Randomized Load Balancing and Oblivious Routing

Peter J. Winzer
Bell Labs, Alcatel-Lucent

Joint work with *F. B. Shepherd, M. K. Thottan, S. Borst, R. Prasad*

DIMACS Tutorial on Algorithms for Next Generation Networks
Rutgers University
August 2007

Other names for the same thing:

- Valiant Load Balancing (VLB)
- Two-phase routing

Full details in:

- F. B. Shepherd and P. J. Winzer, "Selective randomized load balancing and mesh networks with changing demands," J. Opt. Netw. 5, 320-339 (2006)
- R. S. Prasad, P. J. Winzer, S. Borst and M. K. Thottan, "Queuing Delays in Randomized Load Balanced Networks", IEEE INFOCOM (2007)

Other groups looking into this:

- Rui Zhang-Shen, Nick McKeown (Stanford)
- M. Kodialam, T. V. Laskshman (Bell Labs)

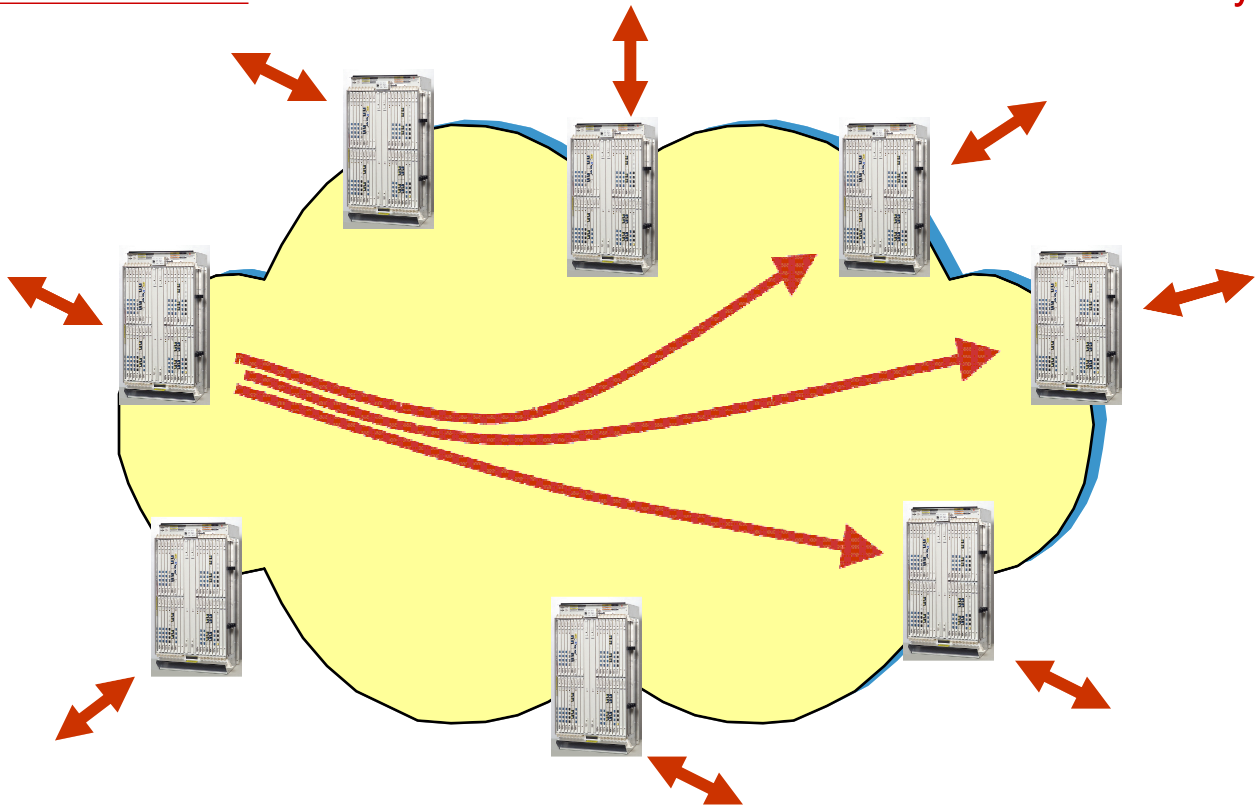
Outline

- Dynamic data traffic and how to cope with it
- Network architectures for dynamic data traffic
  - Circuit-switched networks
  - Packet-switched networks
- Over-provisioning is the price for robustness
- Randomized Load Balancing (RLB):
  A robust network architecture
- How random is ‘random’: Queuing in RLB

1 Dynamic data traffic and how to cope with it

Dynamic data services: Two examples

- Virtual private networks (VPNs)
  - Customer specifies access data rates at multiple business locations (but leaves open the traffic distribution among its sites)
  - Up to the carrier to handle variable traffic demands most efficiently



Alcatel·Lucent

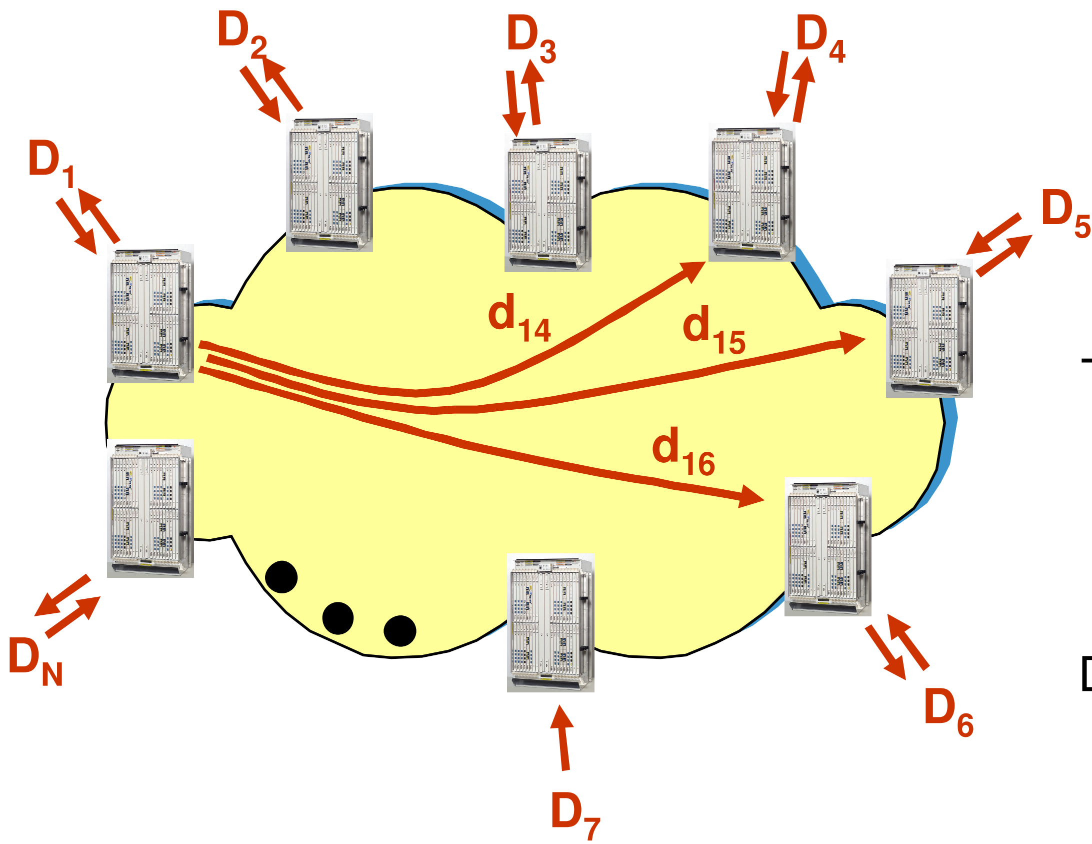
Dynamic data services: Two examples

- Virtual private networks (VPNs)
  - Customer specifies access data rates at multiple business locations (but leaves open the traffic distribution among its sites)
  - Up to the carrier to handle variable traffic demands most efficiently

- Remote storage and computing
  - Customer leases storage space / processor power with service provider (but does not specify times and duration of access)
  - Up to the carrier to handle extended bursts of backup/restore data traffic

How should carriers design their networks to maximize revenue ?

The task – Robust network design



Traffic from node 1 to node 2

$$D = \begin{pmatrix} 0 & d_{12} & d_{13} & d_{14} & d_{15} & \dots & d_{1N} \\ d_{21} & 0 & d_{23} & d_{24} & d_{25} & \dots & d_{2N} \\ & & & \vdots & & & \\ d_{N1} & d_{N2} & d_{N3} & d_{N4} & d_{N5} & \dots & 0 \end{pmatrix}$$

Σ = Traffic originating from node N (=D_N)

- Network of *N* nodes
- Demand distribution specified by ***demand matrix D***
- **A *robust network* has to accommodate all *legal demand matrices***

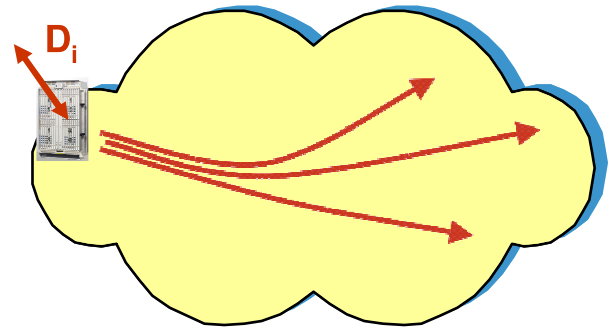
Alcatel·Lucent

What are "legal demand matrices" ?

- Difficult question
  - Depends not only on the present network traffic, …
  - … but also on the traffic likely to be generated by future services

Examples:

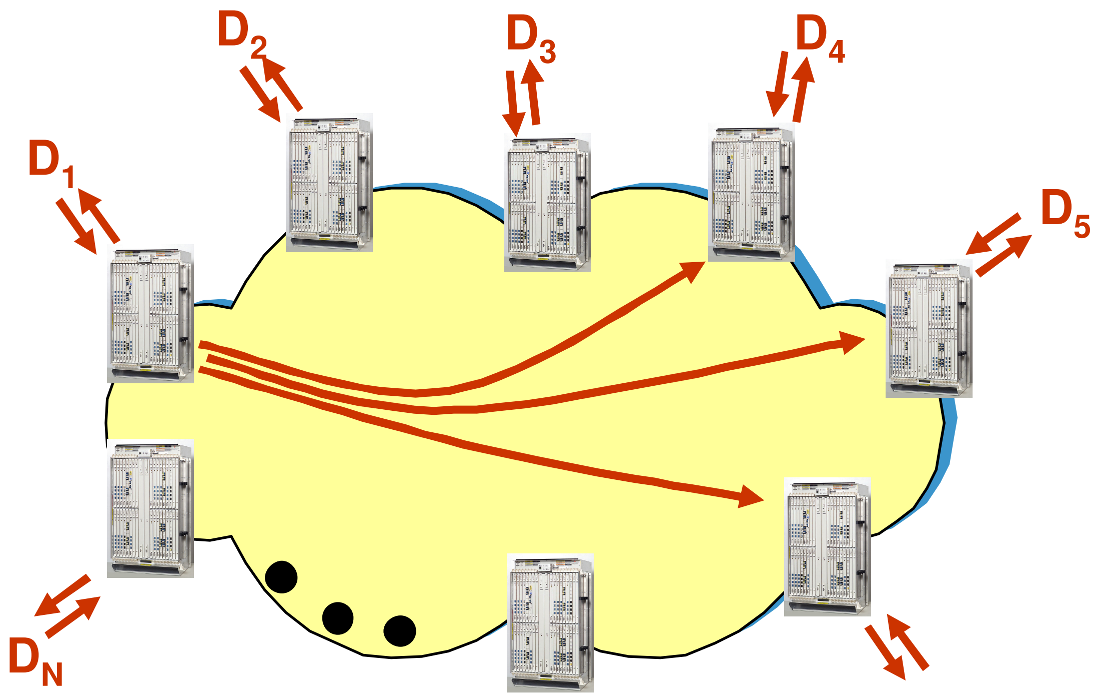
- Demand matrices in the vicinity of some fixed demand matrix
  - Start from some fixed set of projected demands (d_{ij})
  - Allow each demand to vary by some percentage (projected growth)
- Hose matrices (good model for VPNs et al.*)
  - Fixed ingress/egress traffic (D_i) cannot be exceeded (*'hose constraint'*)
  - Individual demands (d_{ij}) may vary, e.g.,
    - from 0 to D_i : complete demand changes
    - from 0 to αD_i : restricted demand changes
    - from αD_i to D_i : static plus changing traffic

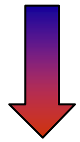


* N. G. Duffield et al., IEEE/ACM Trans. on Networking 10(5), 679-692 (2002).

Alcatel·Lucent

How to deal with dynamic traffic





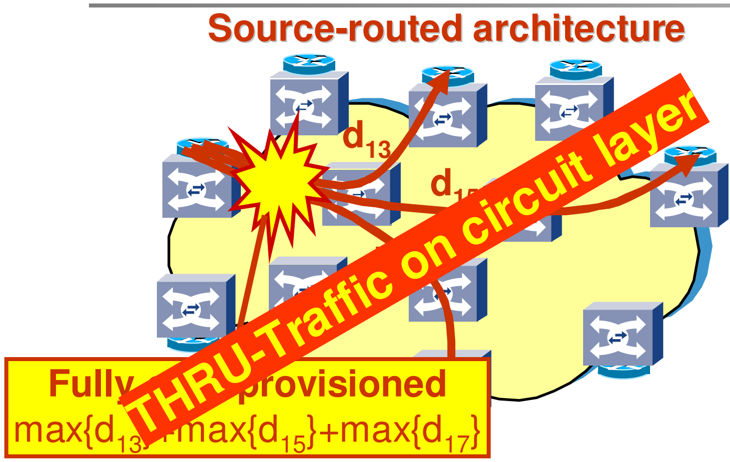
Dynamics	Timescales	Typical solution
Static	Days – months	Management plane
Moderate	Minutes – hours	Fast control plane, ASON
High	Seconds – minutes	MPLS
Packets	Packets – flows	IP network

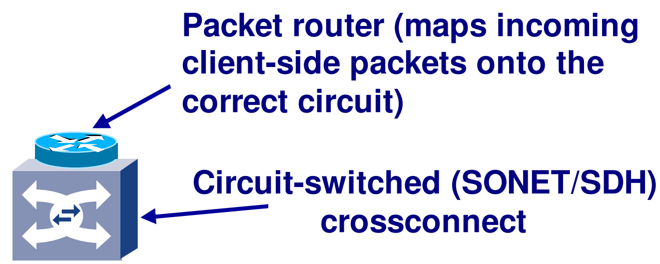
ASON … Automatically switched optical network MPLS … Multi-protocol label switching

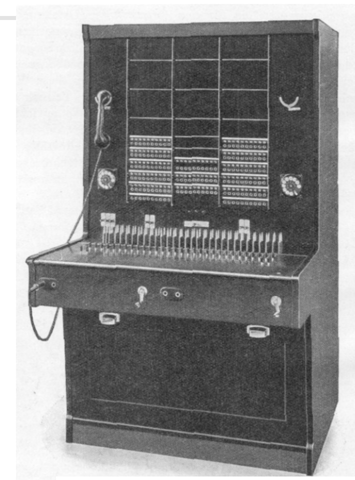
Alcatel·Lucent

2 Network architectures for dynamic data traffic

Traditional approaches – Circuit switching






http://www.s-storbeck.de

- "Source-routed" architecture (routing decisions take place at the *ingress*)
- Single-hop routing (no routing decisions as the packet traverses the network)
- Circuit-switched network core

☺ Network availability, fast protection & restoration

☺ QoS guarantees

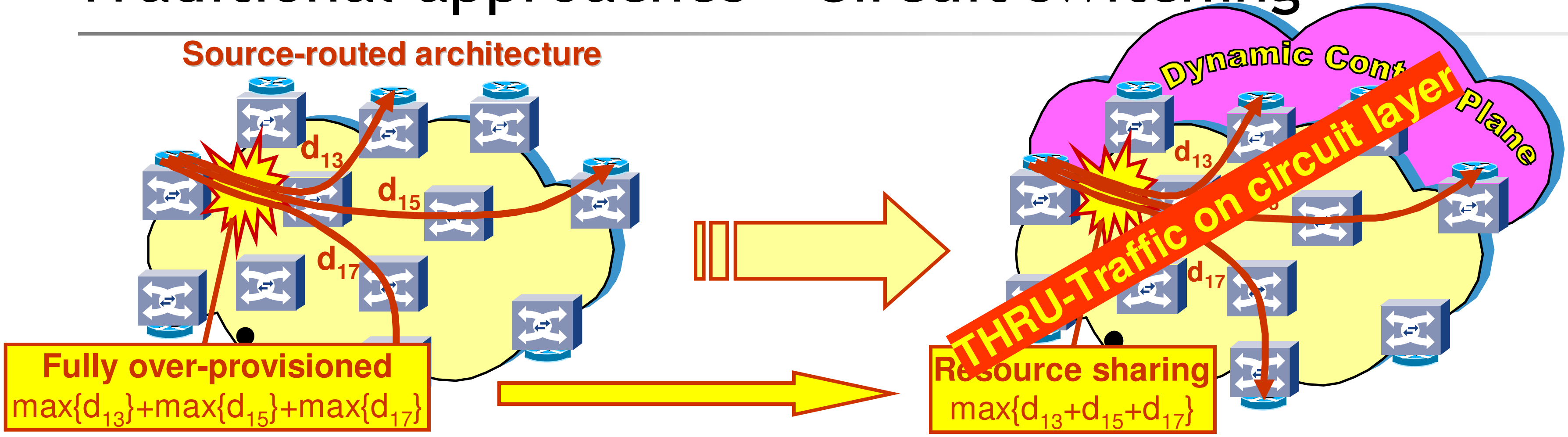
☹ Static circuits do not offer resource sharing

⇨ Vast over-provisioning

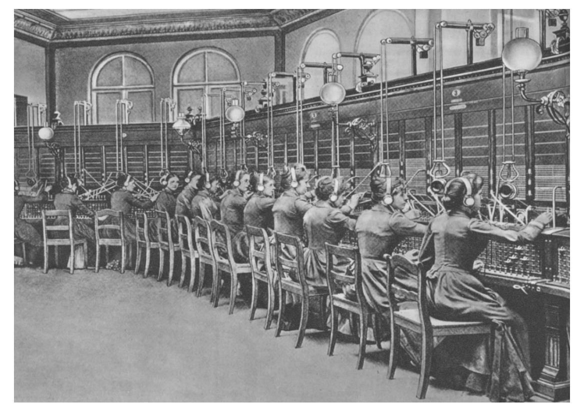
d_{ij} … Demand from node *i* to node *j*

Alcatel·Lucent

Traditional approaches – Circuit switching



- "Source-routed" architecture (routing decisions take place at the *ingress*)
- Single-hop routing (no routing decisions as the packet traverses the network)
- Circuit-switched network core



http://www.s-storbeck.de

☺ Network availability, fast protection & restoration

☺ QoS guarantees

☹ Static circuits do not offer resource sharing

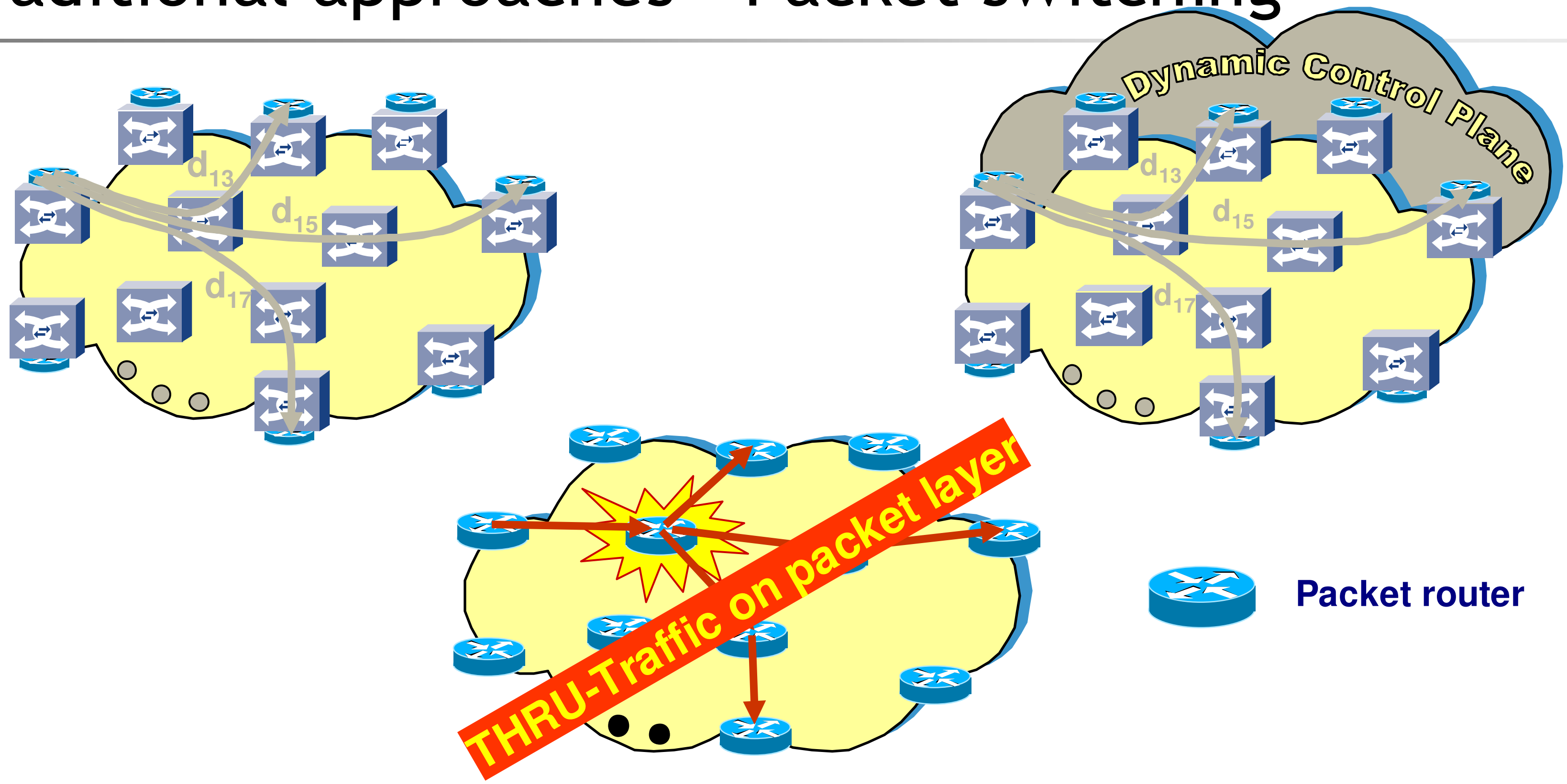
⇨ Vast over-provisioning

Possible solution: *Dynamic control plane*

- "Dynamic" = "Fast enough to follow the changes in traffic patterns"
- Required control plane speed depends on the dynamics of the offered data services !

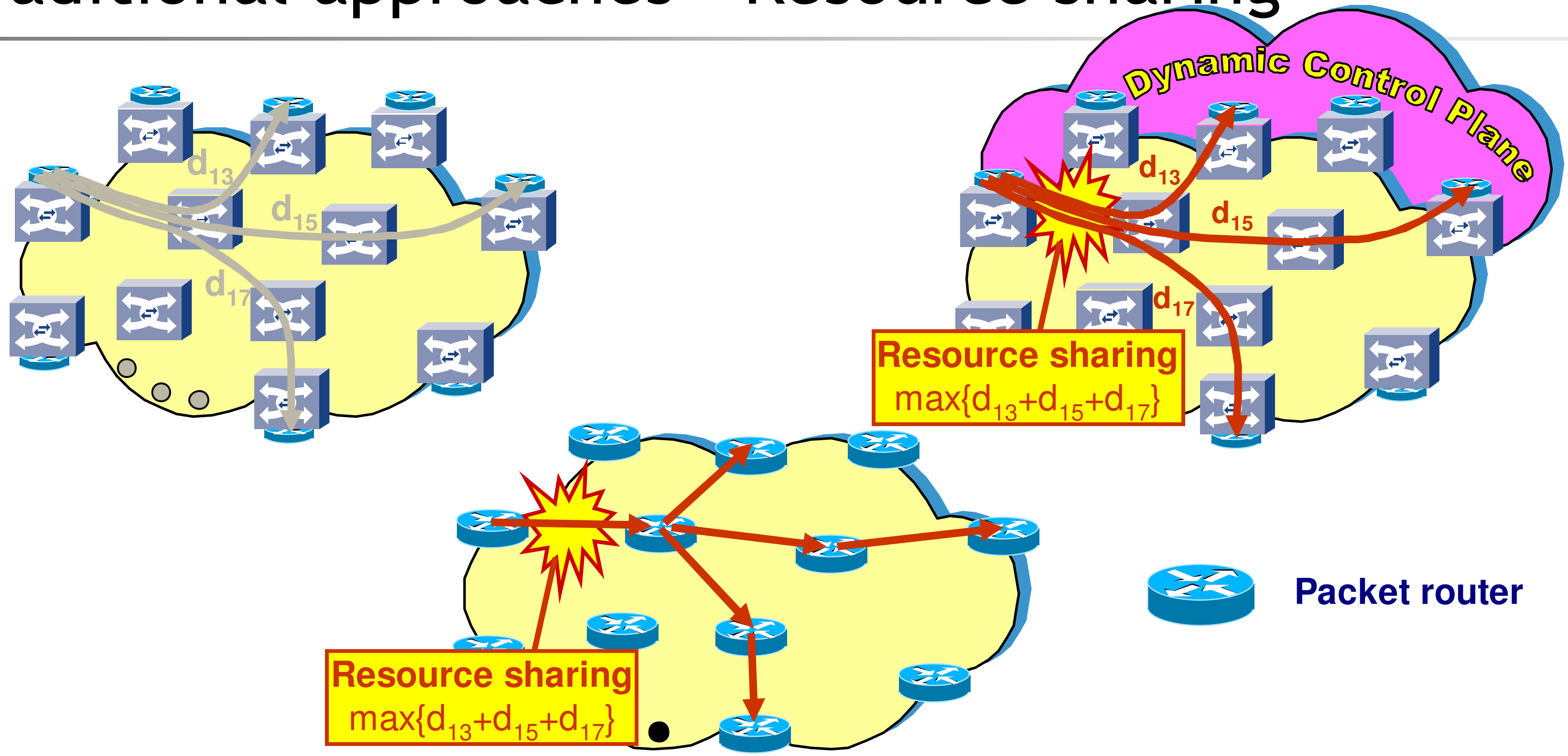
Alcatel·Lucent

Traditional approaches – Packet switching



- Packets get looked up multiple times from source to destination (*multi-hop routing*)
  - ⇨ Problem: ***Thru-traffic*** uses up router capacity
    - Wastes expensive router ports (Router port cost : Crossconnect port cost = 3:1)
    - Leads to scalability problems in large networks
    - Quality of service problems due to multiple buffering (delay and delay jitter !)

Traditional approaches – Resource sharing



- Statistical multiplexing = "Packet-scale re-provisioning"
  (Statistical multiplexing within routers takes the role of distributed dynamic control plane)
  ⇨ *Same amount of resource sharing* for
  - Packet-switched networks
  - Circuit-switched networks with dynamic control-plane

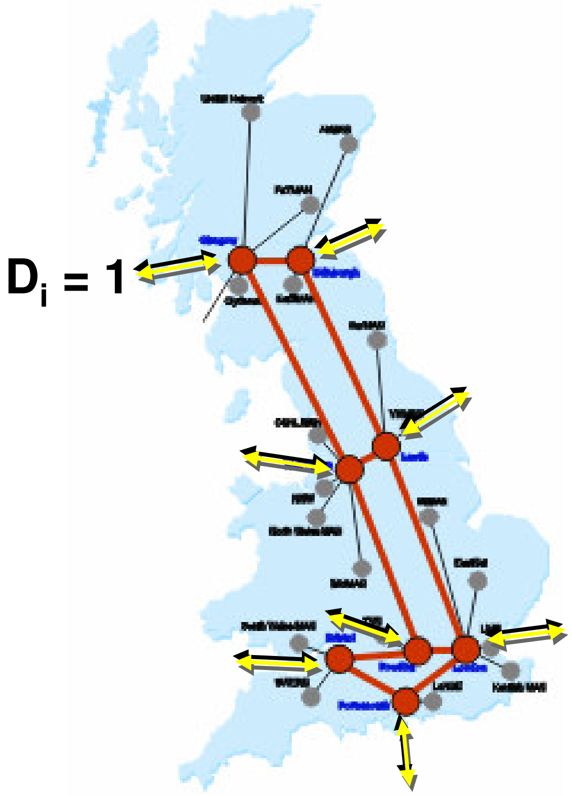
  ⇨ In general, both network types need some *over-provisioning*
  (because $\max\{d_{13}+d_{15}+d_{17}\}$ may be different for different traffic patterns!)

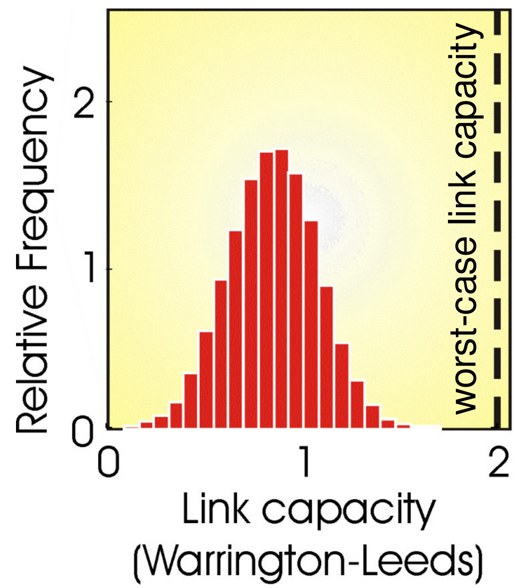
Alcatel·Lucent

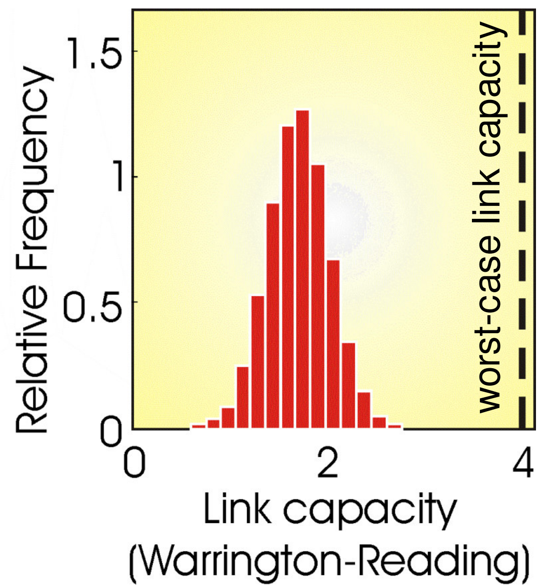
3 Over-provisioning is the price for robustness

Over-provisioning and resource sharing

JANET
(UK research backbone)







Generated 100,000 random demand matrices:

- $\Sigma_i d_{ij} = \Sigma_j d_{ij} = 1$

  (ingress = egress traffic = 1)

- each d_{ij} may vary from 0 to 1

  (full traffic randomness)

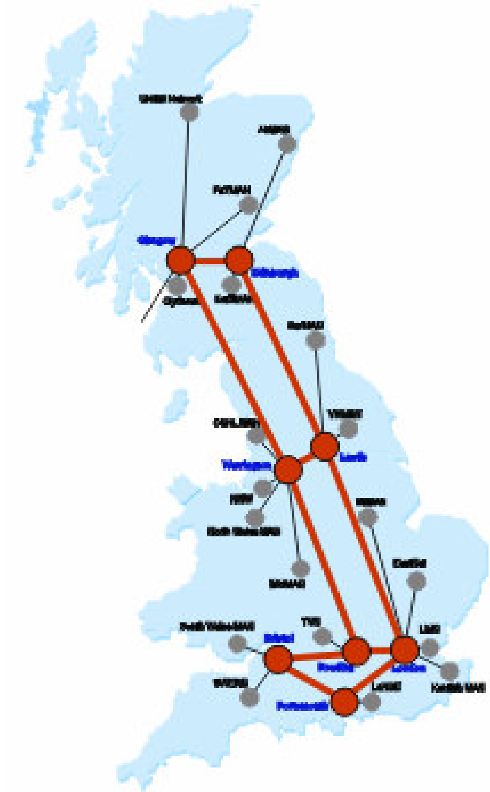
$$D = \begin{pmatrix} 0 & d_{12} & d_{13} & d_{14} & d_{15} & \dots & d_{1N} \\ d_{21} & 0 & d_{23} & d_{24} & d_{25} & \dots & d_{2N} \\ & & & \vdots & & & \\ d_{N1} & d_{N2} & d_{N3} & d_{N4} & d_{N5} & \dots & 0 \end{pmatrix}$$

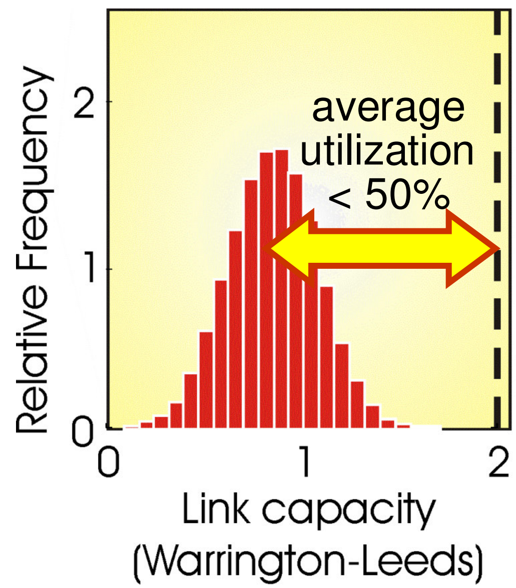
"**Hose model**" for VPN services – Ingress/egress traffic known, but traffic distribution unknown

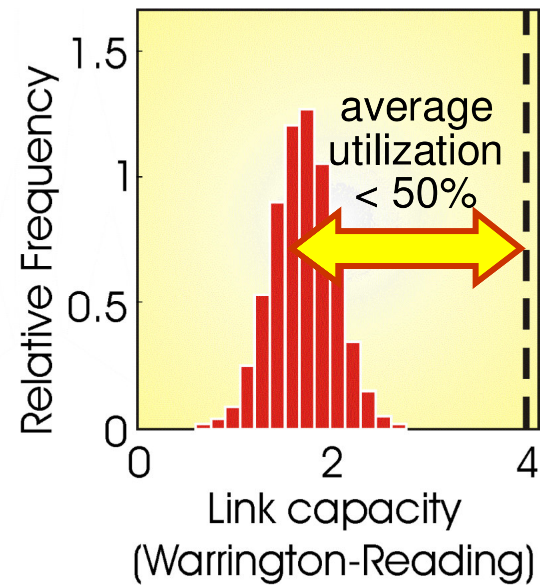
[N. G. Duffield et al., IEEE/ACM Trans. on Networking 10(5), 679-692 (2002).]

Over-provisioning and resource sharing

JANET
(UK research backbone)





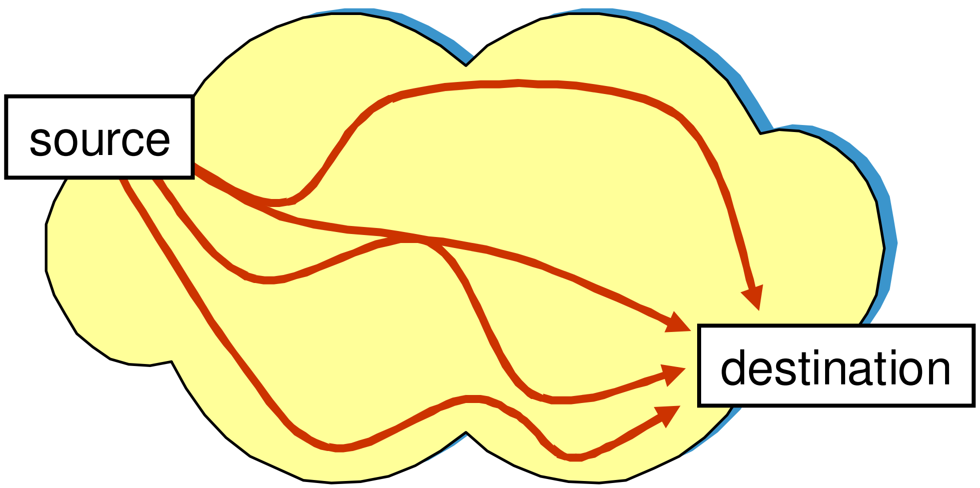


Bottomline: The price for flexibility is over-provisioning (under-utilization)

Alcatel·Lucent

Routing strategies

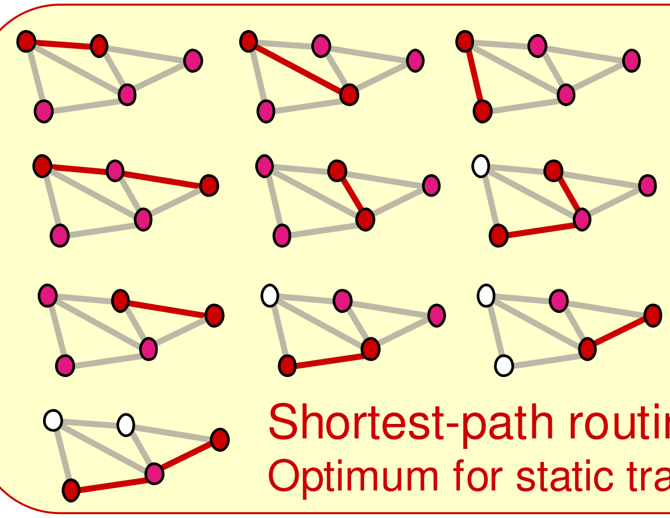
- Oblivious routing
  - Traffic routes do **<u>not</u>** depend on the network state or traffic distribution
  - Design routes ahead of time (*"routing template"*)
- Single-path routing
  - All source-destination traffic follows the same path
- Multi-path routing
  - Traffic may be split and take several parallel routes (e.g., LCAS in SONET)
  - Problem of <u>re-sequencing</u> due to different propagation delays



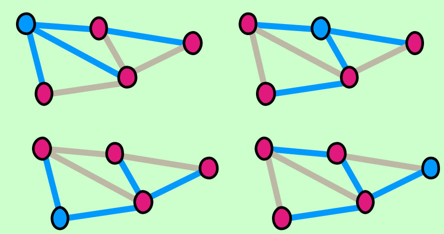
LCAS … Link capacity adjustment scheme

Routing strategies

- Oblivious routing
  - Traffic routes do **<u>not</u>** depend on the network state or traffic distribution
  - Design routes ahead of time (*"routing template"*)
- Single-path routing
  - All source-destination traffic follows the same path
- Multi-path routing
  - Traffic may be split and take several parallel routes (e.g., LCAS in SONET)
  - Problem of <u>re-sequencing</u> due to different propagation delays
- Examples: Shortest-path routing, Tree routing (VPN-Tree)



Shortest-path routing
Optimum for static traffic



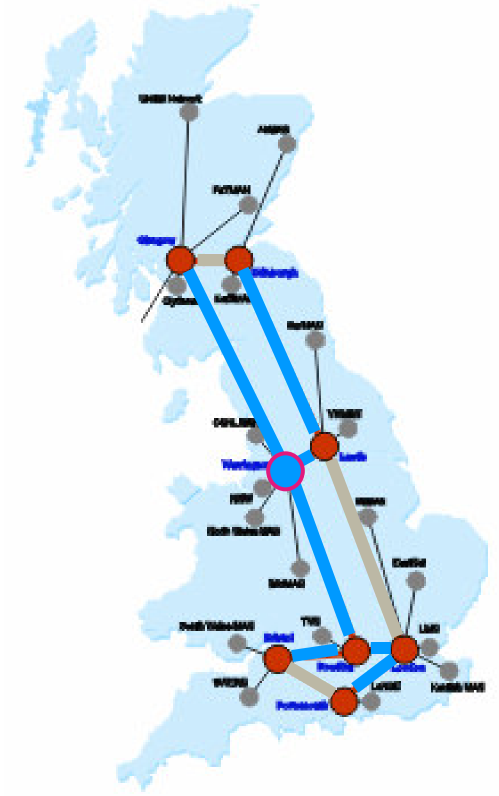
Five examples for trees

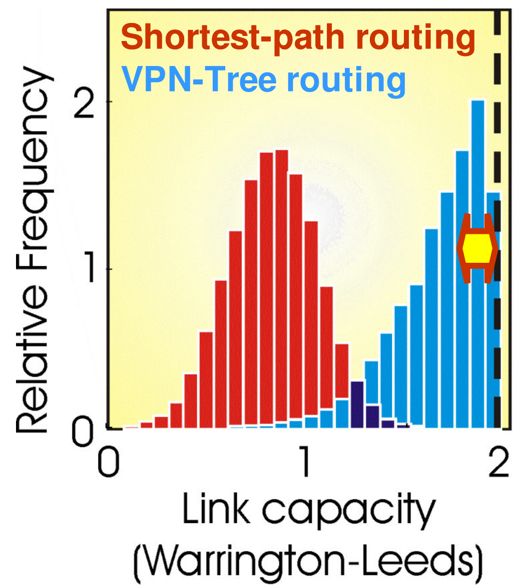
- All traffic is routed along a *<u>single</u>* tree
- Optimum routing of hose traffic by choosing minimum-cost VPN-Tree

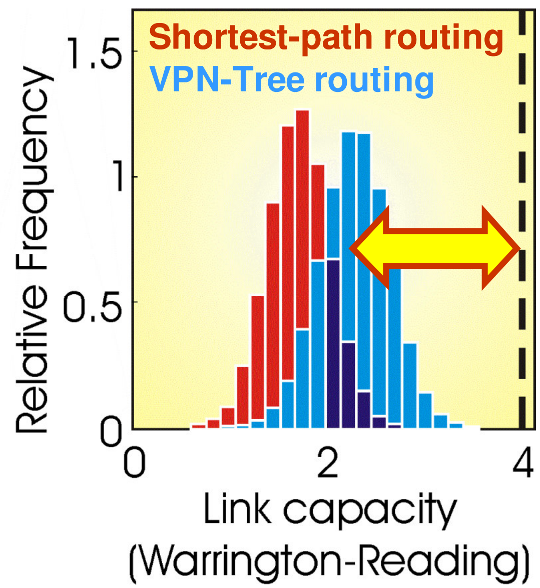
Alcatel·Lucent

VPN-Tree makes better use of resources

JANET
(UK research backbone)







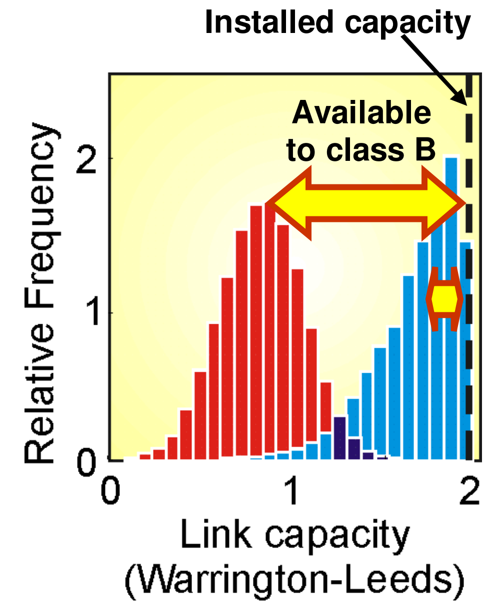
VPN-Tree routing:

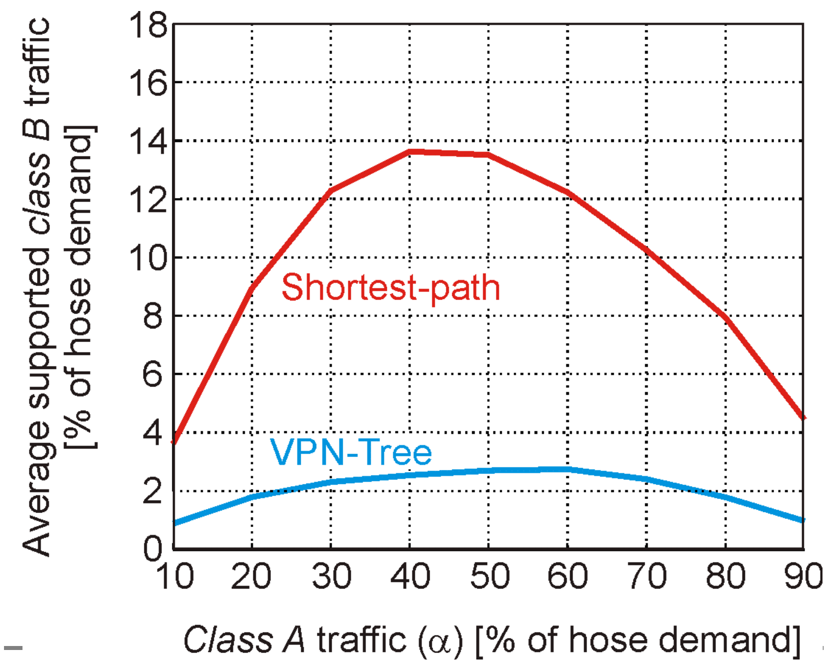
- Find the *cheapest of all possible spanning trees*, and route only across it
- Optimum routing strategy for *hose traffic*
  [A. Gupta et al., ACM STOC'01, (2001).]

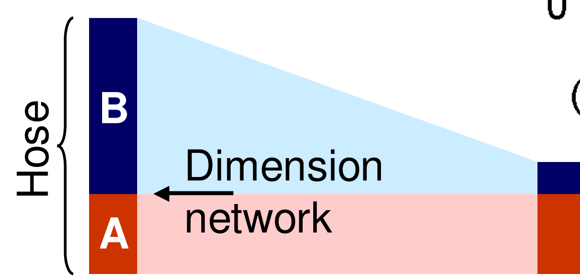
⇨ **VPN Tree increases utilization and lowers cost**

Resource sharing and traffic classes

- The better network resources are utilized by "*class A*" traffic, the less "room" there to statistically multiplex in best-effort "*class B*" traffic (for IP/MPLS networks)
- Expressed differently: The lower network resources are utilized by *class A* traffic, the more resources are available to statistically multiplex in *class B* traffic
  ⇨ ***Here, under-utilization is a good thing !***







- Network dimensioned to fully support αD of *class A* hose traffic
- What fraction β of the hose traffic traffic can ride as *class B* on top of *class A*, on average?
- Goodput = $\alpha D + \beta D$

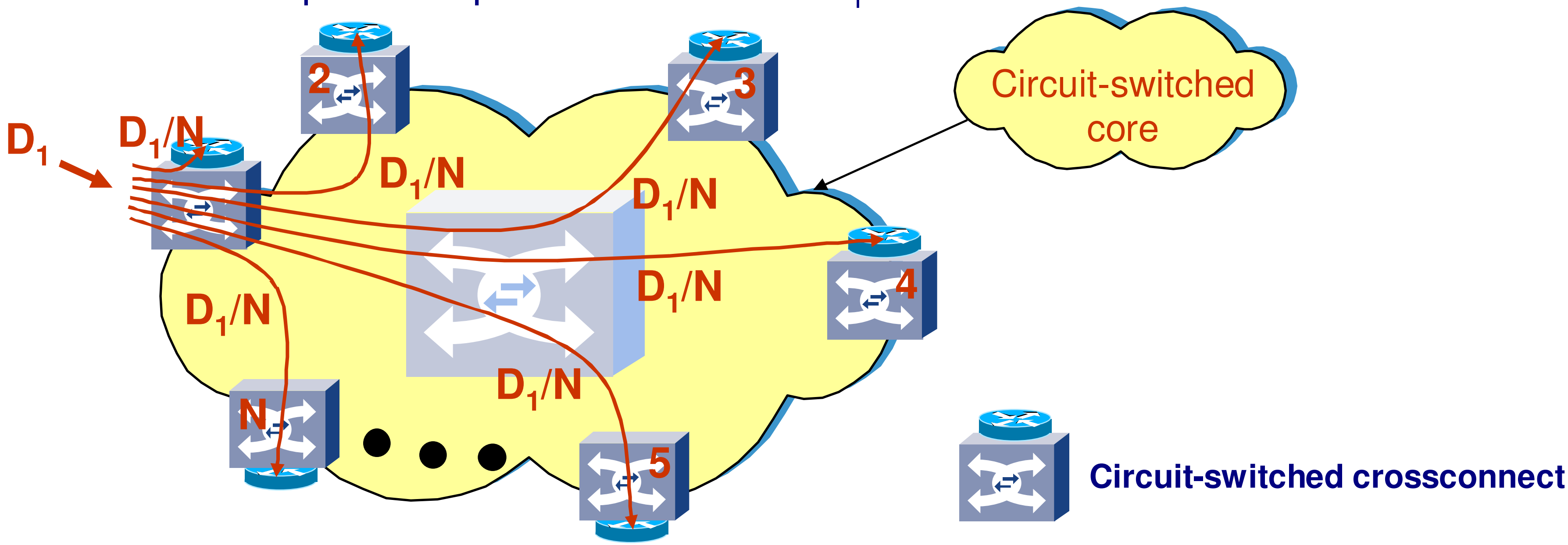
Alcatel·Lucent

4 Randomized Load Balancing: A robust architecture

Randomized Load Balancing

[L. G. Valiant, SIAM J. Comput. 11, 350 (1982).]

Simple example: Demand from D_1



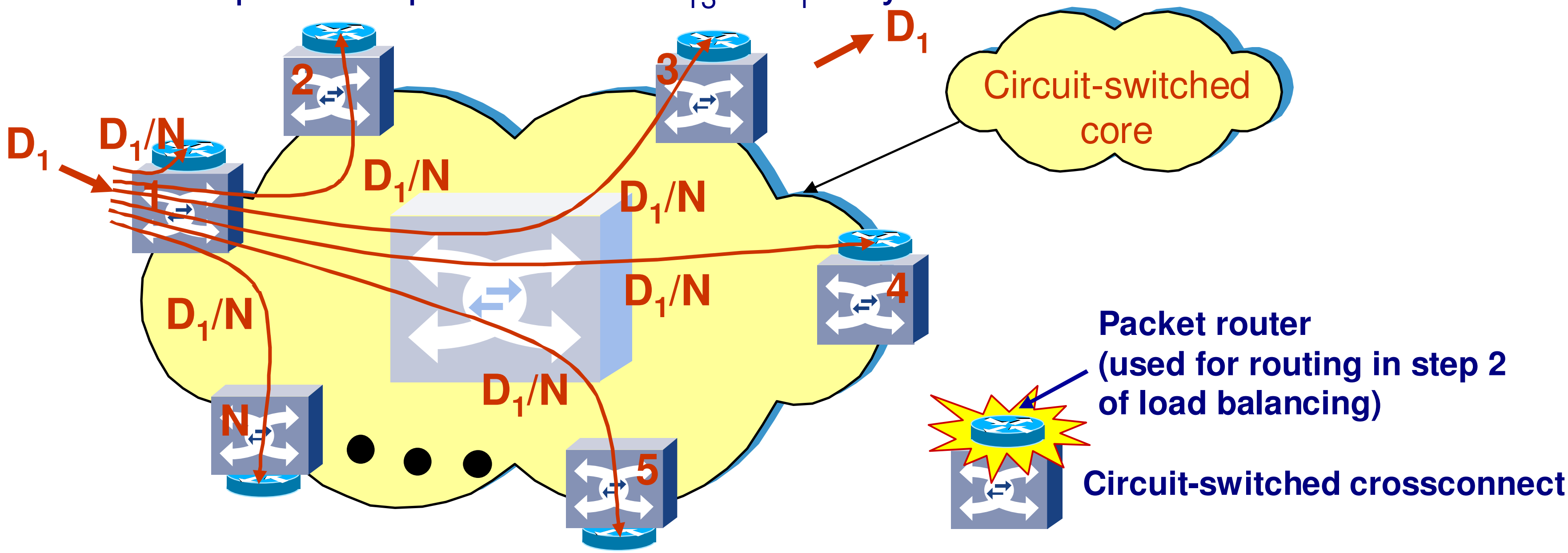
Step 1: Uniform traffic distribution

- Send D_k/N-th of ingress traffic to all other nodes
  - Distribution on a purely random basis (no packet routing in step 1 !)
  - Eliminates burstiness in demand distribution ⇨ strictly uniform traffic
  - Dimension **network for uniform traffic**, but the result is good for all traffic patterns

Randomized Load Balancing

[L. G. Valiant, SIAM J. Comput. 11, 350 (1982).]

Simple example: Demand $d_{13} = D_1$ only

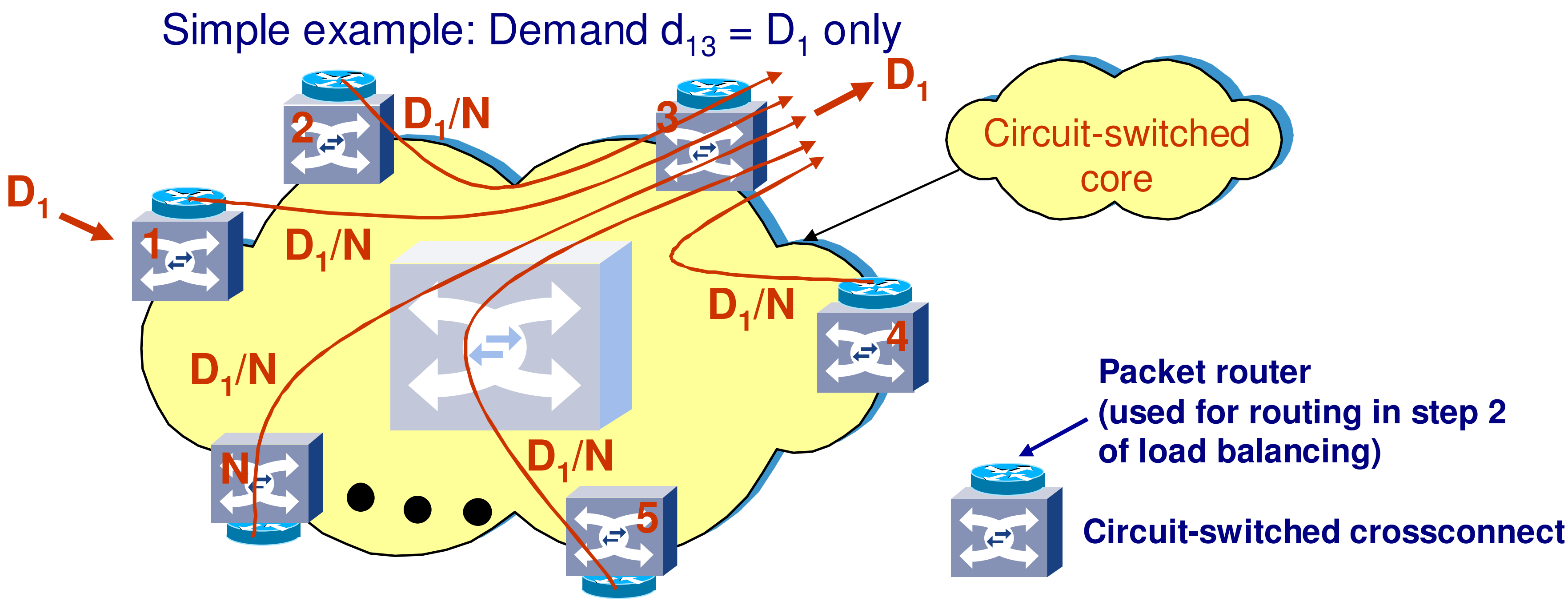


Step 2: Route traffic locally

- Strictly local routing; does not require dynamic topology maps, etc.
- Each packet router needs to process a total of $N \times D / N = D$ only (same as source-routed architecture)

Randomized Load Balancing

[L. G. Valiant, SIAM J. Comput. 11, 350 (1982).]



Step 3: Transport to final destination

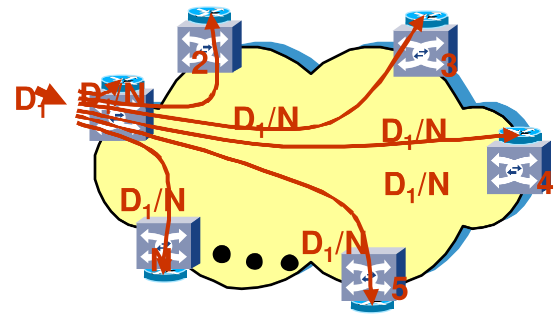
- Like in Step 1 (uniform distribution), only **static** circuits are needed
- **Double-hop routing** (like single-hop: look up header *only once*)

⇨ **No thru-traffic is unnecessarily using expensive IP router ports**

Alcatel·Lucent

Security and coding for resilience

- Additional physical-layer security feature of RLB:
  No node ever sees the full information

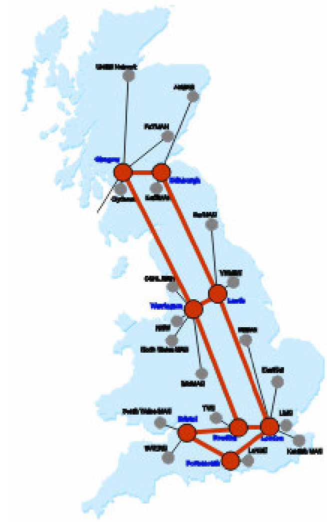


- Resilience by erasure coding:
  - Send $N + k$ packets using, e.g., Reed-Solomon code
  - If k packets are lost, the full information can still be restored
  - Similar to FEC in transport systems

data sub-packets			overhead packet
0	1	0	1
1	1	1	1
1	0	1	0
0	1	1	0
0	0	0	0
1	1	0	0
0	0	0	0

Transport bandwidth requirements

Architecture	Routing	Transport capacity x km
Packet-switching	SP	**3,437**
	VPN	**2,302**
Load bal.	SP	**2,776**



<u>Traffic assumptions:</u>
Hose traffic with $D_i = 1$
Demands allowed to vary between 0 and 1

- Load balancing and packet switching need about the same <u>transport bandwidth</u> (over-provisioning for flexibility [packet] vs. two times uniform & static [load balanced])

⇨ Quantification of over-provisioning: "**<u>Robustness Premium</u>**"

SP … Shortest path routing,   VPN … VPN-Tree routing

The Robustness Premium

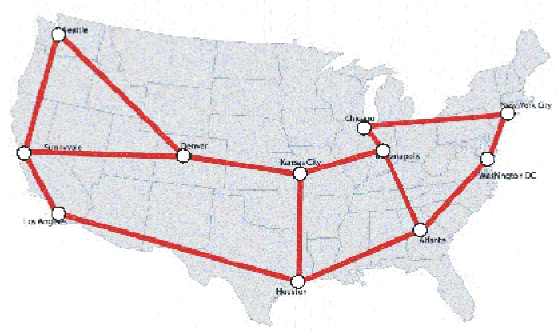
$$\text{Robustness premium} = \frac{\text{“Cost” of supporting all possible demand matrices}}{\text{“Cost” of routing a reference demand matrix}}$$

Architecture & Routing	JANET	ABILENE	GEANT
Static circuit-switched (Shortest-path routing)	8	11	27
Dynamic circuit-switched *or* packet-switched (Shortest-path routing)	2.48	2.46	2.46
Dynamic circuit-switched *or* packet-switched (VPN-Tree routing)	1.66	1.50	1.31
Randomized load balancing (RLB)	2.00	2.00	2.00

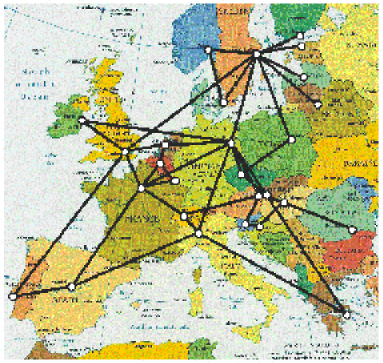
Each step in Randomized Load Balancing requires a uniform full mesh

Assumptions:
“Cost” = Transport capacity
Hose traffic with $D_i = 1$
Demands allowed to vary between 0 and 1
Reference: shortest-path routing of uniform demand matrix

(a) JANET topology



(b) ABILENE topology



(c) GEANT topology

Alcatel·Lucent

The Robustness Premium

$$\text{Robustness premium} = \frac{\text{"Cost" of supporting all possible demand matrices}}{\text{"Cost" of routing a reference demand matrix}}$$

Architecture & Routing	JANET	ABILENE	GEANT
Static circuit-switched (Shortest-path routing)	8	11	27
Dynamic circuit-switched *or* packet-switched (Shortest-path routing)	2.48	2.46	2.46
Dynamic circuit-switched *or* packet-switched (VPN-Tree routing)	1.66	1.50	1.31
Randomized load balancing (RLB)	2.00	2.00	2.00

Assumptions:
"Cost" = Transport capacity
Hose traffic with $D_i = 1$
Demands allowed to vary between 0 and 1
Reference: shortest-path routing of uniform demand matrix

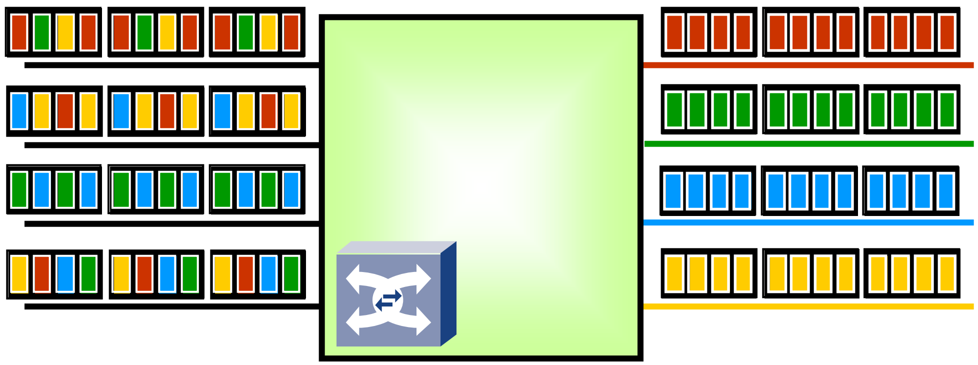
So, does this mean RLB is out?

No! Look at equipment cost!

Alcatel·Lucent

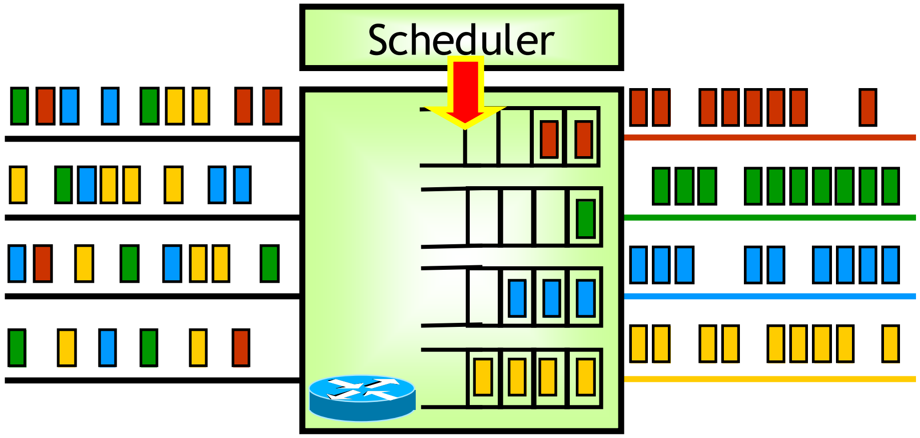
Basic network elements

Circuit-switched crossconnect



- Packets are placed onto the correct output ports based on their position within a frame
- Connections hold for many frames
- No buffering required

Packet router

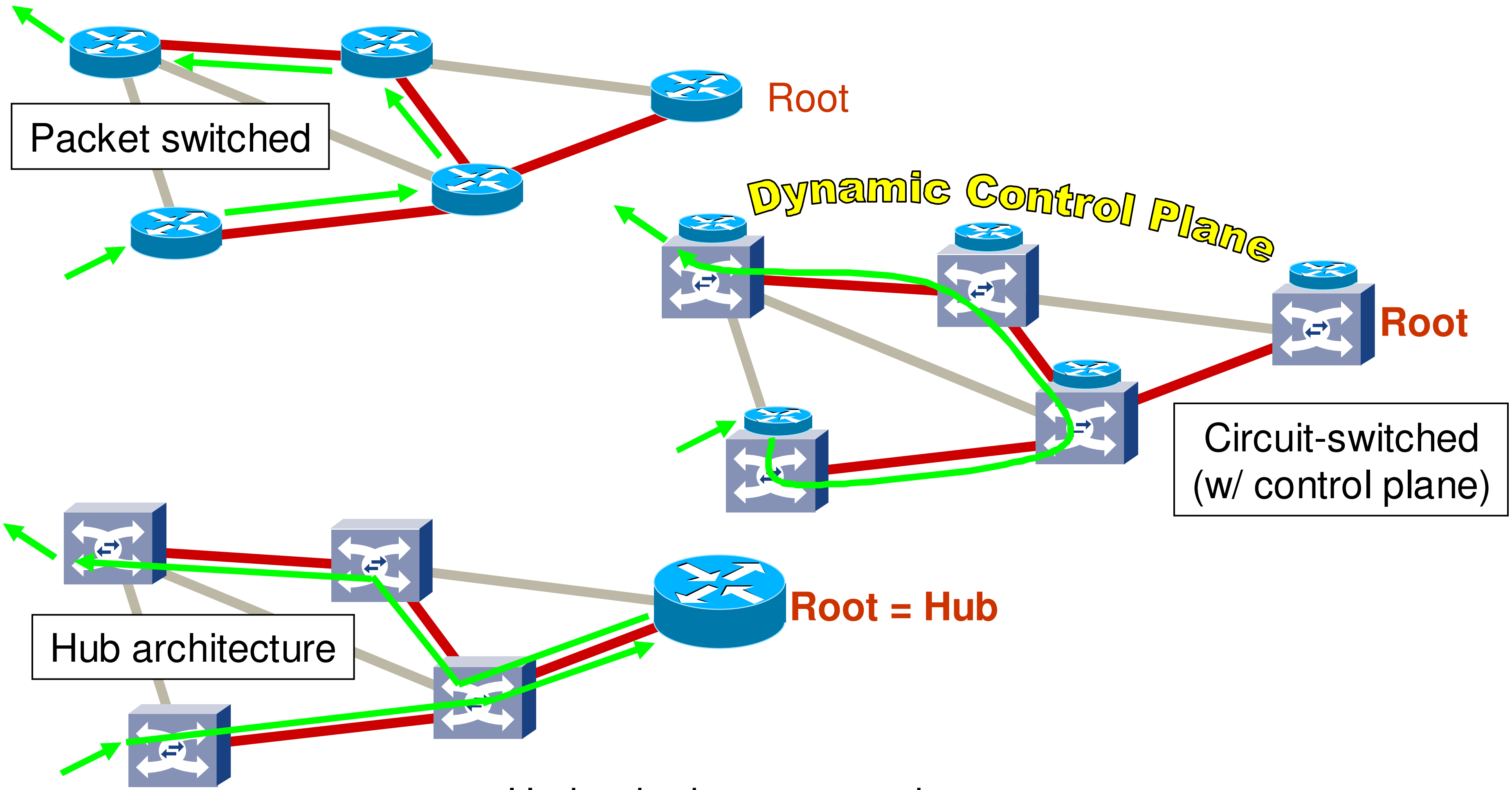


- Packets are placed onto the correct output ports based on their header information
- “Connections” on a per-packet basis
- Buffering ⇨ Delay jitter

Per-port cost ratio: IP router / SONET crossconnect ≈ 3 : 1

Alcatel·Lucent

Tree routing – Architecture options



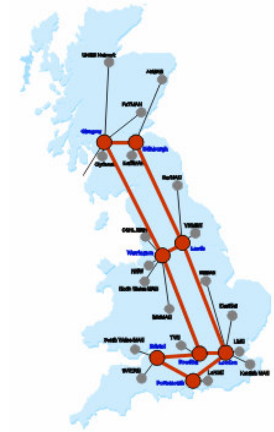
Under the hose constraint:

- All three have *same transport bandwidth* requirements
- They only differ in the type of network elements

Alcatel·Lucent

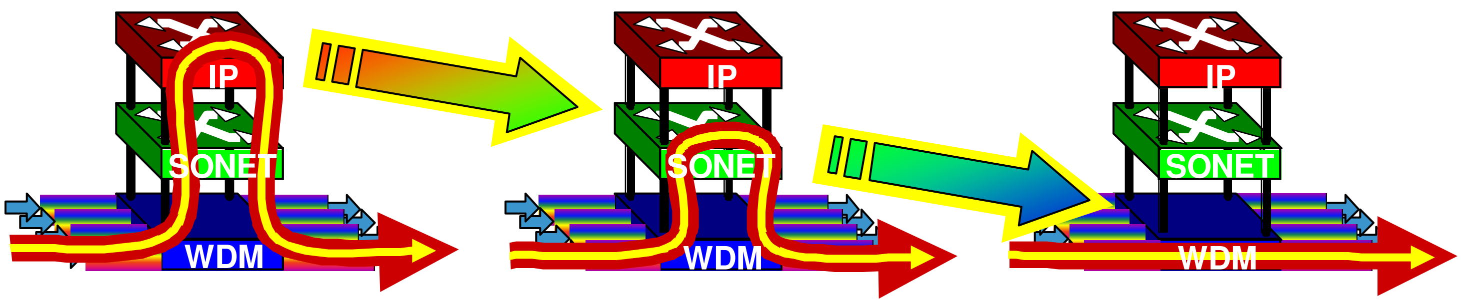
Networking equipment requirements

Architecture	Routing	Transport capacity x km	Circuit-switching capacity	Packet-routing capacity
Packet-switching	SP	**3,437**	**-**	**42**
	VPN-Tree	**2,302**	**-**	**32**
Load bal.	SP	**2,776**	**44**	**8**
Hub routing	VPN-Tree	**2,302**	**40**	**8**



Traffic assumptions:
Hose traffic with $D_i = 1$
Demands allowed to vary between 0 and 1

- Load balancing also trades packet routing for circuit switching
  - ⇨ Much cheaper networking equipment, since no unnecessary thru-traffic processing



The ultimate way to handle thru-traffic is not to handle it at all !

Alcatel·Lucent

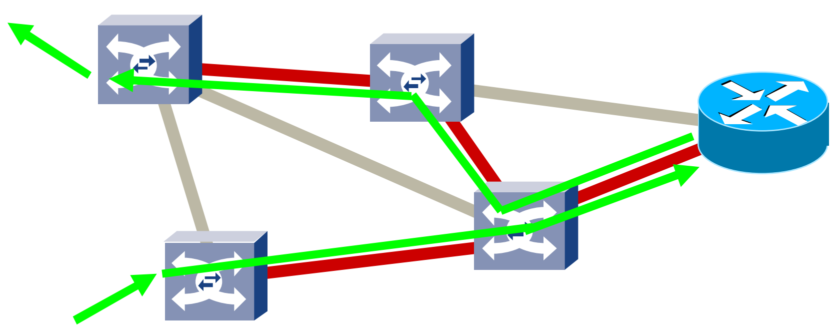
Networking equipment requirements

Architecture	Routing	Transport capacity x km	Circuit-switching capacity	Packet-routing capacity
Packet-switching	SP	**3,437**	**-**	**42**
	VPN-Tree	**2,302**	**-**	**32**
Load bal.	SP	**2,776**	**44**	**8**
Hub routing	VPN-Tree	**2,302**	**40**	**8**

Traffic assumptions:
Hose traffic with $D_i = 1$
Demands allowed to vary between 0 and 1

Hub routing is cheapest if using the optimum (VPN) tree, but is impractical

- Single point of failure
- Single packet router has to handle all network traffic



Alcatel·Lucent

Cost comparison for different networks

- Now include cost of networking equipment

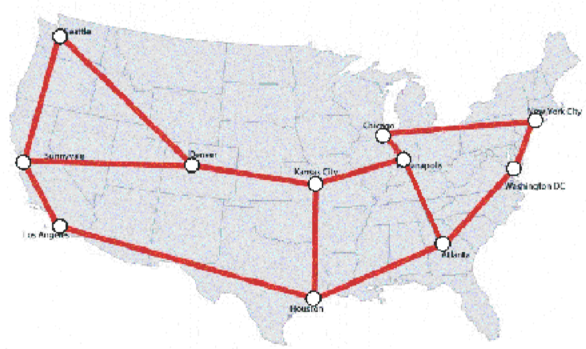
IP router port : SONET crossconnect port : WDM transport per km = 370 : 130 : 1

Architecture	Routing	JANET Rel. cost	ABILENE Rel. cost	GEANT Rel. cost
Packet-switching	SP	**1.59**	**1.43**	**1.59**
	VPN	**1.18**	**0.94**	**0.87**
Load bal.	SP	**1.00**	**1.00**	**1.00**

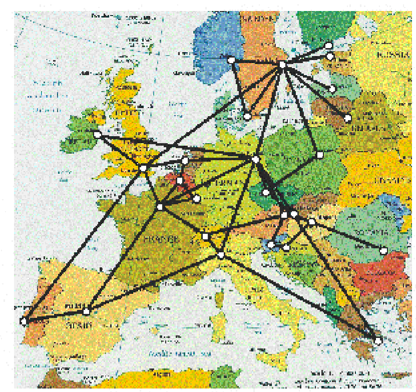
Traffic assumptions:
Hose traffic with $D_i = 1$
Demands allowed to vary between 0 and 1

- Randomized load balancing is always cheaper than shortest-path IP routing (OSPF)
- VPN-Tree routing still beats randomized load balancing on larger networks
  - ⇨ Randomized load-balancing across smaller sub-domains
  - ⇨ **Selective Randomized Load Balancing** (only use M out of N routing nodes)

(a) JANET topology



(b) ABILENE topology


(c) GEANT topology

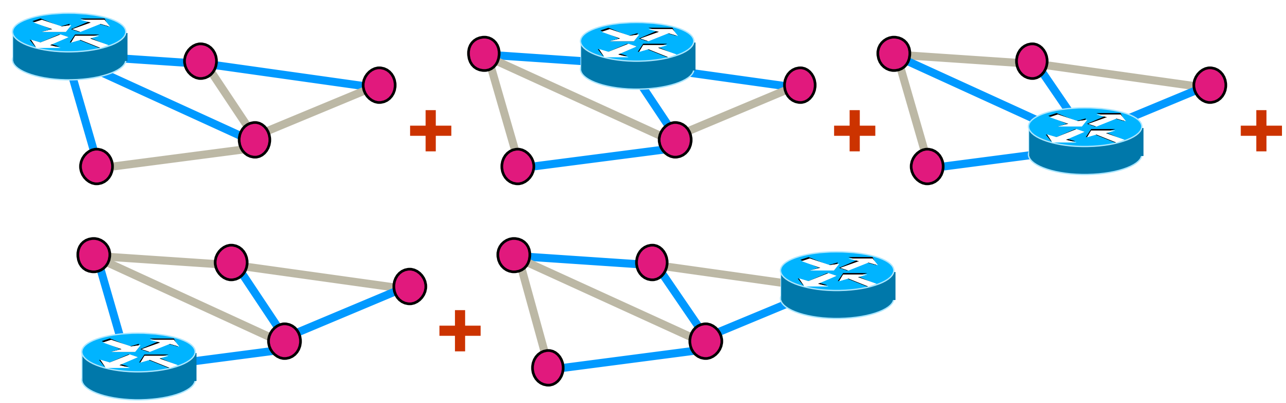
:atel·Lucent

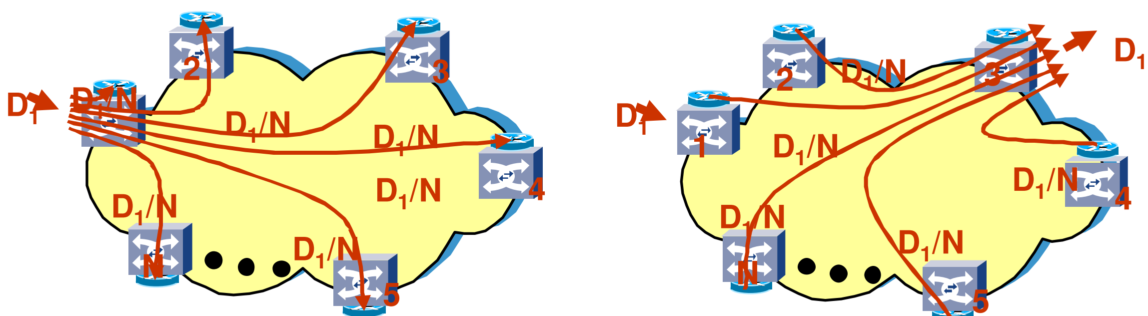
Load balancing and multi-hub routing

Randomized load balancing, as seen from a routing node (step 2):

- Step 1: Each routing node receives traffic from all the other nodes
- Step 2: Traffic received from all the other nodes is routed locally
- Step 3: Traffic is sent from each routing node to its final destination

Load balancing =





Two-step randomized load balancing

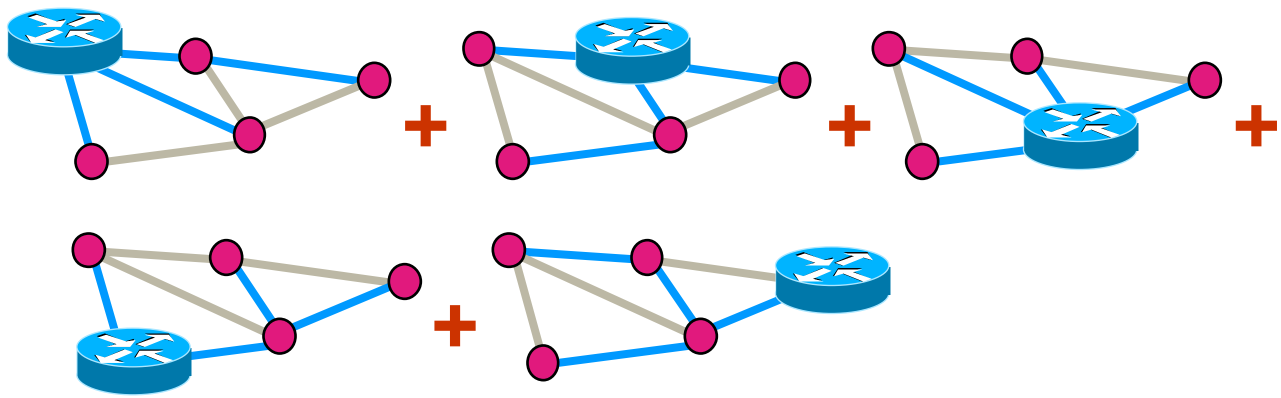
Alcatel·Lucent

Load balancing and multi-hub routing

Randomized load balancing, as seen from a routing node (step 2):

- Step 1: Each routing node receives traffic from all the other nodes
- Step 2: Traffic received from all the other nodes is routed locally
- Step 3: Traffic is sent from each routing node to its final destination

Load balancing =

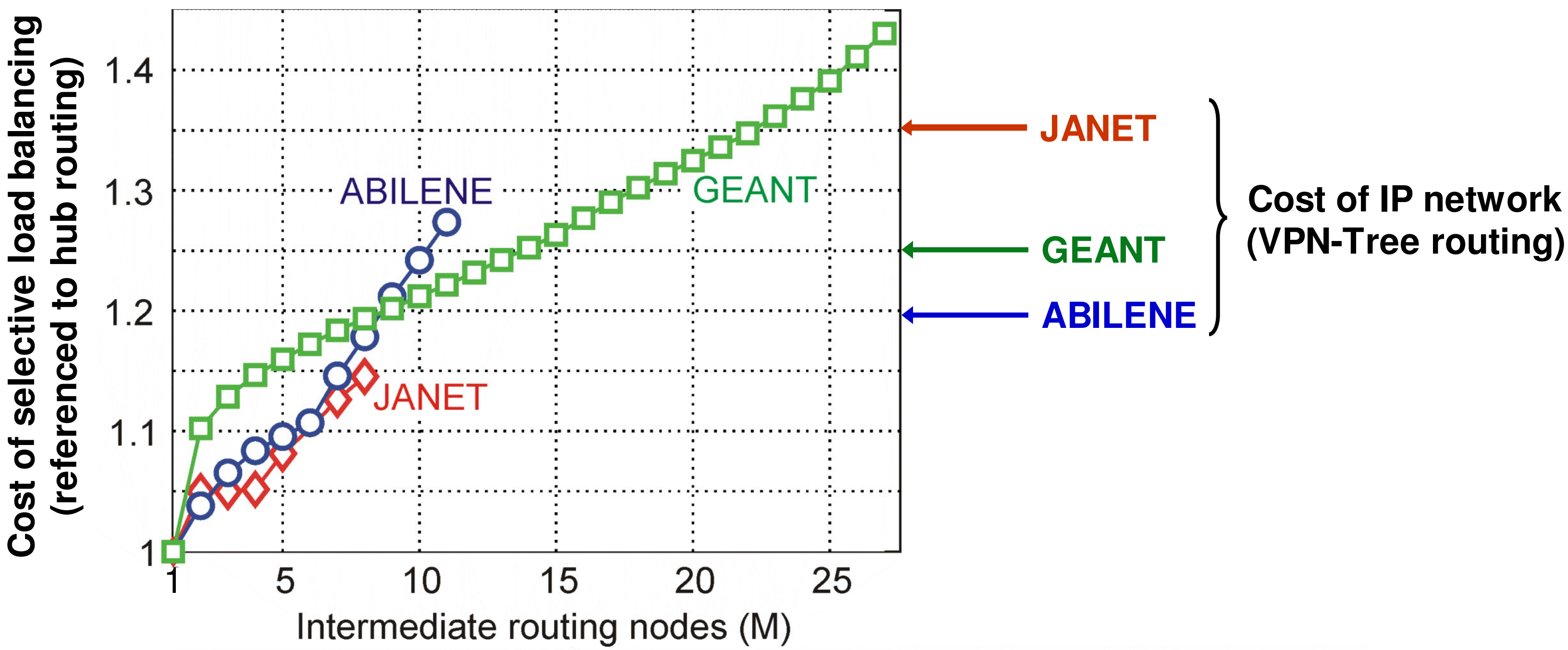


Randomized load balancing = Multi-hub routing

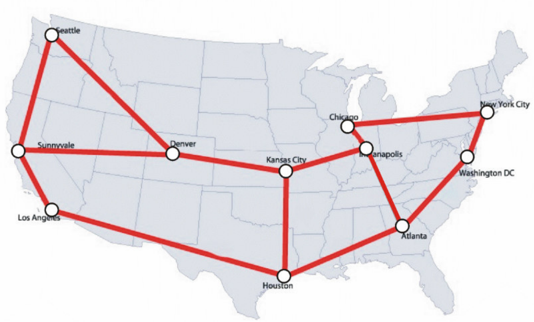
- Cost of load balanced network is the linear average of *N* hub-routed network costs
- Some of the *N* hub-routed networks are more expensive than others
- Don't take all *N* hub-routed networks for load balancing, but only the *M* cheapest ones

Alcatel·Lucent

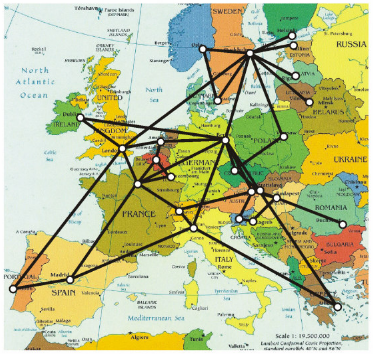
Selective Randomized Load Balancing



(a) JANET topology



(b) ABILENE topology



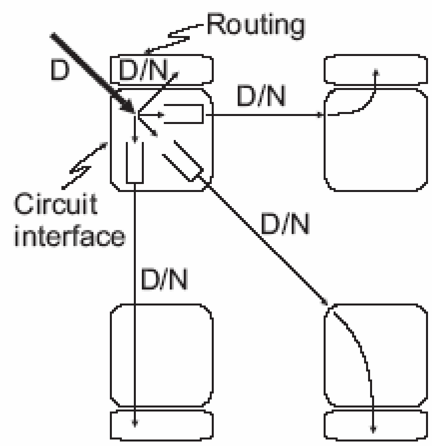
(c) GEANT topology

Alcatel·Lucent

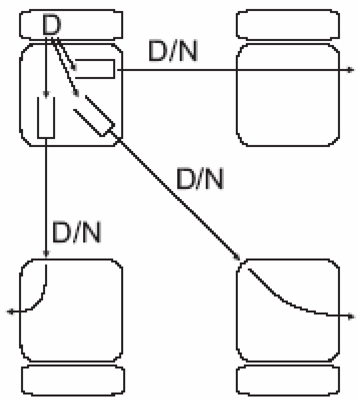
5 How random is 'random': Queuing in RLB

Queues in RLB

- Two RLB steps → Two queues
  - Distribution step
  - Routing step
- Two splitting schemes
  - Purely random split
  - Pseudo-random split (e.g., Round-Robin)
- Queues could have same or different priorities for distribution and routing step traffic



(a) Step 1: Traffic distribution



(b) Step 2: Traffic routing

Alcatel·Lucent

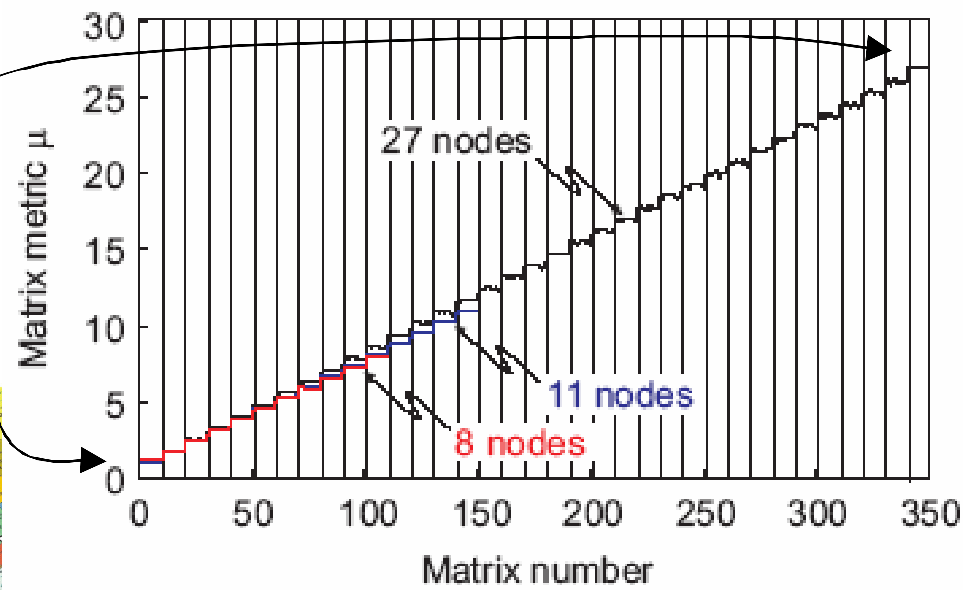
Queuing Analysis

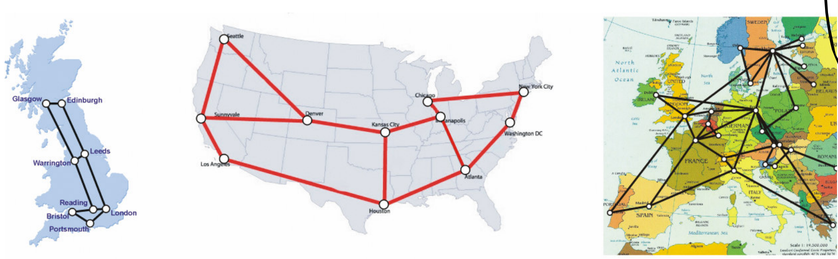
- Pseudo-random traffic split (Round-Robin)
  - For a given offered load, the mean queue sizes depend on the <u>*traffic demand uniformity*</u>
    - – Uniformity quantified by sum of squared traffic demands

$$\mathbb{E}\{Q_{j,2}\} = \frac{\alpha_j^{(2)} - \alpha_j}{2(1-\alpha_j)}, \qquad \alpha_j^{(2)} = \alpha_j(\alpha_j+1) - \underbrace{\sum_{i=1}^{N} \alpha_{ij}^2}_{\mu}, \qquad \alpha_j \equiv \text{mean offered load}$$

Smaller μ implies more uniform traffic

- Uniform demands: $\mu = N/(N\text{-}1)$
- Full point-to-point: $\mu = N$





(a) JANET topology (b) ABILENE topology (c) GEANT topology

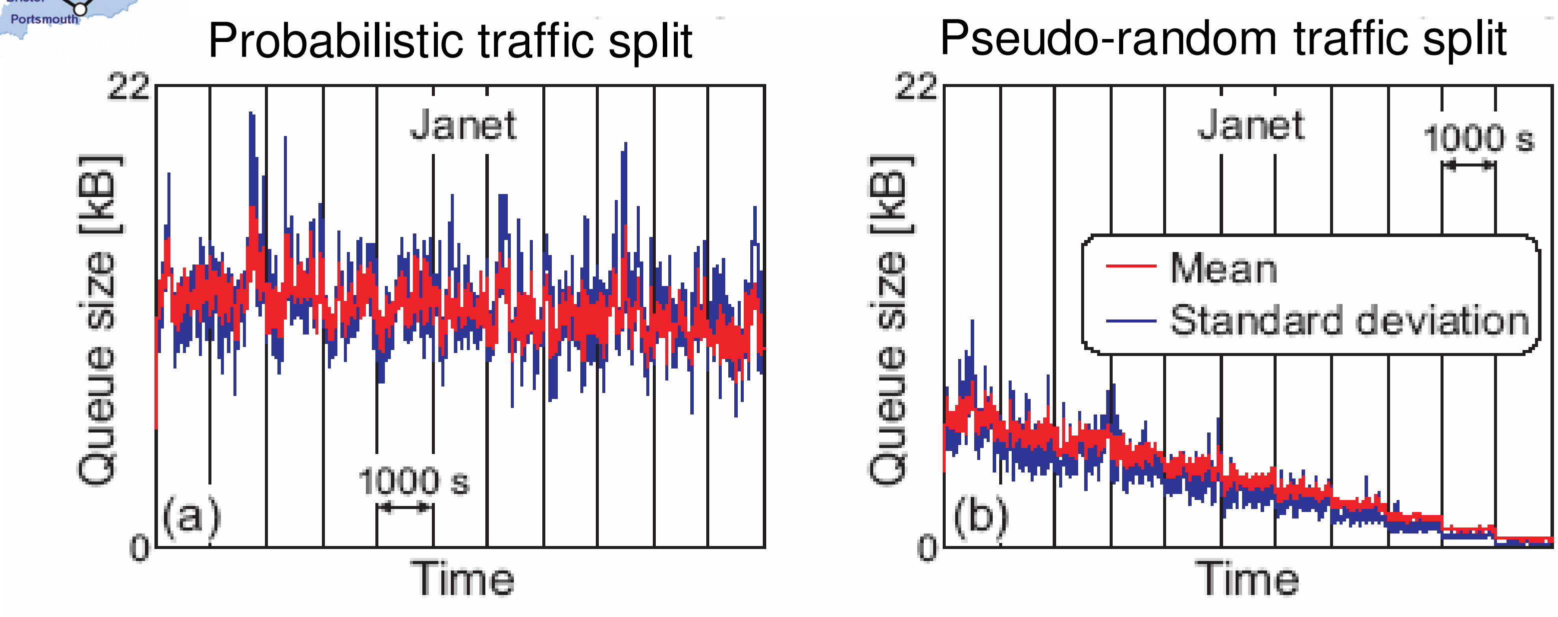
Alcatel·Lucent

Simulation Results

Offered Load: 95%

All queues are equivalent

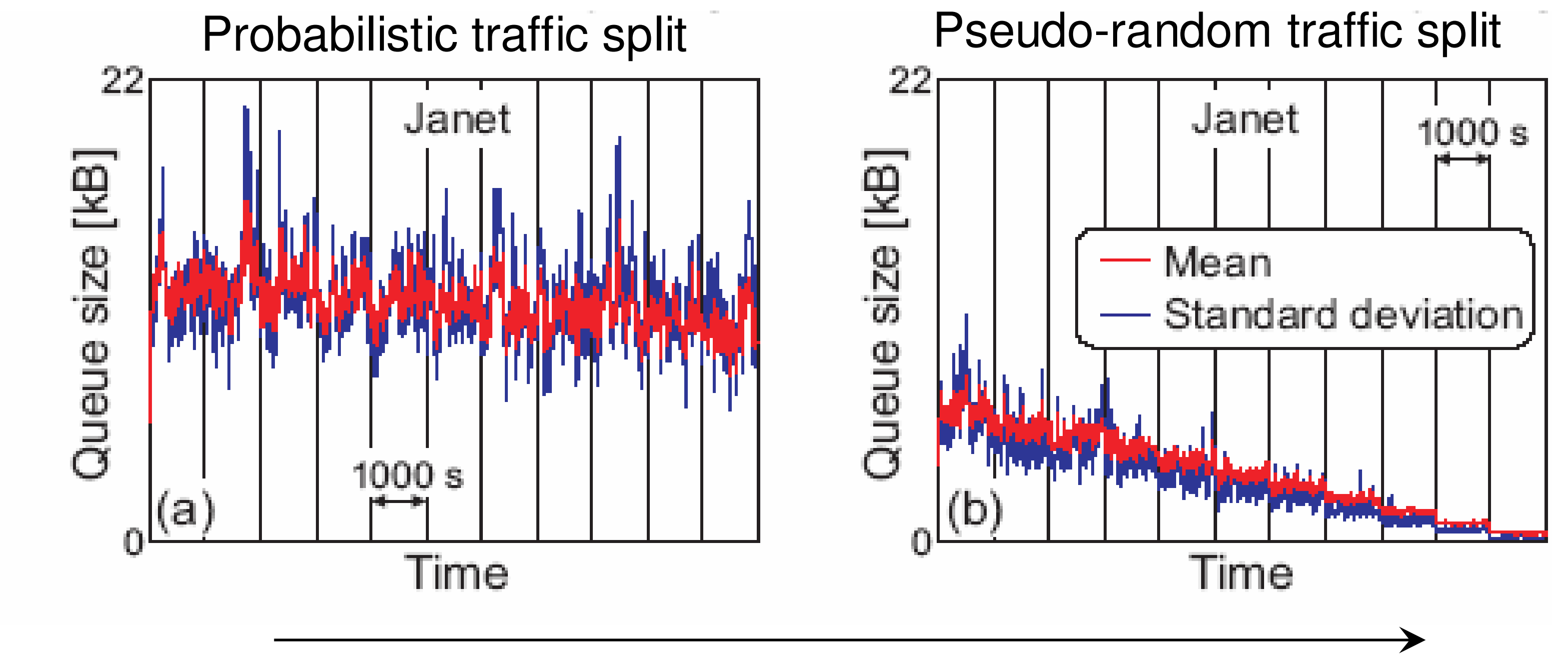
Network-wide average results



Traffic matrices become less and less uniform

Alcatel·Lucent

Simulation Results

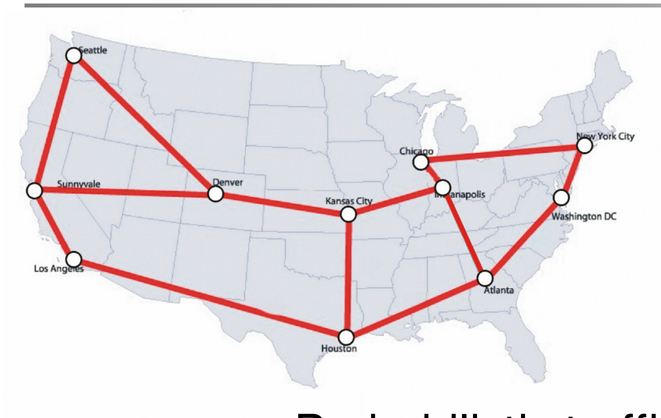


Traffic matrices become less and less uniform

Pseudo-random traffic split:

- Average queue size gets smaller with skewed traffic
  - Pseudo-random splitting maximally smoothens traffic if all traffic is destined to a single destination
- Worst-case queue size is *half* that of random splitting
  - No step 1 queue build-up for pseudo-random splitting

Simulation Results

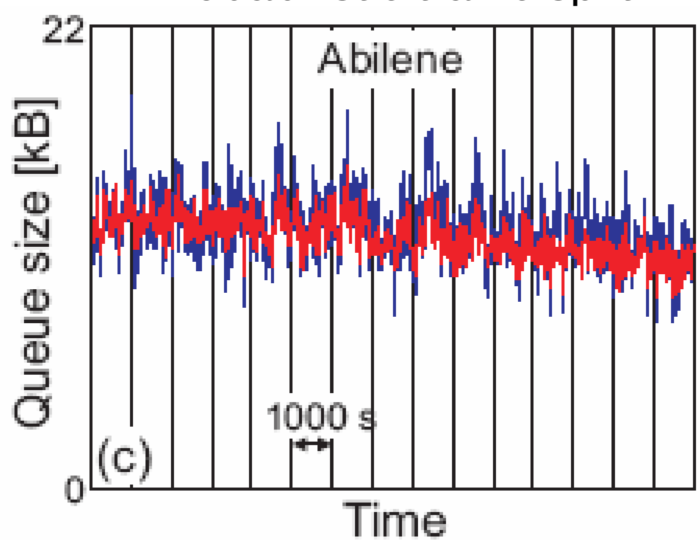


Offered Load: 95%

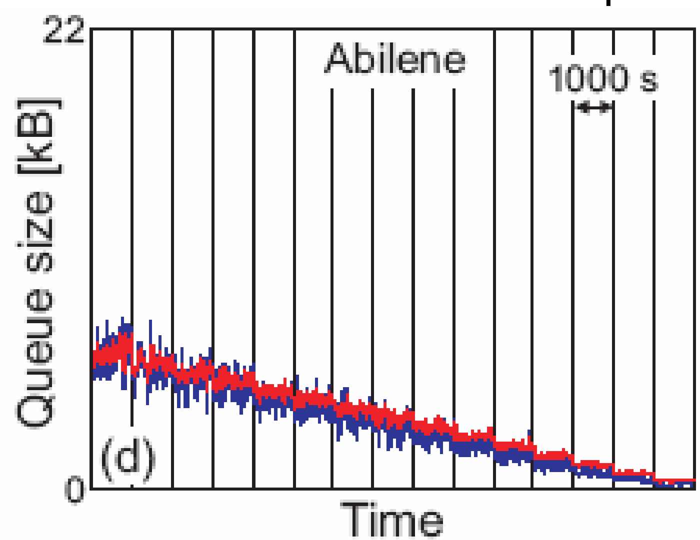
All queues are equivalent

Network-wide average results

Probabilistic traffic split



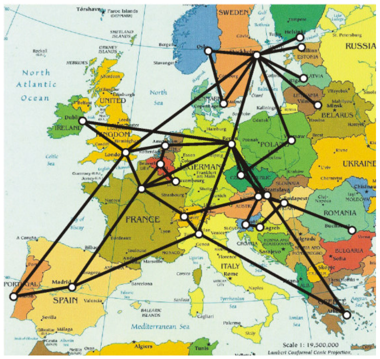
Pseudo-random traffic split



Traffic matrices become less and less uniform

Alcatel·Lucent

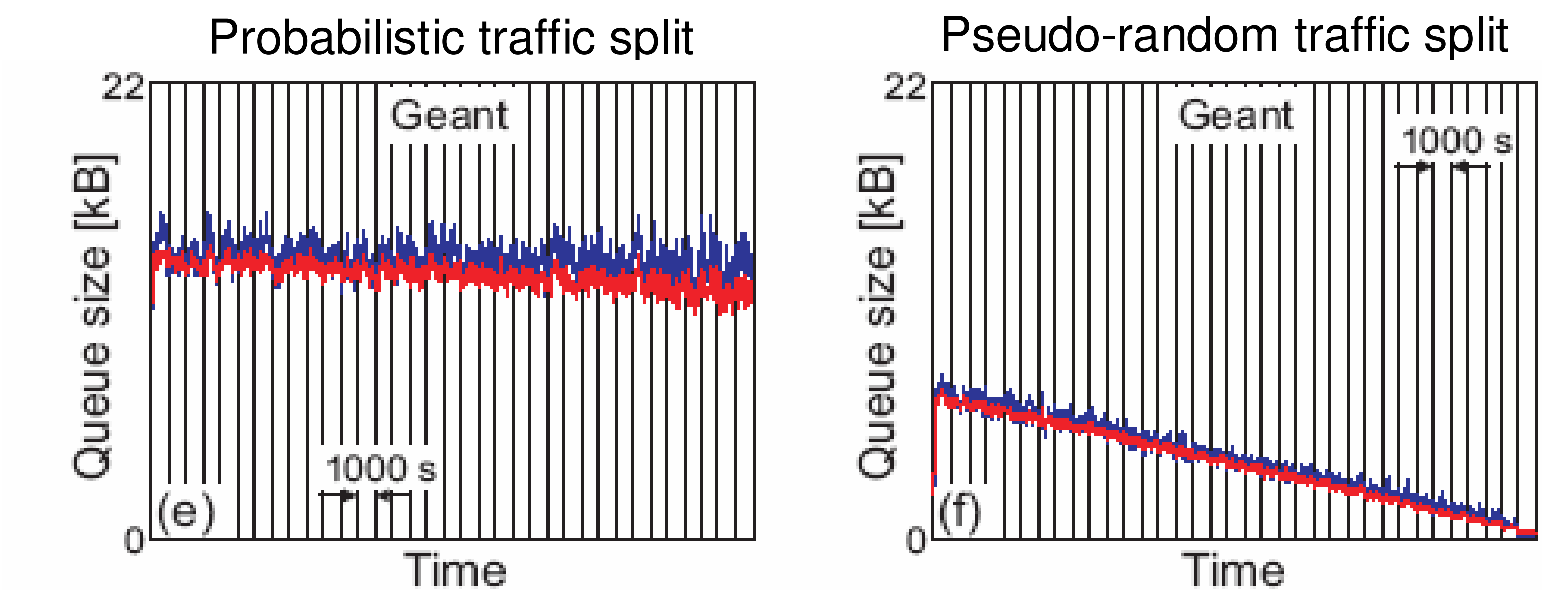
Simulation Results



Offered Load: 95%

All queues are equivalent

Network-wide average results

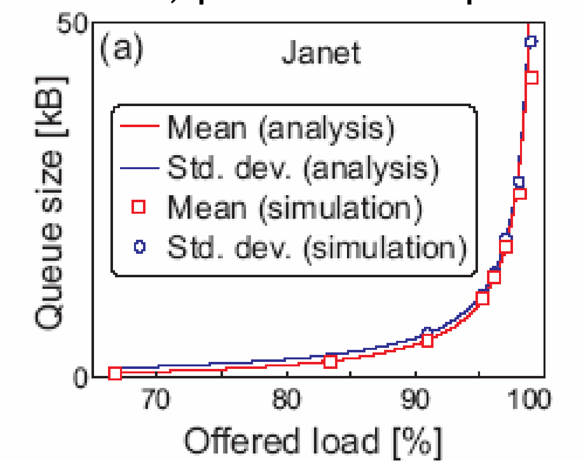


Traffic matrices become less and less uniform

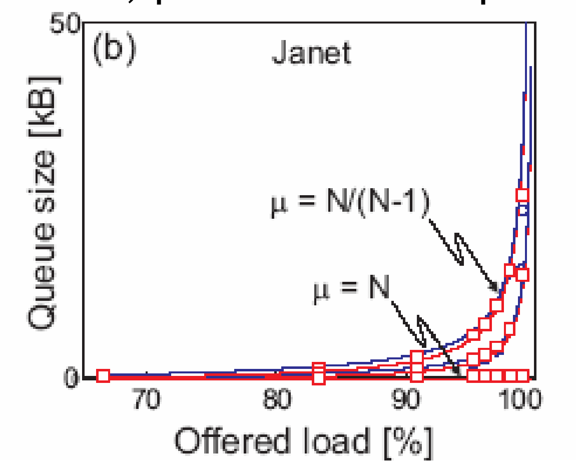
Alcatel·Lucent

Queue Size and offered load

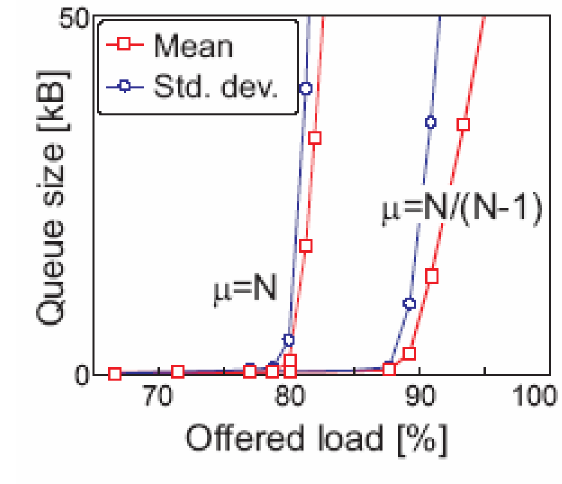
RLB, probabilistic split



RLB, pseudo-random split



Multi-hop, shortest path



- Shortest-path routing shows much larger queue standard deviations than RLB

  → Hot-spots in network !
- Different priorities among RLB queues:

  We see no effect of different priorities between distribution and routing steps

  (Possibly due to traffic being uncorrelated)

Alcatel·Lucent

Summary and proposed future work

- Data services are showing an increasing amount of demand flexibility
- Randomized Load Balancing (RLB) is a robust network architecture
  - Easy to dimension (design for uniform traffic matrices)
    → MORE WORK NEEDED ON RESILIENCE / RESTORATION
  - No control plane, dynamic topology maps, etc.
    → MORE WORK NEEDED ON HYBRID SOURCE ROUTING & RLB
  - Cost efficient and scalable due to the reduction of packet routers
    → MORE WORK NEEDED TO UNDERSTAND RESEQUENCING ISSUES
  - Favorable queuing behavior compared to shortest-path routing
    → MORE WORK NEEDED ON TRAFFIC ENGINEERING FOR RLB
  - Coding for security and resilience
    → MORE WORK NEEDED ON CODING FOR RESILIENCE & SECURITY

→ EXPERIMENTAL DEMONSTRATION ON LIVE TRAFFIC NEEDED !

www.alcatel-lucent.com