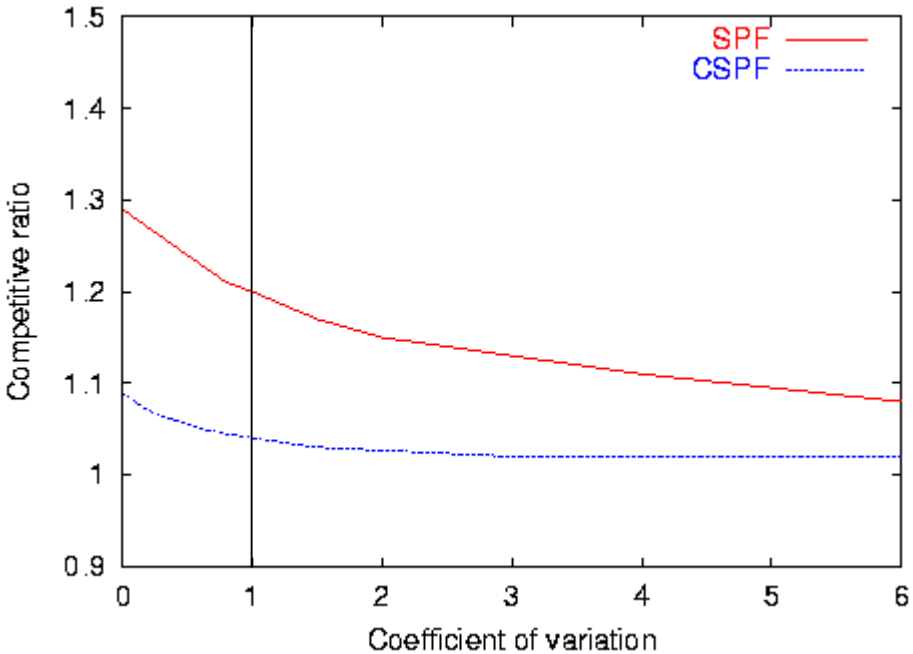
Dynamic case:



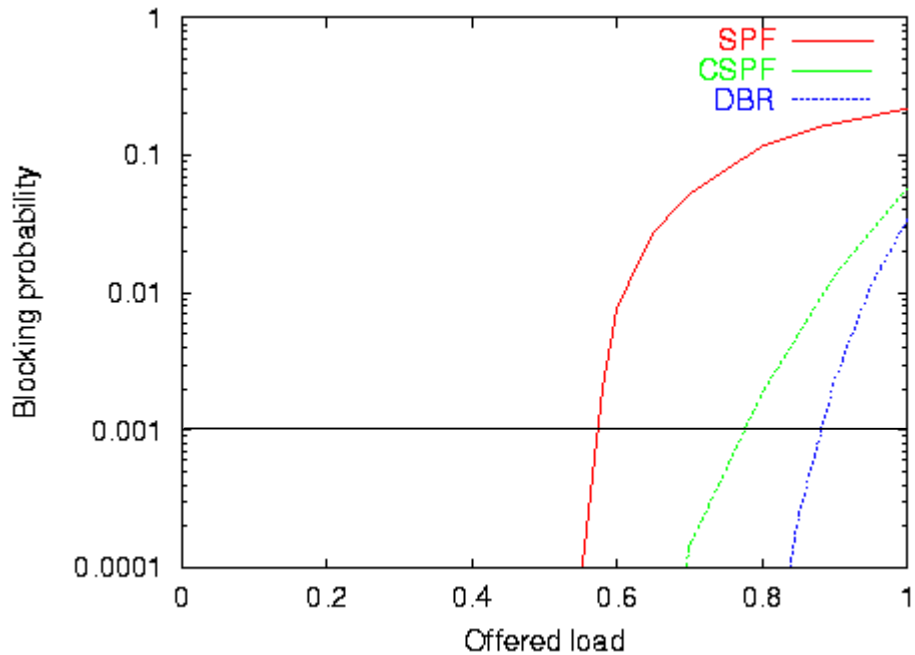
- Disjoint-path constraints make CSPF much less greedy

Shared Protection

Static case:



Dynamic case:

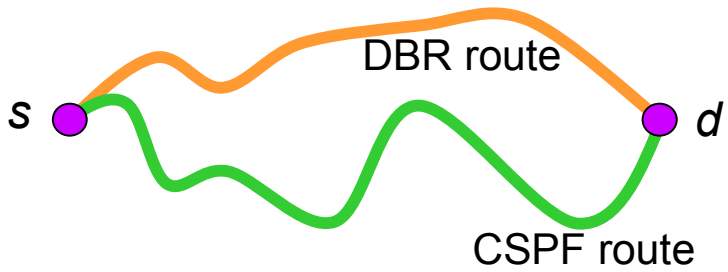


- Network resource is ~27% less than that in dedicated protection

Lucent Technologies
Bell Labs Innovations

Variation: Adaptive DBR

Hybrid DBR and CSPF assisted with online measurement



Which route to choose?

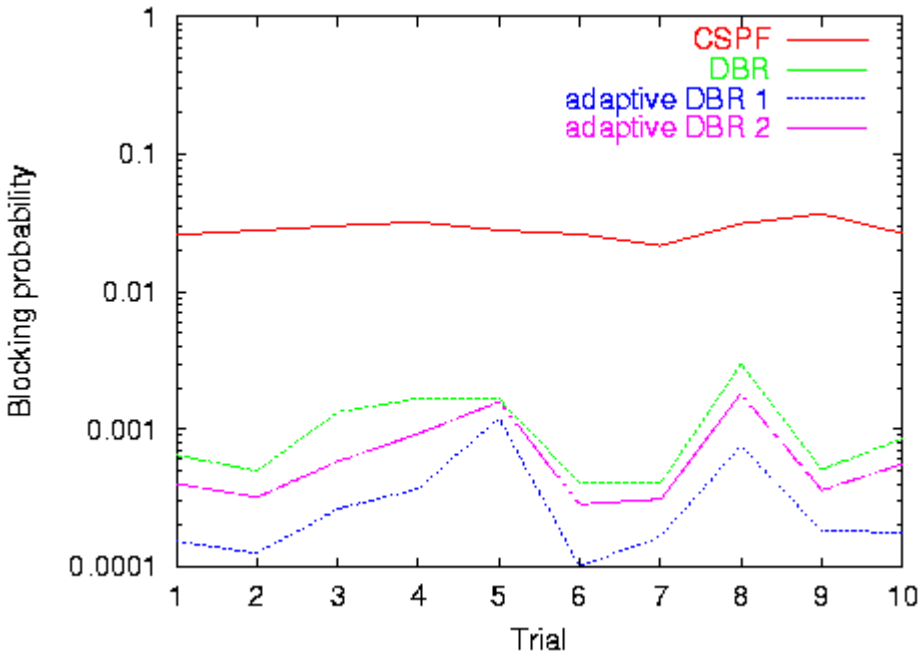
Let v_{sd} = the designed load between *s* and *d* that is used in the computation of DBR routes
Let w_{sd} = the current measured load between *s* and *d*
Path Selection for adaptive DBR is as follows:

```
if  v_sd  - w_sd > 0
        setup the DBR connection if possible
else
        check each link on DBR route and apply TR test
        if pass
                setup the DBR connection if possible
        else
                setup the CSPF connection if possible
```

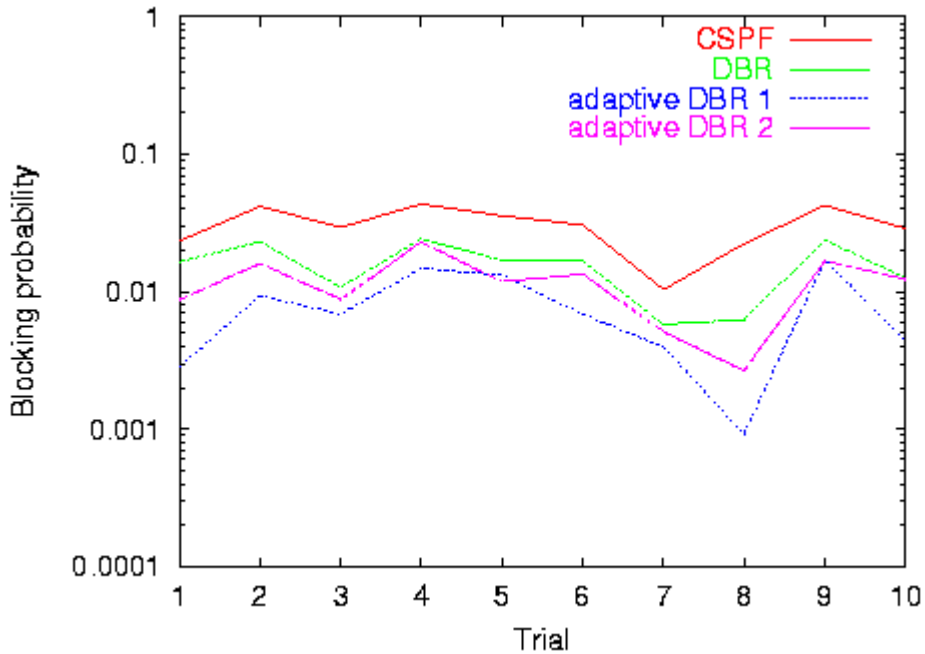
Lucent Technologies
Bell Labs Innovations

Dynamic Case (No Protection)

Moderately abnormal:



Highly abnormal:



- For each trial, the original traffic demand matrix is independently perturbed

Lucent Technologies
Bell Labs Innovations

Resource Requirement

Scheme:	Resource usage (normalized):
DBR with no protection	1.00
CSPF with no protection	1.21
SPF with no protection	1.28
DBR with shared protection	2.03
CSPF with shared protection	2.21
SPF with shared protection	2.62
DBR with dedicated protection	2.77
CSPF with dedicated protection	3.19
SPF with dedicated protection	3.77



Conclusion

- Traffic demand information is a valuable asset for online provisioning
- Even if the information is inaccurate, performance advantage can still be gained
- When used (in our example)
  - CSPF increases utilization over SPF for static loads by ~10%
  - DBR increases utilization over CSPF for static loads by ~20%
  - DBR reduces blocking rate by *orders of magnitude* for dynamic loads
  - For a given level of blocking, DBR increases the carried load by ~40%