Routing in the Future Internet

Marcelo Yannuzzi

Graduate Course (Slideset 2)

Institute of Computer Science

University of the Republic (UdelaR)

August 20th 2012, Montevideo, Uruguay

Department of Computer Architecture
Technical University of Catalonia (UPC), Spain

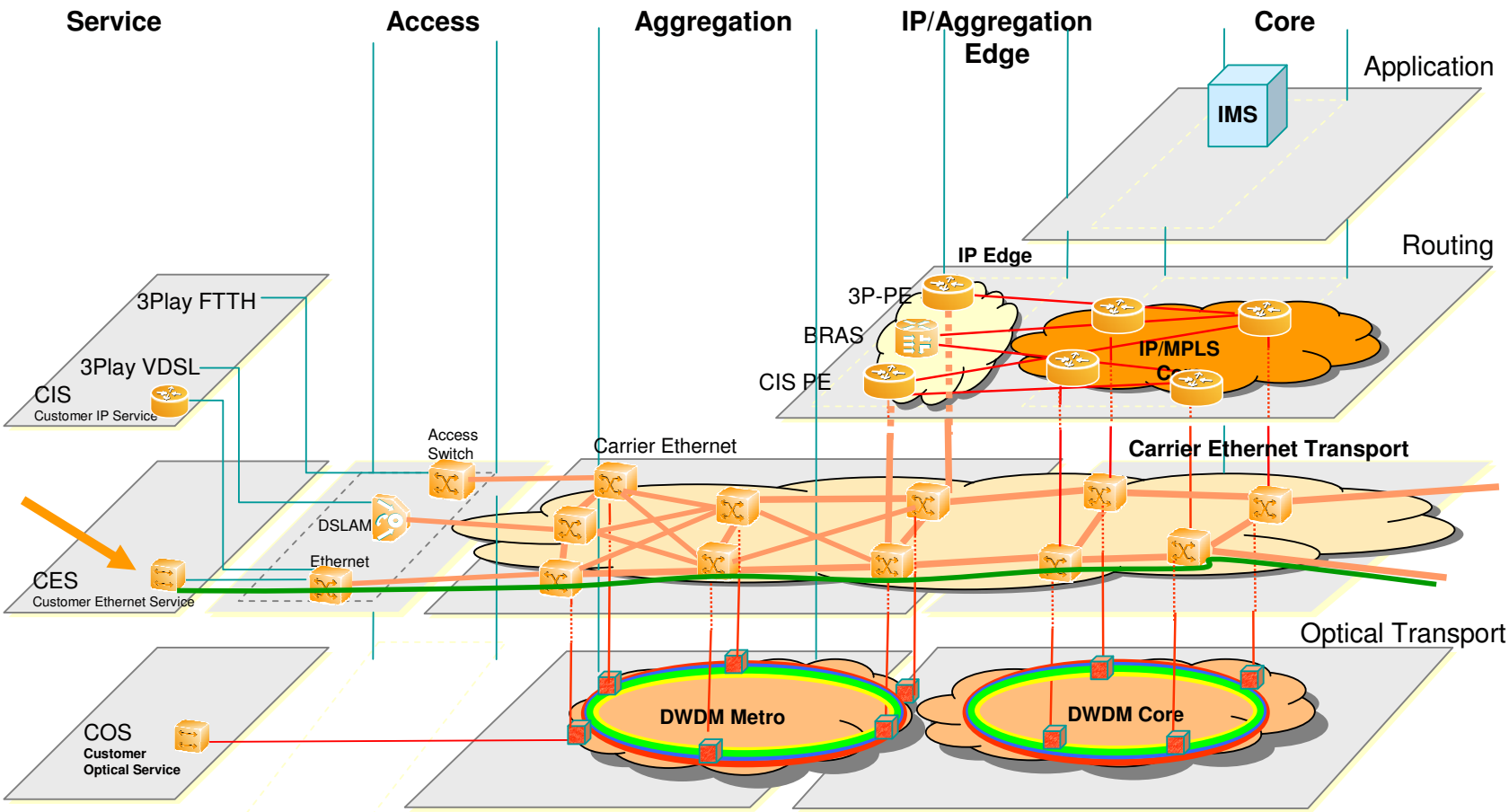
Institute of Computer Science
University of the Republic (UdelaR), Uruguay

Outline

1. **Intradomain aspects**
   - **A look inside carrier-grade networks, and their data and control planes**

Multi-layer Data Planes

The Multi-layer Structure



The Multi-layer Structure (cont.)

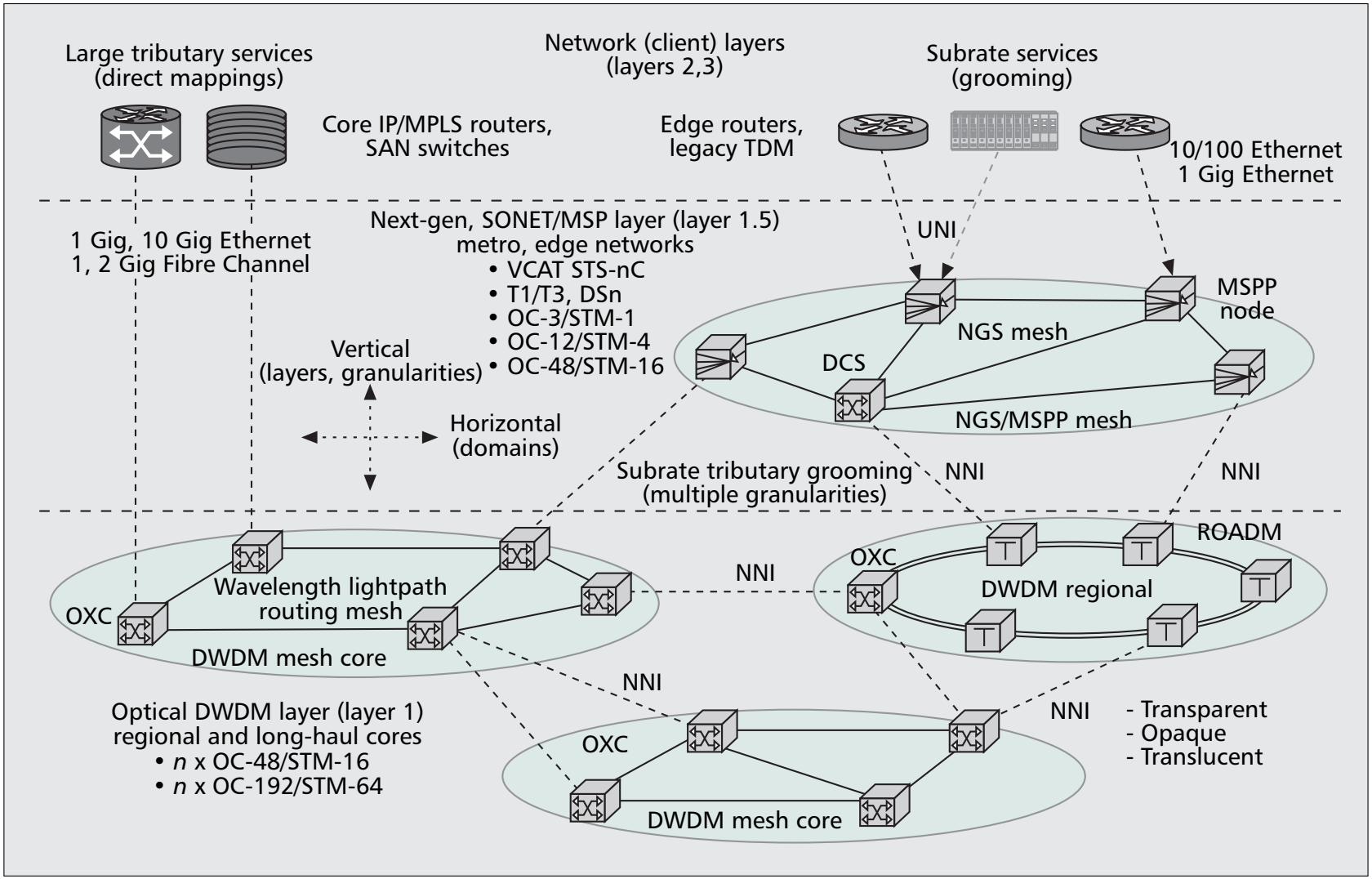
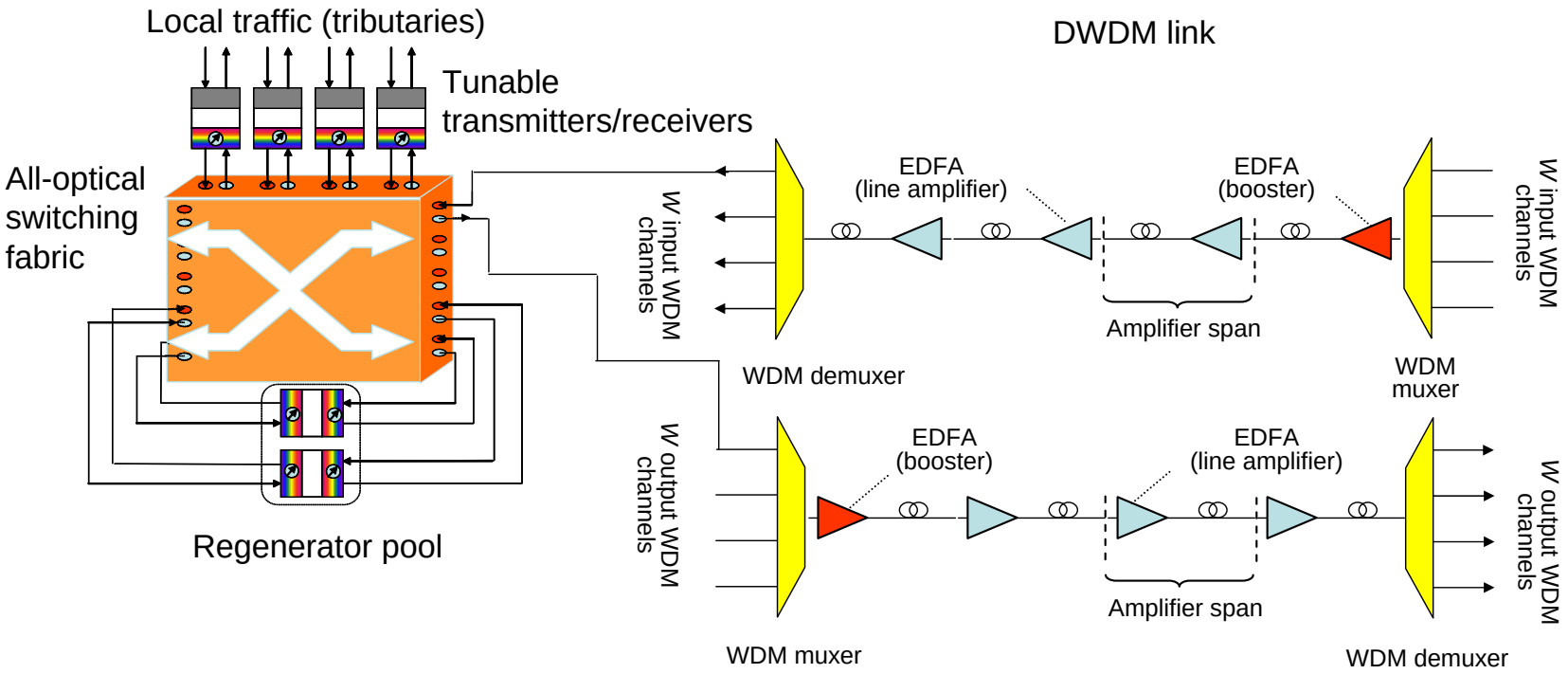


Figure 1. *Multidomain/multilayer optical networks.*

- Source: N. Ghani, et al, "Control Plane Design in Multidomain/Multilayer Optical Networks," IEEE Communications Magazine, Vol. 46, No. 6, June 2008, pp. 78-87.

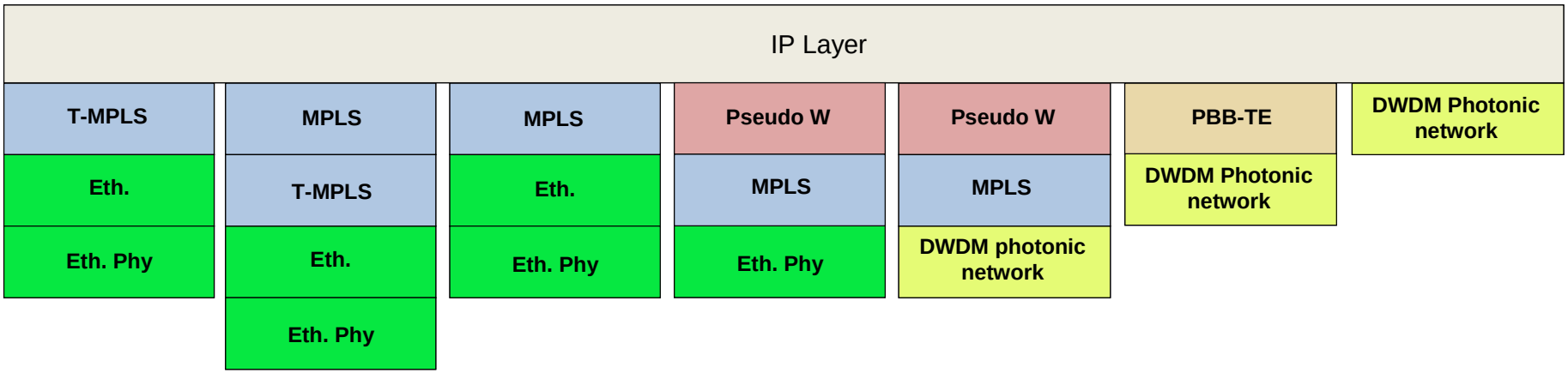
The Optical Layer and its Data and Control Plane

Translucent Optical Transport Network (OTN)



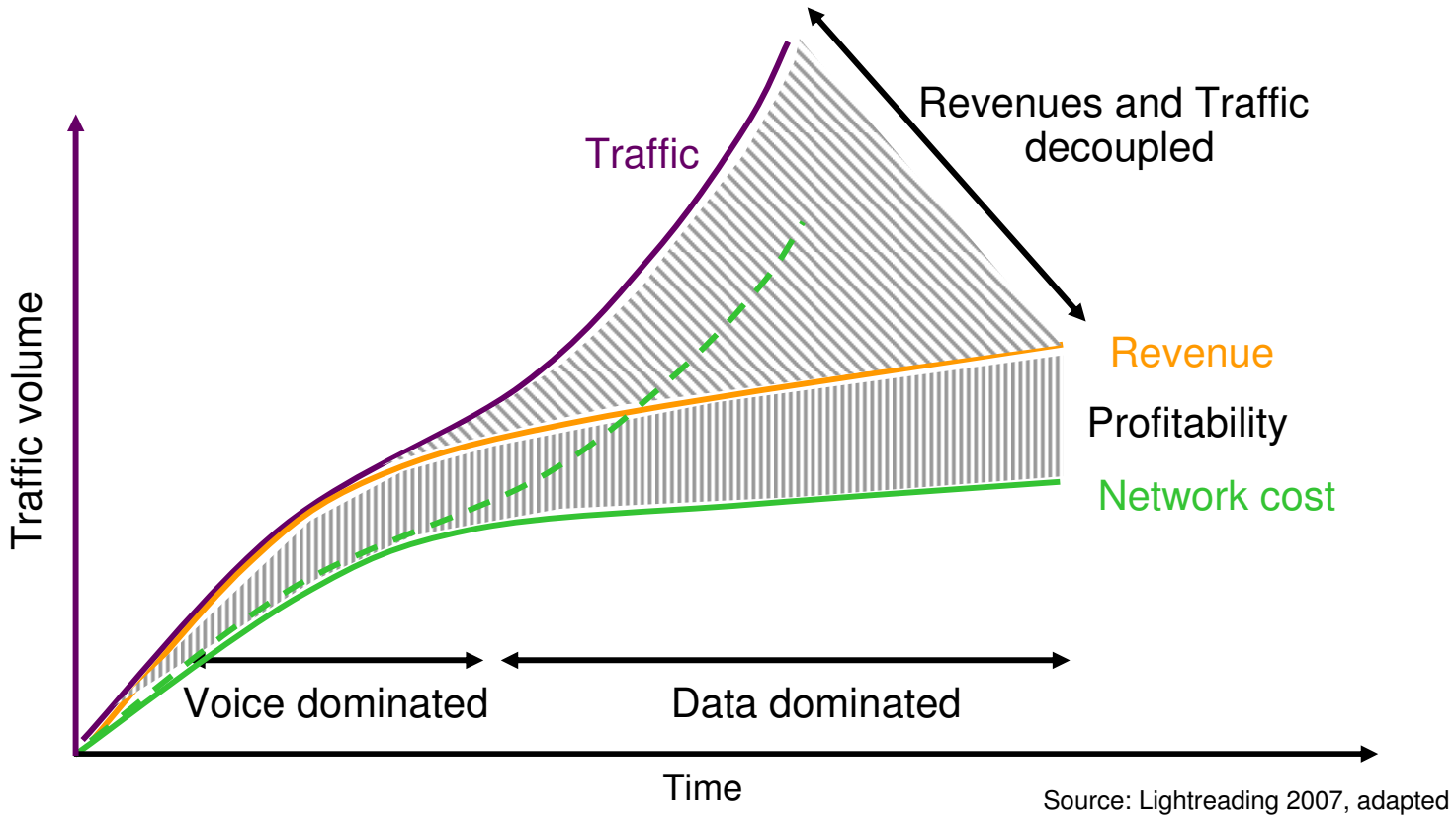
- Source: M. Yannuzzi et al., “Performance of translucent optical networks under dynamic traffic and uncertain physical-layer information,” in Proceedings of the 13th IFIP/IEEE Conference on Optical Network Design and Modelling (ONDM 2009), Braunschweig, Germany, February 2009.

Protocol Stack Zoo



Drivers for more optics: 1) The cost per-bit

Decoupling of revenues and traffic requires low cost per bit technology



Source: Lightreading 2007, adapted

- Source: Dominic Schupke, Nokia Siemens Networks, “Options and Opportunities for Optical Network Resilience,” invited talk at IFIP/IEEE ONDM 2009, Braunschweig, Germany, February 2009.”

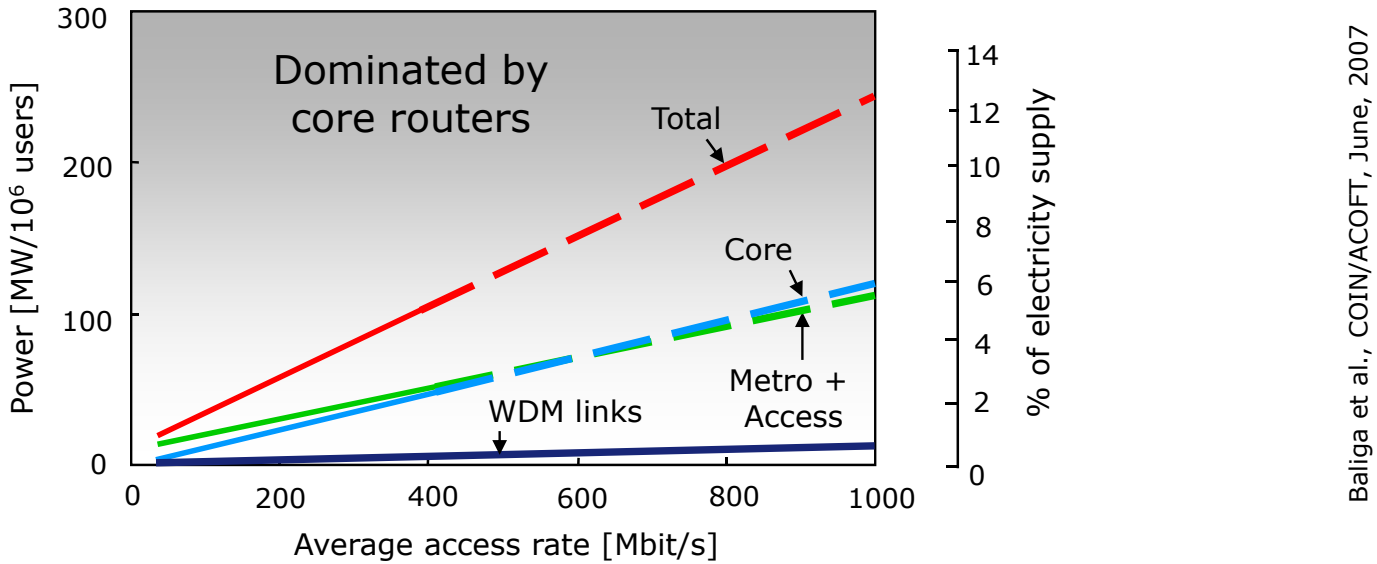
Drivers for more optics: 2) Energy Efficiency

	Implemented current equipment	Next generation equipment
IP/MPLS Router	1 kW / 100Gbit/s	<0.6 kW / 100Gbit/s
L2 Switch	0.5 kW / 100Gbit/s	<0.2 kW / 100Gbit/s
ROADM /PXC Node	0.04 kW / 100Gbit/s at 10Gbit/s in 50 GHz spacing	0.01 kW /100Gbit/s at 40Gbit/s in 50 GHz spacing

- Source: Nokia Siemens Networks, “Greener Networks to Exploit Multilayer Interworking.”

Drivers for more optics: 2) Energy Efficiency

Power requirements:

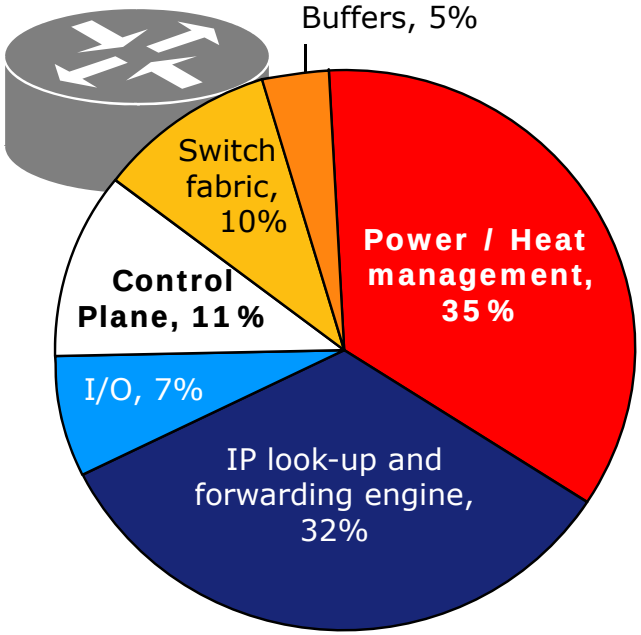


- If ***33%*** of the world's population were to obtain broadband access:

Access rate	1Mbit/s	10Mbit/s
Power consumption	100GW	1TW
Percentage of world's 2007 electricity supply	5%	**50%**

- Source: Christoph Glingener, ADVA Optical Networking, keynote at IFIP/IEEE ONDM 2009, Braunschweig, Germany, February 2009."

Drivers for more optics: 2) Energy Efficiency

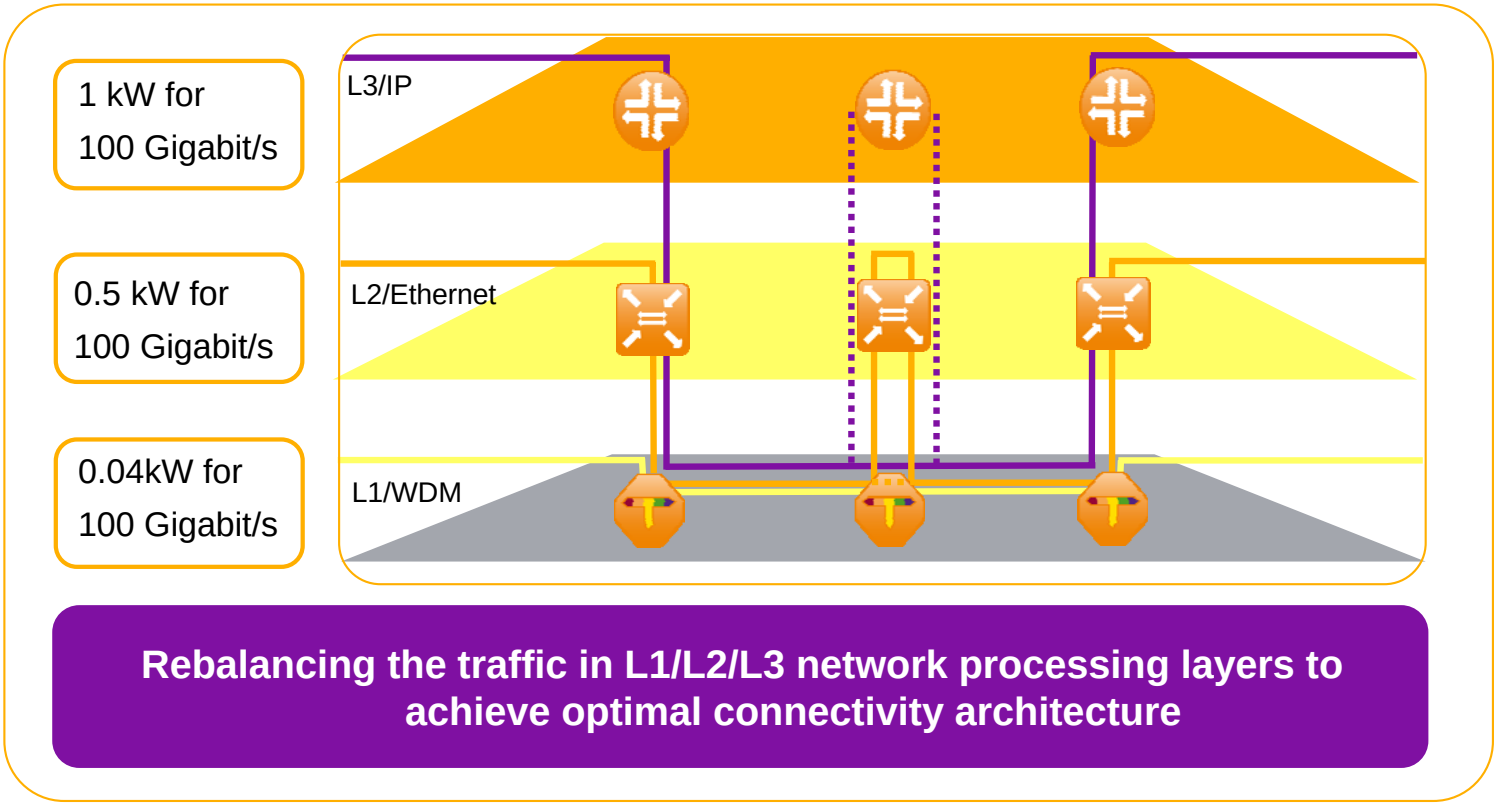


Source: G. Epps, Cisco, 2007

- Power driver : IP look-up/forwarding engine
- I/O – optical transport: is lower in power consumption than switch fabric
- Wireless access power consumption: 10-20 times higher than wired solutions

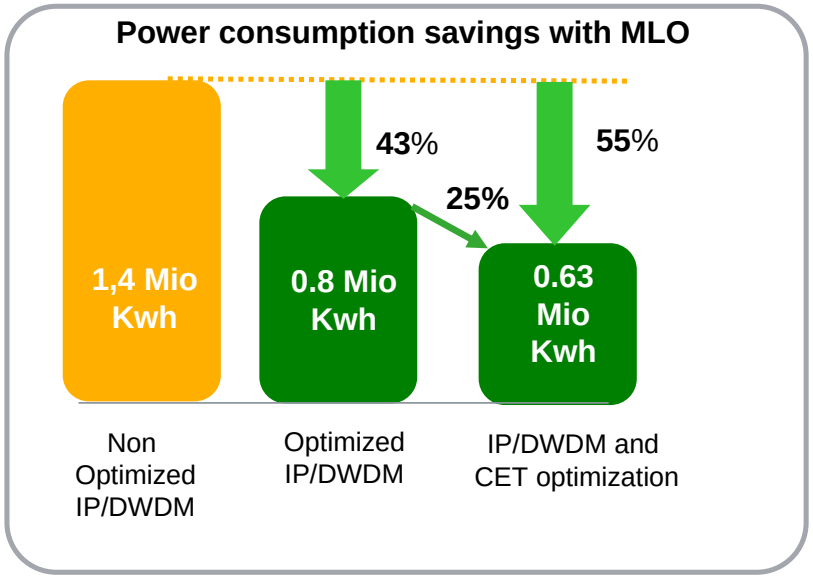
- Source: Christoph Glingener, ADVA Optical Networking, keynote at IFIP/IEEE ONDM 2009, Braunschweig, Germany, February 2009."

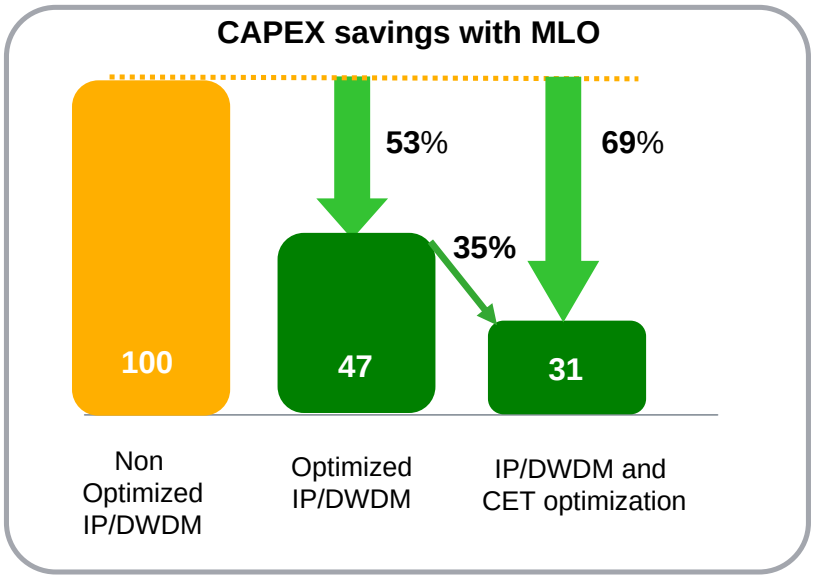
Drivers for more optics: 3) Distributing Traffic



- Source: Nokia Siemens Networks, “Greener Networks to Exploit Multilayer Interworking.”

Drivers for more optics: 4) Multi-layer Optimization

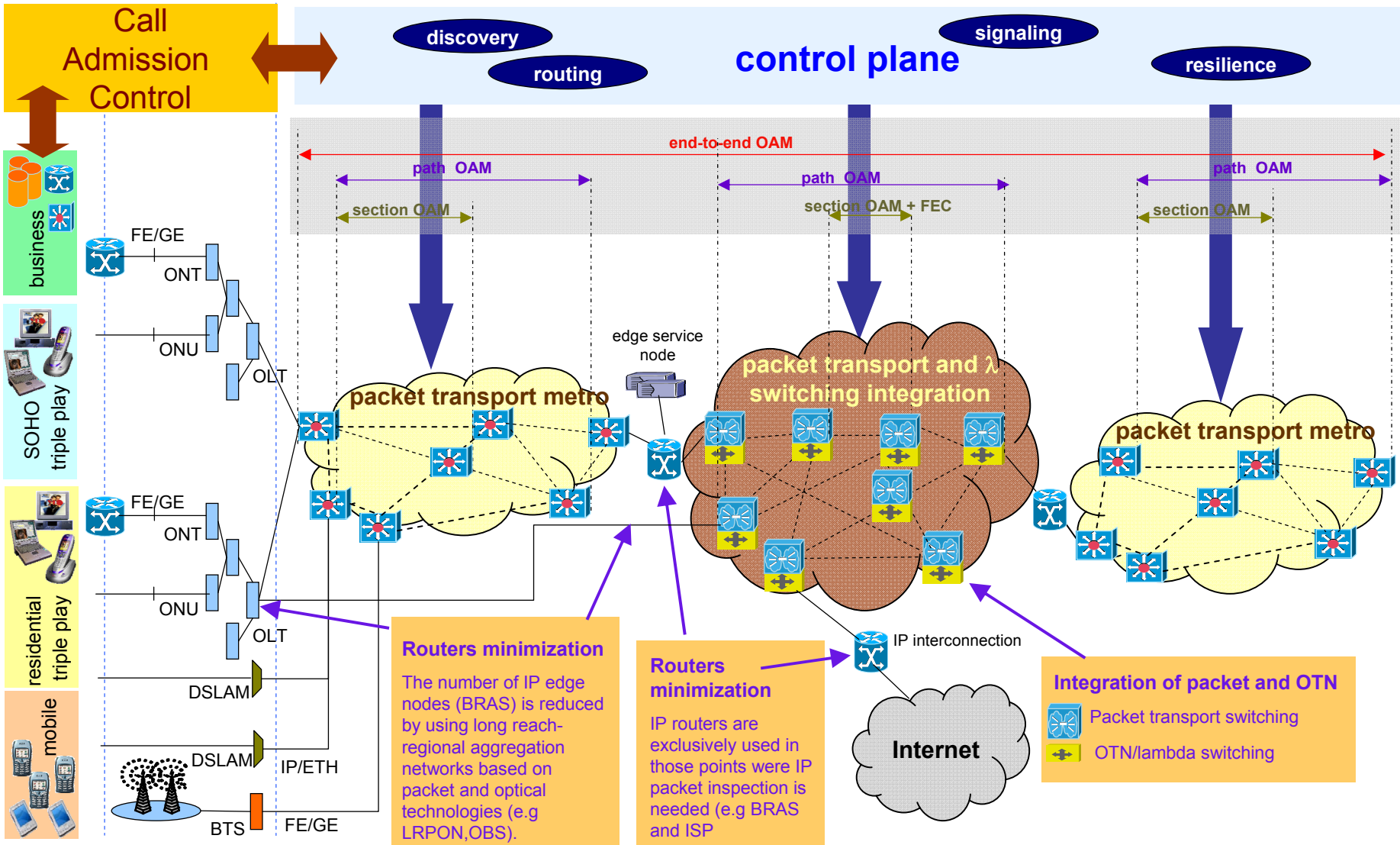




European operator case study, 2008

- Source: Nokia Siemens Networks, “Greener Networks to Exploit Multilayer Interworking.”

Overall....reducing the number of IP routers



- Source: STRONGEST, FP7, ICT EU project.

Control Plane: Advances in standardization

Body	Framework	Routing	Signaling	TE path comp.
ITU-T	Automatically switched transport network (ASTN), formerly ASON (G.8080)	• G.7713.1/Y.1704: distributed call and connection management (PNNI-based) • G.7714/Y.1705: generalized automatic discovery • G.7715/Y.1706: architecture and requirements for routing	• G.7713.2/Y.1704: distributed call and connection management (GMPLS RSVP-TE-based) • G.7713.3/Y.1704: distributed call and connection management (GMPLS CR-LDP-based)	• None specified
IETF	Generalized multiprotocol label switching (GMPLS)	• RFC 4258 (requirements for GMPLS for ASON) • IGP routing extensions for discovery of TE node capabilities • OSPF extensions in support of inter-AS MPLS and GMPLS • Virtual connection aggregation	• RFC 4208 GMPLS UNI • RSVP-TE for overlay model • Interdomain MPLS and GMPLS TE extensions for RSVP-TE	• Policy-enabled PCE framework • PCE protocol (PCEP) • Per-domain path computation for inter-domain TE LSP setup
OIF	User-network interface (UNI), network-network interface (NNI)	• OIF UNI 2.0 • E-NNI-OSPF-01.0 • E-NNI OSPF-based routing 1.0 intra-carrier implementation areement	• OIF-E-NNI-Sig-2.0: intracarrier E-NNI signaling • OIF-UNI-01.0-R2-RSVP: RSVP extensions for UNI 1.0	• None specified

Table 1. *Multidomain optical networking standards.*

- Source: N. Ghani, et al, “Control Plane Design in Multidomain/Multilayer Optical Networks,” IEEE Communications Magazine, Vol. 46, No. 6, June 2008, pp. 78-87.

Control Plane: Research topics

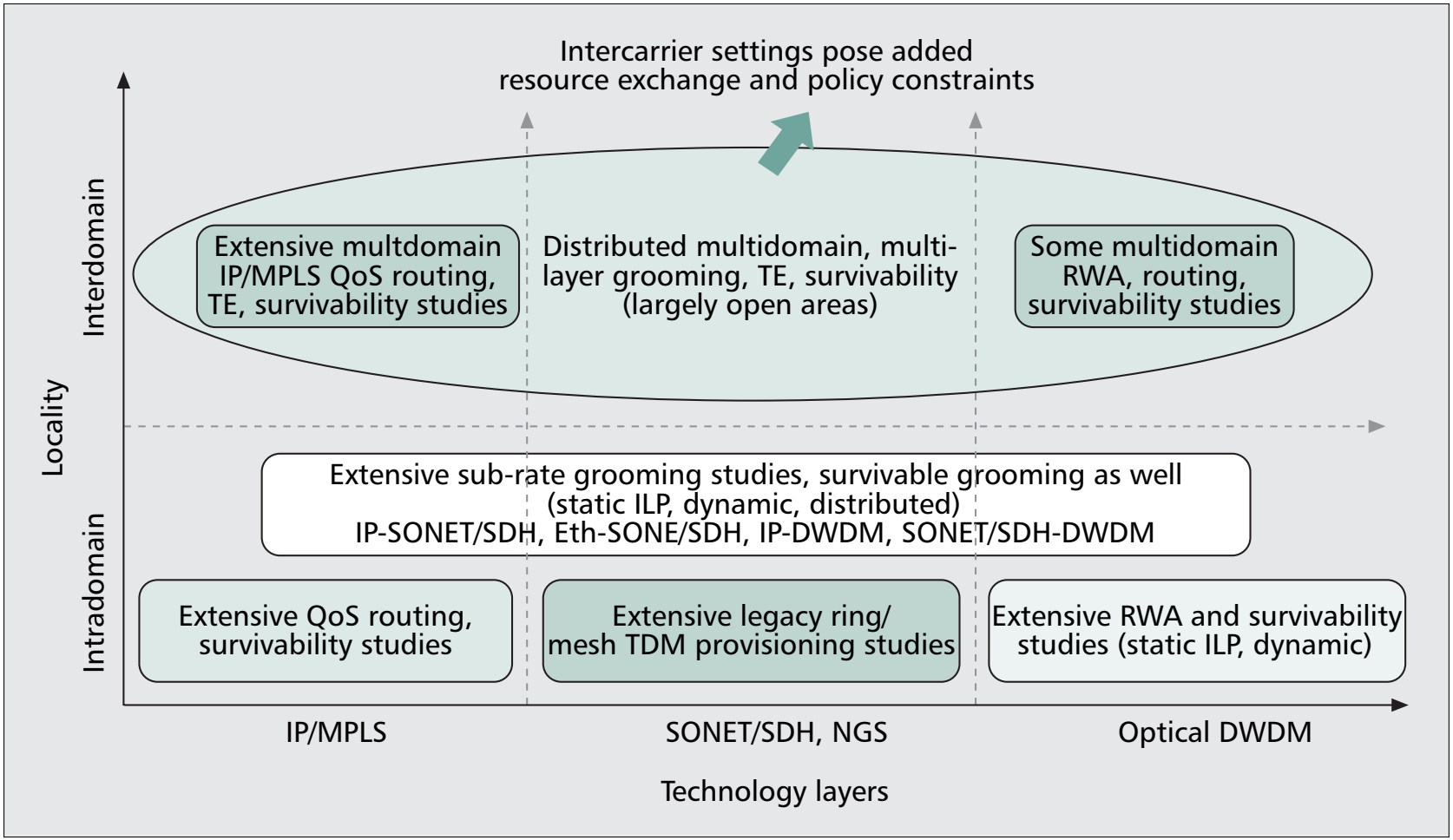


Figure 2. *Overview of research topic areas (wireline networks).*

- Source: N. Ghani, et al, "Control Plane Design in Multidomain/Multilayer Optical Networks," IEEE Communications Magazine, Vol. 46, No. 6, June 2008, pp. 78-87.

Control Plane: Open challenges

Focus area	Open challenges
State dissemination	State aggregation • Multigranularity topology aggregation (mesh, tree, etc.) • Virtual connection link aggregation schemes Update generation • Scalable update policies for aggregated state • Scalable update policies for virtual connection links
TE path computation	Per-domain strategies • Crankback signaling across domains/granularities • Re-optimization/re-grooming of circuit routes PCE-based strategies • Novel TE-based loose route (LR) algorithms • Diversity state in aggregated graphs
Survivability	Protection strategies • Novel TE-based LR primary/back algorithms • Multilayer SLA support (dedicated, shared) • Serial/parallel setup signaling schemes Restoration strategies • Crankback signaling across domains/granularities • Priority/preemption for multi-SLA support

Table 2. *Summary of research challenges.*

- Source: N. Ghani, et al, "Control Plane Design in Multidomain/Multilayer Optical Networks," IEEE Communications Magazine, Vol. 46, No. 6, June 2008, pp. 78-87.

Control Plane: State dissemination models

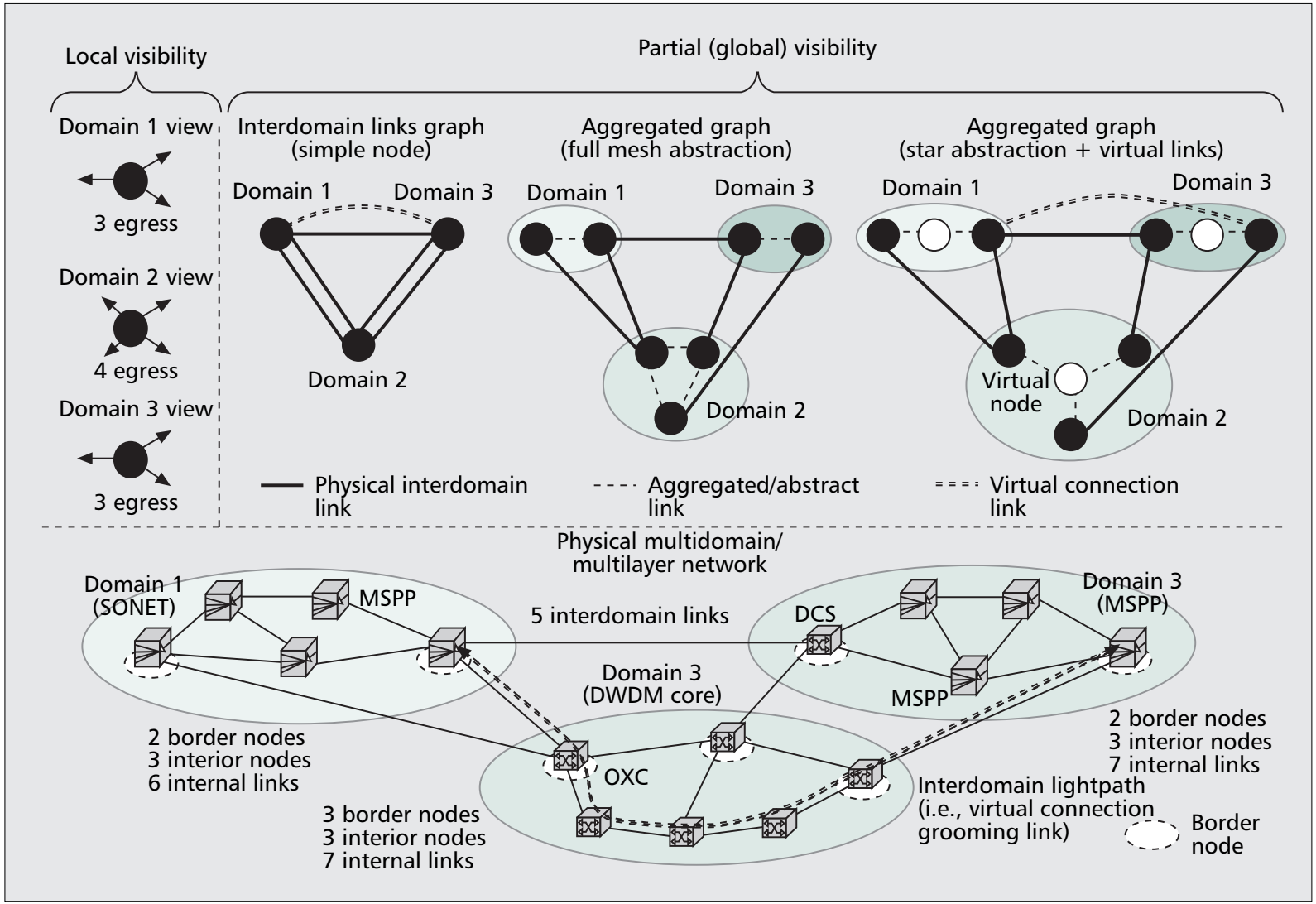


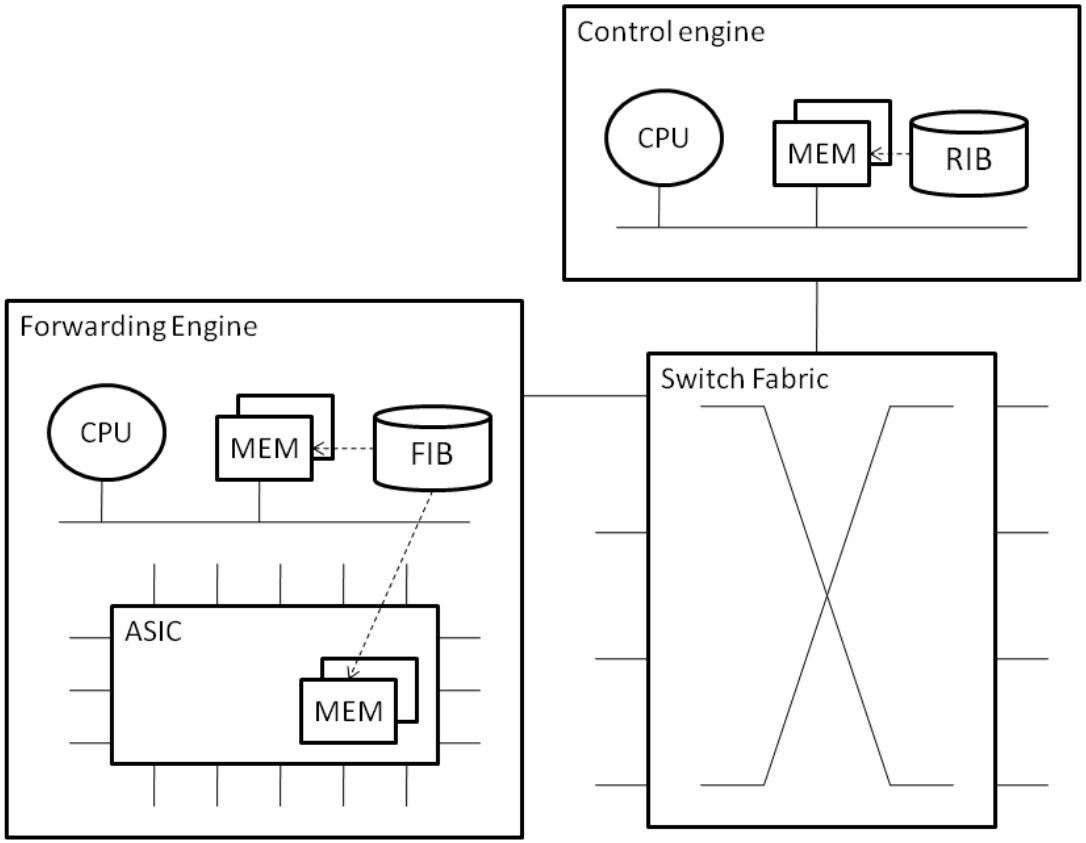
Figure 3. *Interdomain state dissemination models: local, partial.*

- Source: N. Ghani, et al, "Control Plane Design in Multidomain/Multilayer Optical Networks," IEEE Communications Magazine, Vol. 46, No. 6, June 2008, pp. 78-87.

The IP Layer and its Data and Control Plane

Routing and Forwarding

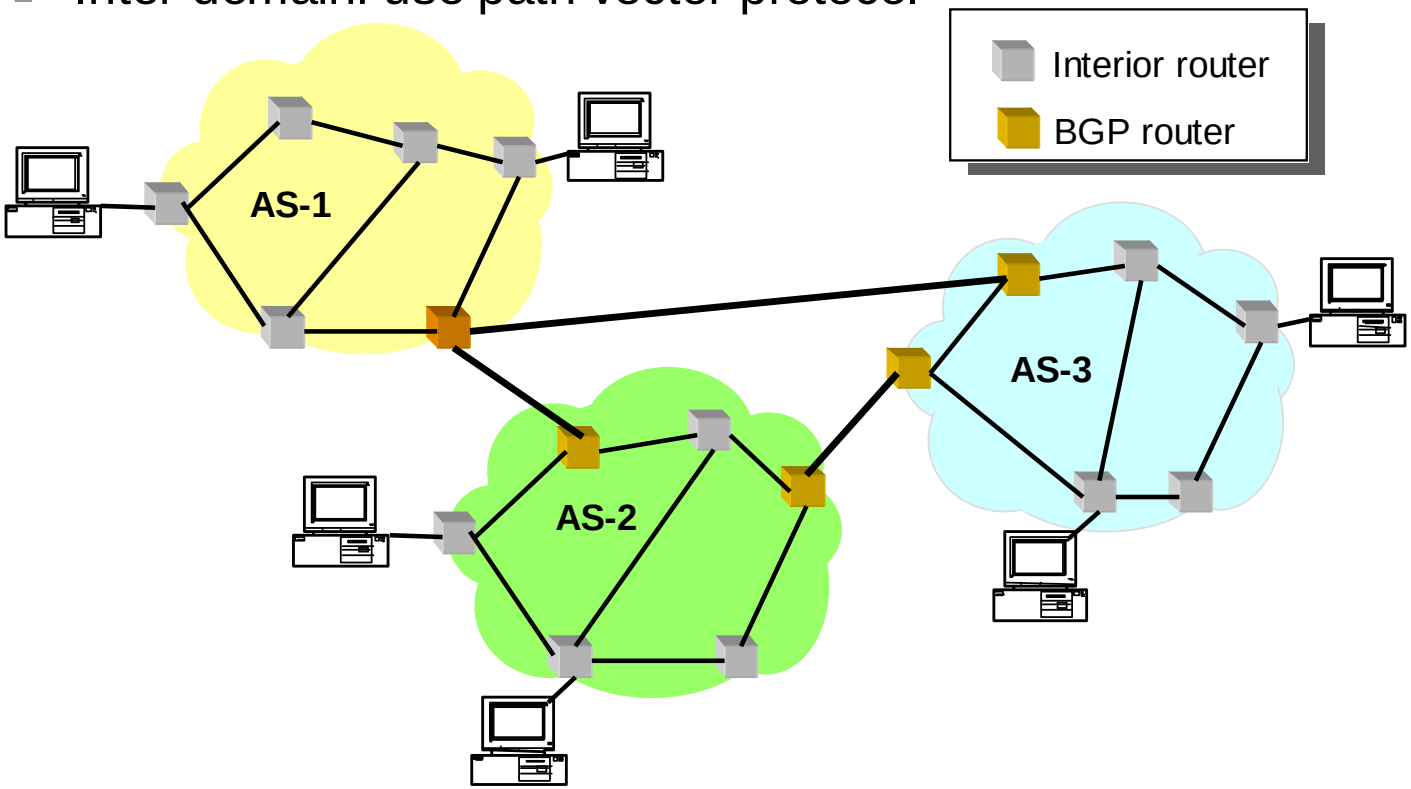
- Router's architecture



- Source: X. Zhao, J. D. Pacella, and J. Schiller, "Routing Scalability: An Operator's View," in IEEE Journal on Selected Areas in Communications, Vol. 28, no. 8, pp. 1262-1270, October 2010.

The intra/inter-domain split

- Intra-domain: use link state or distance vector protocols
- Inter-domain: use path vector protocol



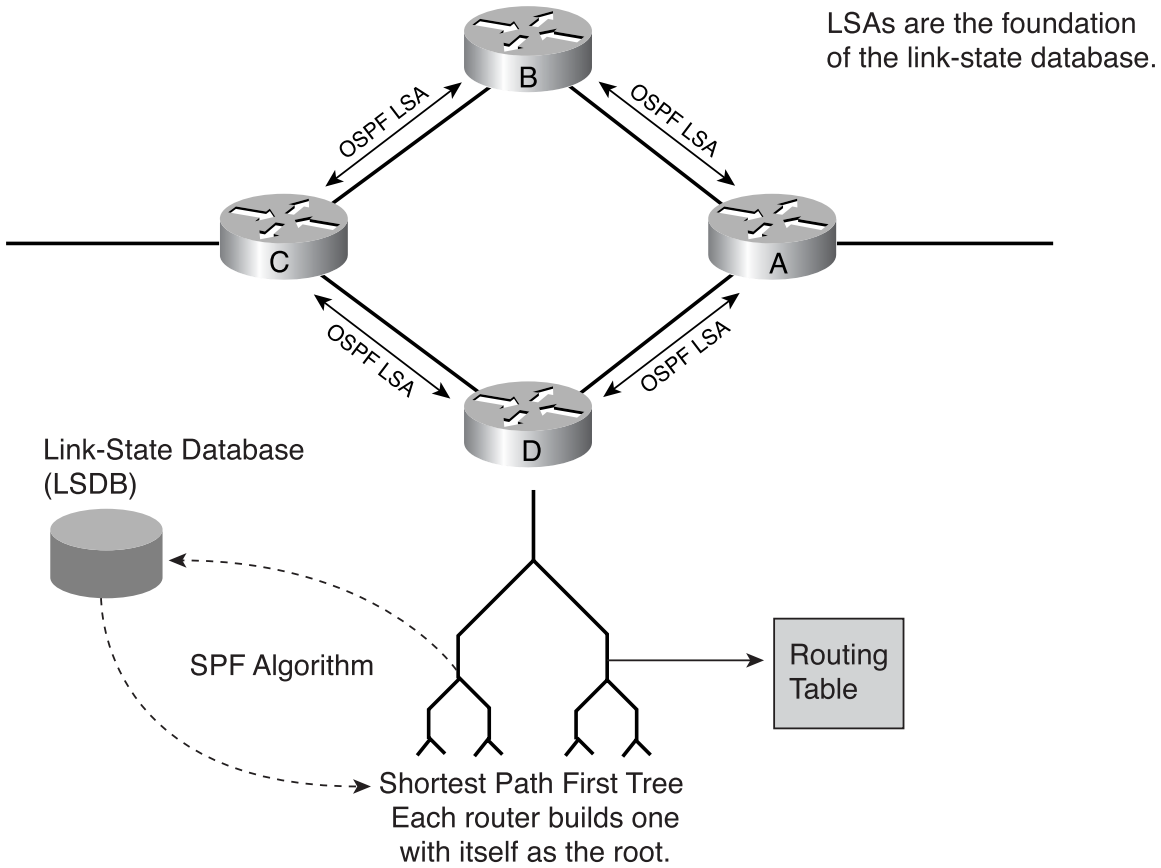
- Source: University of Pennsylvania (CIS)

Two families of intra-domain routing protocols (IGPs)

(1) Link-State Routing

- The principle of link-state routing is that all the routers within an area build a map of the network connectivity in the form of a topological graph.
- The network topology is maintained by each node in a link-state database, and the basic information consists of:
  - Interface identifier
  - Link number
  - Information regarding the state of the link
- The overall goal is that all the nodes in a routing area maintain an identical copy of the network topology.
- Then, each router can independently compute the best path from to every possible destination in the network (Dijkstra's SPF algorithm).
- The collection of best paths will then form the node's routing table.
- Link-state protocols flood all the routing information when they first become active in Link-State Advertisements (LSAs).
- Once the network converges, nodes only exchange incremental updates via LSAs.

(1) Link-State Routing (cont.)



- Source: Thomas M. Thomas, “OSPF Network Design Solutions,” 2nd. Ed., Cisco Press, 2003.

Link-State (LS) Routing Algorithm

Dijkstra's algorithm

- topology and link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node (source) to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k destination nodes

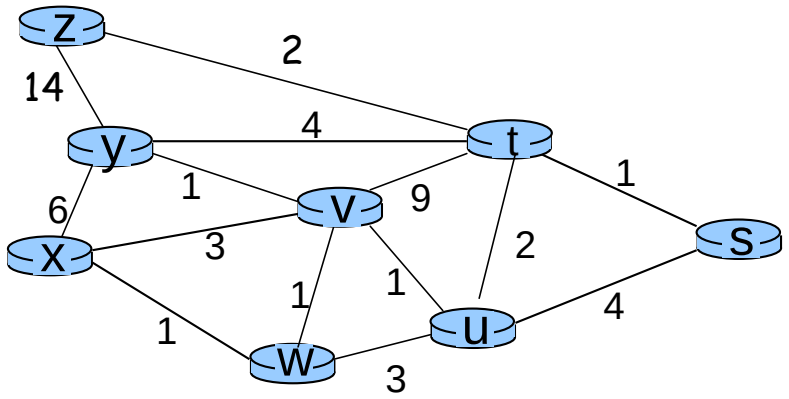
Notation:

- c(x,y): link cost from node x to y; set to ∞ if a and y are not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): v's predecessor node along path from source to v
- N': set of nodes whose least cost path is definitively known

Dijsktra's Algorithm

```
1  Initialization (u = source node):
2    N' = {u}                    /* path to self is all we know */
3    for all nodes v
4      if v adjacent to u
5          then D(v) = c(u,v)    /* assign link cost to neighbours */
6      else D(v) = ∞
7
8  Loop
9      find w not in N' such that D(w) is a minimum
10     add w to N'
11     update D(v) for all v adjacent to w and not in N' :
12        D(v) = min( D(v), D(w) + c(w,v) )
13     /* new cost to v is either old cost to v or known
14      shortest path cost to w plus cost from w to v */
15 until all nodes in N'
```

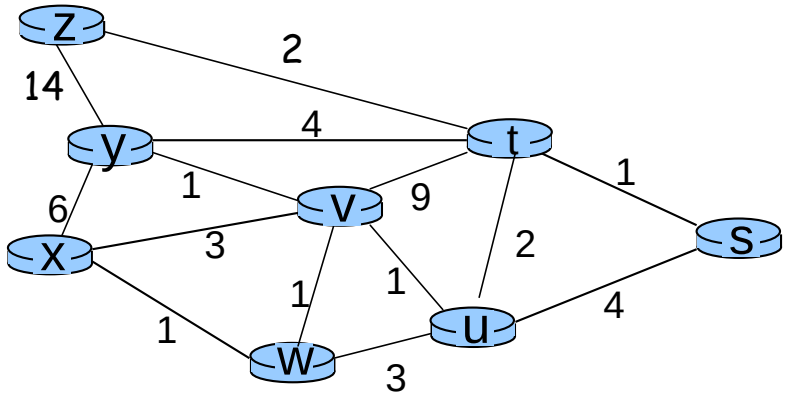
Textbook – Problem 4.21 – x is source



Step	N'	D(s),p(s)	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	∞	3,x	1,x	6,x	∞

Initialization:
- Store source node x in N'
- Assign link cost to neighbours (v,w,y)
- Keep track of predecessor to destination node

Textbook – Problem 4.21 – x is source



Node and its minimum cost are colour-coded in each step

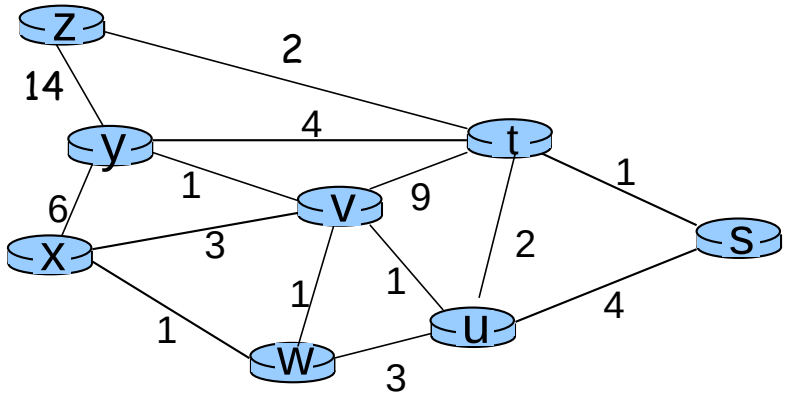
Step	*N'*	*D(s),p(s)*	*D(t),p(t)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	**x**	∞	∞	∞	**3,x**	**1,x**	**6,x**	∞
1	**xw**	∞	∞	**4,w**	**2,w**		**6,x**	∞

Loop – step 1:
- For all nodes not in N', find one that has minimum cost path (1)
- Add this node (w) to N'
- Update cost for all neighbours of added node that are not in N'

repeat until all nodes are in N'

5

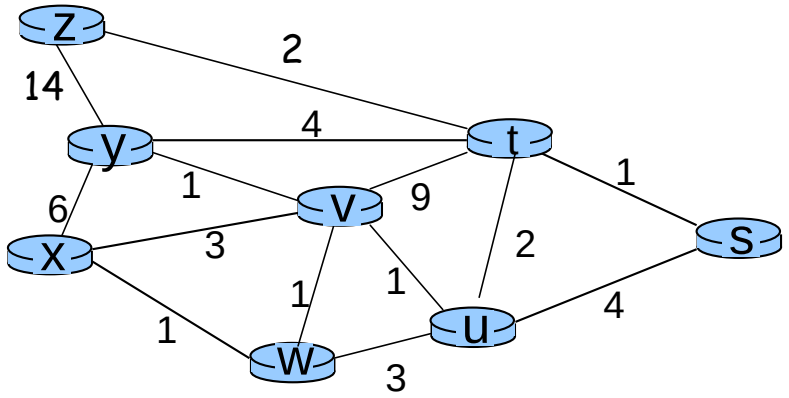
Textbook – Problem 4.21 – x is source



Node and its minimum cost are colour-coded in each step

Step	*N'*	*D(s),p(s)*	*D(t),p(t)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	x	∞	∞	∞	3,x	1,x	6,x	∞
1	xw	∞	∞	4,w	2,w		6,x	∞
2	xwv	∞	11,v	3,v			3,v	∞
3	xwvu	7,u	5,u				3,v	∞
4	xwvuy	7,u	5,u					17,y
5	xwvuyt	6,t						7,t
6	xwvuyts							7,t

Textbook – Problem 4.21 – x is source



We can now build x's forwarding table. E.g. the entry to s will be constructed by looking at predecessors along shortest path: 6,t → 5,u →3,v → 2,w (direct link) So forward to s via w

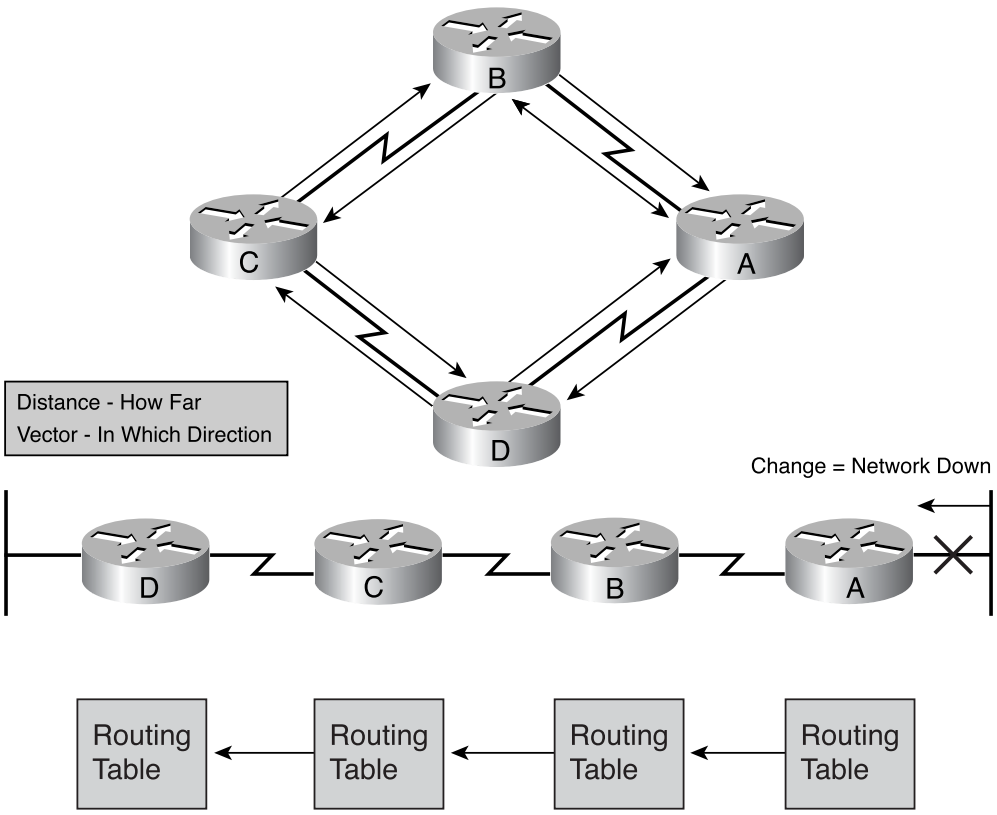
Step	*N'*	*D(s),p(s)*	*D(t),p(t)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	x	∞	∞	∞	3,x	1,x	6,x	∞
1	xw	∞	∞	4,w	2,w		6,x	∞
2	xwv	∞	11,v	3,v			3,v	∞
3	xwvu	7,u	5,u				3,v	∞
4	xwvuy	7,u	5,u					17,y
5	xwvuyt	6,t						7,t
6	xwvuyts							7,t

The second family of intra-domain routing protocols

(2) Distance Vector Routing

- Distance vector means that the information sent from router to router is based on an entry in a routing table that consists of the ***distance*** and ***vector*** to the destination:
  - ***Distance*** being what it “costs” to get there
  - ***Vector*** being the “direction” to get there — direction strictly means the next hop address and exit interface to which packets must be forwarded.
- Distance vector algorithms call for each router to send its entire routing table, but only to its neighbors.
- The neighbor then forwards its entire routing table to its neighbors, and so on.
- Notice that the routers using a distance vector protocol do not have knowledge of the entire path to a destination.

(2) Distance Vector Routing (cont.)



Every router must be told of the change when a network goes down.

- Source: Thomas M. Thomas, “OSPF Network Design Solutions,” 2nd. Ed., Cisco Press, 2003.

(2) Distance Vector Routing (cont.)

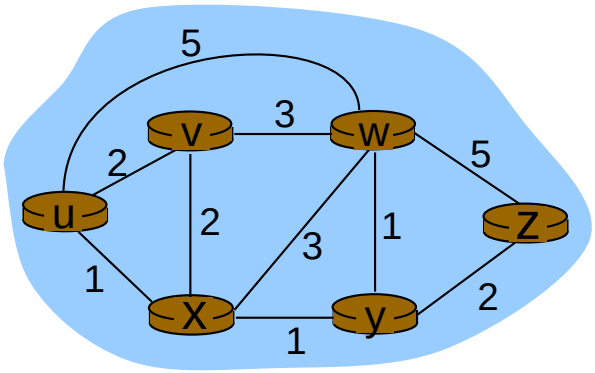
Based on Bellman-Ford distance

- $d_x(y)$: cost of least-cost path from node x to node y
- $c(x, v)$: cost of the direct link from node x to node v
- Then, $\forall$ node v that is a neighbor of node x do

$$d_x(y) = \min_v \{c(x, v) + d_v(y)\}$$

Bellman-Ford Equation Example

Consider a path from *u* to *z*
By inspection, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$



B-F equation says:

$$d_u(z) = \min \{ c(u,v) + d_v(z), \; c(u,x) + d_x(z), \; c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

Node that achieves minimum is next hop in shortest path ➜ entry in forwarding table

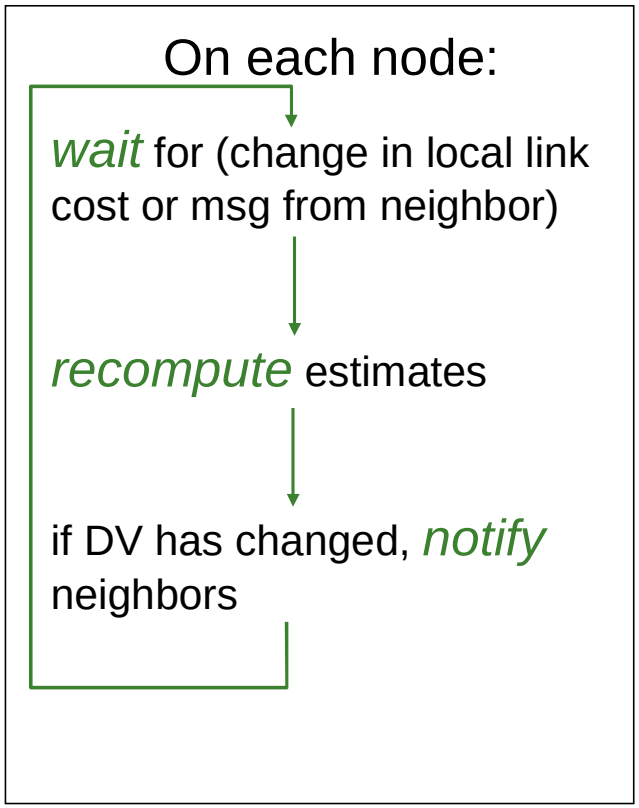
Distance Vector Algorithm

Basic idea:

- Nodes keep vector (DV) of least costs to other nodes
  - These are estimates, $D_x(y)$
- Each node periodically sends its own DV to neighbors
- When node *x* receives DV from neighbor, it keeps it and updates its own DV using B-F:

  $D_x(y) \leftarrow min_v\{c(x,v) + D_v(y)\}$

  for each node $y \in N$
- Ideally, the estimate $D_x(y)$ *converges to the actual least cost* $d_x(y)$



(2) Distance Vector Routing (Source: U. of Calgary, CPSC 441)

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

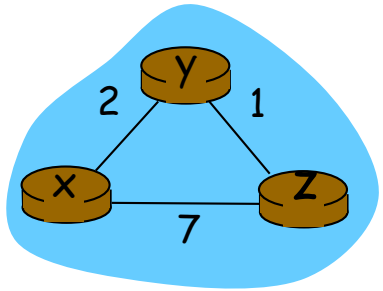
node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

Step 1: Initialization

Initialize costs of direct links

Set to ∞ costs from neighbours



time

(2) Distance Vector Routing (Source: U. of Calgary, CPSC 441)

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

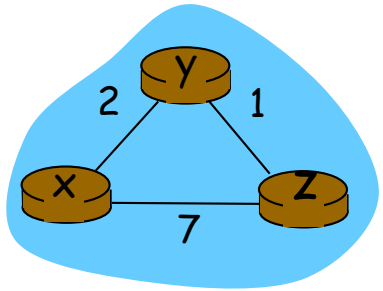
from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

Step 2: Exchange DV and iterate

-In first iteration, node x saves neighbours' DVs

-Then, it checks path costs to all nodes using received DVs

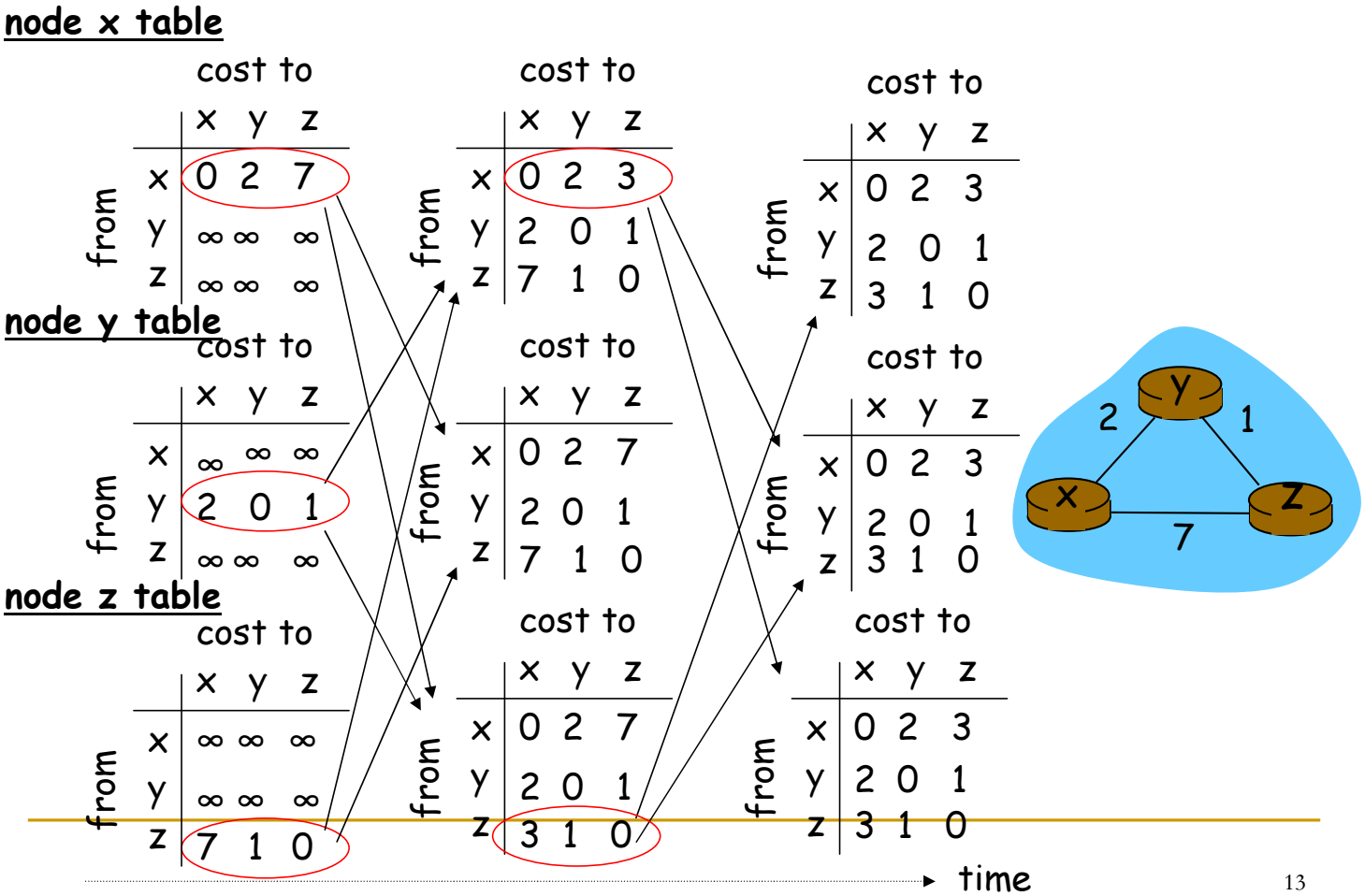
-E.g. new cost $D_x(z)$ is obtained by adding costs marked red



time

(2) Distance Vector Routing (Source: U. of Calgary, CPSC 441)

In similar fashion, algorithm proceeds until all nodes have updated tables



One family of inter-domain routing protocols (EGPs)

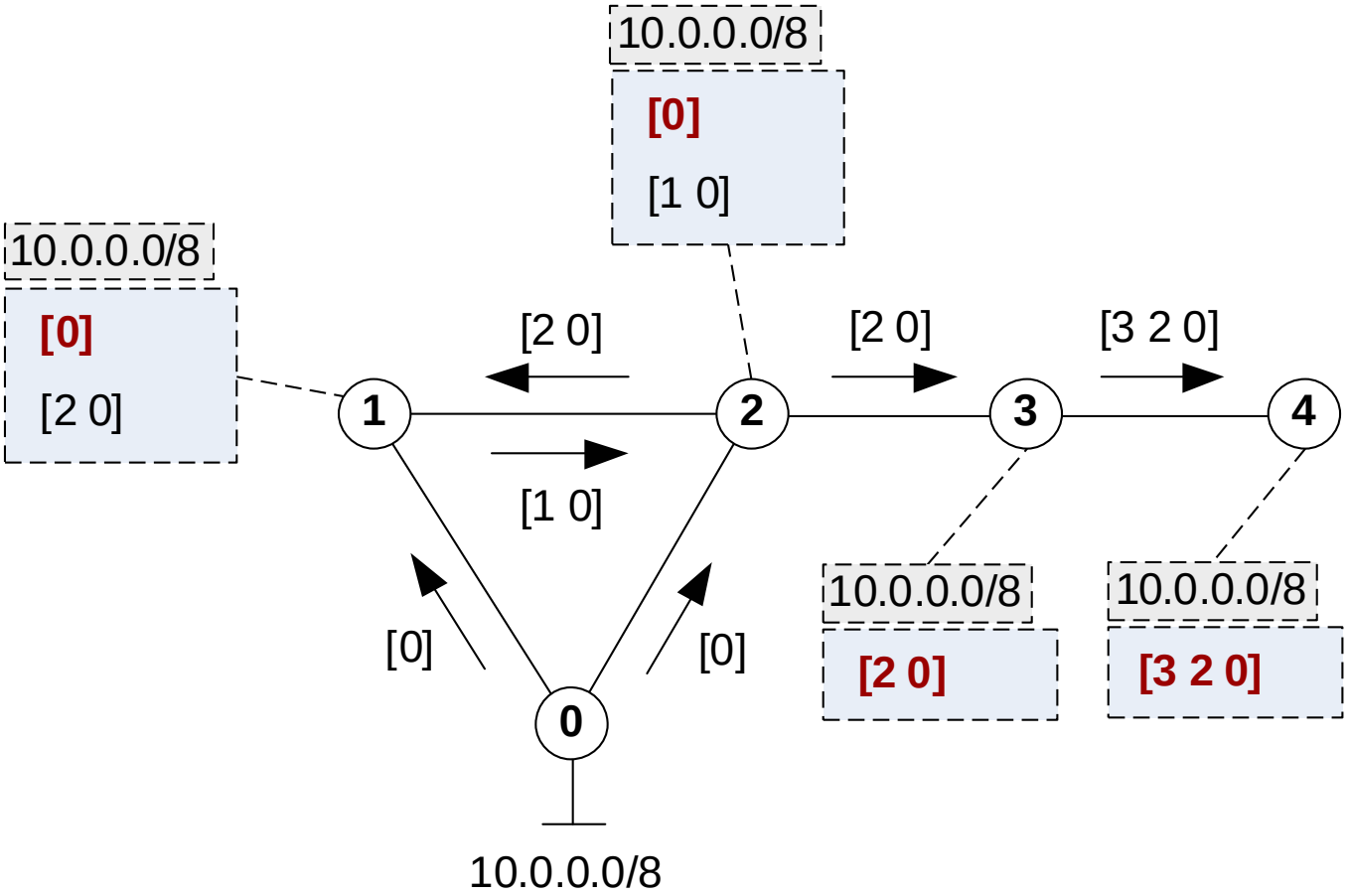
(3) Path Vector Routing: the basics

- For scalability and confidentiality reasons, the routing information managed and exchanged among ASs is highly condensed.
- Differently from link-state routing protocols, which maintain the topological state of the network, path routing protocols only handle AS-level paths for any possible destination.
- An AS-level path is composed of a set of attributes, including an ordered sequence of AS numbers (a vector of ASs) that need to be traversed to reach a destination. This routing paradigm is thus called *path vector routing*.

The two main goals of path vector routing

- To distribute reachability information among domains in a **"highly scalable way"**.
- To find loop-free paths among domains.

(3) Path Vector Routing



(3) Path Vector Routing (cont.)

Similarities between distance and path Vector Routing

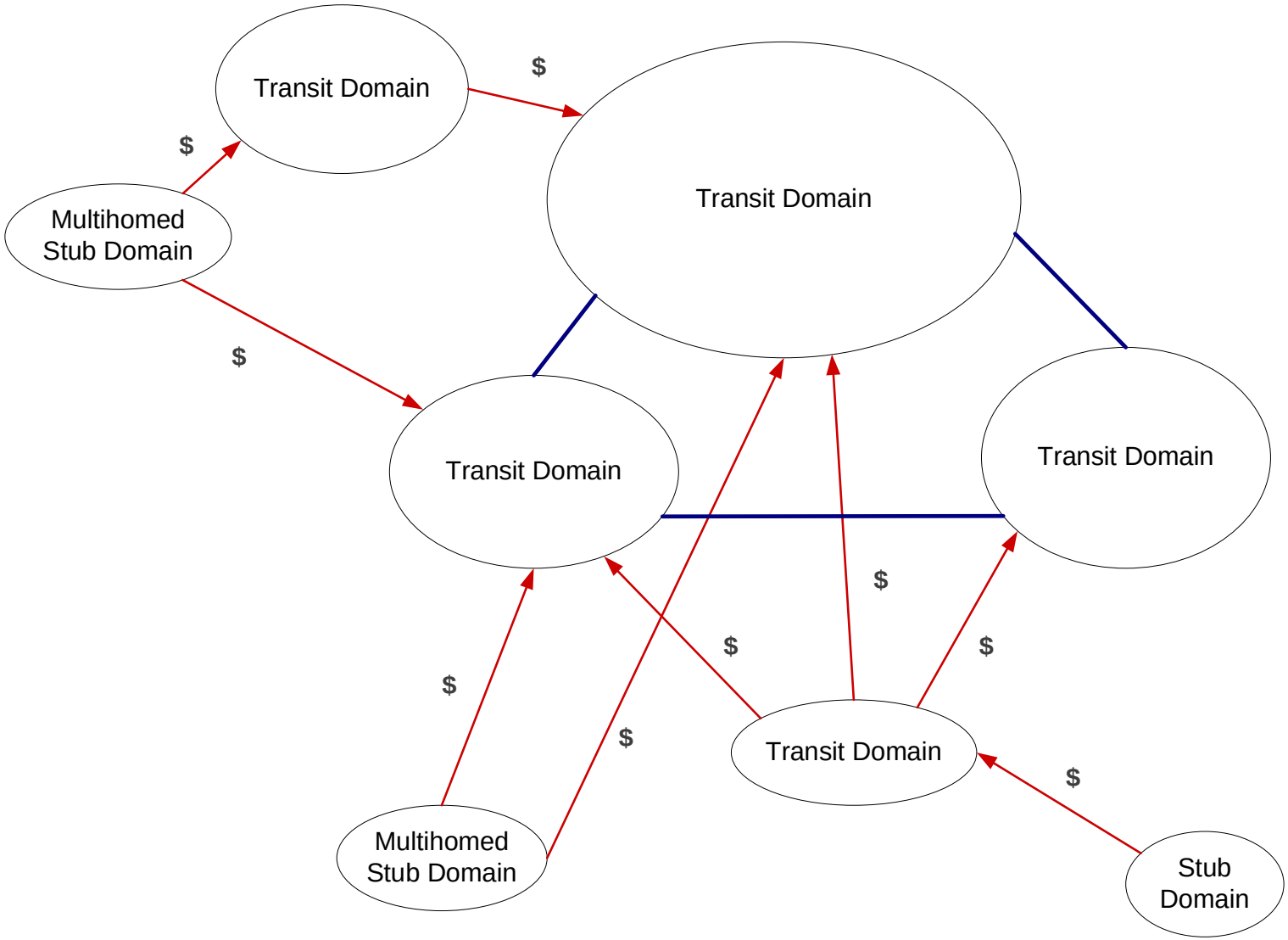
- Distance vector protocols choose routes according to the shortest distance to a destination (e.g., the least number of routers to be traversed)
- Path vector protocols will generally choose the route that traverses the least number of ASs
- The term “generally” is because the AS-path length (a rough sense of distance) is the attribute that is typically considered during the route selection process, but is not the only one.
- Route selection in path vector protocols is much more complex than in distance vector routing.
- Path vector routers can filter routes based on multiple and elaborated criteria.
- They can change the preference of a route and override the AS hop count, and even change the attributes of the routes they use and advertise to other devices based on commercial interests and the policies locally configured on each router.
- The combination of these features allows domains to enforce their routing policies, enabling control over their traffic according to their criteria.

Relationships between ASs **(Myth)**

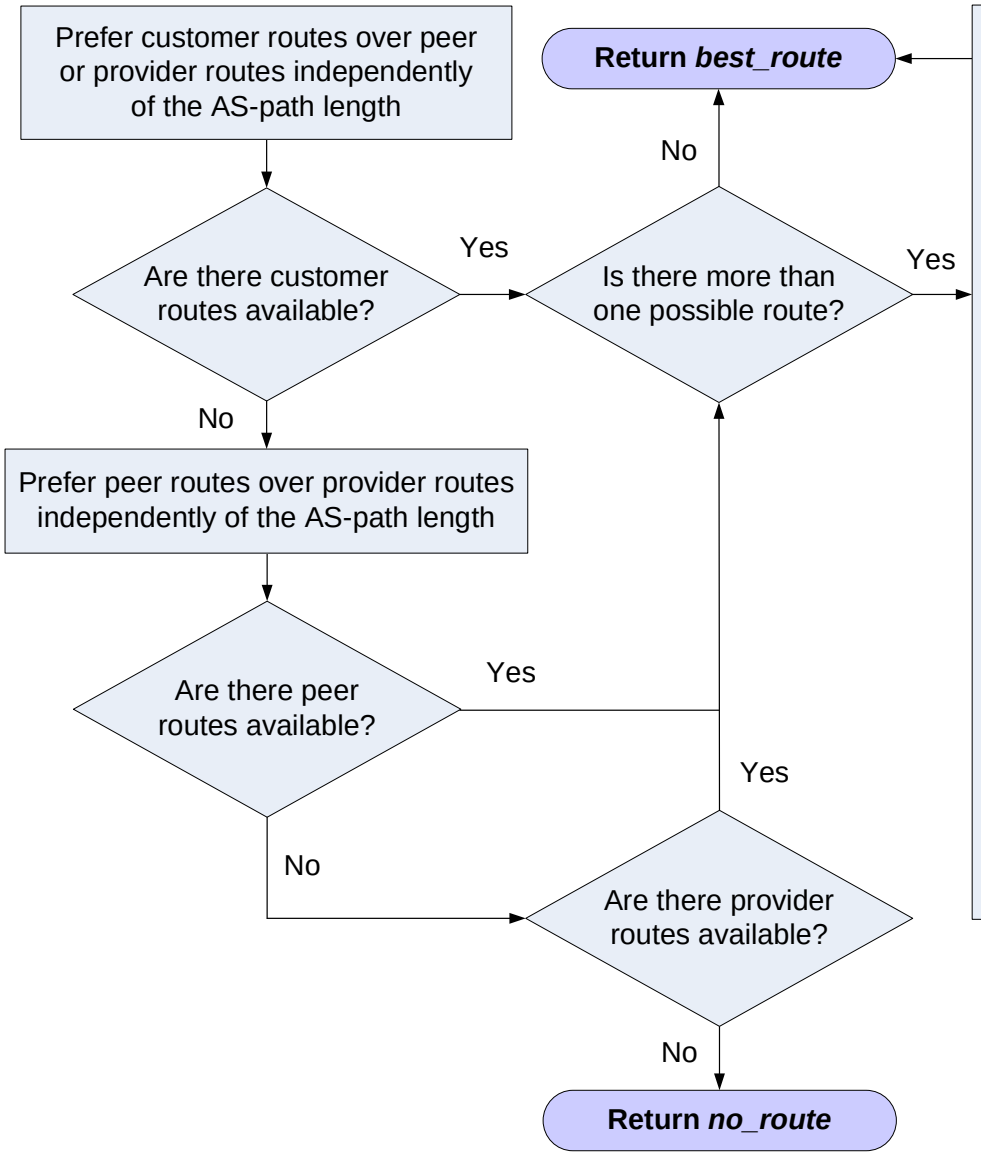
There are three types of relationships ...

- ... which correspond to the different traffic exchange agreements between neighboring domains
- ***customer-provider***: applies when a domain buys Internet connectivity from a provider.
- ***peer-peer***: applies when two providers that exchange a significant amount of traffic, agree to connect directly to each other to avoid transiting through, and thus pay, a third-party provider. Peers share the costs of the connection between them, so there is no customer-provider relationship in this case.
- ***sibling-sibling***: this relationship is quite infrequent, and are sometimes used between merging companies. According to data from CAIDA's AS Relationships Dataset, less than 0.3% of the total number of relationships between Internet domains were siblings in March of 2010.

Tiered Hierarchy of Autonomous Systems (Myth)



(3) Path Vector Routing: path selection process



Decision Process of a Path Vector Protocol

Input: A set of candidate routes to reach *d*
Output: The best route to reach *d*

1: Choose the route with the highest preference according to the outbound policy locally configured

2: If the outbound preferences are equal, choose the route with the shortest AS-path (i.e., the shortest path vector)

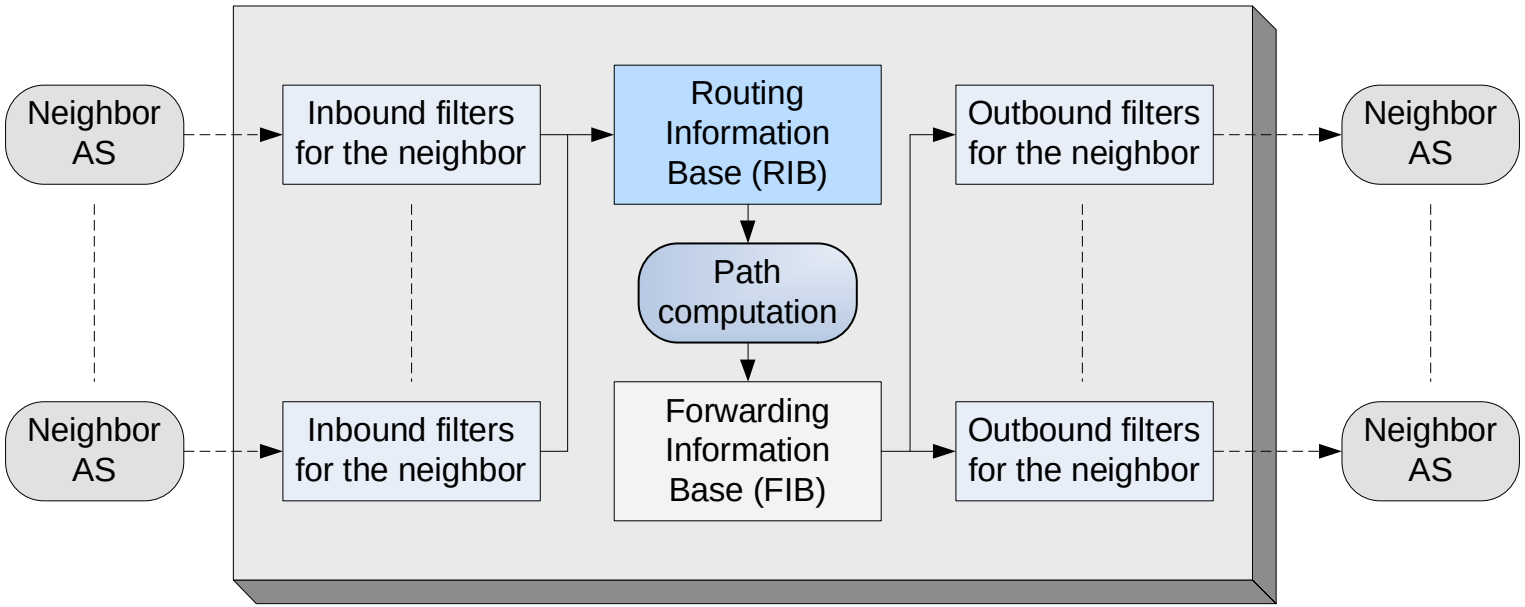
3: If the AS-paths lengths are equal, choose the route with the highest preference according to the inbound traffic policies advertised by the neighbors

4: If the neighbors' inbound preferences are equal, prefer external routes over internal routes

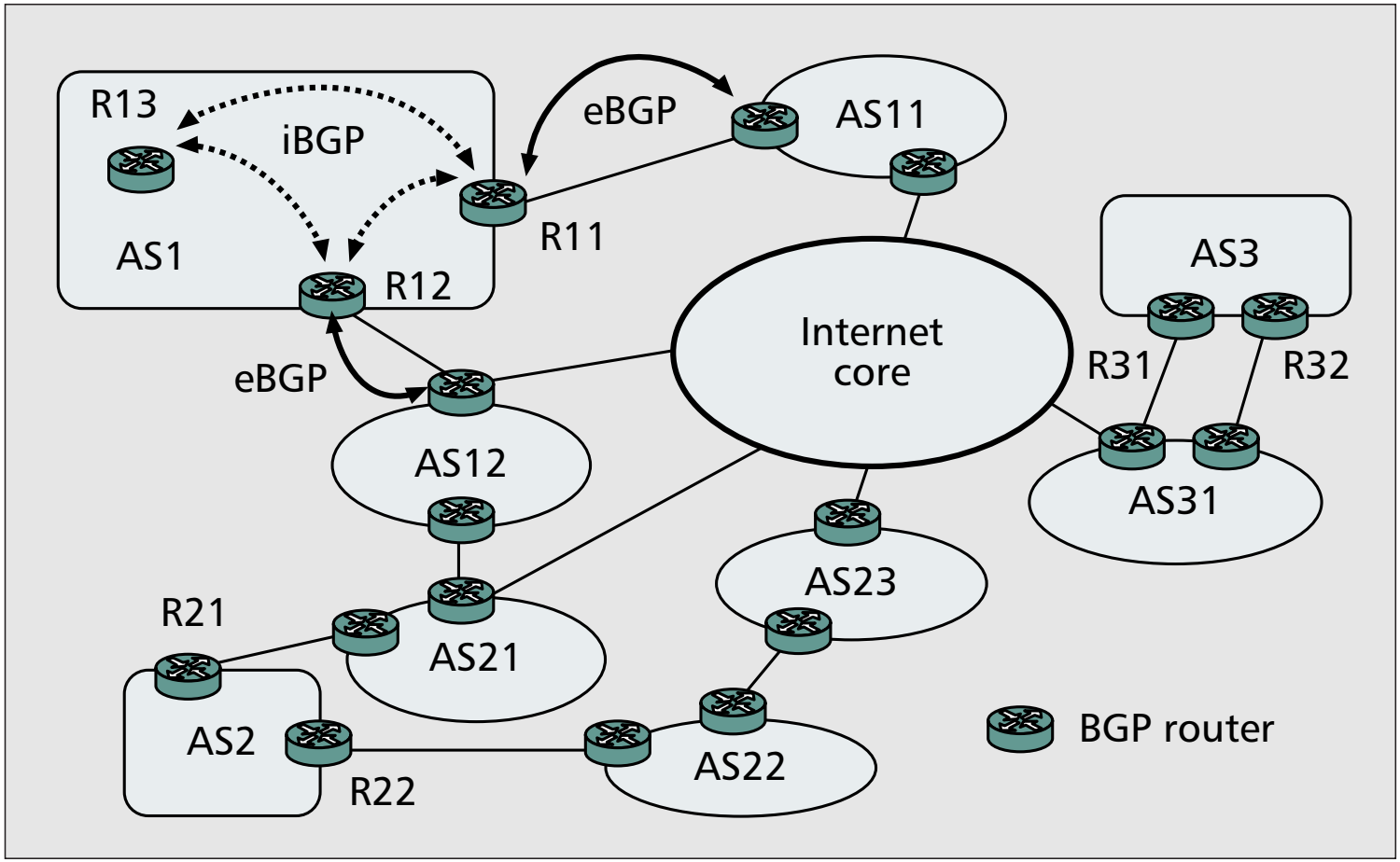
5: If the routes are still equally preferred, choose the one with the lowest metric to the next-hop router according to the internal routing cost

6. Run tie-breaking rules

(3) Path Vector Routing: inside the router ...

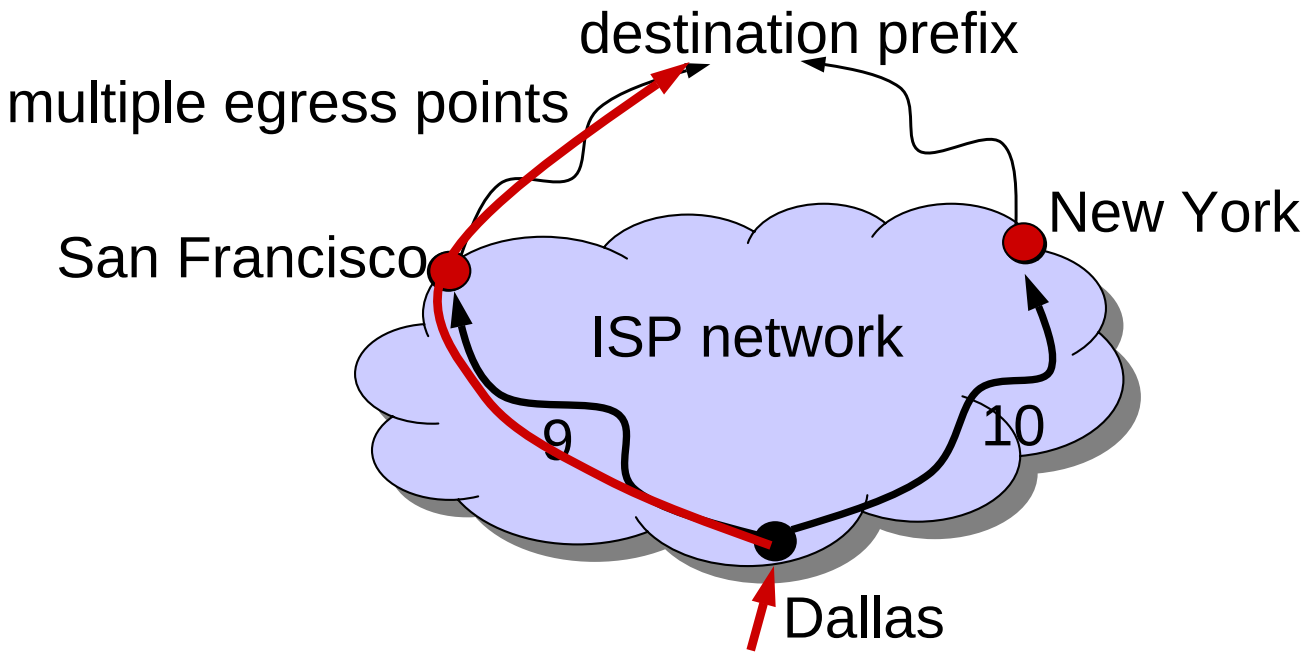


The Border Gateway Protocol (BGP): RFC 4271, 2006

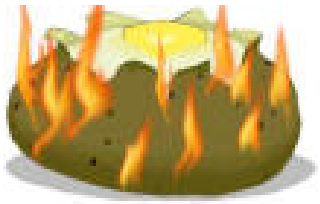


- eBGP: External Border Gateway Protocol
- iBGP Internal Border Gateway Protocol

The intra/inter-domain merge: Transit Providers

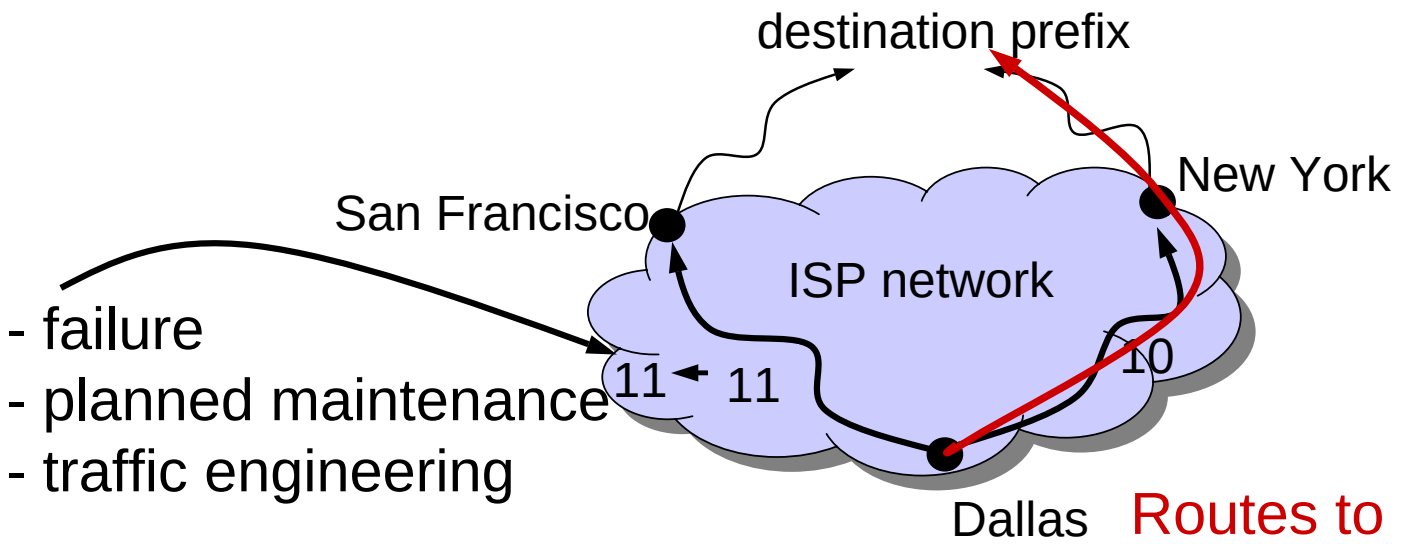


Hot-potato routing = select closest egress point when there is more than one route to destination



- Source: R. Texeira, “Hot Potatoes Heat Up BGP Routing,” RIPE 51, Amsterdam, Netherlands, October 2005.

The intra/inter-domain merge: Transit Providers



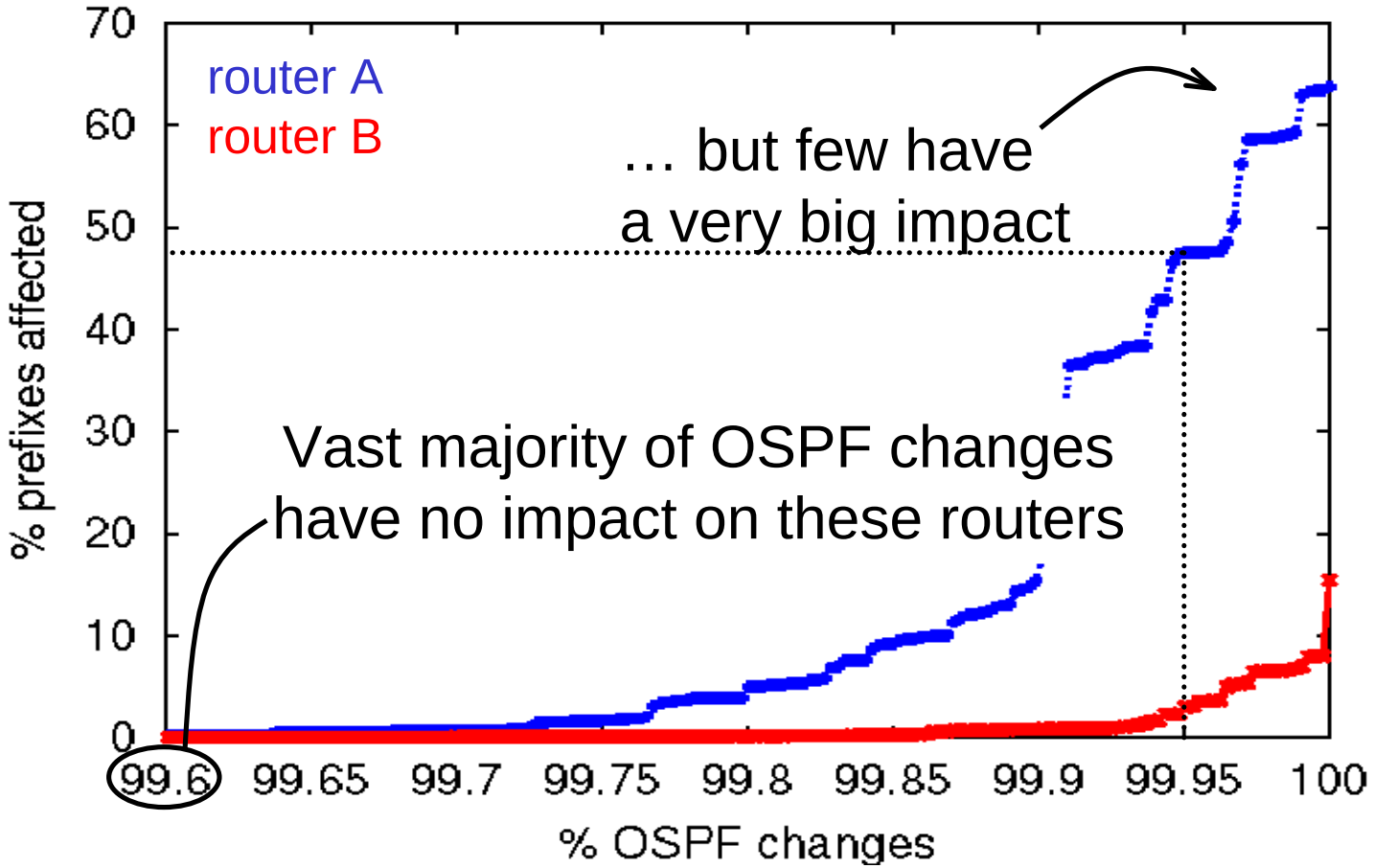
Routes to thousands of prefixes switch egress points!!!

Consequences:

- Transient forwarding instability
- Traffic shift
- Interdomain routing changes

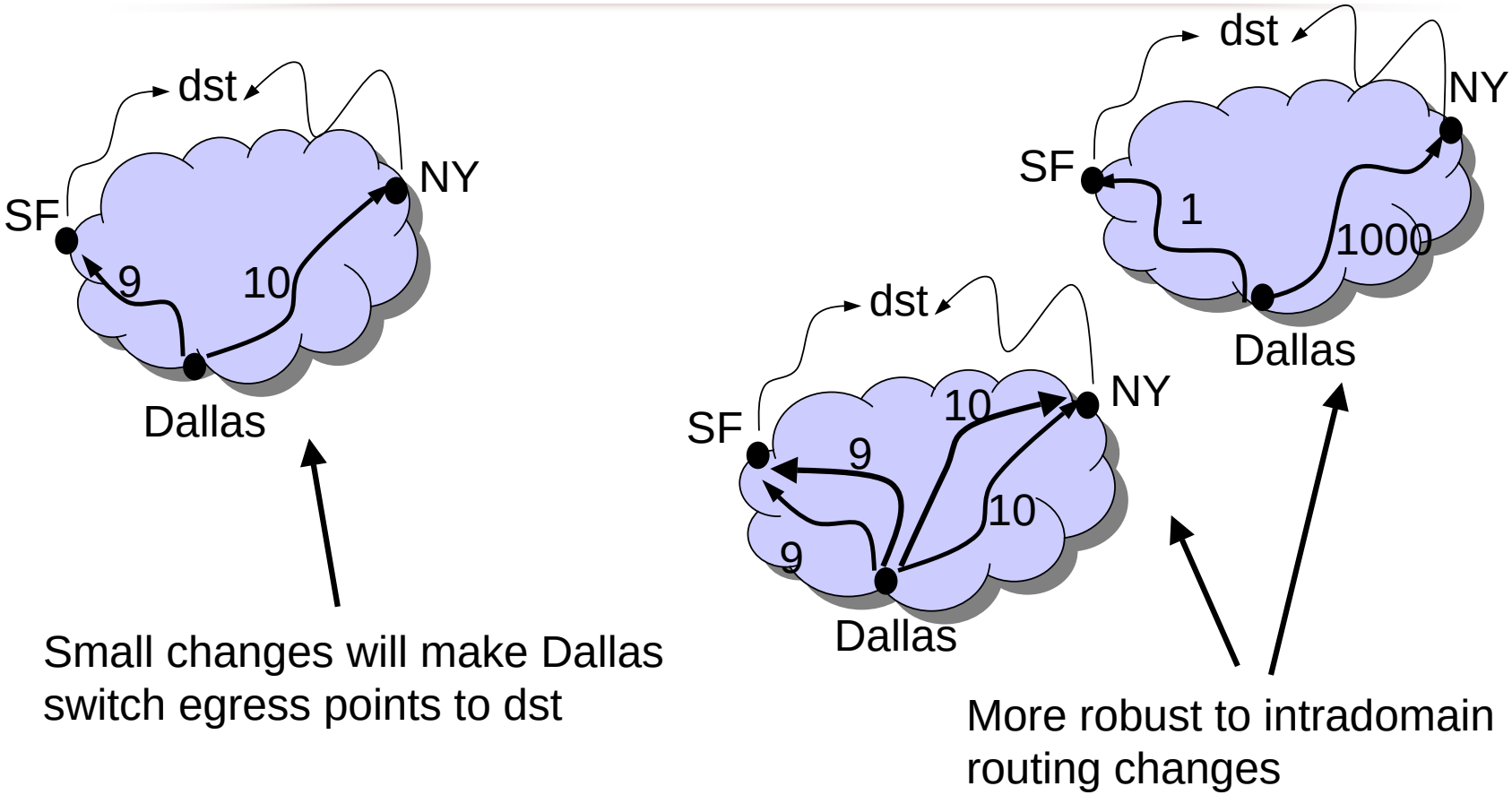
- Source: R. Texeira, “Hot Potatoes Heat Up BGP Routing,” RIPE 51, Amsterdam, Netherlands, October 2005.

The intra/inter-domain merge: Transit Providers



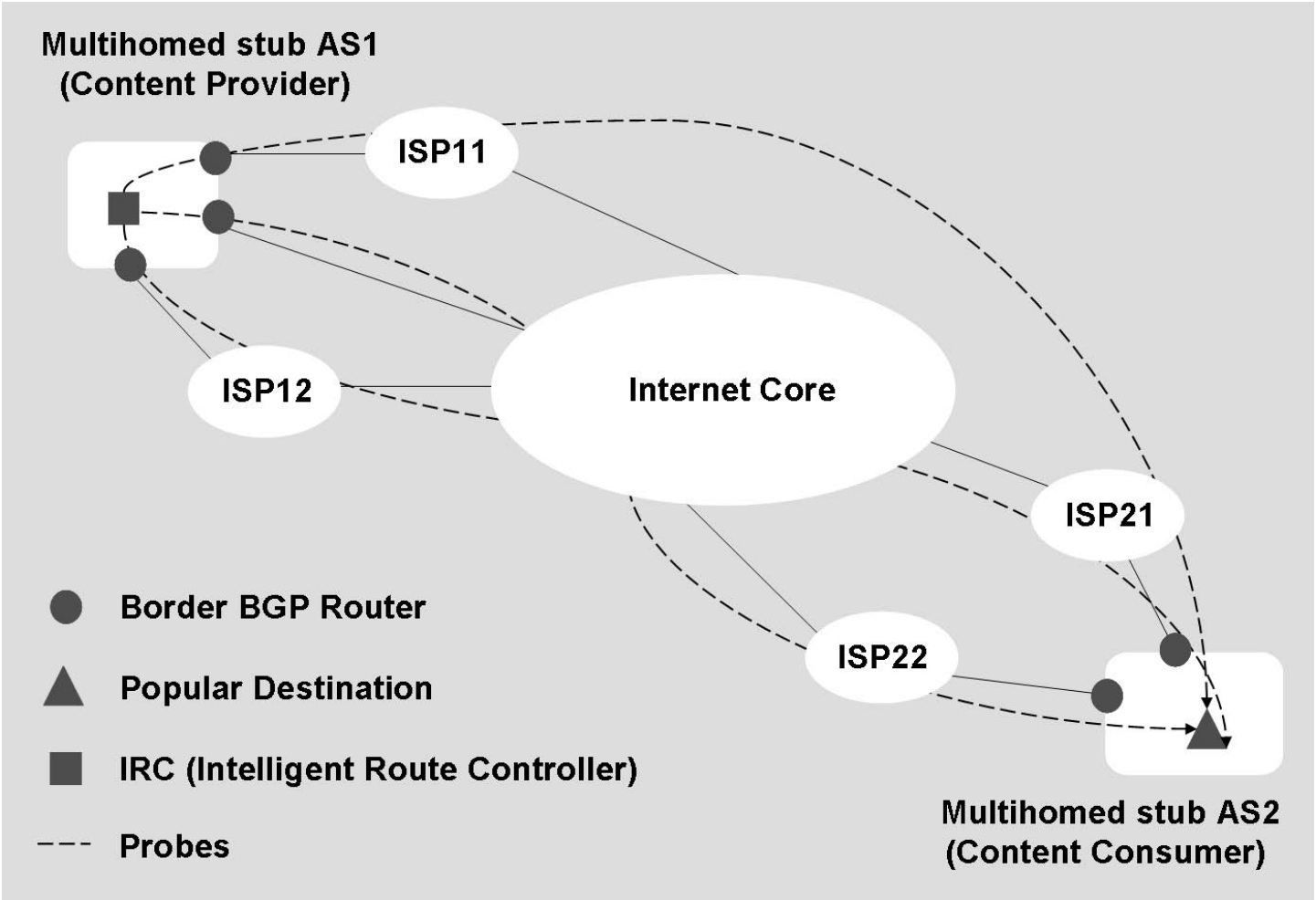
- Source: R. Texeira, “Hot Potatoes Heat Up BGP Routing,” RIPE 51, Amsterdam, Netherlands, October 2005.

The intra/inter-domain merge: Transit Providers



- Source: R. Texeira, “Hot Potatoes Heat Up BGP Routing,” RIPE 51, Amsterdam, Netherlands, October 2005.

The intra/inter-domain merge: Non-transit domains



Questions?