Computer Networks : Protocols and Practice

Part 4 : Network Layer

Olivier Bonaventure
http://inl.info.ucl.ac.be/

© O. Bonaventure 2008

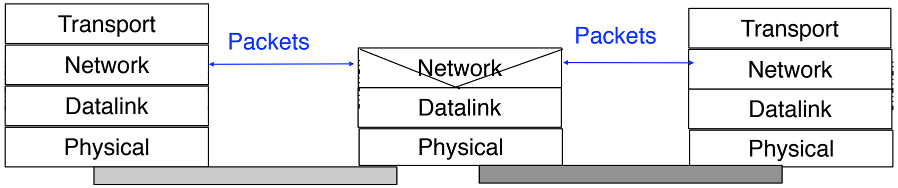
These slides are licensed under the creative commons attribution share-alike license 3.0. You can obtain detailed information
about this license at http://creativecommons.org/licenses/by-sa/3.0/

Network layer

- Basics
  - Datagram mode
  - Virtual circuits
- Routing
- IP : Internet Protocol
- Routing in IP networks

© O. Bonaventure, 2007

The network layer



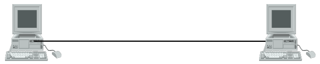
- Goal
  - Allow packets to be forwarded from any source to any destination through heterogeneous networks and routers

- Services
  - Unreliable connectionless service
  - Reliable connection-oriented service

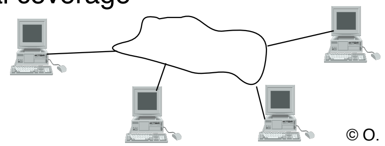
© O. Bonaventure, 2007

Two types of datalink layers

- WAN type datalink layer
  - PPP, HDLC
    - Reliable exchange of frames between two hosts attached to the same “link”
    - Mainly used by wide area networks



- LAN type datalink layer
  - Ethernet, Token Ring, FDDI, WiFi, Wimax,
    - Exchange of frames between hosts attached to the same LAN
    - limited geographical coverage



CNPP/2008.4.

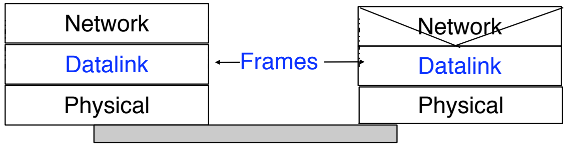
© O. Bonaventure, 2007

PPP is defined in several documents, including :

RFC1661 The Point-to-Point Protocol (PPP). W. Simpson, Ed.. July 1994

Local area networks are described in part 5.

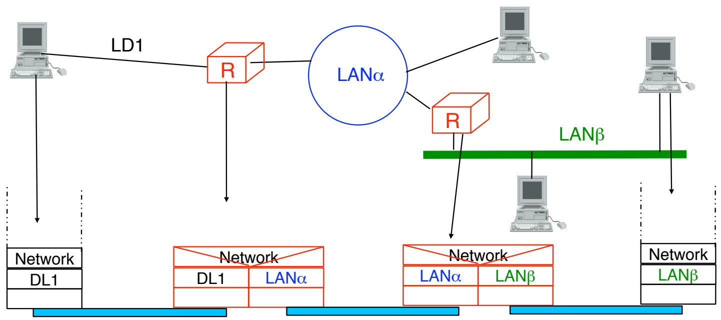
The datalink service



- Service of datalink layer
  - Unreliable connectionless service
    - Transmission of frames between hosts directly connected at the physical layer or directly attached to the same LAN
    - Unreliable transmission (frames can be lost but usually transmission errors are detected)
    - Most datalink layers have maximum frame length
  - Connection-oriented service, reliable or not
    - Transmission of frames between hosts directly connected at the physical layer or directly attached to the same LAN
    - Reliable or unreliable transmission

© O. Bonaventure, 2007

Routers



- Router
  - Relay within the network layer
- packet is unit of transmission

© O. Bonaventure, 2007

Network layer
Basic principles

- Each host/router must be identified by a network layer address which is independent from its datalink layer address
- Network layer forwards packets from source to destination through multiple routers
- Network layer service must be completely independent from the service provided by the datalink layer
- Network layer user should not need to know anything about the internal structure of the network layer to be able to send packets

© O. Bonaventure, 2007

Internal organisation of the network layer

- Tow possible organisations
  - datagrams
  - virtual circuits

- The internal organisation of the network is orthogonal to the service provided, but often
  - datagram mode is used to provide a connectionless service
  - virtual circuits are used to provide a connection-oriented service

© O. Bonaventure, 2007

Datagram transmission mode

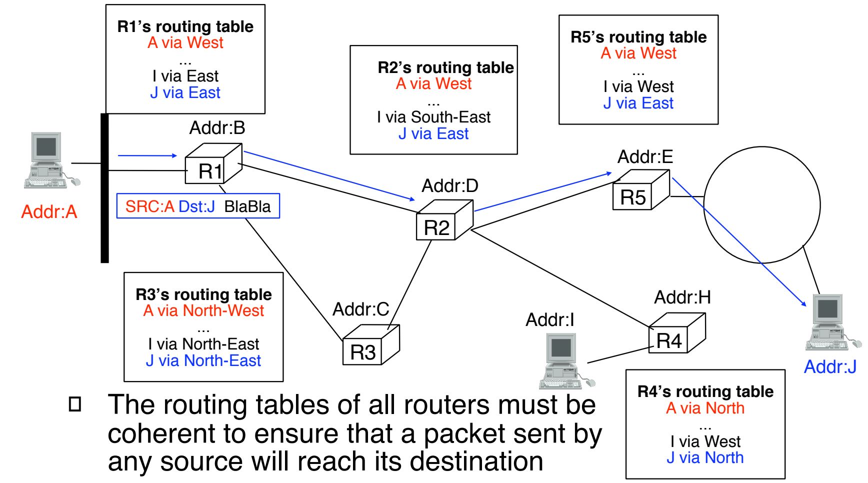
- Basics
  - Each route/host is identified by an address
  - Information is divided in packets
  - Each packet contains
    - Source address
    - Destination address
    - Payload
  - Router behavior
    - Upon packet arrival look at destination address and routing table to decide where the packet should be forwarded
      - hop-by-hop forwarding, each routers takes a forwarding decision
- Examples
  - IP (IPv4 and IPv6)
  - CLNP
  - IPX

CNPP/2008.4.

© O. Bonaventure, 2007

- The datagram mode is used in other networking protocols
  - Internet Protocol
  - IPX (used by Novell)
  - ConnectionLess Network Protocol (CLNP), developed by ISO and used in some networks

Datagram transmission mode (2)



- The routing tables of all routers must be coherent to ensure that a packet sent by any source will reach its destination

© O. Bonaventure, 2007

Virtual circuit organisation

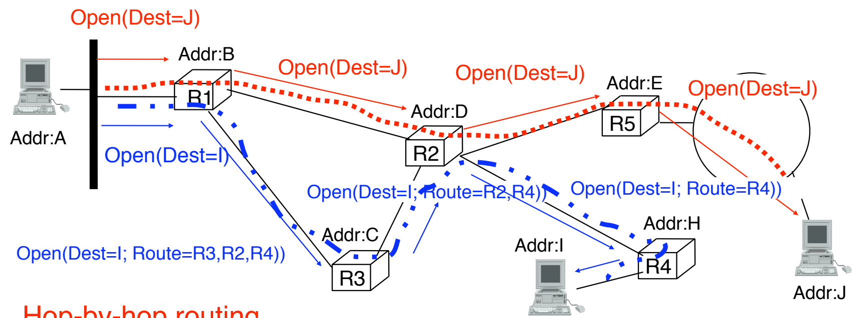
- Goals
  - Keep forwarding on the routers as simple as possible
    - consulting a routing table for each packet is costly from a performance viewpoint
- Solution
  - Before transmitting packets containing data, create a virtual circuit that links source and destination through the network
    - During the virtual circuit establishment, efficient datastructures are updated on each transit router to simplify forwarding
  - Use the virtual circuits to forward the packets
    - All packets will follow the same path
- Example
  - ATM, X.25, Frame Relay, MPLS, gMPLS

© O. Bonaventure, 2007

Virtual circuits are used by several networking technologies, including :
- Asynchronous Transfer Mode (ATM)
- Frame Relay

MultiProtocol Label Switching (MPLS) is a way to integrate virtual circuits with IP. It will be described later.

Establishment of a virtual circuit



Hop-by-hop routing

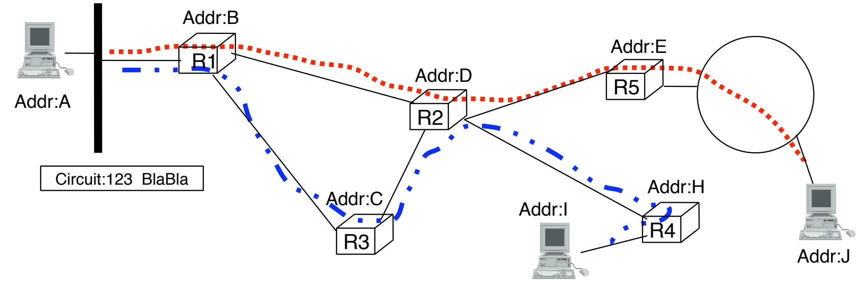
Each router consults its routing table to forward vc establishment

Source routing/ explicit routing

Source (or first hop router) indicates in vc establishment packet the path to be followed

© O. Bonaventure, 2007

Packet transmission



- Packet contents
  - virtual circuit identifier
  - packet payload
- What kind of virtual circuit identifier
  - Naive solution
    - unique identifier for all virtual circuits inside network
      - How to coordinate allocation of vc identifiers ?

© O. Bonaventure, 2007

Packet transmission (2)

- How unique should virtual circuits identifiers be ?
  - globally unique
    - unrealistic
  - unique inside a given network
    - then coordination among routers is necessary
  - unique on a given link
    - easier to manage, no coordination required, but
    - virtual circuit identifier may need to be changed from link to link

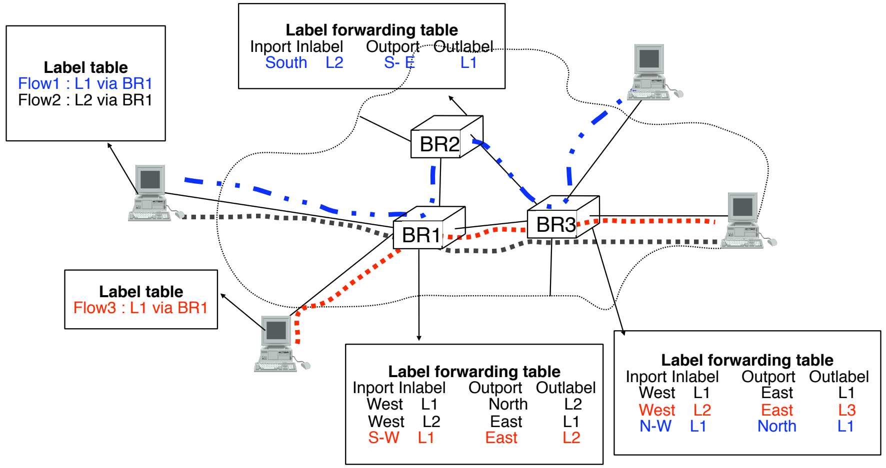
- How to update the virtual circuit identifier of packets
  - All routers must contain a label forwarding table
    - this table is updated every time a virtual circuit is established

Label forwarding table

Inport	Inlabel	Outport	Outlabel
West	L1	East	L1
West	L2	East	L3
N-W	L1	North	L1

© O. Bonaventure, 2007

Virtual circuits : example



© O. Bonaventure, 2007

Network layer

- Basics
- Routing
  - Static routing
  - Distance vector routing
  - Link state routing
- IP : Internet Protocol
- Routing in IP networks

© O. Bonaventure, 2007

Routing and Forwarding

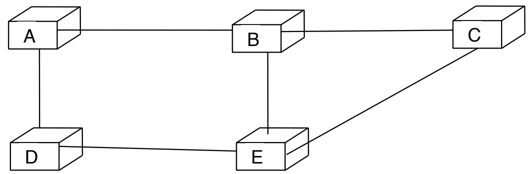
- Main objective of network layer
  - transport packets form source to destination

- Two mechanisms are used in network layer
  - forwarding
    - algorithm use by each router to determine on which interface each packet should be sent to reach its destination or follow its virtual circuit
      - relies on the routing table maintained by each router
  - routing
    - algorithm (usually distributed) that distributes to all routers the information that allows them to build their routing tables

© O. Bonaventure, 2007

Routing (2)

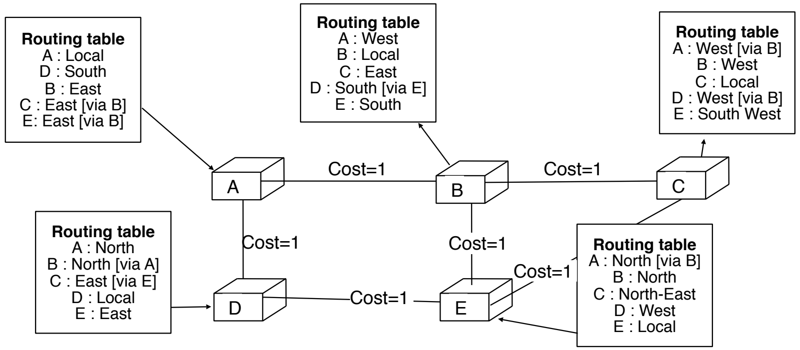
- How to build the routing tables of each router ?



- Principle
  - Include in the routing table of each router the path to allow it to reach each destination
    - Which path to be included in the routing table
      - From A to C ?
      - From D to B

© O. Bonaventure, 2007

Selection of the shortest paths



- Principle
  - Associate a weight/cost to each link
  - Each router chooses the lowest cost path
    - How to ensure that the routing tables of all routers are coherent ?

© O. Bonaventure, 2007

Static Routing

- Principle
  - Network manager or network management station computes all routing tables and downloads them on all routers
  - How to compute routing tables ?
    - shortest path algorithms
    - more complex algorithms to provide load balancing or traffic engineering
  - Advantages of static routing
    - Easy to use in a small network
    - routing tables can be optimised
  - Drawbacks of static routing
    - does not adapt dynamically to network load
    - how to deal with link and router failures ?

© O. Bonaventure, 2007

Dynamic or distributed routing

- Principle
  - routers exchange messages and use a distributed algorithm to build their routing tables
  - used in almost all networks

- Advantages
  - can easily adapt routing tables to events
- Drawbacks
  - more complex to implement than static routing

- Most common distributed routing methods
  - Distance vector routing
  - Link state routing

© O. Bonaventure, 2007

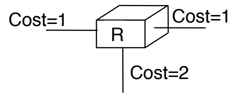
Network layer

- Basics
- Routing
  - Static routing
  - Distance vector routing
  - Link state routing
- IP : Internet Protocol
- Routing in IP networks

© O. Bonaventure, 2007

Distance vector routing

- Basic principles
  - Configuration of each router
    - Cost of each link



  - When it boots, a router only knows itself
  - Each router sends periodically to all its neighbours a vector that contains <u>for each destination that it knows</u>
    1. Destination address
    2. Distance between transmitting router and destination
       - distance vector is a summary of the router's routing table
  - Each router will update its routing table based on the information received from its neighbours

© O. Bonaventure, 2007

Distance vector routing (2)

- Routing table *maintained by router*
  - For each destination d inside routing table
    - R[d].cost = total cost of shortest path to reach d
    - R[d].link = outgoing link used to reach d via shortest path
- Distance vector *sent to neighbours*
  - For each destination d
    - V[d].cost = total cost of shortest path to reach d

```
Every N seconds:
 Vector=null;
 for each destination=d in R[]
 {
  Vector=Vector+Pair(d,R[d].cost);
 }
 for each interface
 {
  Send(Vector);
 }
```


© O. Bonaventure, 2007

Distance vector routing (3)

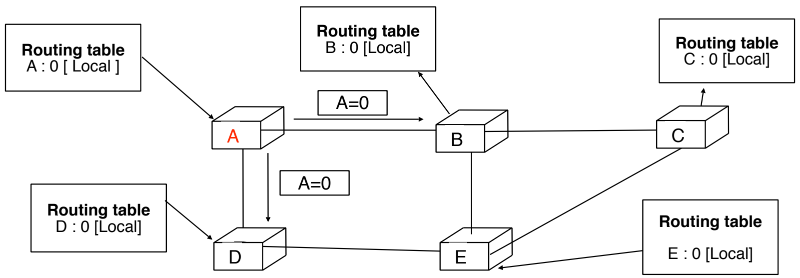
- Processing of received distance vectors

```
Received(Vector V[],link l)
{ /* received vector from link l */
for each destination=d in V[]
{
  if (d isin R[])
  { if ((V[d].cost+l.cost) < R[d].cost)
     { /* shorter path */
       R[d].cost=V[d].cost+l.cost;
       R[d].link=l;
     }
   }
   else
   { /* new route */
     R[d].cost=V[d].cost+l.cost;
     R[d].link=l;
   }
}
```


© O. Bonaventure, 2007

Distance vectors example

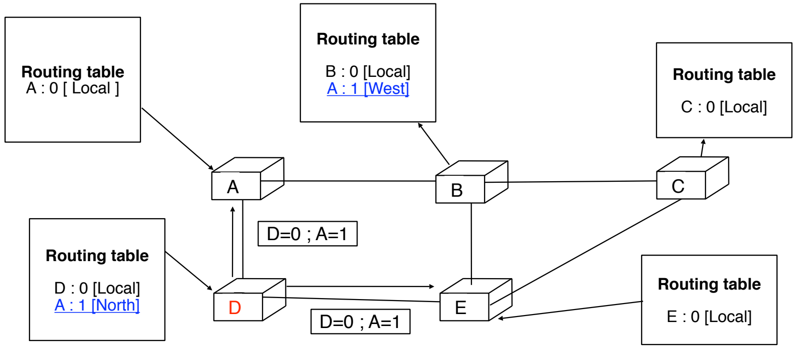
- All links have a unit cost



When a router boots, it only knows itself. Its distance vector thus contains only its address

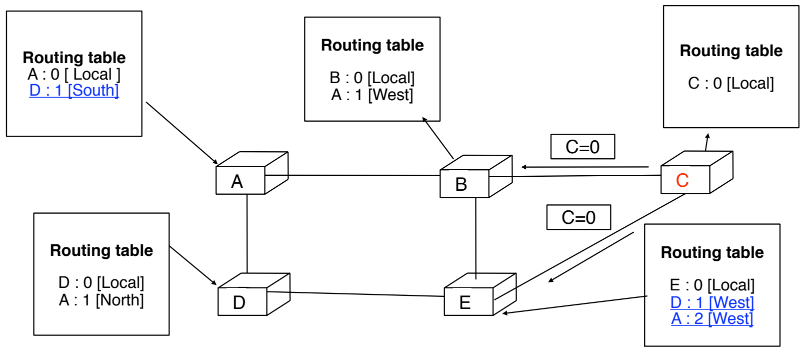
© O. Bonaventure, 2007

Distance vectors example (2)



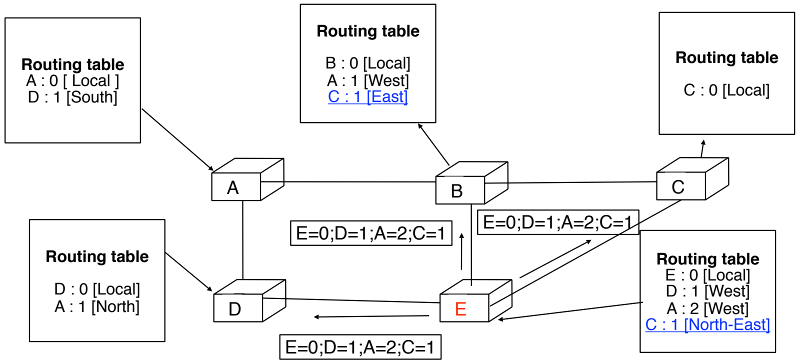
© O. Bonaventure, 2007

Distance vectors example (3)



© O. Bonaventure, 2007

Distance vectors example (4)

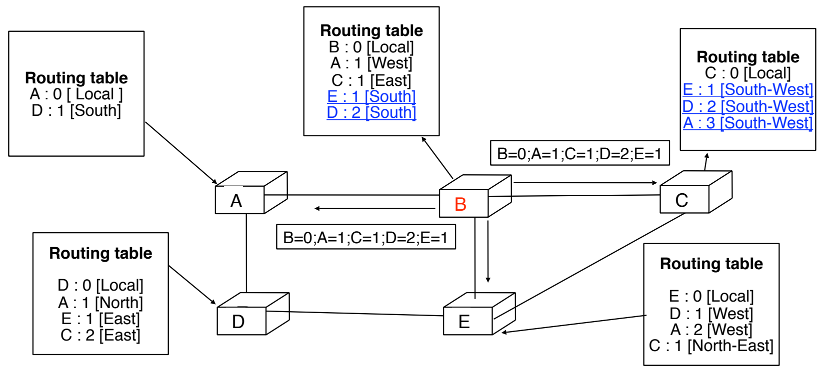


- Reception of distance vector on B
  - New route to reach E and D, longer route for A
- Reception of distance vector on C
  - New routes to reach D, A and E
- Reception of distance vector on D
  - New routes to reach E and C, longer route for A

© O. Bonaventure, 2007

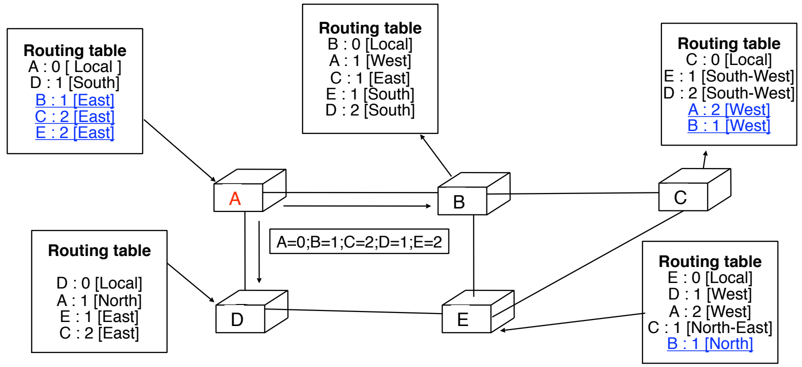
Distance vectors example (5)

- B is the first to send its vector



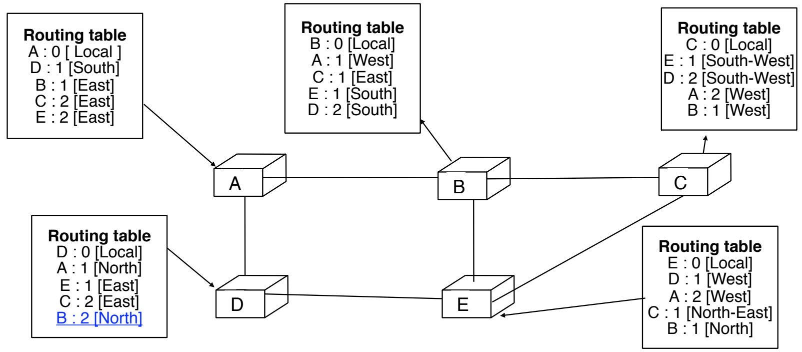
© O. Bonaventure, 2007

Distance vectors example (6)



© O. Bonaventure, 2007

Distance vectors example (7)

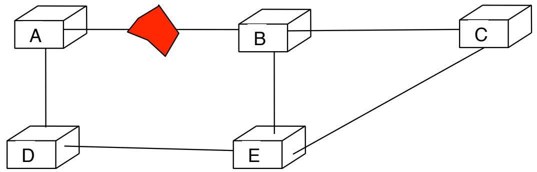


- All routers know how to reach all other routers
- Routing tables are stable
  - If a distance vector is sent by one router, it will not cause any change to the routing table of other routers in the network

© O. Bonaventure, 2007

Distance vectors
Link failures

- How to deal with link failures ?



- Two problems must be solved for failures
  - How to detect that the link has failed ?
  - How to indicate to all routers that they should update their routing table since the paths that use link A-B do not work anymore ?

© O. Bonaventure, 2007

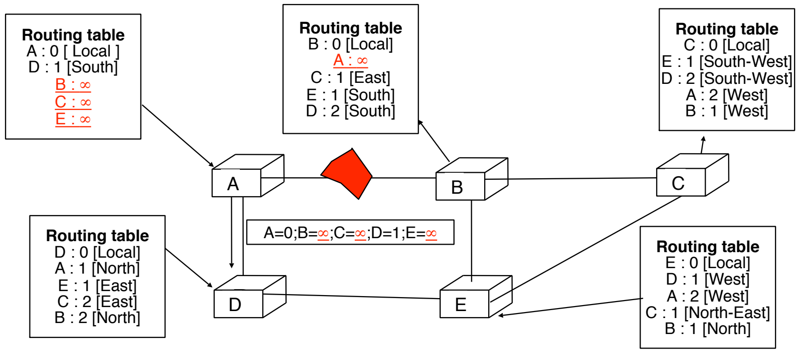
Detection of link failures

- Two types of solutions
  - rely on failure information from datalink or physical layer
    - fast and reliable
    - unfortunately not supported by all datalink/physical layers
  - ask each router to regularly send its distance vector (e.g. every 30 seconds)
    - If a router does not receive a refresh for a route in a distance vector from one of its neighbours during some time (e.g. 90 seconds), it assumes that the route is not available anymore

© O. Bonaventure, 2007

How to update the routing table ?

- All routes that use a failed link are marked with an infinite cost



© O. Bonaventure, 2007

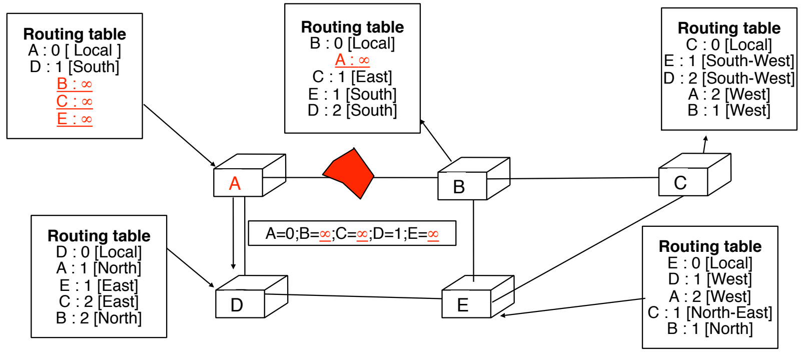
How to update the routing table ? (2)

- Reception of a distance vector

```
Received(Vector V[],link l)
{ /* received vector from link l */
for each destination=d in V[]
{
  if (d isin R[])
  { if ( ((V[d].cost+l.cost) < R[d].cost) OR
        ( R[d].link == l) )
     { /* better route or change to current route */
       R[d].cost=V[d].cost+l.cost;
       R[d].link=l;
     }
   }
   else
   { /* new route */
     R[d].cost=V[d].cost+l.cost;
     R[d].link=l;
   }
}
```


© O. Bonaventure, 2007

How to update the routing table ? (3)



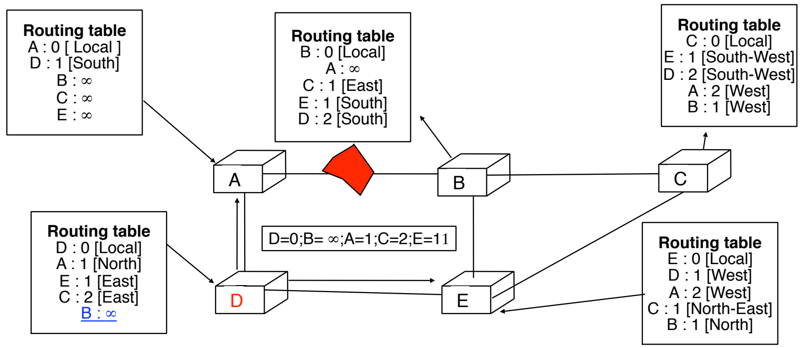
- D must remove from its routing tables all the routes that it learned from its North link and are announced now with an ∞ cost

CNPP/2008.4.

© O. Bonaventure, 2007

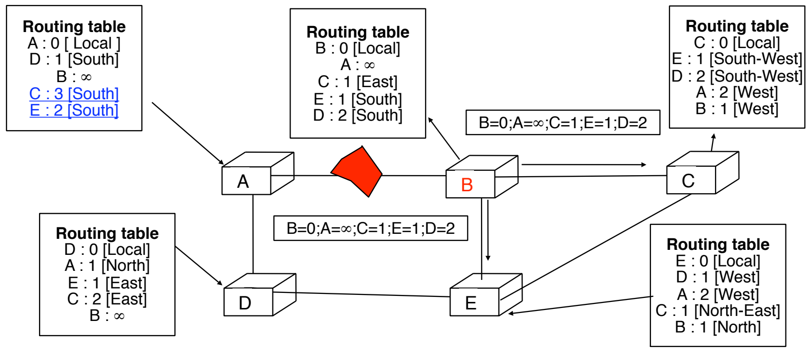
En outre, un temporisateur East associé à chaque entrée de la table de routage de tout routeur. Ce temporisateur East remis à zéro chaque fois qu'un vecteur de distance contenant cette route East reçu par le routeur. Si le temporisateur expire, la route East considérée comme étant invalide et elle East supprimée de la table de routage.

How to update the routing table ? (4)



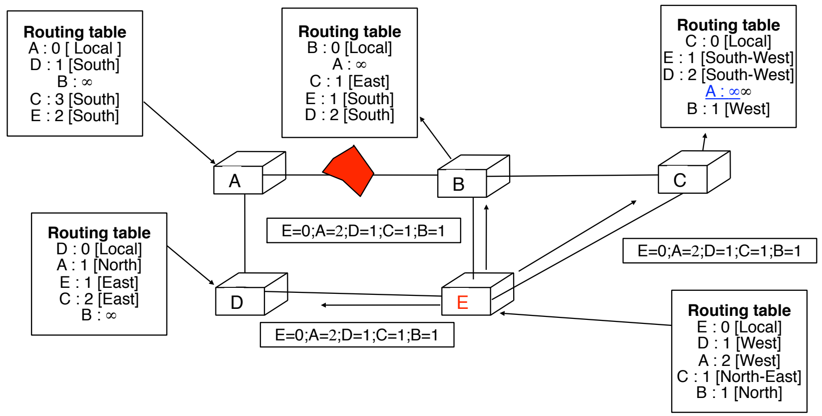
© O. Bonaventure, 2007

How to update the routing table ? (5)



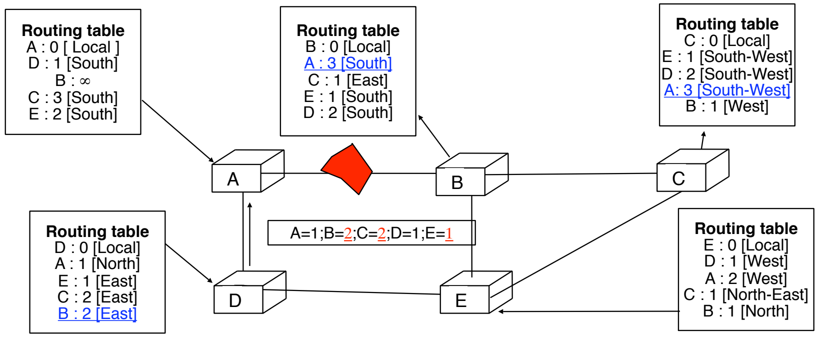
© O. Bonaventure, 2007

How to update the routing table ? (6)



© O. Bonaventure, 2007

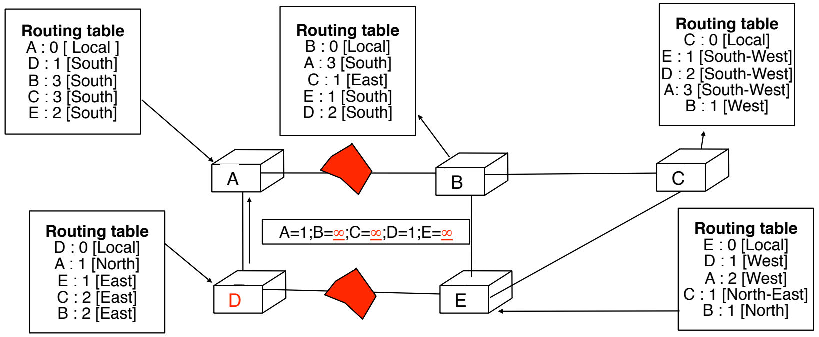
How to update the routing table ? (7)



- Failure has been recovered, all routers are now reachable again from any router

© O. Bonaventure, 2007

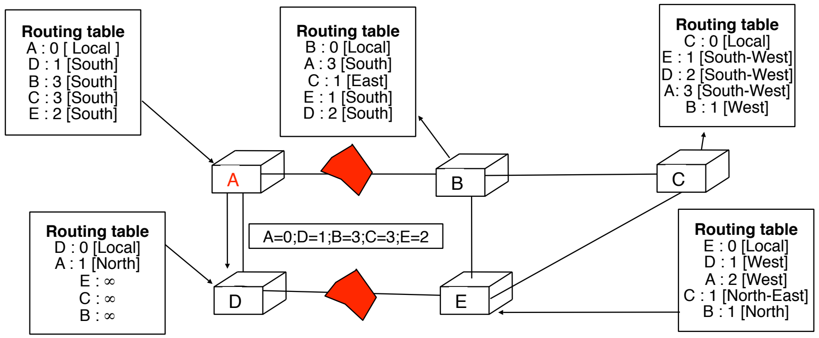
Second link failure



- D detects the failure
  - If it is the first to send its distance vector, failure is detected and router A updates its routing table

© O. Bonaventure, 2007

Second link failure (2)

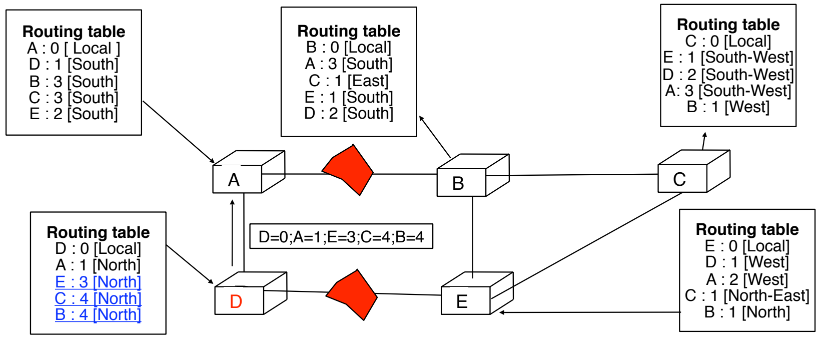


- But if A sends its distance vector before having received or processed D’s updated distance vector ...

© O. Bonaventure, 2007

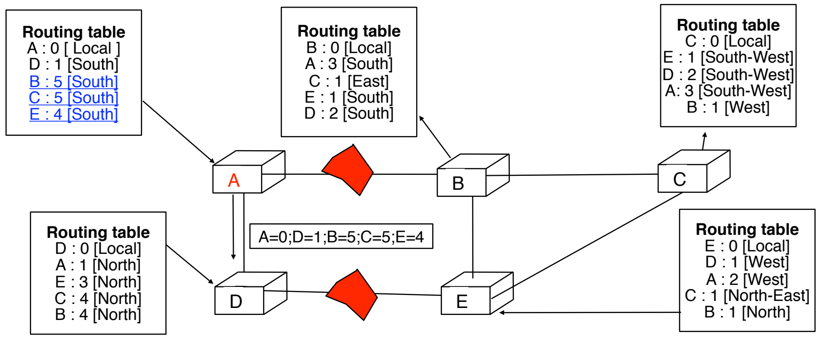
Second link failure (3)

- Upon reception of A's vector, D updates its routing table



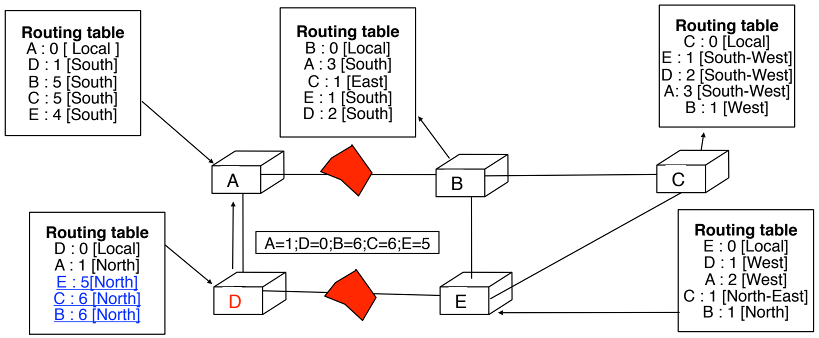
© O. Bonaventure, 2007

Second link failure (4)



© O. Bonaventure, 2007

Second link failure (5)



- This problem is called counting to infinity
  - How can we avoid it ?

© O. Bonaventure, 2007

Second link failure (6)

- Where does counting to infinity comes form ?
  - A router announces on a link routes that it has already learned via this link
- How to avoid counting to infinity ?
  - split horizon
    - each router creates a distance vector <u>for each link</u>
      - on link i, router does not announce the routers learned over link i

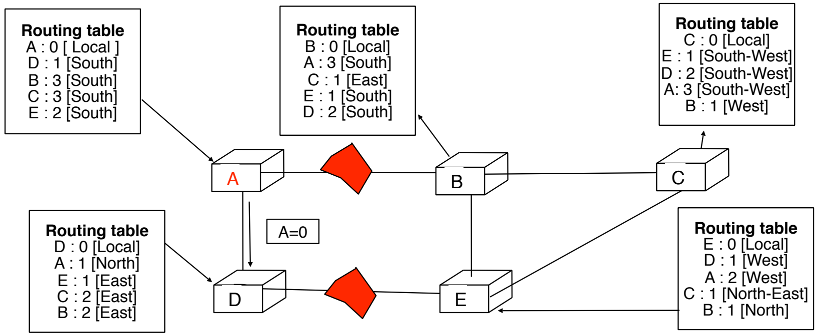
```
Pseudocode
Every N seconds:
 for each link=l
 { /* one different vector for each link */
  Vector=null;
  for each destination=d in R[]
  {
   if (R[d].link<>l)
    { Vector=Vector+Pair(d,R[d].cost); }
  }
  Send(Vector);
 }
```


© O. Bonaventure, 2007


```
Pseudocode
Every N seconds:
 for each link=l
 { /* one different vector for each link */
  Vector=null;
  for each destination=d in R[]
  {
   if (R[d].link<>l)
   {
    Vector=Vector+Pair(d,R[d].cost);
   }
  }
  Send(Vector);
 }
```

Split horizon

- Back to previous example

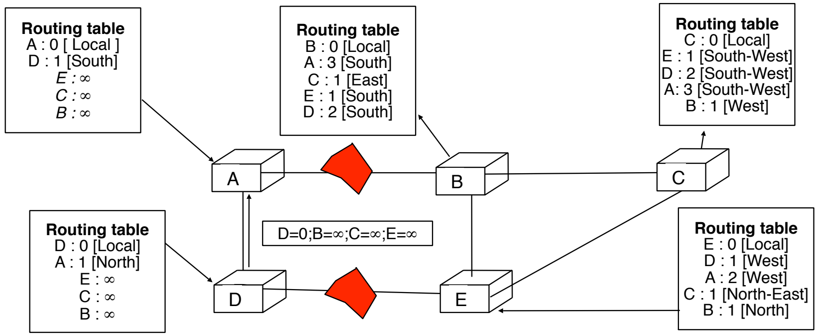


- A will not pollute D's routing table with split horizon

© O. Bonaventure, 2007

Split horizon (2)

- D can also send its distance vector



- Does split horizon allows to avoid all counting to infinity problems ?

© O. Bonaventure, 2007

Split horizon with poisoning

- Improvement
  - Instead of not advertising a route over the link from which it was learned, advertise it with an infinite cost

```
Pseudocode

Every N seconds:
 for each link=l
 { /* one different vector for each link */
  Vector=null;
  for each destination=d in R[]
  {
   if (R[d].link<>l)
   {
    Vector=Vector+Pair(d,R[d].cost);
   }
   else
   {
    Vector=Vector+Pair(d,∞);
   }
  }
  Send(Vector);
 }
```

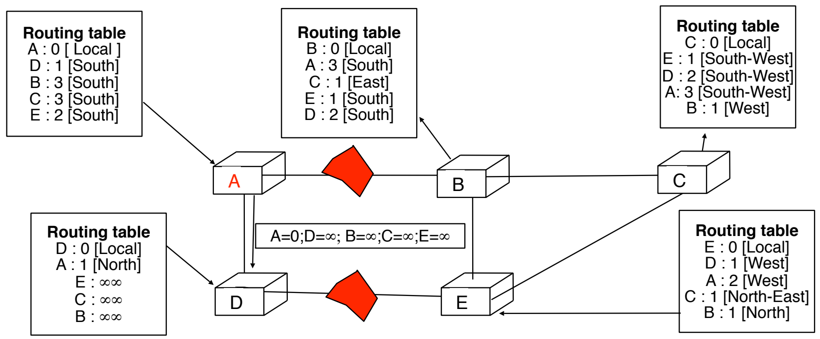

© O. Bonaventure, 2007


```
Pseudocode

Every N seconds:
 for each link=l
 { /* one different vector for each link */
  Vector=null;
  for each destination=d in R[]
  {
   if (R[d].link<>l)
   {
    Vector=Vector+Pair(d,R[d].cost);
   }
   else
   {
    Vector=Vector+Pair(d,∞);
   }
  }
  Send(Vector);
 }
```

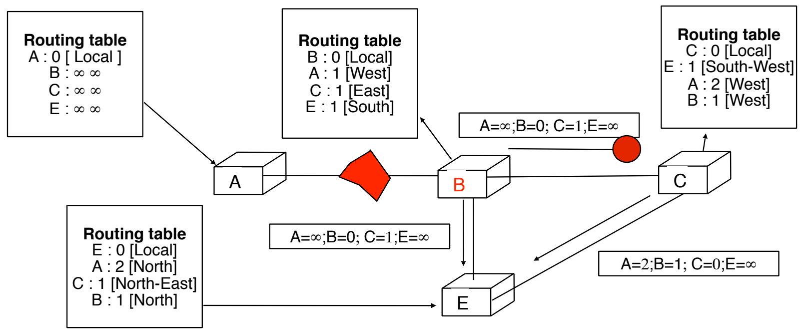
Split horizon with poisoning (2)

- Back to previous example



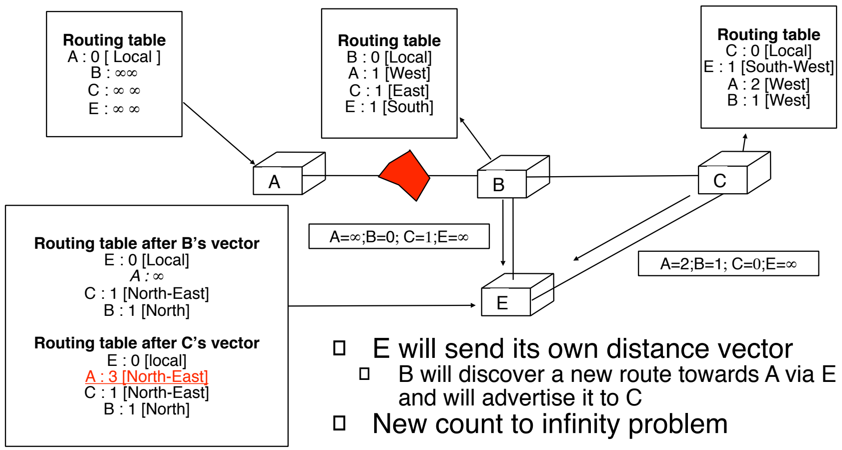
© O. Bonaventure, 2007

Limitations to split horizon



© O. Bonaventure, 2007

Limitations to split horizon (2)



- E will send its own distance vector
  - B will discover a new route towards A via E and will advertise it to C
- New count to infinity problem

© O. Bonaventure, 2007

Network layer

- Basics
- Routing
  - Static routing
  - Distance vector routing
  - Link state routing
- IP : Internet Protocol
- Routing in IP networks

© O. Bonaventure, 2007

Link state routing

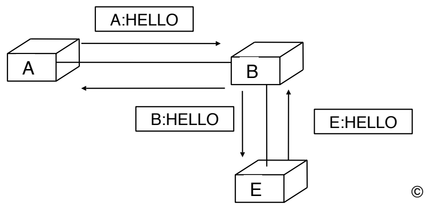
- Idea
  - Instead of distributing summaries of routing tables, wouldn't it be better to distribute network map ?

- How to build such as network map ?
  - Each router must discover its neighbours
  - It should be possible to associate a cost to each link since all links are not equal
  - Each router sends its local topology to all routes and assembles the information received from other routers
  - Routers build the network graph and used Dijkstra's algorithm to compute shortest paths

© O. Bonaventure, 2007

Neighbour discovery

- How does a router discover its neighbours ?
  - By manual configuration
    - Unreliable and difficult to manage
  - By using HELLO packets
    - Every N seconds, each router sends a HELLO packet on each link with its address
    - Neighbours replay by sending their own address
    - Periodic transmission allows to verify that the link remains up and detect failures



© O. Bonaventure, 2007

How to determine link costs ?

- Principle
  - one cost is associated with each link direction
- Commonly configured link costs
  - Unit cost
    - simplest solution but only suitable for homogeneous networks
  - Cost depends on link bandwidth
    - high cost for low bandwidth links
    - low cost for high bandwidth links
  - Cost depends on link delays
    - often used to avoid satellite links
- Cost based on measurements
  - Use HELLO to measure link rtt
    - allows to track link load, but be careful if the measurement is not stable enough as each delay change will cause a topology change ...

© O. Bonaventure, 2007

Different costs can be used for different link directions

Assembling the network topology

- How to assemble the network topology
  - By receiving HELLOs, each routers builds its local part of the network map
  - Each router summarises its local topology inside one link state packet that contains
    - router identification
    - pairs (neighbour identification,cost to reach neighbour)
- When should a router send its link state packet ?
  - in case of modification to its local topology
    - allows to inform all other routers of the change
  - Every N seconds
    - allows to refresh information in all routers and makes sure that if an invalid information was stored on a router due to memory errors it will not remain in the router forever

© O. Bonaventure, 2007

Contents of the LSP

LSP.Router : Identification of the sender of the LSP

LSP.Links[] : links advertised in the LSP

LSP.Links[i].Id : identification of the neighbour

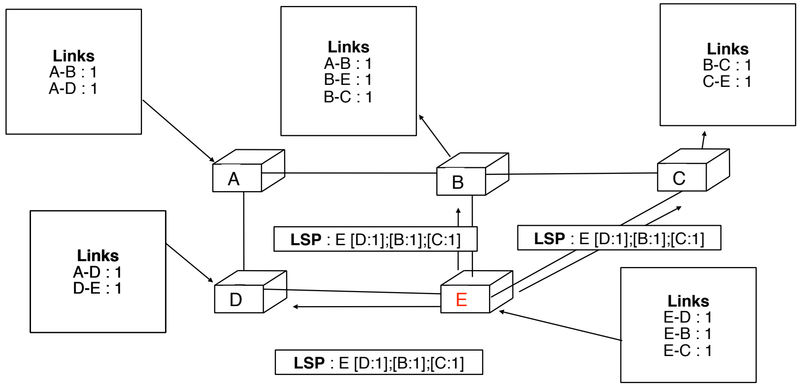
LSP.Links[i].delay : cost of the link

How to distribute the link state packets ?

- Naive solution
  - Each router sends one packet to each other router in the network
    - This solution can only work if
      - All routers know the address of all other routers in network
      - All routers already have routing tables that allow them to forward packets to any destination
- Realistic solution
  - Does not rely on pre-existing routing tables
  - Each router must receive entire topology
  - First solution
    - Each router sends local topology in link state packet and sends it to all its outgoing links
    - When a router receives an LSP, it forwards it to all its outgoing links except the link from which it received it

© O. Bonaventure, 2007

LSP flooding

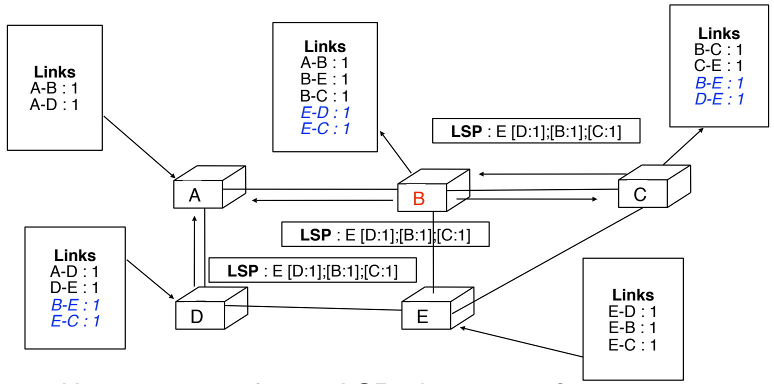


- Assumes that all links have a unit cost

CNPP/2008.4. © O. Bonaventure, 2007

Il faut noter que le LSP envoyé par le routeur E décrit les liens dirigés du routeur E vers les routeurs D, B et C. Le LSP du routeur D contiendra l'information relative au liens dirigés D->E et D->A.

LSP flooding (2)



- How to ensure that an LSP will not loop ?

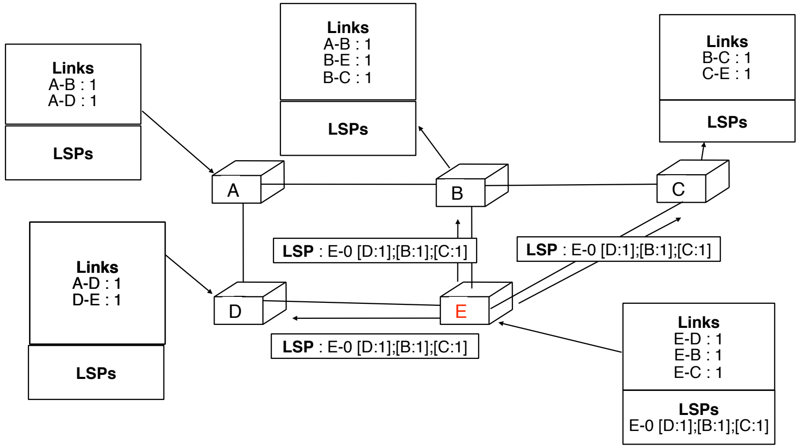
© O. Bonaventure, 2007

LSP flooding (3)

- How to avoid LSP flooding loops ?
  - A router should not reflood an LSP that it has already and flooded
- Solution
  - LSP contents
    - sequence number
      - incremented every time an LSP is generated by a router
    - address of LSP originator
    - pairs address:distance for all neighbours of the originator
  - Each router must store the last LSP received from each router of the network
  - A received LSP is processed and flooded only if is it more recent than the LSP stored in the LSDB

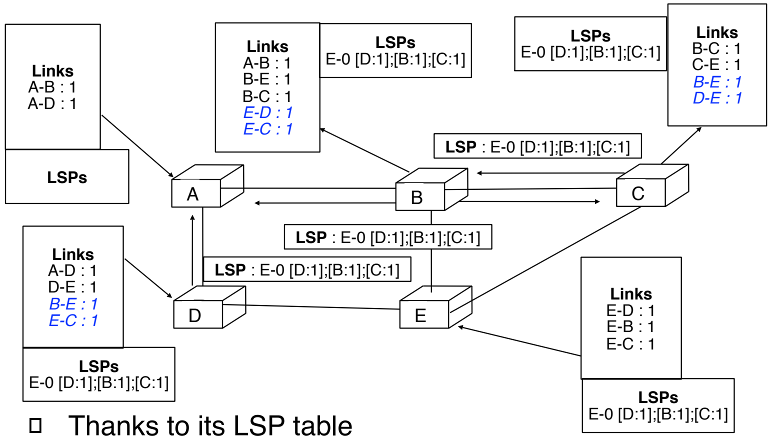
© O. Bonaventure, 2007

LSP flooding (4)



© O. Bonaventure, 2007

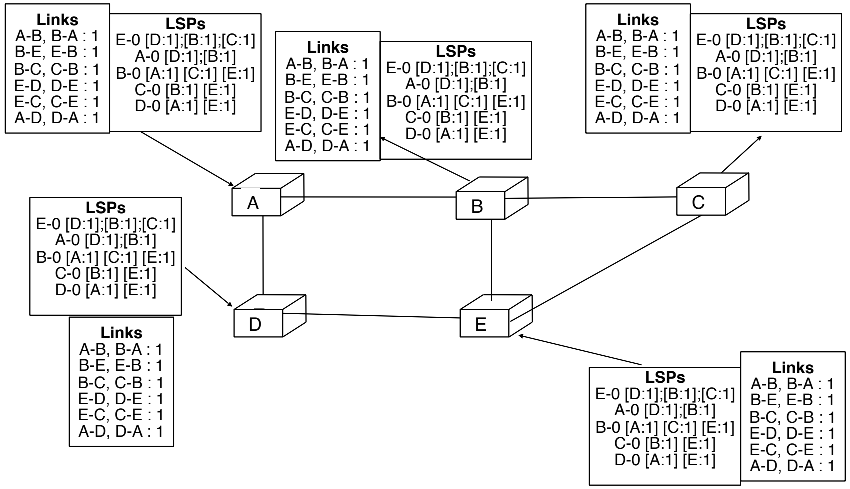
LSP flooding (5)



- Thanks to its LSP table
  - A can detect that it received same LSP via B and D
  - C can detect that it received same LSP via B and E

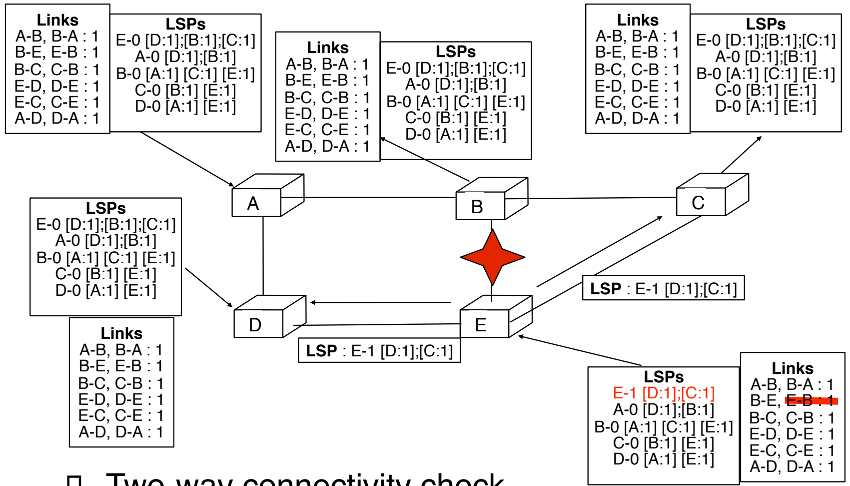
© O. Bonaventure, 2007

Full topology



© O. Bonaventure, 2007

How to deal with link failures ?



- Two-way connectivity check
  - A link is only considered useable if both directions have been advertised

© O. Bonaventure, 2007

Router failures

- What happens if a router fails ?
  - All its interfaces become unusable and do not reply anymore to HELLO packets
- What happens when the router reboots ?
  - It will send its LSP with its sequence number set to zero
    - If older LSPs from same router were still in network, then the new LSP will not be flooded
- Solution
  - Add "age" field inside each LSP
  - Each router must decrement age regularly
    - even for the LSPs stored in its LSDB
  - LSP having age=0 is too old and must be deleted
  - Each router must flood regularly its own LSP with age>0 to ensure that it remains inside network

© O. Bonaventure, 2007

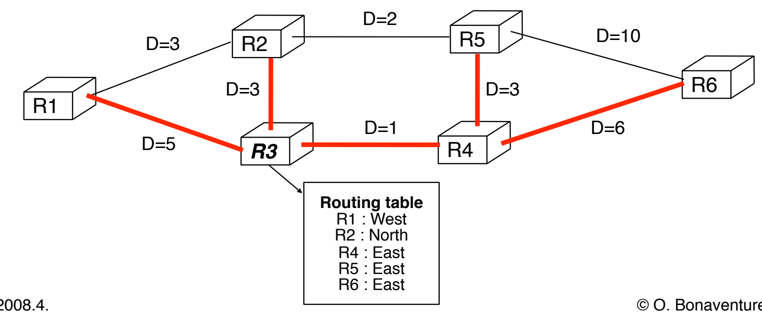
Improvements to LSP flooding

- Avoid sending twice same LSP on a link
  - When an LSP needs to be flooded on a link, wait some time to let other router flood the LSP
    - reduces number of LSPs exchanged on a link but
    - increases flooding time
- Reliable flooding
  - CRC inside each LSP to detect transmission errors
  - Acknowledgements on each link for the LSPs exchanged on this link
    - each transmission is protected by a timer
- Link state database exchange/synchronisation
  - Routers can compare the content of their LSDB and exchange only missing LSPs form neighbour
    - useful when the router boots and wants to receive quickly all LSPs from the network

© O. Bonaventure, 2007

Computation of routing table

- Principle
  - Each router uses the received LSPs to build a graph and then computes the shortest spanning tree rooted on itself
    - From this spanning tree, it is easy to compute the routing table



© O. Bonaventure, 2007

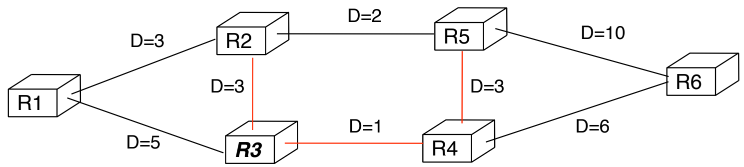
Link state routing protocols apply the two way connectivity check before starting the computation. This check verifies that a link is advertised by both endpoints of the link. If this is not the case, then the link is removed from the graph. This check is important to deal with node failures, but it also allows to speedup the convergence after a link failure.

Dijkstra's shortest path

- Computing the shortest path tree
  - At the beginning, the tree only contains the root node
  - Adjacent routers are placed with the cost of their link in the candidates list
  - Candidate router with lowest cost is chosen and added to the tree
  - Consider the neighbours of the chosen candidate router and update the candidate router list if
    - one of the new neighbours was not already in the candidates list
    - one of the new neighbours was already in the candidates list but with a longer path than the one in the current list
  - Algorithm continues with the new candidates list and ends when all routers belong to shortest path tree

© O. Bonaventure, 2007

Dijkstra's shortest path (2)



1) Routers : [R1, R2, R4, R5,R6] ; Candidates : [-] ; Tree : R3
2) Routers : [R5, R6] ; Candidates : [R1(5) ; R2(3) ; R4 (1)]
   selected candidate : R4
   New tree : R3 - R4
   New Candidates ? [R1(5) ; R2 (3) ; R5(R4-4) ; R6(R4-7)]
3) Routers [-]
   Selected candidate : R2 ; New tree : R2 - R3 - R4
   New Candidates ? [R1(5) ; R5(R4-4) ; R6 (R4-7)]
4) Selected candidate : R5 ; New tree : R2 - R3 - R4 - R5
   New candidates ? [R1(5) ; R6(R4-7)]
5) …

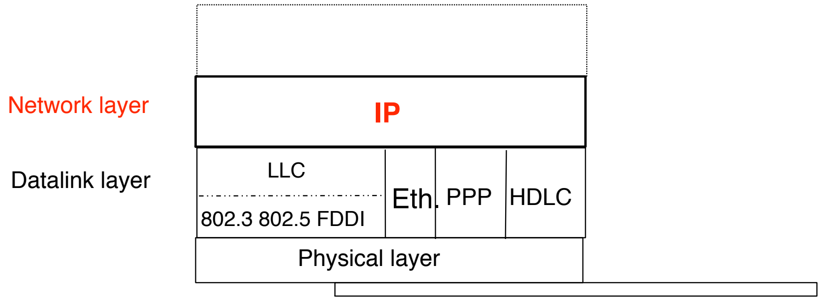
© O. Bonaventure, 2007

Network layer

- Basics
- Routing
  - Static routing
  - Distance vector routing
  - Link state routing
- IP : Internet Protocol
  - IP version 4
  - IP version 6
- Routing in IP networks

© O. Bonaventure, 2007

IP : Internet Protocol



- Internet network layer
  - provides unreliable connectionless service
    - some packets can be lost
    - packets can suffer from transmission errors
    - packets can be misordered

IPv4 is defined in RFC791

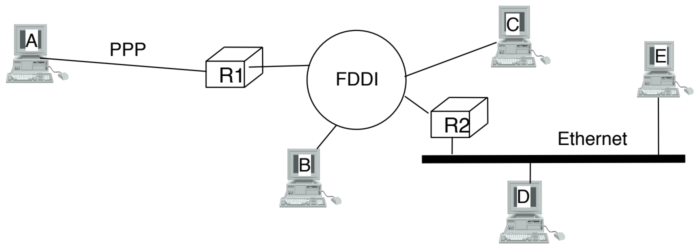
© O. Bonaventure, 2007

IP is defined in

RFC791 Internet Protocol. J. Postel. Sep-01-1981.

Basic principles

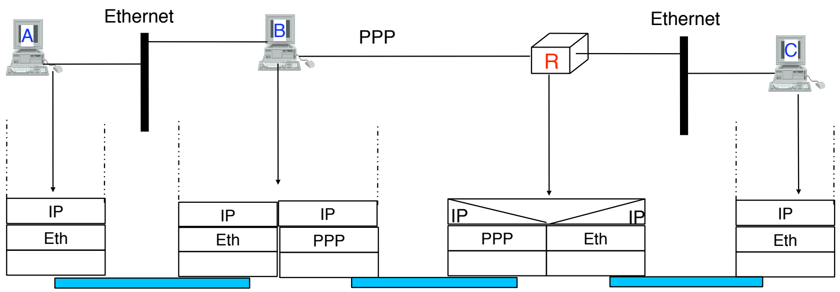
- Datagram mode



  - Each host is identified by one IP address (encoded as 32 bits number)
  - Each host knows how to reach at least one router
  - Routers know how to reach other routers

© O. Bonaventure, 2007

Basic principles (2)



- Endhost
  - equipment able to send and receive packets originated by or destined to it
- Router
  - equipment able to send and receive packets originated by or destined to it
  - equipment able to forward toward theirs destination packets that it did not originate

© O. Bonaventure, 2007

Il East utile de remarquer que la différence fondamentale entre les routeurs et les stations terminales n'East pas le nombre d'interfaces. Même si une station terminale dispose en général d'une seule interface physique, rien ne l'empêche d'en avoir plusieurs. Même dans ce cas, la station pourra très bien seulement recevoir et envoyer des paquets qui sont dEastinés à l'une de ces interfances. C'East par exemple le cas d'un serveur qui serait connecté physiquement à deux réseaux distincts. Une telle station ne deviendra un routeur que si elle accepte de retransmettre vers leur destinations des paquets qui ne lui sont pas directement dEastinés. En pratique, un routeur n'aura que peu d'utilité si il n'a qu'une interface.

IP Addressing

- Utilisation of IP address
  - identify a host/router that implements IP
    - usually, one IP address identifies one (physical) interface on one endhost or router
      - (physical) interface is access point to datalink layer
      - usually endhosts have a single interface
      - routers have more than one interface
  - Encoding of 32 bits IP address
    - 10001010 00110000 00011010 00000001
    - **138 . 48 . 26 . 1**
- How to allocate IP addresses to hosts in a campus network
  - Naive solution
    - First come first served

© O. Bonaventure, 2007

Hierarchical allocation of IP addresses

- Allocation of IP addresses
  - one address per interface
  - each address composed of two parts
    1. subnetwork identifier
       - M high order bits of IP address
    2. equipment identifier inside the subnetworks
       - *32-M* bits low order bits of IP address

  Example

  $\underbrace{\text{10001010 00110000 0001101}}_{\text{subnetwork id}}\underbrace{\textit{0 00000001}}_{\textit{host id}}$

  Notation 138.48.26.1/23 or 138.48.26.1 255.255.254.0

  - All hosts that belong to the same subnetwork can directly exchange frames through datalink layer

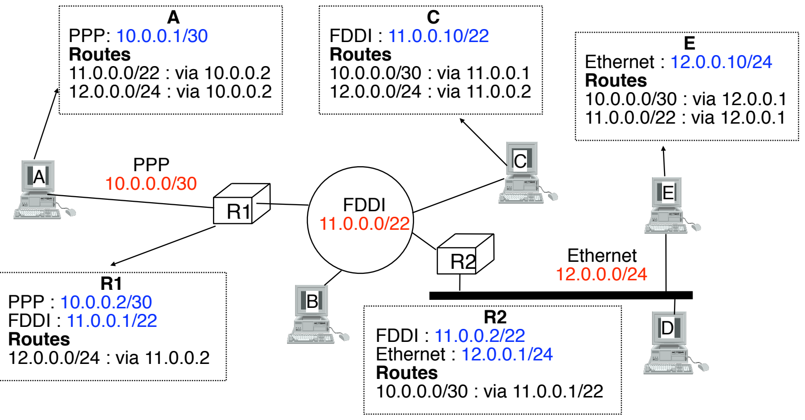
CNPP/2008.4.

© O. Bonaventure, 2007

Une autre façon très fréquente de représenter le nombre de bits correspondant au sous-réseau East de faire suivre l'adresse IP d'un masque. Si le sous-réseau correspond aux M bitsde poids forts de l'adresses, alors les M bits de poids fort du masque sont mis à un tandis que les 32-M bits de poids faible sont mis à zéro.

Par exemple,
Sous-réseau /8 : Masque =255.0.0.0 (aussi appelé réseau de classe A pour des raisons historiques)
Sous-réseau /16: Masque = 255.255.0.0 (aussi appelé réseau de classe B pour des raisons historiques)
Sous-réseau /23: Masque = 255.255.254.0
Sous-réseau /24: Masque = 255.255.255.0 (aussi appelé réseau de classe C pour des raisons historiques)
Sous-réseau /30: Masque = 255.255.255.253 souvent, les lignes point-à-point utilisent des sous-réseaux /30

IP addressing : examples



- Drawbacks of subnetworks
  - most subnetworks are not fully occupied
  - a campus network will need more IP addresses than the number of hosts attached to the network

© O. Bonaventure, 2007

IP addresses

- Most addresses are allocated by IANA
  - and the regional registries RIPE, ARIN, ...
- But some addresses play a special role
  - 127.0.0.1
    - Loopback address on each host
      - Allows to reach servers on the local host
  - 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16
    - used for private networks (not directly attached to Internet)
  - 218.0.0.0/8 - 223.0.0.0/8 and 240.0.0.0/8 - 255.0.0.0/8
    - reserved for further utilization
  - 224.0.0.0/8 - 239.0.0.0/8
    - used by IP multicast
  - 255.255.255.255
    - broadcast address
  - 0.0.0.0
    - used when a host is booting and does not yet know its address

© O. Bonaventure, 2007

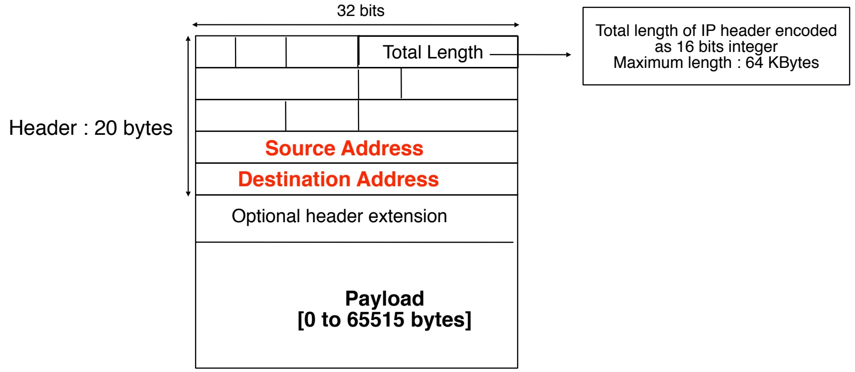
The addresses 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16 are called private addresses or RFC1918 addresses

RFC1918 Address Allocation for Private Internets. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear. February 1996.

More information about the allocation of IP addresses may be found in
http://www.ripe.net
http://www.iana.org

IP Packets

- IP packet format

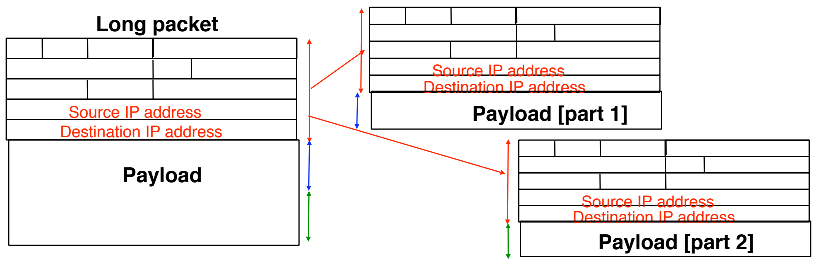


- How can we transmit a 64 KBytes packet ?

© O. Bonaventure, 2007

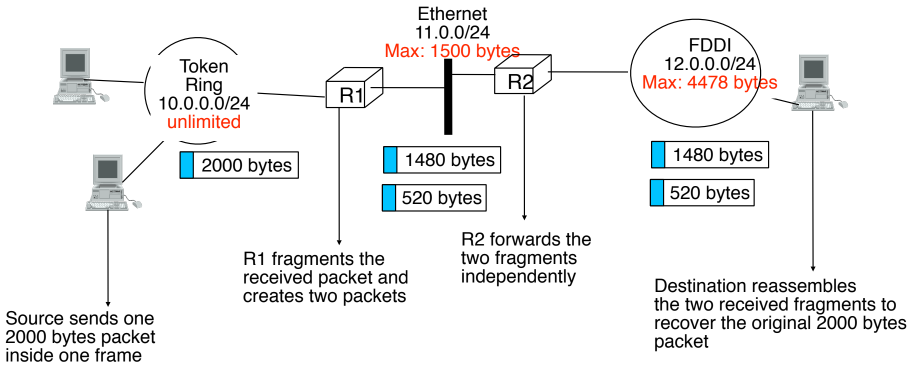
Transmission of long IP packets

- Principe
  - Each host and each router can fragment packets
    - Each fragment is a complete IP packet that contains source and destination IP addresses
  - Only the destination host performs reassembly



© O. Bonaventure, 2007

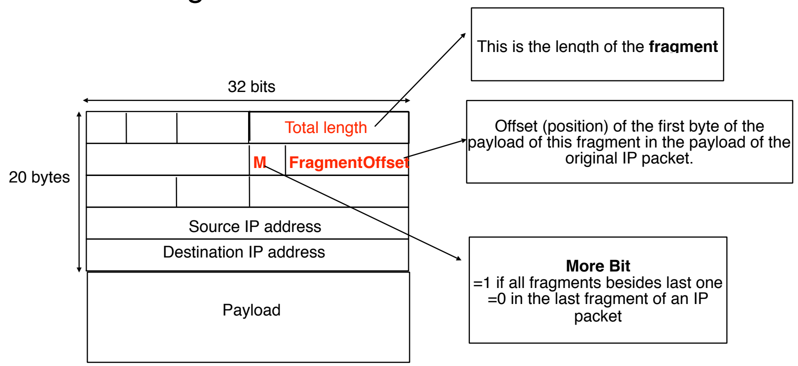
Transmission of long IP packets (2)



© O. Bonaventure, 2007

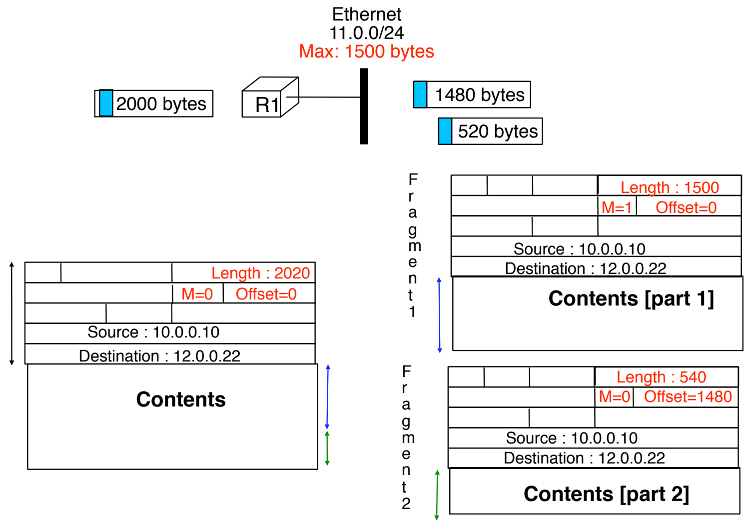
How to deal with limited MTU links ?

- IP fragmentation
  - Fragment the payload of IP packet
  - Each fragment must be number to recover from misordering



© O. Bonaventure, 2007

Fragmentation : example



© O. Bonaventure, 2007

L'algorithme permettant de fragmenter un paquet IP peut se construire de façon quasi immédiate sur base de cet exemple. Lorsque les différents fragments d'un paquet ont été construits, il rEaste à les transmettre vers la destination. Certaines implémentations transmettent les fragments dans l'ordre croissant tandis que d'autres les transmettent dans l'ordre décroissant. La deuxième solution, lorsque les fragments arrivent en séquence, facilite la gEastion des buffers de la destination. En effet, si la destination reçoit d'abord le dernier fragment d'un paquet, elle connait immédiatement la taille exacte du buffer nécessaire pour réassembler ce paquet. Alors que si la destination reçoit d'abord le premier fragment, elle n'a aucune information sur la longueur totale du paquet.

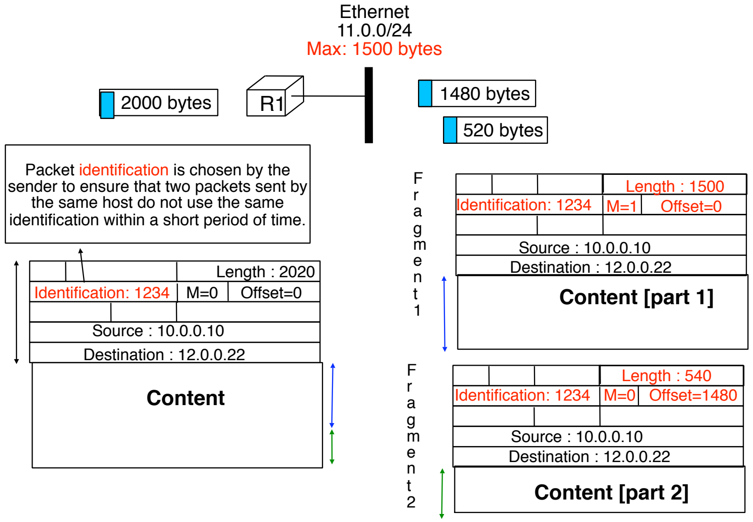
Reassembly

- Issues
  - When does the destination has received all fragments ?
    - Last fragment contains bit More=0
    - How to handle lost fragments ?
      - the IP packet will not be reassembled by destination and received fragments of this packet will be discarded
  - How to deal with misordering
    - Offset field allows to reorder fragments from same packet
    - But misordering can cause fragments from multiple packets to be mixed
      - Each fragment must contain an identification of the original packet from which is was created

© O. Bonaventure, 2007

En pratique, la conception d'un algorithme de réassemblage est un peu plus complexe que ce qui est décrit ci dessus car il faut prendre en compte divers éléments comme l'arrivée de fragments hors séquence et les pertes de segments.

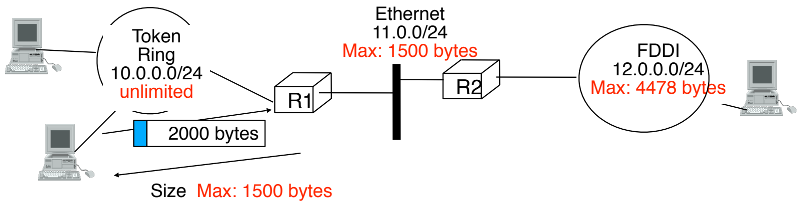
Packets and fragments identification



© O. Bonaventure, 2007

How to avoid fragmentation ?

- Problem
  - How can a host determine the maximum packet size that he can use to reach a destination ?
  - Solution
    - Instead of performing fragmentation, the router could indicate the maximum packet size that it supports



    - Knowing this maximum packet size, the endhost can send correctly sized packets

CNPP/2008.4.

© O. Bonaventure, 2007

Cette solution east utilisée dans l'Internet par certains protocoles utilisateurs de la couche IP. TCP par exemple est capable d'adapter la taille des segments qu'il envoie en fonction de la longueur maximale des paquets acceptées dans le réseau. Cette technique porte le nom de PathMTU discovery (MTU signifie Maximum Transmission Unit, c'East-à-dire la taille maximale des paquets IP généralement sur une ligne donnée)

Voir :

RFC1191 Path MTU discovery. J.C. Mogul, S.E. Deering. Nov-01-1990

Transmission errors

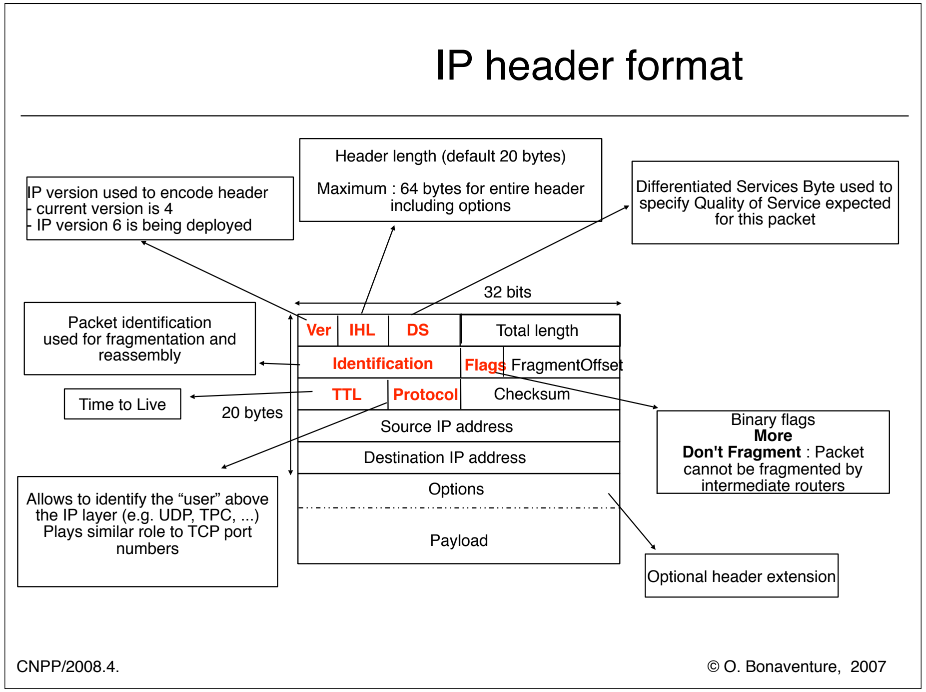
- How should IP react to transmission errors ?
  - Transmission error inside packet content
    - some applications may continue to work despite this error
    - IP : no detection of transmission errors in packet payload
  - Transmission error inside packet header
    - could cause more problems
      - imagine that the transmission error changes the source or destination IP address
    - IP uses a checksum to detect transmission errors in header
      - 16 bits checksum (same as TCP/UDP) computed only on header
      - each router and each end host verifies the chacksum of all packets that it receives. A packet with an errored header is immediately discarded

© O. Bonaventure, 2007

Transient and permanent loops

- Problem
  - Loops can occur in an IP network
    - permanent loops due to configuration errors
    - transient loops while routing tables are being updated
- Solution
  - Each packet contains a Time-to-Live (TTL) that indicates the maximum number of intermediate routers that the packet can cross
    - many hosts set the initial TTL of their packets to 32 or 64
  - each router checks the TTL of all packets
    - If TTL=1, packet is discarded and source is notified
    - If TTL>1, packet is forwarded and TTL is decremented by at least 1
      - routers thus must recompute checksum of all forwarded packets
  - Utilisation of TTL is a means to bound the lifetime of packets inside the Internet

© O. Bonaventure, 2007



1 ICMP Internet Control Message [RFC792]
2 IGMP Internet Group Management [RFC1112]
4 IP IP in IP (encapsulation) [RFC2003]
6 TCP Transmission Control [RFC793]
17 UDP User Datagram [RFC768]

See also http://www.iana.org/assignments/protocol-numbers

IP options are rarely used in practice

IP Options

- Sample IP header options
  - Strict source route option
    - allows the source to list IP addresses of all intermediate routers to reach destination between source and destination
  - Loose source route option
    - allows the source to list IP addresses of some intermediate routers to reach destination between source and destination
  - Record route option
    - allows each router to insert its IP address in the header
      - rarely used because limited header length
  - Router alert
    - allows the source to indicate to routers that there is something special to be done when processing this packet

Constraint : maximum header size with option 64 bytes

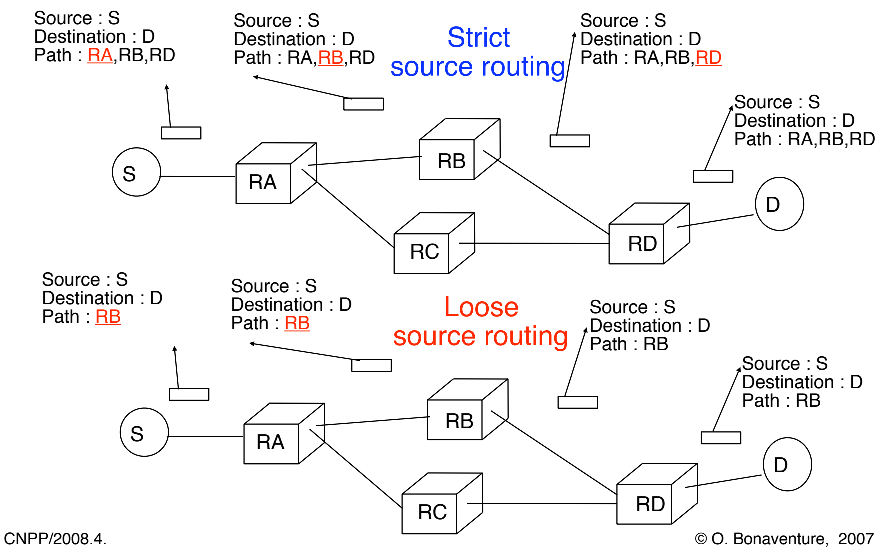
 © O. Bonaventure, 2007

The first three options have been defined in
RFC791 Internet Protocol. J. Postel. Sep-01-1981.

The last is discussed in
RFC2113 IP Router Alert Option. D. Katz. February 1997

However, there have been proposals to remove it

IP Source Routing



CNPP/2008.4.

© O. Bonaventure, 2007

Operation of an IP endhost

- Required information on an IP endhost
  - IP addresses of its interfaces
    - For each address, the subnet mask allows the endhost to determine the addresses that are directly reachable through the interface
  - (small) routing table
    - Directly connected subnets
      - From the subnet mask of its own IP addresses
    - Default router
      - Router used to reach any unknown address
      - By convention, default route is 0.0.0.0/0
    - Other subnets known by endhost
      - Could be manually configured or learned through routing protocols are special packets (see later)

CNPP/2008.4. © O. Bonaventure, 2007

Exemple de configuration d'une station IP :

configuration des interfaces (une loopback et une Ethernet)

```
/sbin/ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 130.104.229.58 netmask ffffff80 broadcast 130.104.229.127
```

Cette station dispose de deux interfaces, l'interface loopback East lo0 et l'interface Ethernet hme0.

table de routage

netstat -rnv

```
IRE Table:
  Destination          Mask           Gateway          Device Mxfrg  Rtt  Ref Flg  Out  In/Fwd
-------------------- --------------- ------------------- ------ ----- ----- --- --- ----- ------
130.104.229.0       255.255.255.128 130.104.229.58       hme0    1500*    0   3 U     5750     0
224.0.0.0           240.0.0.0       130.104.229.58       hme0    1500*    0   3 U        0     0
default             0.0.0.0         130.104.229.126              1500*    0   0 UG   42564     0
127.0.0.1           255.255.255.255 127.0.0.1            lo0     8232*  315   0 UH   65966     0
```

IP address configuration

- How does a host know its IP address
  - Manual configuration
    - Used in many small networks
  - Server-based autoconfiguration RARP
    - DHCP
      - Dynamic Host Configuration Protocol
      - Principle
      - When it attaches to a subnet, endhost broadcasts a request to find DHCP server
      - DHCP server replies and endhost can contact it to obtain IP address
      - DHCP server allocates an IP address for some time period and can also provide additional information (subnet, default router, DNS resolver, ...)
        - DHCP servers can be configured to always provide the same IP address to a given endhost or not
        - Endhost reconfirms its allocation regularly
  - Serverless autoconfiguration
    - Used by IPv6

CNPP/2008.4.
© O. Bonaventure, 2007

RARP est rarement utilisé en pratique aujourd'hui. Il servait essentiellement pour permettre à des stations ne disposant que d'un ROM limitée d'obtenir leur adresse IP afin de booter depuis un serveur situé sur le même réseau local.

DHCP East le protocole d'attribution d'adresses le plus couramment utilisé. Il East le successeur de BOOTP. DHCP est défini dans : RFC2131 Dynamic Host Configuration Protocol. R. Droms. March 1997.

Operation of an IP router

- Required information on an IP router
  - IP addresses of its interfaces
    - For each address, the subnet mask allows the endhost to determine the addresses that are directly reachable through the interface
  - Routing table
    - Directly connected subnets
      - From the subnet mask of its own IP addresses
    - Other known subnets
      - Usually learned via routing protocols, sometimes manually configured
    - Default router
      - Router used to reach any unknown address
      - By convention, default route is 0.0.0.0/0

CNPP/2008.4.

© O. Bonaventure, 2007

En pratique, le nexthop sera l'adresse IP d'un routeur, généralement directement joignable via la couche liaison de données, auquel le routeur local devra envoyer les paquets pour rejoindre un réseau distant.

Operation of an IP router (2)

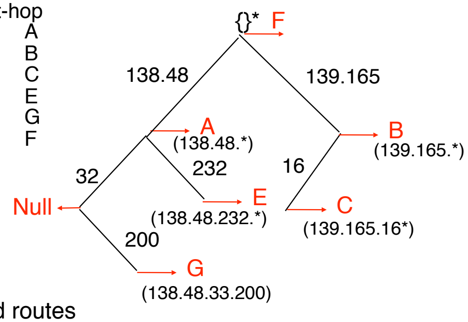
- Operations performed for each packet
  1. Check whether the packet's destination address is one of the router's addresses
     - If yes, packet reached destination
  2. Query Forwarding Information Base that contains
     - list of directly connected networks with masks
     - list of reachable networks and intermediate router
  3. Lookup the most specific route in FIB
     - For each route A.B.C.D/M via Rx
     - compare M higher order bits of destination address with M higher order bits of routes to find longest match
     - forward packet along this route

© O. Bonaventure, 2007

Forwarding Information Base Lookup

- How to find most specific route ?
  - similar to longest match in a text
  - Trie

Subnet	Prefix	Next-hop
138.48.0.0	16	A
139.165.0.0	16	B
139.165.16.0	24	C
138.48.232.0	24	E
138.48.32.200	32	G
0.0.0.0	0	F



Cost of lookup
- f(average length of prefixes)
  - comparisons
  - memory accesses
- caches for most frequently used routes

© O. Bonaventure, 2007

Pour une présentation des méthodes permettant de localiser la route la plus spécifique dans une table de routage, voir :
R. Perlman, Interconnections : bridges and routers, Second Edition, Addison Wesley

Handling IP packets in error

- Problem
  - What should a router/host do when it receives an errored packet
    - Example
      - Packet whose destination is not the current endhost
      - Packet containing a header with invalid syntax
      - Packet received with TTL=1
      - Packet destined to protocol not supported by host
- Solutions
  - Ignore and discard the errored packet
  - Send a message to the packet's source to warn it about the problem
    - ICMP : Internet Control Message Protocol
    - ICMP messages are sent inside IP packets by routers (mainly) and hosts
      - To avoid performance problems, most hosts/routers limit the amount of ICMP messages that they send

ICMP is defined in RFC792

© O. Bonaventure, 2007

ICMP is defined in

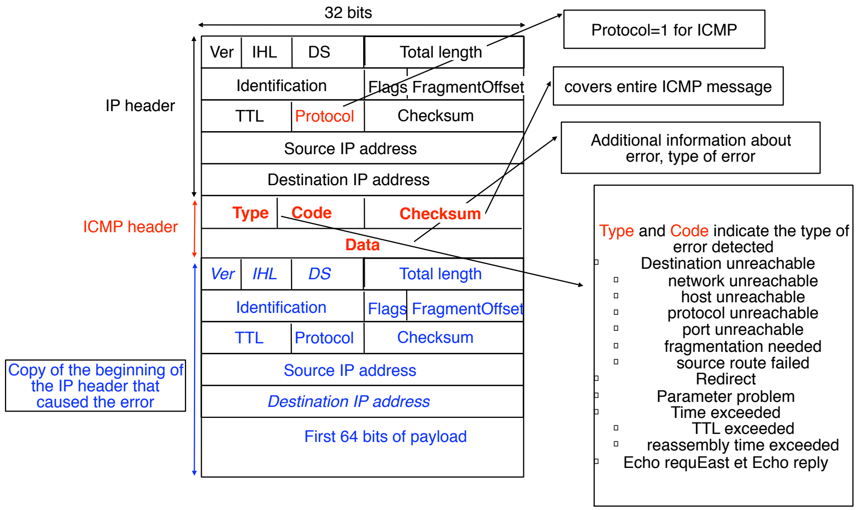
RFC792 Internet Control Message Protocol. J. Postel. Sep-01-1981.

Sample ICMP messages

- Routing error
  - Destination unreachable
    - Final destination of packet cannot be reached
      - Network unreachable for entire subnet
      - Host unreachable for an individual host
      - Protocol/Port unreachable for protocol/port on a reachable host
    - Redirect
      - The packet was sent to an incorrect first-hop router and should have been instead sent to another first-hop router
- Error in the IP header
  - Parameter Problem
    - Incorrect format of IP packet
  - TTL Exceeded
    - Router received packet with TTL=1
  - Fragmentation
    - the packet should have been fragmented, but its DF flag was true

© O. Bonaventure, 2007

ICMP messages



© O. Bonaventure, 2007

Usage of ICMP messages

- Examples
  - destination unreachable
    - the router sending this message did not have a route to reach the destination
  - time exceeded
    - the router sending the message received an IP packet with TTL=0
    - used by `traceroute`
  - redirect
    - to reach destination, another router must be used and ICMP message provides address of this router
  - echo request / echo reply
    - used by `ping`
  - fragmentation impossible
    - the packet should have been fragmented by the router sending the ICMP message by this packet had "Don't Fragment" set to true

CNPP/2008.4. © O. Bonaventure, 2007

Example of ping usage

```
ping astrolabe
PING astrolabe (130.104.229.109) 56(84) bytes of data.
64 bytes from astrolabe (130.104.229.109): icmp_seq=1 ttl=245 time=20.7 ms
64 bytes from astrolabe (130.104.229.109): icmp_seq=2 ttl=245 time=20.2 ms
64 bytes from astrolabe (130.104.229.109): icmp_seq=3 ttl=245 time=20.1 ms

--- astrolabe ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2016ms
rtt min/avg/max/mdev = 20.156/20.383/20.722/0.244 ms
```

Example of traceroute

```
] traceroute www.geant.net
traceroute: Warning: ckecksums disabled
traceroute to newweb.dante.org.uk (62.40.101.34), 30 hops max, 40 byte packets
 1  accelar-1 (130.104.229.126)  1.890 ms  1.752 ms  1.723 ms
 2  XVLX-CR.fsa.ucl.ac.be (130.104.233.233)  1.620 ms  1.620 ms  1.603 ms
 3  CsPythagore.sri.ucl.ac.be (130.104.254.221)  1.317 ms  1.305 ms  1.302 ms
 4  CsHalles.sri.ucl.ac.be (130.104.254.201)  1.512 ms  1.425 ms  1.415 ms
 5  193.191.11.9 (193.191.11.9)  0.891 ms  0.780 ms  0.780 ms
 6  193.191.1.197 (193.191.1.197)  1.166 ms  1.263 ms  1.079 ms
 7  193.191.1.2 (193.191.1.2)  1.329 ms  1.107 ms  1.100 ms
 8  belnet.be1.be.geant.net (62.40.103.13)  1.341 ms  1.490 ms  1.323 ms
```

Network layer

- Basics
- Routing
  - Static routing
  - Distance vector routing
  - Link state routing
- IP : Internet Protocol
  - IP version 4
  - IP version 6
- Routing in IP networks

© O. Bonaventure, 2007

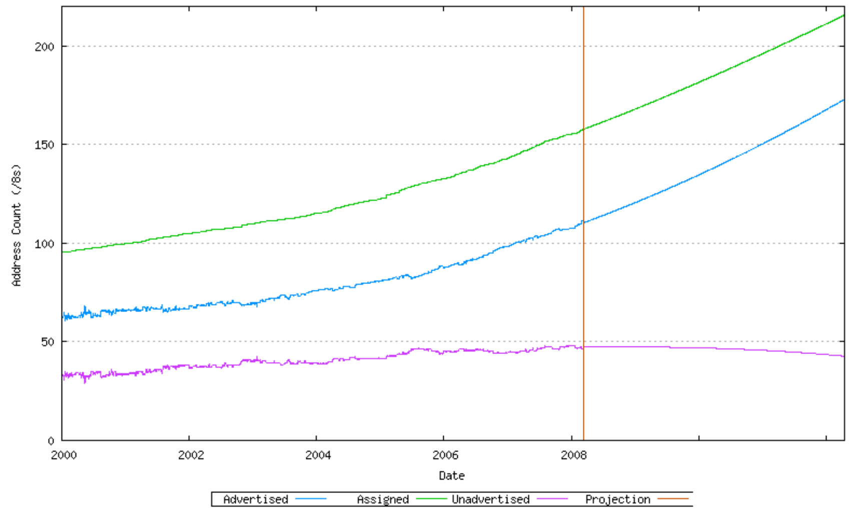
Issues with IPv4

- Implementation issues - 1990s
  - IPv4 packet format is complex
  - IP forwarding is difficult in hardware

- Missing functions - 1990s
  - IPv4 requires lots of manual configuration
    - Competing protocols (CLNP, Appletalk, IPX, ...) already supported autoconfiguration in 1990s
  - How to support Quality of Service in IP ?
    - Integrated services and Differentiated services did not exist then
  - How to better support security in IP ?
    - Security problems started to appear but were less important than today
  - How to better support mobility in IP ?
    - GSM started to appear and some were dreaming of mobile devices attached to the Internet

© O. Bonaventure, 2008

Main motivation today IPv4 address exhaustion



CNPP/2008.4.

Source http://www.potaroo.net/tools/ipv4/index.html

© O. Bonaventure, 2008

This figure shows the number of IPv4 prefixes used on the global Internet. In addition, some networks, e.g. large cable networks, have had difficulties in using IPv4 due to the limited number of available addresses. For example, comcast is planning to use IPv6 to manage its cable modems mainly because IPv4 does not allow them to have enough addresses to identify all their potential cable modems in a scalable manner, see http://www.nanog.org/mtg-0606/durand.html

IPv6 addresses

IPv4

IP version 6

- Each IPv6 address is encoded in 128 bits
  - 3.4 x 10^38 possible addressable devices
    - 340,282,366,920,938,463,463,374,607,431,768,211,456
  - ~ 5 x 10^28 addresses per person on the earth
  - 6.65 x 10^23 addresses per square meter
  - Looks unlimited.... today
- Why 128 bits ?
  - Some wanted variable size addresses
    - to support IPv4 and 160 bits OSI NSAP
  - Some wanted 64 bits
    - Efficient for software, large enough for most needs
  - Hardware implementers preferred fixed size

CNPP/2008.4.

© O. Bonaventure, 2008

IP version 4 supports 4,294,967,296 distinct addresses, but some are reserved for :
private addresses (RFC1918)
loopback (127.0.0.1)
multicast
...

The IPv6 addressing architecture

- Three types of IPv6 addresses
  - Unicast addresses
    - An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address
  - Anycast addresses
    - An identifier for a set of interfaces. A packet sent to an
    - anycast address is delivered to the “nearest” one of the interfaces identified by that address
  - Multicast addresses
    - An identifier for a set of interfaces. A packet sent to a multicast address is delivered to all interfaces identified by that address.

CNPP/2008.4. © O. Bonaventure, 2008

The IPv6 addressing architecture is defined in :

R. Hinden, S. Deering, IP Version 6 Addressing Architecture, RFC4291, February 2006

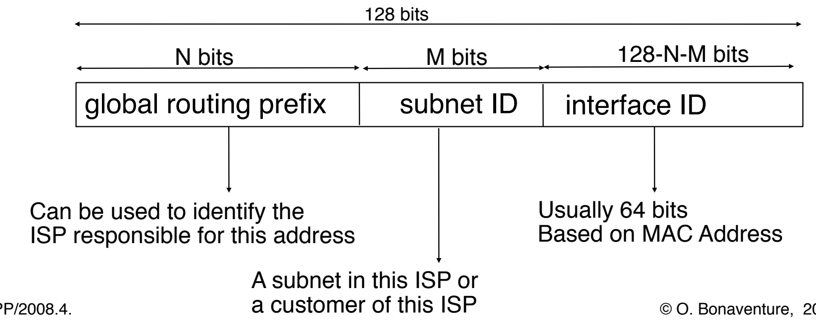
Representation of IPv6 addresses

- How can we write a 128 bits IPv6 address ?
  - Hexadecimal format
    - FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
    - 1080:0:0:0:8:800:200C:417A
  - Compact hexadecimal format
    - Some IPv6 addresses contain lots of zero
      - utilize "::" to indicate one or more groups of 16 bits of zeros. The "::" can only appear once in an address
      - Examples
        - 1080:0:0:0:8:800:200C:417A = 1080::8:800:200C:417A
        - FF01:0:0:0:0:0:0:101 = FF01::101
        - 0:0:0:0:0:0:0:1 = ::1

© O. Bonaventure, 2008

The IPv6 unicast addresses

- Special addresses
  - Unspecified address : 0:0:0:0:0:0:0:0
  - Loopback address : 0:0:0:0:0:0:0:1
- Global unicast addresses
  - Addresses will be allocated hierarchically



© O. Bonaventure, 2008

Today, the default encoding for global unicast addresses is to use :

48 bits for the global routing prefix (first three bits are set to 001)

16 bits for the subnet ID

64 bits for the interface ID

Allocation of IPv6 addresses

- IANA controls all IP addresses and delegates assignments of blocks to Regional IP Address Registries (RIR)
  - RIPE, ARIN, APNIC, AFRINIC, ...
- An organisation can be allocated two different types of IPv6 addresses
  - Provider Independent (PI) addresses
    - Usually allocated to ISPs or very large enterprises directly by RIRs
    - Default size is /32
  - Provider Aggregatable (PA) addresses
    - Smaller prefixes, assigned by ISPs from their PI block
    - Size
      - /48 in the general case, except for very large subscribers
      - /64 when t one and only one subnet is needed by design
      - /128 when it is absolutely known that one and only one device is connecting.

© O. Bonaventure, 2008

See http://www.ripe.net/ripe/docs/ripe-388.html for the policy used by RIPE to allocate IP prefixes in Europe

The IPv6 link-local addresses

- Used by hosts and routers attached to the same LAN to exchange IPv6 packets when they don't have/need globally routable addresses

128 bits

10 bits	54 bits	64 bits
FE80	0000000000.....00000000000	interface ID

- Each host must generate one link local address for each of its interfaces
  - Each IPv6 host will use several IPv6 addresses
- Each routers must generate one link local address for each of its interfaces

CNPP/2008.4. © O. Bonaventure, 2008

Site-local addresses were defined in the first IPv6 specifications, but they are now deprecated and should not be used.

Recently "private" addresses have been defined as Unique Local IPv6 Addresses as a way to allow entreprise to obtain IPv6 addresses without being forced to request them from providers or RIRs.

The way to choose such a ULA prefix is defined in :

R. Hinden, B. Haberman, Unique Local IPv6 Unicast Addresses, RFC4193, October 2005

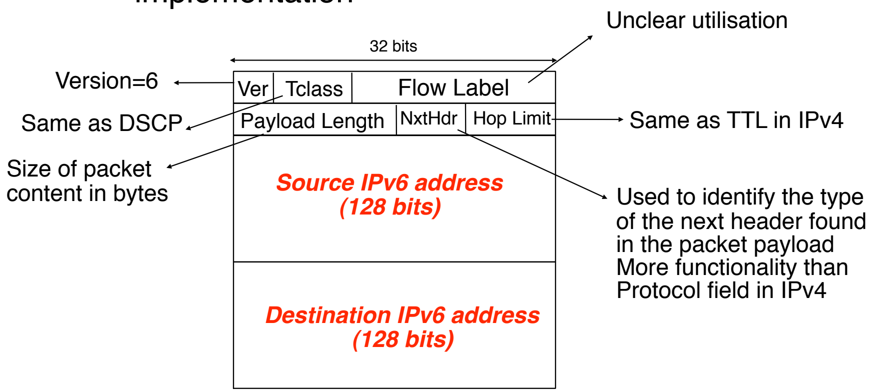
Recently, the case for a registration of such addresses has been proposed, see :
R. Hinden, G. Huston, T. Narten, Centrally Assigned Unique Local IPv6 Unicast Addresses, internet draft, <draft-ietf-ipv6-ula-central-02.txt>, work in progress, June 2007

See also

http://www.ripe.net/ripe/policies/proposals/2007-05.html -

The IPv6 packet format

- Simplified packet format
  - Fields aligned on 32 bits boundaries to ease implementation



- No checksum in IPv6 header
  - rely on datalink and transport checksums

© O. Bonaventure, 2008

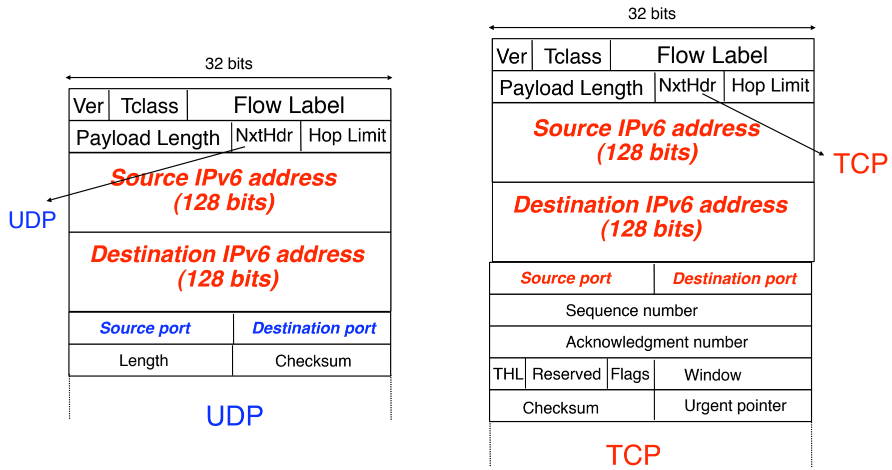
The IPv6 packet format is described in
S. Deering, B. Hinden, Internet Protocol, Version 6 (IPv6) Specification , RFC2460, Dec 1998

Several documents have been written about the usage of the Flow label. The last one is

J. Rajahalme, A. Conta, B. Carpenter, S. Deering, IPv6 Flow Label Specification, RFC3697, 2004

However, this proposal is far from being widely used and deployed.

Sample IPv6 packets



- Identification of a TCP connection
  - IPv6 source, IPv6 destination, Source and Destination ports

© O. Bonaventure, 2008

IPv6 does not require changes to TCP and UDP for IPv4. The only modification is the computation of the checksum field of the UDP and TCP headers since this checksum is computed by concerning a pseudo header that contains the source and destination IP addresses.

The IPv6 extension headers

- Several types of extension headers
  - Hop-by-Hop Options
    - contains information to be processed by each hop
  - Routing (Type 0 and Type 2)
    - contains information affecting intermediate routers
  - Fragment
    - used for fragmentation and reassembly
  - Destination Options
    - contains options that are relevant for destination
  - Authentication
    - for IPSec
  - Encapsulating Security Payload
    - for IPSec
- Each header must be encoded as n*64 bits

© O. Bonaventure, 2008

An example hop-by-hop option is the router alert option defined in
A. Jackson, C. Partridge, IPv6 Router Alert Option RFC2711, 1999

Hop-by-hop and destination option headers

- TLV format of these options

NxtHdr	HLen	Type	Len
Data (var. length)			

- Two leftmost bits
  - How to deal with unknown option ?
    - 00 ignore and continue processing
    - 01 silently discard packet
    - 10 discard packet and send ICMP parameter problem back to source
    - 11 discard packet and send ICMP parameter problem to source if destination isn't multicast
  - Third bit
    - Can option content be changed en-route
  - Five rightmost bits
    - Type assigned by IANA

© O. Bonaventure, 2008

The Len field encodes the size of the data field in bytes. Furthermore, special options have been defined to allow hosts using the options to pad the size of vairable length options to multiples of 64 bits.

Pad1 option (alignment requirement: none)

```
+-+-+-+-+-+-+-+-+
|       0       |
+-+-+-+-+-+-+-+-+
```

NOTE! the format of the Pad1 option is a special case -- it does not have length and value fields.

The Pad1 option is used to insert one octet of padding into the Options area of a header. If more than one octet of padding is required, the PadN option, described next, should be used, rather than multiple Pad1 options.

IPv6 jumbograms

- IPv6 packet format only supports 64 KBytes packets
  - packet size is encoded in 16 bits field
- on most hosts throughput increases with packet size

- Hop-by-hop jumbogram option
  - Increases packet size to 32 bits
    - when used, packet size in IPv6 header should be set to zero

NxtHdr	HLen	C2	Len:4
Packet size			

C2 : 11 0 00020
11 -> ICMP must be sent if option is unrecognised
0 -> content of option does not change en-route

CNPP/2008.4. © O. Bonaventure, 2008

As of today, it is unclear whether the jumbogram option has been implemented in practice. Using it requires link layer technologies that are able to support frames larger than 64 KBytes.

The jumbogram option has been defined in

D. Borman, S. Deering, B. Hinden, IPv6 Jumbograms, RFC2675, August 1999

The Kame (http://www.kame.net) implementation on FreeBSD supports this option, but there is no link-layer that supports large frames.

Packet fragmentation

- IPv4 used packet fragmentation on routers
  - All hosts must handle 576+ bytes packets
  - experience showed fragmentation is costly for routers and difficult to implement in hardware
  - PathMTU discovery is now widely implemented
- IPv6
  - IPv6 requires that every link in the internet have an MTU of 1280 octets or more
    - otherwise link-specific fragmentation and reassembly must be provided at a layer below IPv6
  - Routers do not perform fragmentation
    - Only end hosts perform fragmentation and reassembly by using the fragmentation header
    - But PathMTU discovery should avoid fragmentation most of the time

© O. Bonaventure, 2008

Path MTU discovery is defined in

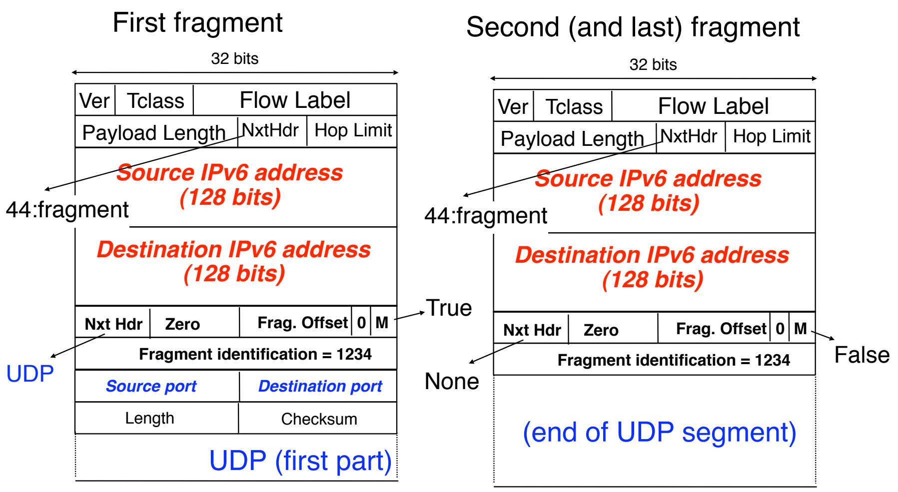
J. Mogul, S. Deering, Path MTU Discovery, RFC1191, 1996

and in

J. McCann, S. Deering, J. Mogul, Path MTU Discovery for IP version 6, RFC1981, 1996

for IPv6

A fragmented IPv6 packet



© O. Bonaventure, 2008

In IPv6, the fragment identification field is much larger than in IPv4. Furthermore, it is only used in packets that really need fragmentation. IPv6 header does not contain a fragmentation information for each unfragmented packet unlike IPv4.

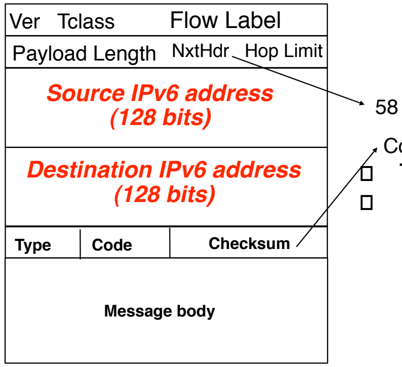
ICMPv6

- Provides the same functions as ICMPv4, and more
- Types of ICMPv6 messages
  - Destination unreachable
  - Packet too big
    - Used for PathMTU discovery
  - Time expired (Hop limit exhausted)
    - Traceroute v6
  - Echo request and echo reply
    - Pingv6
  - Multicast group membership
  - Router advertisments
  - Neighbor discovery
  - Autoconfiguration

© O. Bonaventure, 2008

ICMPv6 is defined in :
A. Conta, S. Deering, M. Gupta, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC4443, March 2006

ICMPv6 packet format



- Type
- ICMPv6 error messages (0<type<127)
  - 1 Destination Unreachable
  - 3 Time Exceeded
  - 2 Packet Too Big
  - 4 Parameter Problem
  - 100 Private experimentation
  - 101 Private experimentation
  - 127 Reserved for expansion
- ICMPv6 informational messages:
  - 128 Echo Request
  - 129 Echo Reply
  - 200 Private experimentation
  - 201 Private experimentation
  - 255 Reserved for expansion

of ICMPv6 informational

© O. Bonaventure, 2008

ICMPv6 uses a next header value of 58 inside IPv6 packets

Network layer

- Basics
- Routing
- IP : Internet Protocol
- Routing in IP networks
  - Internet routing organisation
  - Intradomain routing : RIP
  - Intradomain routing : OSPF
  - Interdomain routing : BGP

© O. Bonaventure, 2007

Internet organisation

- Internet is an internetwork with a large number of Autonomous Systems (AS)
  - an AS is a set of routers that are managed by the same administrative entity
    - Examples : BELNET, UUNET, SKYNET, ...
    - about 20000 ASes in 2007
  - Autonomous Systems are interconnected to allow the transmission of IP packets from any source to any destination
    - On the Internet, most packets need to travel through several transit Autonomous Systems

© O. Bonaventure, 2007

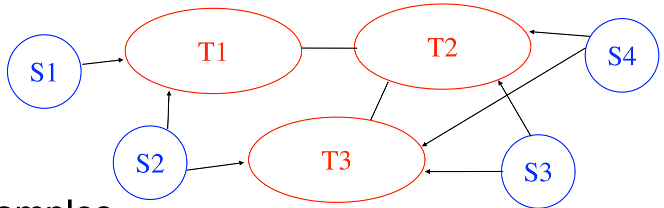
Organisation of the Internet

- Internet is composed of about 30.000 autonomous routing domains
  - A domain is a set of routers, links, hosts and local area networks under the same administrative control
    - A domain can be very large...
      - AS568: SUMNET-AS DISO-UNRRA contains 73154560 IP addresses
    - A domain can be very small...
      - AS2111: IST-ATRIUM TE Experiment a single PC running Linux...
  - Domains are interconnected in various ways
    - The interconnection of all domains should in theory allow packets to be sent anywhere
    - Usually a packet will need to cross a few ASes to reach its destination

© O. Bonaventure, 2003

Types of domains

- Transit domain
  - A transit domain allows external domains to use its own infrastructure to send packets to other domains

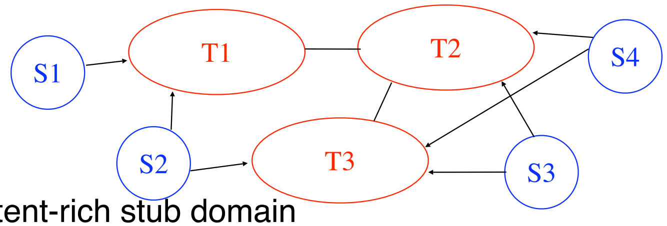


- Examples
  - UUNet, OpenTransit, GEANT, Internet2, RENATER, EQUANT, BT, Telia, Level3,...

© O. Bonaventure, 2003

Types of domains (2)

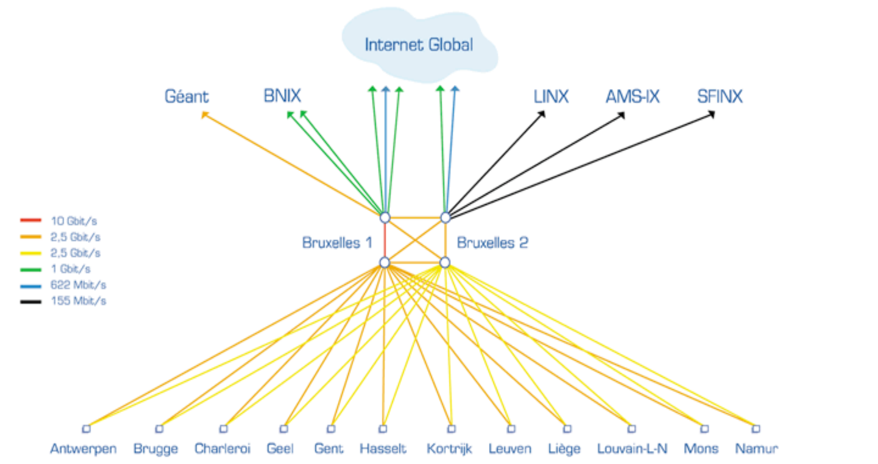
- Stub domain
  - A stub domain does not allow external domains to use its infrastructure to send packets to other domains
    - A stub is connected to at least one transit domain
      - Single-homed stub : connected to one transit domain
      - Dual-homed stub : connected to two transit domains



- Content-rich stub domain
  - Large web servers : Yahoo, Google, MSN, TF1, BBC,...
- Access-rich stub domain
  - ISPs providing Internet access via CATV, ADSL, ...

© O. Bonaventure, 2003

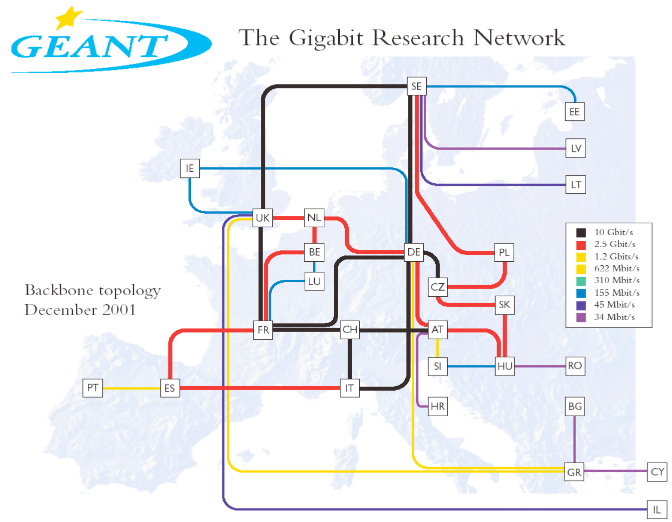
Sample network : Belnet



© O. Bonaventure, 2007

Source : http://www.belnet.be

Sample network : GEANT

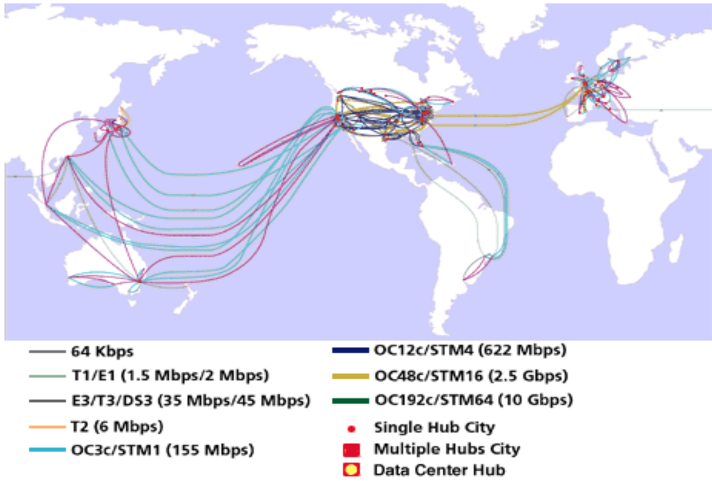


CNPP/2008.4.

© O. Bonaventure, 2007

Source http://www.dante.net

A large worldwide network : UUNet

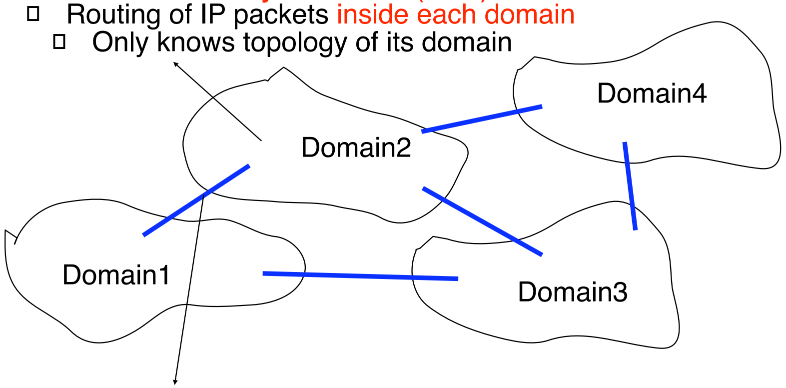


© O. Bonaventure, 2007

Source http://www.uu.net

Internet routing

- Interior Gateway Protocol (IGP)
  - Routing of IP packets inside each domain
    - Only knows topology of its domain



- Exterior Gateway Protocol (EGP)
  - Routing of IP packets between domains
    - Each domain is considered as a blackbox

© O. Bonaventure, 2003

Intradomain routing

- Goal
  - Allow routers to transmit IP packets along the best path towards their destination
    - best usually means the shortest path
      - Shortest measured in seconds or as number of hops
    - sometimes best means the less loaded path
  - Allow to find alternate routes in case of failures

- Behaviour
  - All routers exchange routing information
    - Each domain router can obtain routing information for the whole domain
    - The network operator or the routing protocol selects the cost of each link

© O. Bonaventure, 2003

Three types of Interior Gateway Protocols

- Static routing
  - Only useful in very small domains
- Distance vector routing
  - Routing Information Protocol (RIP)
    - Still widely used in small domains despite its limitations
- Link-state routing
  - Open Shortest Path First (OSPF)
    - Widely used in enterprise networks
  - Intermediate System- Intermediate-System (IS-IS)
    - Widely used by ISPs

© O. Bonaventure, 2003

Network layer

- Basics
- Routing
- IP : Internet Protocol
- Routing in IP networks
  - Internet routing organisation
  - Intradomain routing : RIP
  - Intradomain routing : OSPF
  - Interdomain routing : BGP

© O. Bonaventure, 2007

RIP
Routing Information Protocol

- Simple routing protocol that relies on distance vectors
  - Defined in RFC2453
- Principle
  - Each router periodically sends its distance vectors
    - default period : 30 seconds
    - distance vector is sent in UDP message with TTL=1 to all routers in local subnets (via IP multicast)
  - Optional extension : send a distance vector when the routing table changes
    - simple solution : send distance vector after each change
    - but some links flaps...
      - solution : send a distance vector if routing table changed and we did not send another vector within the last 5 seconds

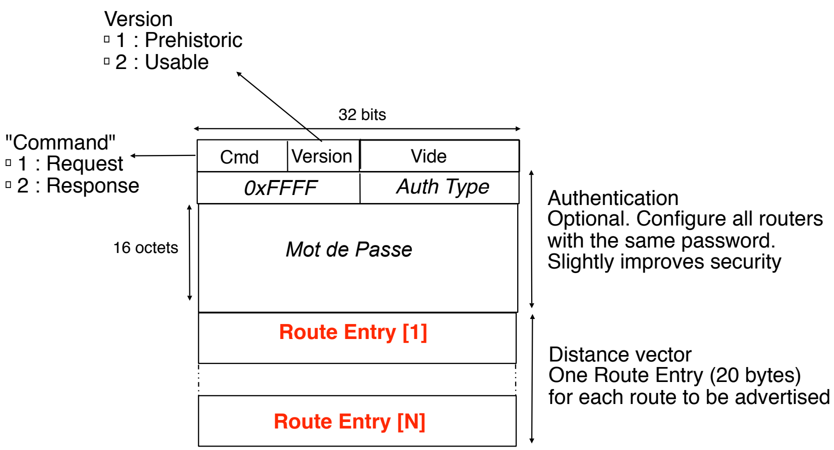
© O. Bonaventure, 2007

La version actuelle de RIP East définie dans
RFC2453 RIP Version 2. G. Malkin. November 1998

Une autre description de RIP East disponible dans :
Gary Malkin, RIP : an intra-domain routing protocol, Addison-Wesley, 2002

IP multicast East couvert dans le cours avancé. A ce stade, il suffit de considérer IP multicast (avec TTL=1) comme étant un mécanisme permettant à un routeur connecté sur un réseau local d'envoyer, en une seule transmission, un paquet qui sera reçu par tous les routeurs RIP connectés à ce réseau local.

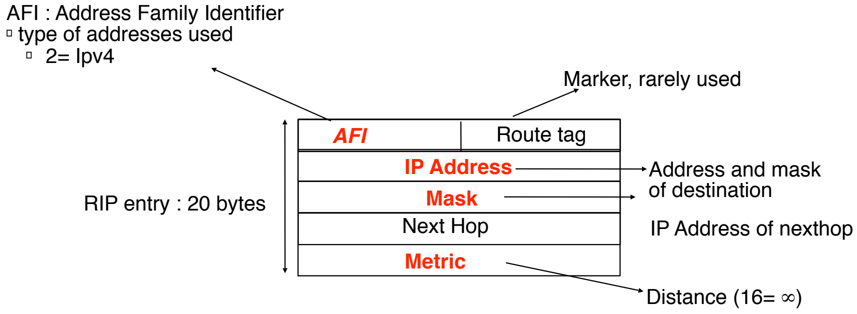
RIP : message format



- RIP messages are sent by UDP
  - port 520

© O. Bonaventure, 2007

RIP : Route Entries



- Default route
  - IP Address = 0.0.0.0, Mask = 0
- Each RIP message can contain up to 25 route entries (24 with authentication)
- If the routing table is larger than 25 entries, router will need to send several RIP messages

© O. Bonaventure, 2007

RIP timers

- Operation
  - At each expiration of its 30-sec timer, each router sends its distance vector and restarts its timer
- Problem
  - After a power failure, all routers might restart at same time and have synchronised RIP timers
    - Each router will need to process bursts of RIP messages
- Solution
  - Add some randomness to the timers
    - Restart timer after random[27.5, 32.5] instead of 30 seconds
      - commonly used technique to avoid synchronisation problems in distributed protocols

© O. Bonaventure, 2007

Ce problème de synchronisation des messages échangés par les protocoles de routage a été décrit dans :

The Synchronization of Periodic Routing Messages , Floyd, S., and Jacobson, V. IEEE/ACM Transactions on Networking, V.2 N.2, p. 122-136, April 1994.

Network layer

- Basics
- Routing
- IP : Internet Protocol
- Routing in IP networks
  - Internet routing organisation
  - Intradomain routing : RIP
  - Intradomain routing : OSPF
  - Interdomain routing : BGP

© O. Bonaventure, 2007

OSPF

- Standardised link state routing protocol

- Operation
  - Router startup
    - HELLO packets to discover neighbours
  - Update of routing tables
    - Link state packets
      - acknowledgements, sequence numbers, age
    - periodic transmission
    - transmission upon link changes
  - Database description
    - provides the list of sequence numbers of all LSPs stored by router
  - Link state Request
    - used when a router boots to request link state packets from neighbours

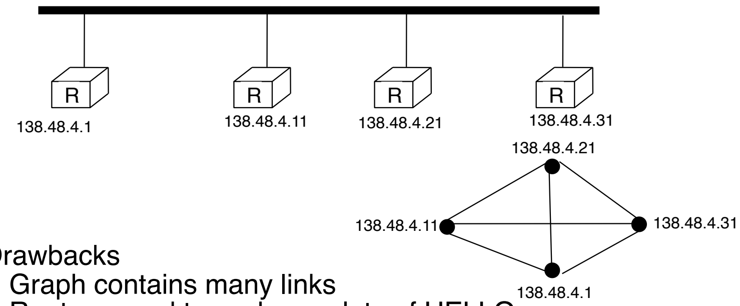
OSPF is defined in RFC2328

© O. Bonaventure, 2007

Pour plus d 'informations sur OSPF, voir
RFC2328 OSPF Version 2. J. Moy. April 1998.
ou
J. Moy, OSPF: Anatomy of an Internet Routing Protocol, Addison Wesley, 1998

OSPF details

- Routers are often attached to LANs
  - How to describe a LAN full of routers as a graph

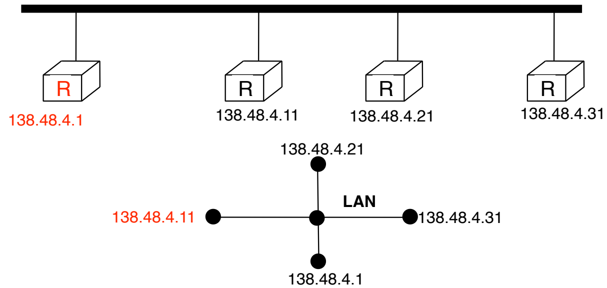


- Drawbacks
  - Graph contains many links
  - Routers need to exchange lots of HELLOs
  - Does not really describe the LAN
    - a failure of the LAN would cause a disconnection of all routers while the graph indicates a redundant topology

© O. Bonaventure, 2007

OSPF details (2)

- Solution
  - represent the LAN as a star with one router acting as the LAN
    - Designated router
      - One router is elected in the LAN to originate link state packets for the LAN
    - Adjacent router
      - Maintain adjacencies with the designated router



© O. Bonaventure, 2007

OSPF details (3)

- OSPF in large networks
  - avoid too large routing tables in OSPF routers
- Solution
  - Divide network in areas
  - Backbone area : network backbone
    - all routers connected to two or more areas belong to the backbone area
  - All non-backbone areas must be attached to the backbone area
    - at least one router inside each area must be attached to the backbone
- OSPF routing must allow any router to send packets to any other router

© O. Bonaventure, 2007

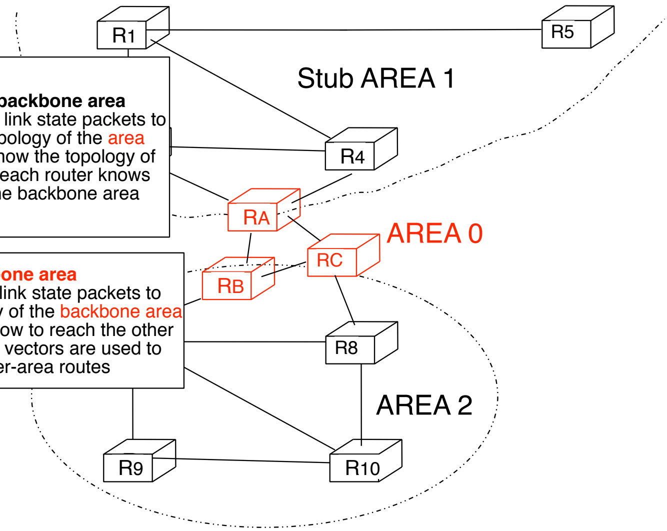
OSPF details (4)

Inside each non-backbone area

- Routers exchange link state packets to distribute the topology of the area
- Routers do not know the topology of other areas, but each router knows how to reach the backbone area

Inside backbone area

- Routers exchange link state packets to distribute the topology of the backbone area
- Each router knows how to reach the other areas and distance vectors are used to distribute inter-area routes



© O. Bonaventure, 2007

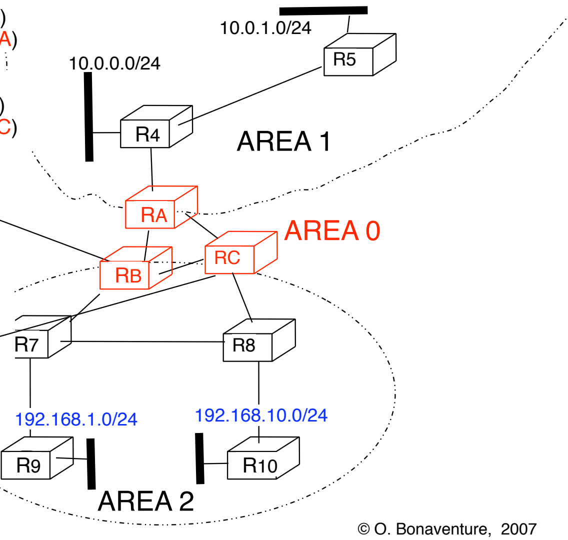
OSPF areas : Example

Routes learned by R4
- 192.168.1.0/24, distance 3 (via RA)
- 192.168.10.0/24, distance 3 (via RA)

Routes chosen par RA
- 192.168.1.0/24, distance 3 (via RB)
- 192.168.10.0/24, distance 3 (via RC)

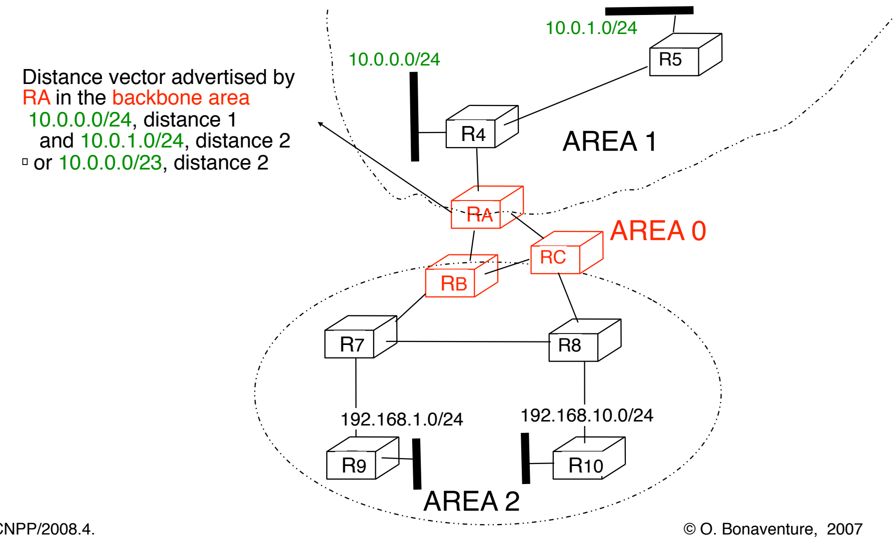
Distance vectors advertised by RB in backbone area
- 192.168.1.0/24, distance 2 and 192.168.10.0/24, distance 3

Distance vectors advertised by RB in backbone area
- 192.168.1.0/24, distance 3 and 192.168.10.0/24, distance 2



© O. Bonaventure, 2007

Areas OSPF : Example (2)



CNPP/2008.4.

© O. Bonaventure, 2007

Cet exemple ne présente que les informations de routage concernant les sous-réseaux indiqués sur le schéma. Les adresses des routeurs sont ignorées pour simplifier l'exemple.

Network layer

- Basics
- Routing
- IP : Internet Protocol
- Routing in IP networks
  - Internet routing organisation
  - Intradomain routing : RIP
  - Intradomain routing : OSPF
  - Interdomain routing : BGP

© O. Bonaventure, 2007

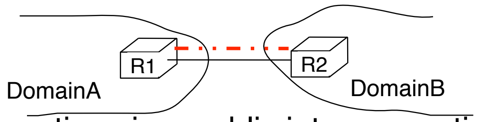
Interdomain routing

- Goals
  - Allow to transmit IP packets along the best path towards their destination through several transit domains while taking into account the routing policies of each domain without knowing the detailed topology of those domains
    - From an interdomain viewpoint, best path often means *cheapest* path
    - Each domain is free to specify inside its routing policy the domains for which it agrees to provide a transit service and the method it uses to select the best path to reach each destination

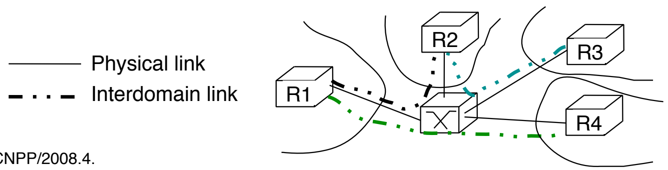
© O. Bonaventure, 2003

Types of interdomain links

- Two types of interdomain links
  - Private link
    - Usually a leased line between two routers belonging to the two connected domains



  - Connection via a public interconnection point
    - Usually Gigabit or higher Ethernet switch that interconnects routers belonging to different domains



© O. Bonaventure, 2003

For more information on the organization of the Internet, see :

G. Huston, Peerings and settlements, Internet Protocol Journal, Vol. 2, N1 et 2, 1999, http://www.cisco.com/warp/public/759/ipj_Volume2.html

For more information on interconnection points or Internet exchanges, see :

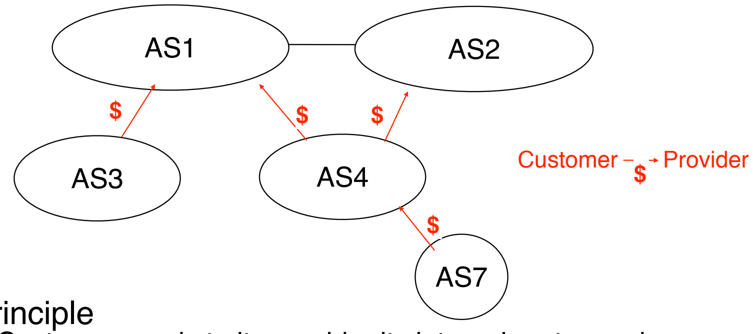
http://www.euro-ix.net/
http://www.ripe.net/ripe/wg/eix/index.html
http://www.ep.net/ep-main.html

Routing policies

- In theory BGP allows each domain to define its own routing policy...
- In practice there are two common policies
  - customer-provider peering
    - **Customer c** buys Internet connectivity from **provider P**
  - shared-cost peering
    - **Domains x** and **y** agree to exchange packets by using a direct link or through an interconnection point

© O. Bonaventure, 2003

Customer-provider peering

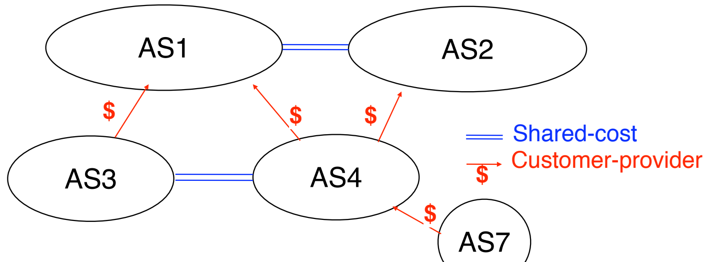


- Principle
  - Customer sends to its provider its internal routes and the routes learned from its own customers
    - Provider will advertise those routes to the entire Internet to allow anyone to reach the Customer
  - Provider sends to its customers all known routes
    - Customer will be able to reach anyone on the Internet

© O. Bonaventure, 2003

On link AS7-AS4
AS7 advertises its own routes to AS4
AS4 advertises to AS7 the routes that allow to reach the entire Internet
On link AS4-AS2
AS4 advertises its own routes and the routes belonging to AS7
AS2 advertises the routes that allow to reach the entire Internet

Shared-cost peering



- Principle
  - PeerX sends to PeerY its internal routes and the routes learned from its own customers
    - PeerY will use shared link to reach PeerX and PeerX's customers
    - PeerX's providers are not reachable via the shared link
  - PeerY sends to PeerX its internal routes and the routes learned from its own customers
    - PeerX will use shared link to reach PeerY and PeerY's customers
    - PeerY's providers are not reachable via the shared link

© O. Bonaventure, 2003

On link AS3-AS4

AS3 advertises its internal routes

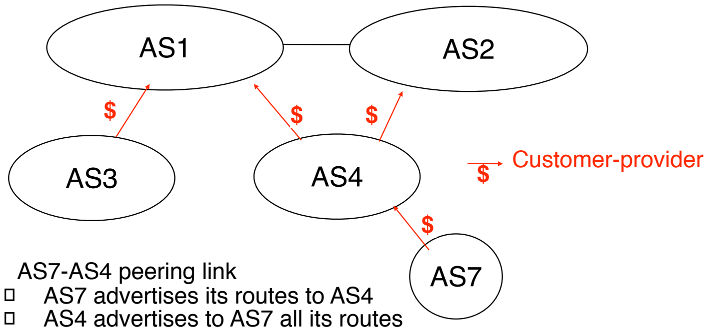
AS4 advertises its internal routes and the routes learned from AS7 (its customer)

On link AS1-AS2

AS1 advertises its internal routes and the routes received from AS3 and AS4 (its customers)

AS2 advertises its internal routes and the routes learned from AS74(its customer)

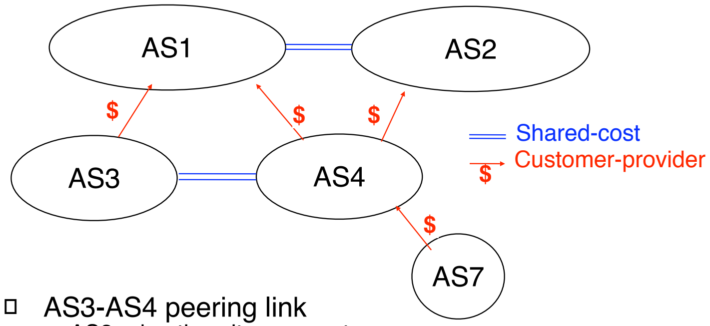
Customer-provider peering : example



- AS7-AS4 peering link
  - AS7 advertises its routes to AS4
  - AS4 advertises to AS7 all its routes
- AS4-AS2 peering link
  - AS4 advertises its own routes et those of its customers (AS7)
  - AS2 advertises to AS2 all known routes

© O. Bonaventure, 2007

Shared-cost peering : example



- AS3-AS4 peering link
  - AS3 advertises its own routes
  - AS4 advertises its own routes and those received from its clients (AS7)
- AS1-AS2 peering link
  - AS1 advertises its own routes and those received from its clients (AS3 and AS4)
  - AS1 advertises its own routes and those received from its clients (AS4)

© O. Bonaventure, 2007

Routing policies

- A domain specifies its routing policy by defining on each BGP router two sets of filters for each peer
  - Import filter
    - Specifies which routes can be accepted by the router among all the received routes from a given peer
  - Export filter
    - Specifies which routes can be advertised by the router to a given peer
- Filters can be defined in RPSL
  - Routing Policy Specification Language
    - defined in RFC2622 and examples in RFC2650
      - See also http://www.ripe.net/ripencc/pub-services/whois.html

CNPP/2008.4.

© O. Bonaventure, 2003

RFC 2622 Routing Policy Specification Language (RPSL). C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, M. Terpstra. June 1999.

RFC 2650 Using RPSL in Practice. D. Meyer, J. Schmitz, C. Orange, M. Prior, C. Alaettinoglu. August 1999.

Internet Routing Registries contain the routing policies of various ISPs, see :

http://www.ripe.net/ripencc/pub-services/whois.html
http://www.arin.net/whois/index.html
http://www.apnic.net/apnic-bin/whois.pl

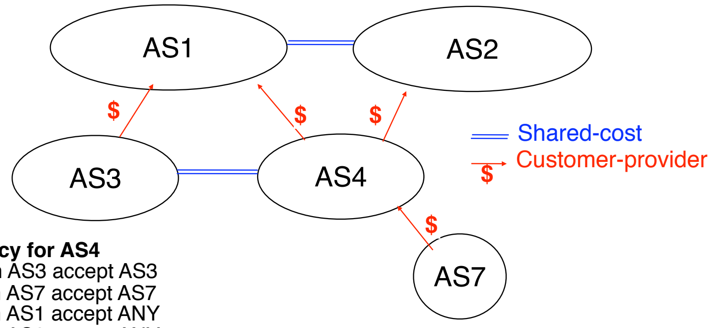
RPSL

- Simple import policies
  - Syntax
    - `import: from AS# accept list_of_AS`
  - Examples
    - `Import: from Belgacom accept Belgacom WIN`
    - `Import: from Provider accept ANY`

- Simple export policies
  - Syntax
    - `Export: to AS# announce list_of_AS`
  - Example
    - `Export: to Customer announce ANY`
    - `Export: to Peer announce Customer1 Customer2`

© O. Bonaventure, 2003

Routing policies
Simple example with RPSL



Import policy for AS4
Import: from AS3 accept AS3
import: from AS7 accept AS7
import: from AS1 accept ANY
import: from AS2 accept ANY

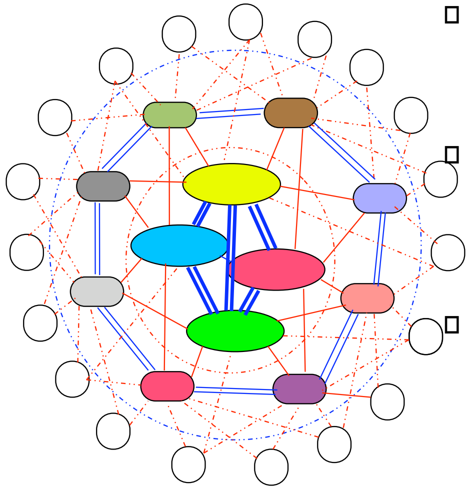
Export policy for AS4
export: to AS3 announce AS4 AS7
export: to AS7 announce ANY
export: to AS1 announce AS4 AS7
export: to AS2 announce AS4 AS7

Import policy for AS7
Import: from AS4 accept ANY

Export policy for AS4
export: to AS4 announce AS7

© O. Bonaventure, 2003

The organisation of the Internet



- Tier-1 ISPs
  - Dozen of large ISPs interconnected by shared-cost
  - Provide transit service
    - Uunet, Level3, OpenTransit, ...
- Tier-2 ISPs
  - Regional or National ISPs
  - Customer of T1 ISP(s)
  - Provider of T3 ISP(s)
  - shared-cost with other T2 ISPs
    - France Telecom, BT, Belgacom
- Tier-3 ISPs
  - Smaller ISPs, Corporate Networks, Content providers
  - Customers of T2 or T1 ISPs
  - shared-cost with other T3 ISPs

CNPP/2008.4.

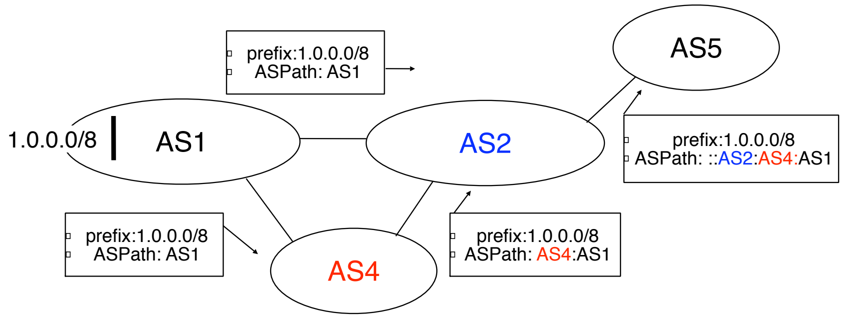
© O. Bonaventure, 2003

See :

L. Subramanian, S. Agarwal, J. Rexford, and RH Katz. Characterizing the Internet hierarchy from multiple vantage points. In IEEE INFOCOM, 2002

The Border Gateway Protocol

- Principle
  - Path vector protocol
    - BGP router advertises its best route to each destination



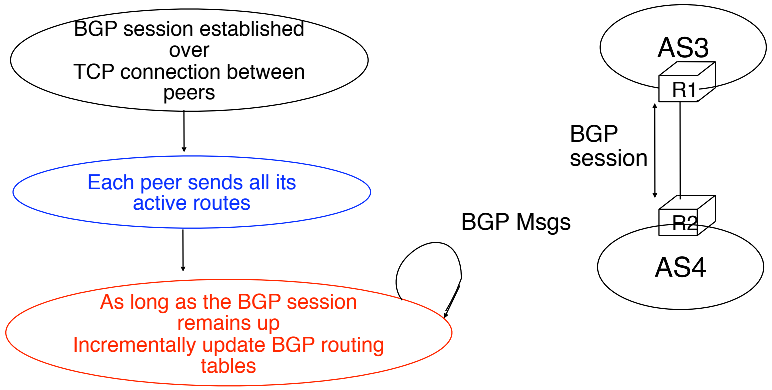
- ... with incremental updates
  - Advertisements are only sent when their content changes

BGP is defined in RFC4271

© O. Bonaventure, 2003

BGP : Principles of operation

- Principles
  - BGP relies on the incremental exchange of path vectors



© O. Bonaventure, 2003

BGP : Principles of operation (2)

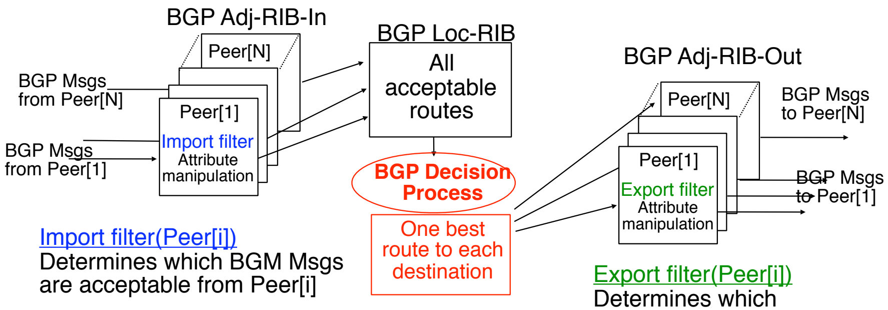
- Simplified model of BGP
  - 2 types of BGP path vectors

  - UPDATE
    - Used to announce a route towards one prefix
    - Content of UPDATE
      - Destination address/prefix
      - Interdomain path used to reach destination (AS-Path)
      - Nexthop (address of the router advertising the route)

  - WITHDRAW
    - Used to indicate that a previously announced route is not reachable anymore
    - Content of WITHDRAW
      - Unreachable destination address/prefix

© O. Bonaventure, 2003

Conceptual model of a BGP router



Import filter(Peer[i])
Determines which BGM Msgs are acceptable from Peer[i]

Export filter(Peer[i])
Determines which routes can be sent to Peer[i]

BGP Routing Information Base
Contains all the acceptable routes learned from all Peers + internal routes

- BGP decision process selects *the* best route towards each destination

© O. Bonaventure, 2003

Where do the routes advertised by BGP routers come from ?

- Learned from another BGP router
  - Each BGP router advertises best route towards each destination
- Static route
  - Configured manually on the router
    - Ex : The BGP router at UCL advertises 130.104.0.0/16
    - Drawback
      - Requires manual configuration
    - Advantage
      - BGP advertisements are stable
- Learned from an intradomain routing protocol
  - BGP might try to aggregate the route before advertising it
  - Advantage :
    - BGP advertisements correspond to network status
  - Drawback
    - Routing instabilities inside a domain might propagate in Internet

© O. Bonaventure, 2007

BGP : Session Initialization

```
Initialize_BGP_Session(RemoteAS, RemoteIP)
{ /* Initialize and start BGP session */
/* Send BGP OPEN Message to RemoteIP on port 179*/
/* Follow BGP state machine */

/* advertise local routes and routes learned from peers*/
foreach (destination=d inside RIB)
 {
  B=build_BGP_UPDATE(d);
  S=apply_export_filter(RemoteAS,B);
  if (S<>NULL)
     { /* send UPDATE message */
       send_UPDATE(S,RemoteAS, RemoteIP)
     }
 }
/* entire RIB was sent */
/* new UPDATE will be sent only to reflect local or distant
   changes in routes */
...
}
```


© O. Bonaventure, 2007

Events during a BGP session

1. Addition of a new route to RIB
   - A new internal route was added on local router
     - static route added by configuration
     - Dynamic route learned from IGP
   - Reception of UPDATE message announcing a new or modified route
2. Removal of a route from RIB
   - Removal of an internal route
     - Static route is removed from router configuration
     - Intradomain route declared unreachable by IGP
   - Reception of WITHDRAW message
3. Loss of BGP session
   - All routes learned from this peer removed from RIB

© O. Bonaventure, 2003

Export and Import filters

```
BGPMsg Apply_export_filter(RemoteAS, BGPMsg)
{ /* check if Remote AS already received route */
if (RemoteAS isin BGPMsg.ASPath)
   BGPMsg==NULL;
/* Many additional export policies can be configured : */
/* Accept or refuse the BGPMsg */
/* Modify selected attributes inside BGPMsg */
}

BGPMsg apply_import_filter(RemoteAS, BGPMsg)
{ /* check that we are not already inside  ASPath */
 if (MyAS isin BGPMsg.ASPath)
   BGPMsg==NULL;
/* Many additional import policies can be configured : */
/* Accept or refuse the BGPMsg */
/* Modify selected attributes inside BGPMsg */
}
```


© O. Bonaventure, 2003

In the above export filter, we assume that the BGP sender does not send to PeerX the routes learned from this peer. This behavior is not required by the BGP specification, but is a common optimization, often called sender-side loop detection.

The check for the presence of the localAS number in the routes learned is specified in the BGP RFC.

BGP : Processing of UPDATES

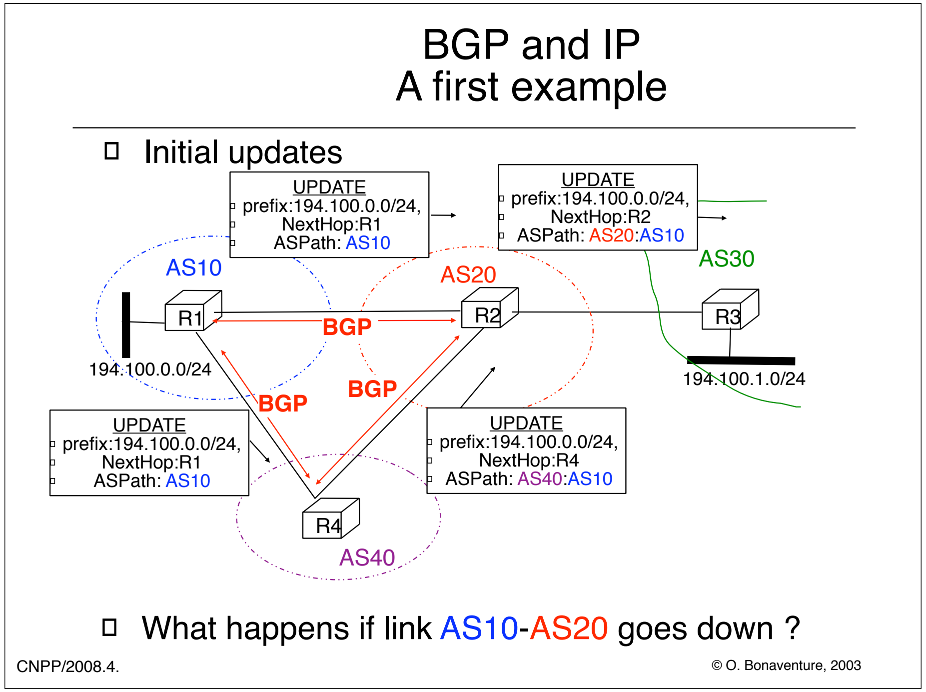
```
Recvd_BGPMsg(Msg, RemoteAS)
{
 B=apply_import_filer(Msg,RemoteAS);
 if (B==NULL) /* Msg not acceptable */
     exit();
 if IsUPDATE(Msg)
 {
  Old_Route=BestRoute(Msg.prefix);
  Insert_in_RIB(Msg);
  Run_Decision_Process(RIB);
  if (BestRoute(Msg.prefix)<>Old_Route)
  { /* best route changed */
    B=build_BGP_Message(Msg.prefix);
    S=apply_export_filter(RemoteAS,B);
    if (S<>NULL) /* announce best route */
     send_UPDATE(S,RemoteAS);
    else if (Old_Route<>NULL)
     send_WITHDRAW(Msg.prefix);
 } ...
```


© O. Bonaventure, 2003

BGP : Processing of WITHDRAW

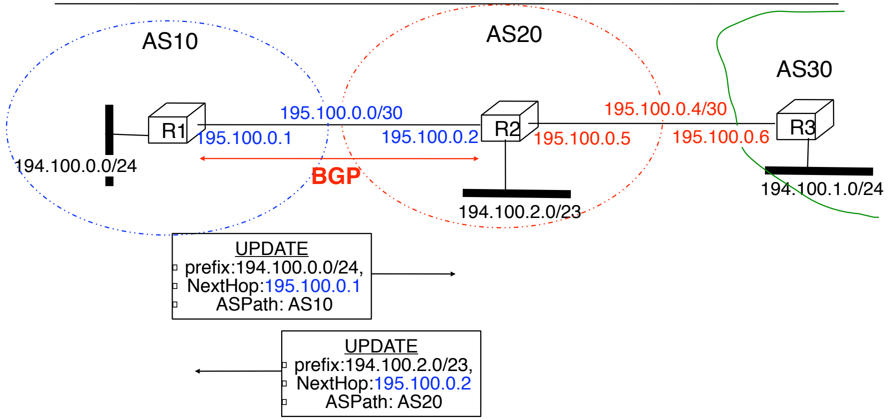
```
Recvd_Msg(Msg, RemoteAS)
...
if IsWITHDRAW(Msg)
 {
  Old_Route=BestRoute(Msg.prefix);
  Remove_from_RIB(Msg);
  Run_Decision_Process(RIB);
  if (Best_Route(Msg.prefix)<>Old_Route)
  { /* best route changed */
    B=build_BGP_Message(d);
    S=apply_export_filter(RemoteAS,B);
    if (S<>NULL) /* still one best route */
       send_UPDATE(S,RemoteAS, RemoteIP);
    else if(Old_Route<>NULL)/* no best route anymore */
       send_WITHDRAW(Msg.prefix,RemoteAS,RemoteIP);
  }
 }
}
```


© O. Bonaventure, 2003



If link AS10-AS20 goes down, AS20 will not consider anymore the path learned from AS10. It will thus remove this path from its routing table and will instead select the path learned from AS40. This will force AS20 to send the following UPDATE to AS30 :

BGP and IP
A second example



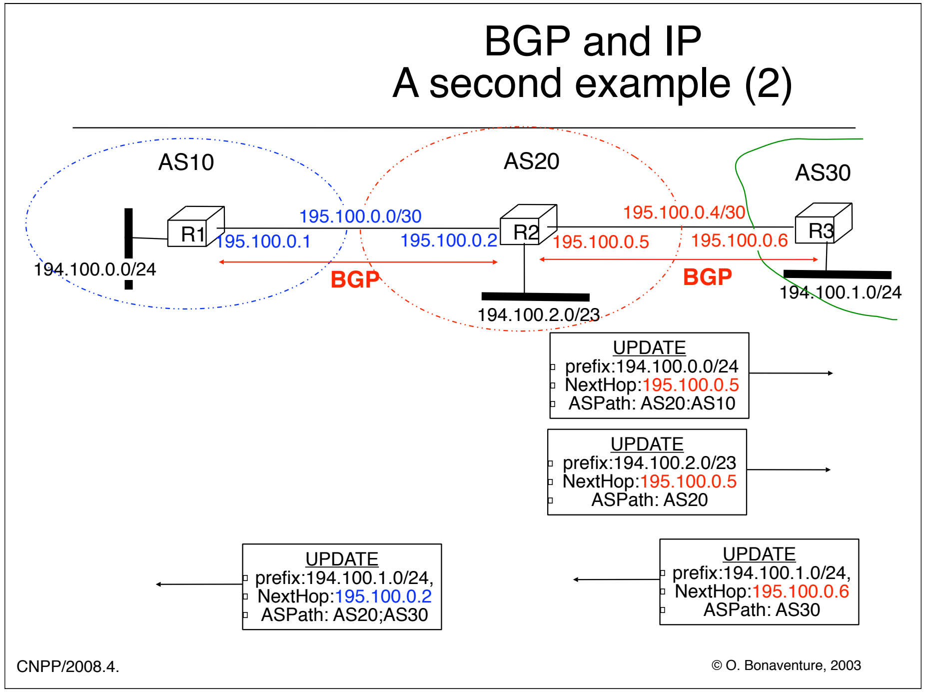
- Main Path attributes of UPDATE message
  - NextHop : IP address of router used to reach destination
  - ASPath : Path followed by the route advertisement

© O. Bonaventure, 2003

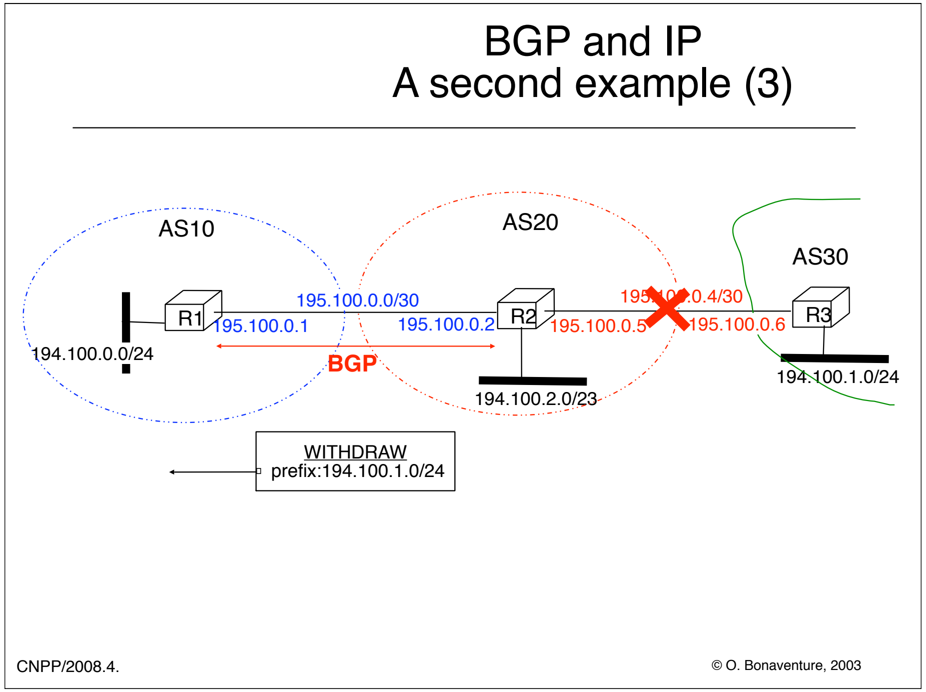
In this example, we only consider the BGP messages concerning the following IP networks :194.100.0.0/24, 194.100.1.0.0/24 and 194.100.2.0/23. Routes concerning networks 195.100.* also need to be distributed in practice, but they are not considered in the example.

The UPDATE message carries the ASPath in order to be able to detect routing loops.

The nexthop information in the UPDATE is often equal to the IP address of the router advertising the route, but it can be sometimes useful to advertise as a next hop another IP address than the address of the router producing the BGP UPDATE message. For example, a router supporting BGP could advertise a route on behalf of another router who cannot run the BGP protocol.

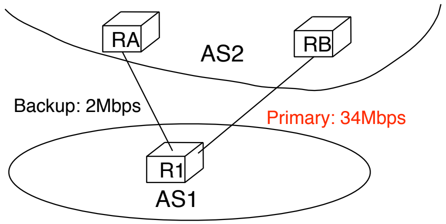


In this example, we only consider the BGP messages concerning the following IP networks :194.100.0.0/24, 194.100.1.0.0/24 and 194.100.2.0/23. Routes concerning networks 195.100.* also need to be distributed, but they are not considered in the example.

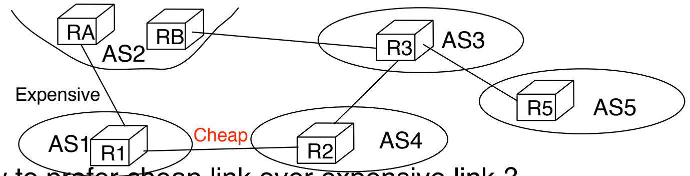


In this example, we only consider the BGP messages concerning the following IP networks :194.100.0.0/24, 194.100.1.0.0/24 and 194.100.2.0/23. Routes concerning networks 195.100.* also need to be distributed, but they are not considered in the example.

How to prefer some routes over others ?



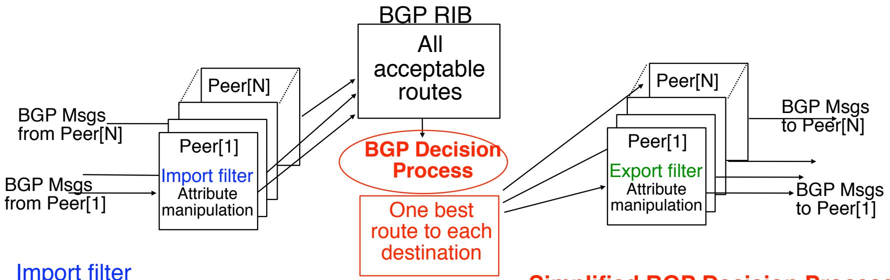
- How to ensure that packets will flow on primary link ?



- How to prefer cheap link over expensive link ?

© O. Bonaventure, 2003

How to prefer some routes over others (2) ?

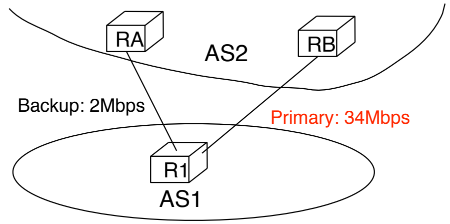


Import filter
- Selection of acceptable routes
- Addition of `local-pref` attribute inside received BGP Msg
  - Normal quality route : `local-pref`=100
  - Better than normal route :`local-pref`=200
  - Worse than normal route :`local-pref`=50

Simplified BGP Decision Process
- Select routes with highest `local-pref`
- If there are several routes, choose routes with the shortest ASPath
- If there are still several routes tie-breaking rule

© O. Bonaventure, 2003

How to prefer some routes over others (3) ?



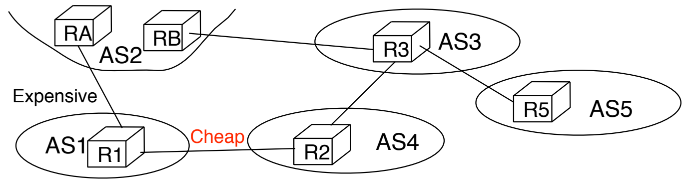
RPSL-like policy for AS1
aut-num: AS1
import: from AS2 RA at R1 set localpref=100;
        from AS2 RB at R1 set localpref=200;
        accept ANY
export: to AS2 RA at R1 announce AS1
        to AS2 RB at R1 announce AS1

RPSL-like policy for AS2
aut-num: AS2
import: from AS1 R1 at RA set localpref=100;
        from AS1 R1 at RB set localpref=200;
        accept AS1
export: to AS1 R1 at RA announce ANY
        to AS2 R1 at RB announce ANY

© O. Bonaventure, 2003

Note that in RPSL, the set localpref construct does not exist. It is replaced with action preference=x. Unfortunately, in RPSL the routes with the lowest preference are preferred. RPSL uses thus the opposite of local-pref....

How to prefer some routes over others (4) ?



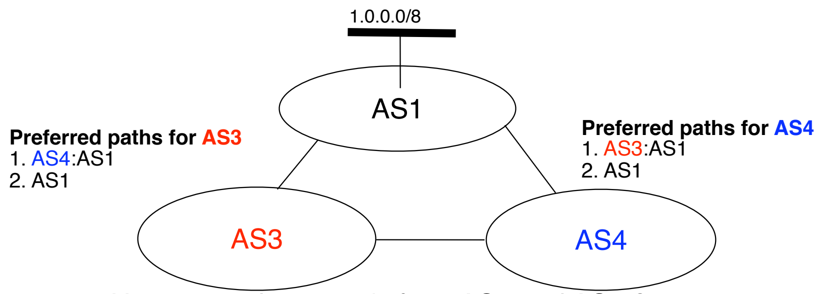
RPSL policy for AS1
aut-num: AS1
import: from AS2 RA at R1 set localpref=100;
from AS4 R2 at R1 set localpref=200;
accept ANY
export: to AS2 RA at R1 announce AS1
to AS4 R2 at R1 announce AS1

- AS1 will prefer to send packets over the cheap link
- But the flow of the packets destined to AS1 will depend on the routing policy of the other domains

© O. Bonaventure, 2003

Limitations of `local-pref`

- In theory
  - Each domain is free to define its order of preference for the routes learned from external peers

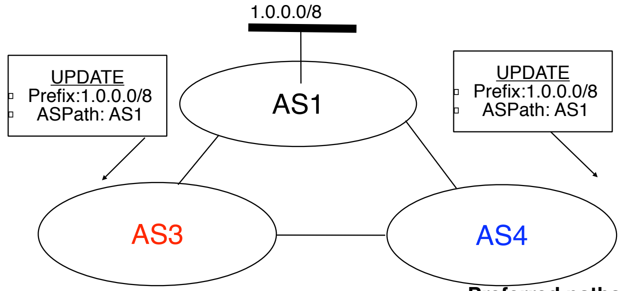


  - How to reach 1.0.0.0/8 from AS3 and AS4 ?

© O. Bonaventure, 2003

Limitations of `local-pref` (2)

- AS1 sends its UPDATE messages ...



Preferred paths for AS3
1. AS4:AS1
2. AS1

Preferred paths for AS4
1. AS3:AS1
2. AS1

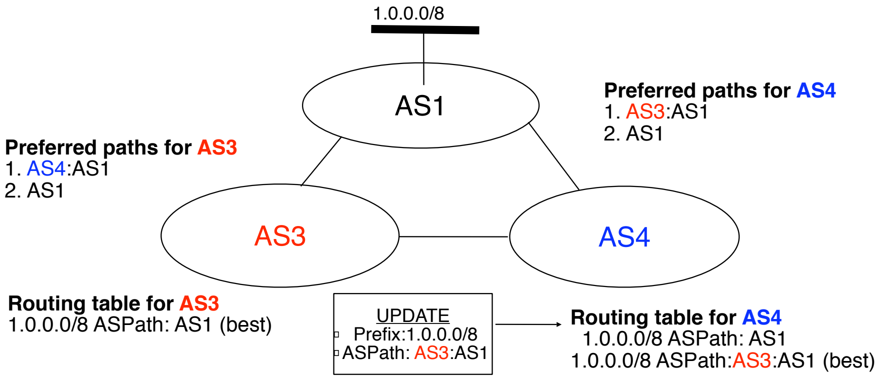
Routing table for AS3
1.0.0.0/8 ASPath: AS1 (best)

Routing table for AS4
1.0.0.0/8 ASPath: AS1 (best)

© O. Bonaventure, 2003

Limitations of `local-pref` (3)

- First possibility
  - AS3 sends its UPDATE first...

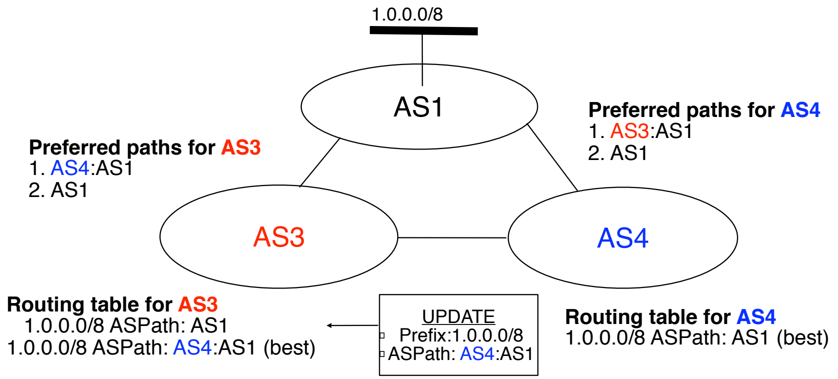


- Stable route assignment

© O. Bonaventure, 2003

Limitations of `local-pref` (4)

- Second possibility
  - AS4 sends its UPDATE first...

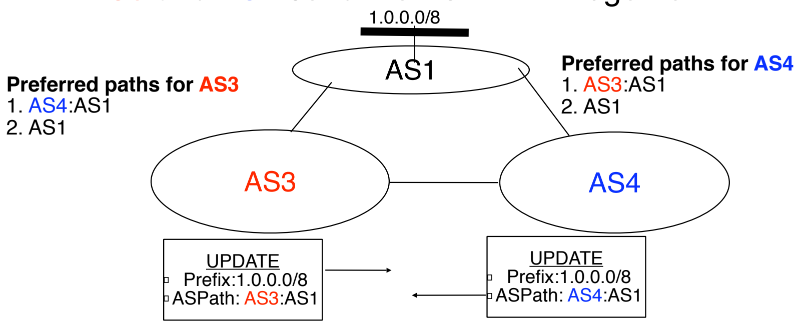


- Another (but different) stable route assignment

© O. Bonaventure, 2003

Limitations of `local-pref` (5)

- Third possibility
  - AS3 and AS4 send their UPDATE together...

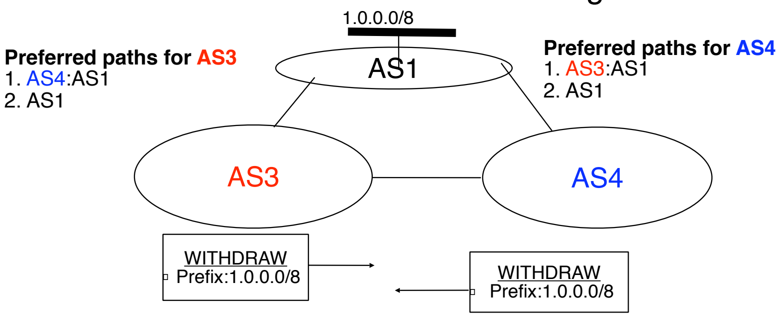


- AS3 prefers the indirect path and will thus send withdraw since the chosen best path is via AS4
- AS4 prefers the indirect path and will thus send withdraw since the chosen best path is via AS3

© O. Bonaventure, 2003

Limitations of `local-pref` (6)

- Third possibility (cont.)
  - AS3 and AS4 send their UPDATE together...

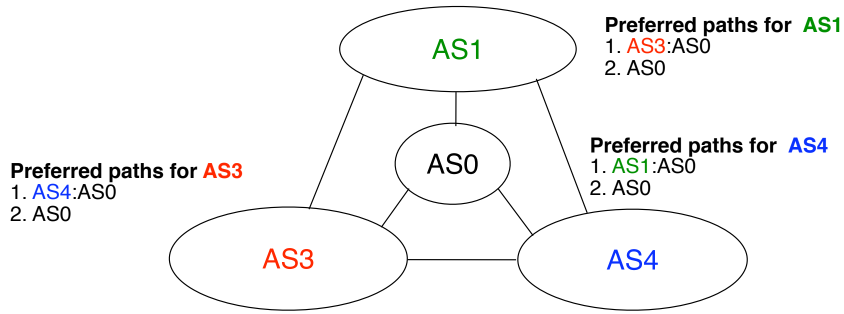


  - AS3 learns that the indirect route is not available anymore
    - AS3 will reannounce its direct route...
  - AS4 learns that the indirect route is not available anymore
    - AS4 will reannounce its direct route...

© O. Bonaventure, 2003

More limitations of `local-pref`

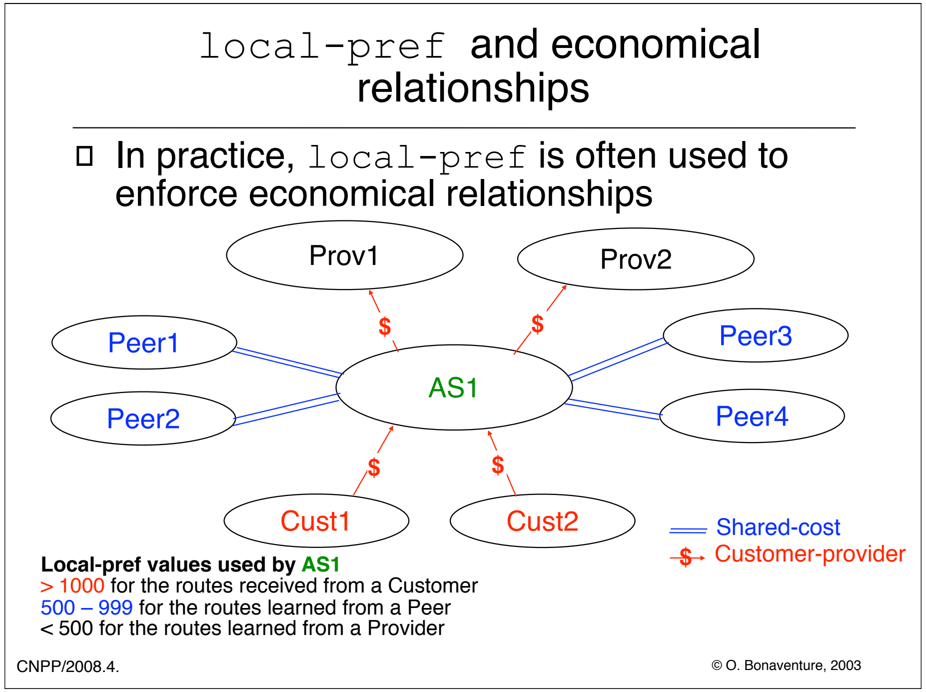
- Unfortunately, interdomain routing may not converge at all in some cases...



- How to reach a destination inside AS0 in this case ?

© O. Bonaventure, 2003

In practice, the exchange of BGP UPDATE messages will cease due to the utilization of timers by BGP routers and the routing will stabilize on one of the two stable route assignments.



181

This local-pref settings corresponds to the economical relationships between the various ASes.
Since AS1 is paid to carry packets towards Cust1 and Cust2, it will select a route towards those networks whenever possible.
Since AS1 does not need to pay to carry packets towards Peer1-4, AS1 will select a route towards those networks whenever possible.
AS1 will only utilize the routes receive from its providers when there is no other choice.

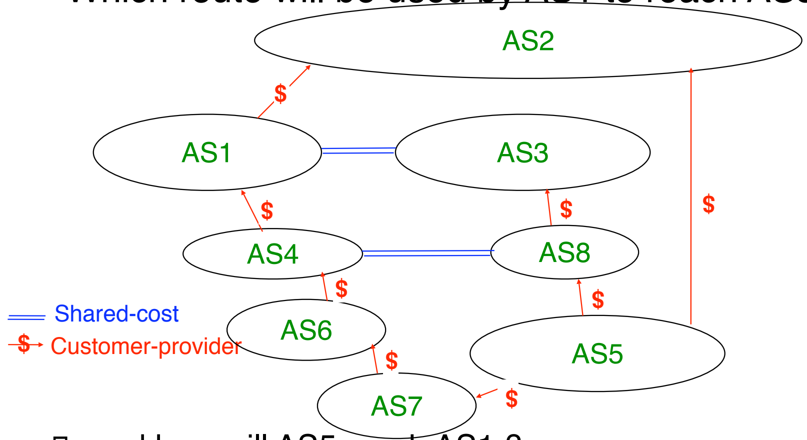
It is shown in the following papers that this way of utilizing the local-pref attribute leads to stable BGP routes :
Lixin Gao, Timothy G. Griffin, and Jennifer Rexford, "Inherently safe backup routing with BGP," Proc. IEEE INFOCOM, April 2001
Lixin Gao and Jennifer Rexford, "Stable Internet routing without global coordination," IEEE/ACM Transactions on Networking, December 2001, pp. 681-692

The RPSL policy of AS1 could be as follows :
RPSL policy for AS1

```
aut-num: AS1
import:     from  Cust1 action set localpref=200; accept Cust1
            from  Cust2 action set localpref=200; accept Cust2
            from  Peer1 action set localpref=150; accept Peer1
            from  Peer2 action set localpref=160; accept Peer2
            from  Peer3 action set localpref=170; accept Peer3
            from  Peer4 action set localpref=180; accept Peer4
            from  Prov1 action set localpref=100; accept ANY
```

Consequence of this utilisation of `local-pref`

- Which route will be used by AS1 to reach AS5 ?



- and how will AS5 reach AS1 ?

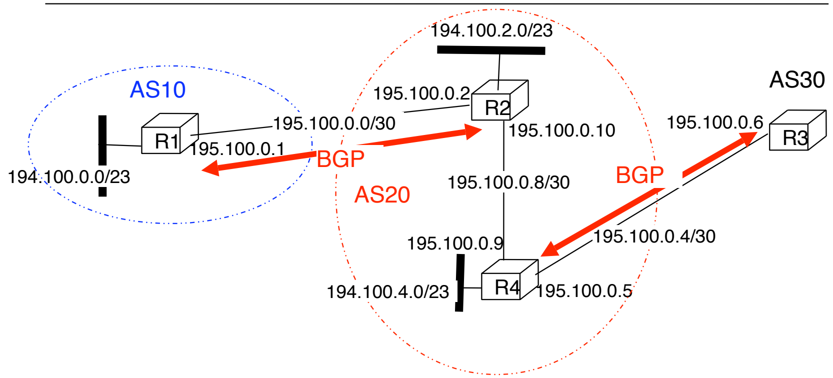
Internet paths are often asymmetrical

© O. Bonaventure, 2003

Due to the utilization of the local-pref attribute, some paths on the Internet are longer than their optimum length, see :

Lixin Gao and Feng Wang , The Extent of AS Path Inflation by Routing Policies, GlobalInternet 2002

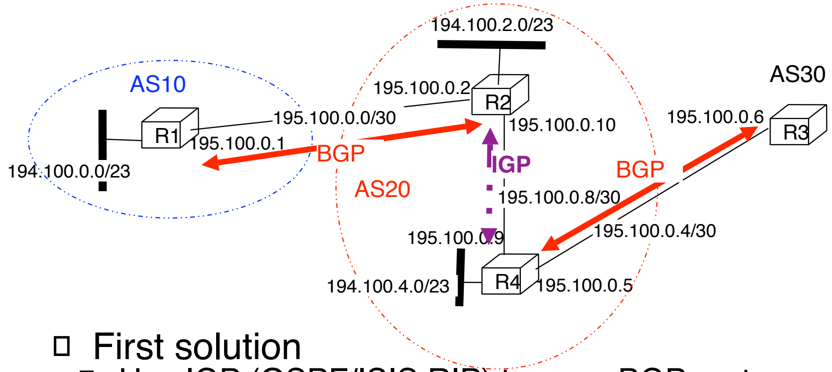
BGP and IP
Second example



- Problem
  - How can R2 (resp. R4) advertise to R4 (resp. R2) the routes learned from AS10 (resp. AS30) ?

© O. Bonaventure, 2003

BGP and IP
Second example (2)

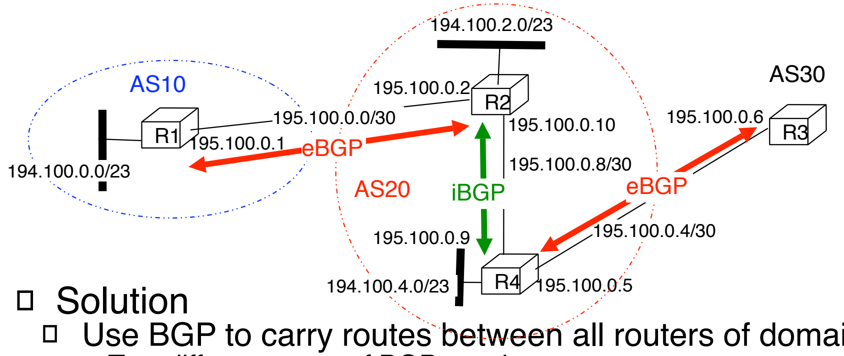


- First solution
  - Use IGP (OSPF/ISIS,RIP) to carry BGP routes
- Drawbacks
  - IGP may not be able to support so many routes
  - IGP does not carry BGP attributes like ASPath !

© O. Bonaventure, 2003

There are regularly discussions on whether the redistribution of BGP routes in an IGP should be removed from BGP implementations. See e.g. http://www.irbs.net/internet/nanog/0210/0140.html

iBGP and eBGP



- Solution
  - Use BGP to carry routes between all routers of domain
    - Two different types of BGP sessions
    - eBGP between routers belonging to different ASes
    - iBGP between each pair of routers belonging to the same AS
      - Each BGP router inside ASx maintains an iBGP session with all other BGP routers of ASx (full iBGP mesh)
      - Note that the iBGP sessions do not necessarily follow physical topology

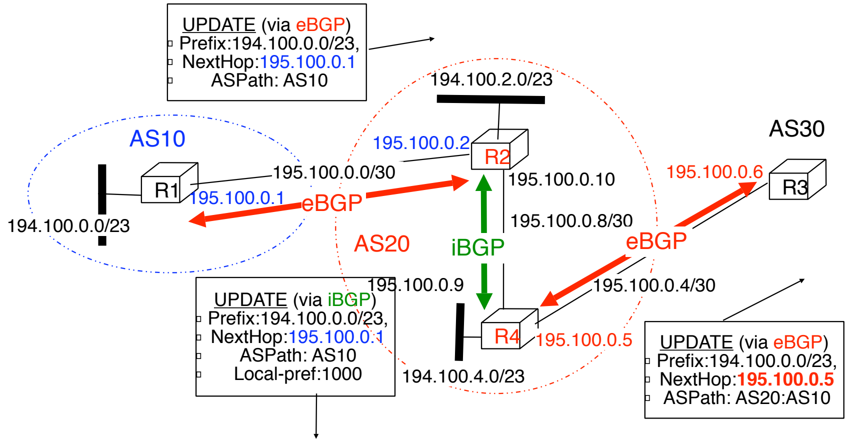
© O. Bonaventure, 2003

iBGP versus eBGP

- Differences between iBGP and eBGP
  - `local-pref` attribute is only carried inside messages sent over iBGP session
  - Over an eBGP session, a router only advertises its best route towards each destination
    - Usually, import and export filters are defined for each eBGP session
  - Over an iBGP session, a router advertises only its best routes learned over eBGP sessions
    - A route learned over an iBGP session is *never* advertised over another iBGP session
    - Usually, no filter is applied on iBGP sessions

© O. Bonaventure, 2003

iBGP and eBGP : Example



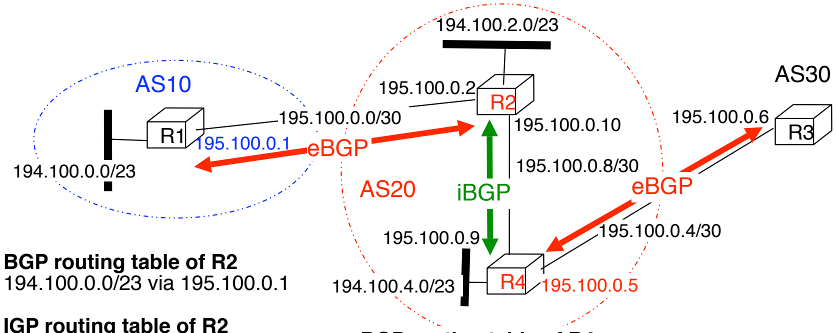
- Note that the next-hop and the AS-Path of BGP update messages are only updated when sent over an eBGP session

CNPP/2008.4.

© O. Bonaventure, 2003

In some cases, it is useful to update the value of BGP nexthop when an UPDATE message is received over an eBGP session. Most BGP implementations support this feature with a command often called “nexthop-self”. Although this command is useful in some practical situations, we do not discuss its utilization in this course.

iBGP and eBGP Packet Forwarding



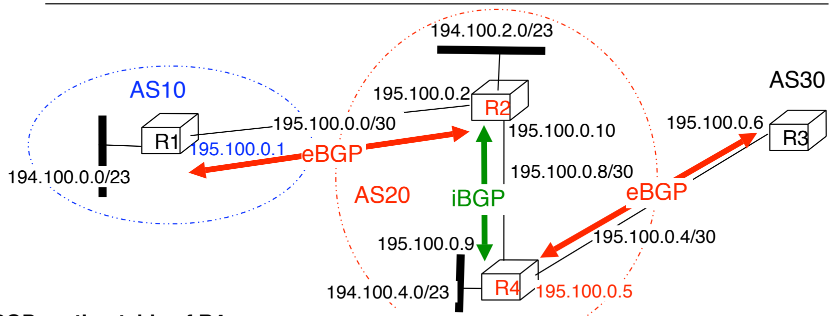
BGP routing table of R2
194.100.0.0/23 via 195.100.0.1

IGP routing table of R2
195.100.0.0/30 West
195.100.0.4/30 via 195.100.0.9
195.100.0.8/30 South
194.100.0.4/23 via 195.100.0.9
194.100.2.0/23 North

BGP routing table of R4
194.100.0.0/23 via 195.100.0.1

IGP routing table of R4
195.100.0.0/30 via 195.100.0.10
195.100.0.4/30 East
195.100.0.8/30 North
194.100.2.0/23 via 195.100.0.10
194.100.4.0/23 West

iBGP and eBGP Packet Forwarding (2)



BGP routing table of R4
194.100.0.0/23 via 195.100.0.1

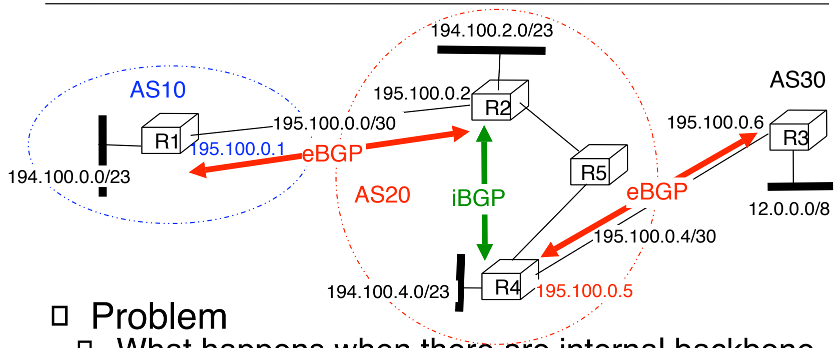
IGP routing table of R4
195.100.0.0/30 via **195.100.0.10**
195.100.0.4/30 East
195.100.0.8/30 North
194.100.2.0/23 via 195.100.0.10
194.100.4.0/23 West

Forwarding of R4
194.100.0.0/23 via **195.100.0.10**
195.100.0.0/30 via 195.100.0.10
195.100.0.4/30 East
195.100.0.8/30 North
194.100.2.0/23 via 195.100.0.10
194.100.4.0/23 West

© O. Bonaventure, 2003

The Forwarding table of a router is thus built on the basis of both the IGP table and the BGP table.

Using non-BGP routers

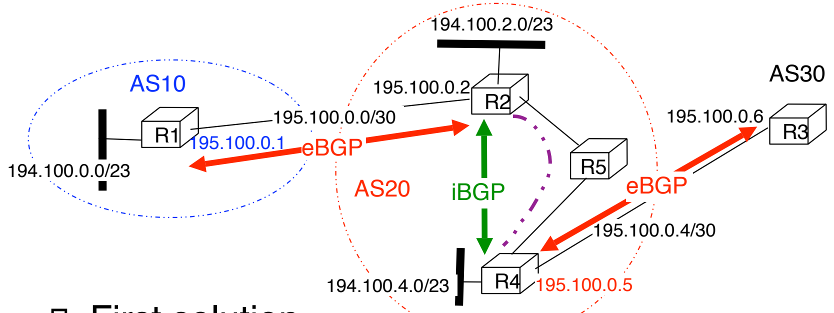


- Problem
  - What happens when there are internal backbone routers between BGP routers inside an AS ?
    - iBGP session between BGP routers is easily established when IGP is running since iBGP runs over TCP connection
    - How to populate the routing table of the backbone routers to ensure that they will be able to route any IP packet ?

© O. Bonaventure, 2003

In this example, the iBGP session between R2 and R4 would be established over a TCP connection. The packets of this connection with source/dest R2 or R4 would be routed from R2 to R4 and the opposite via R5 by using the IGP table. Thus, the IP addresses of the routers must be distributed by the IGP.

Using non-BGP routers (2)



- First solution
  - Use tunnels between BGP routers to encapsulate interdomain packets
    - GRE tunnel
      - Needs static configuration and be careful with MTU issues
    - MPLS tunnel
      - Can be dynamically established in MPLS enabled backbone

CNPP/2008.4.

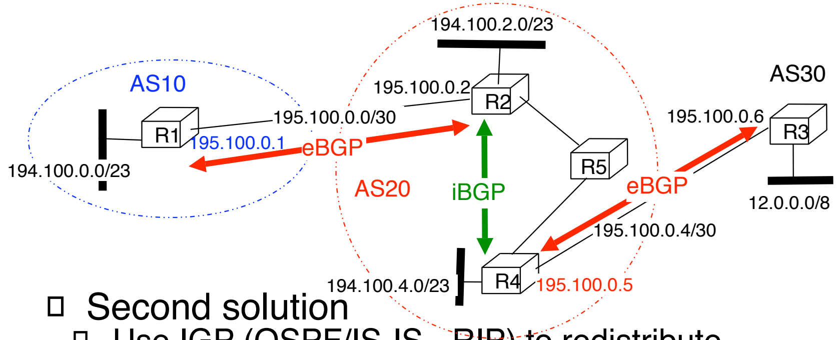
© O. Bonaventure, 2003

The solution of using tunnels inside an AS to forward transit packets was discused in the BGP4 applicability RFC :

Y. Rekhter, P. Gross (Eds.), Application of the Border Gateway Protocol in the Internet, RFC1772, March 1995

However, it only became widespread with the deployment of MPLS. It should be noted that today IP tunnels could also be used inside ASes to transit packets.

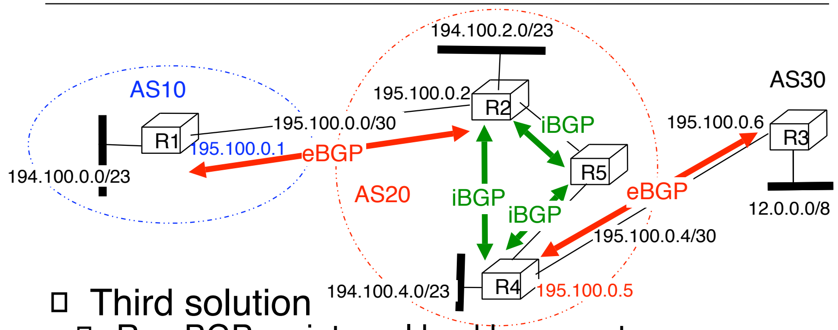
Using non-BGP routers (3)



- Second solution
  - Use IGP (OSPF/IS-IS - RIP) to redistribute interdomain routes to internal backbone routers
  - Drawbacks
    - Size of BGP tables may completely overload the IGP
    - Make sure that BGP routes learned by R2 and injected inside IGP will not be re-injected inside BGP by R4 !

© O. Bonaventure, 2003

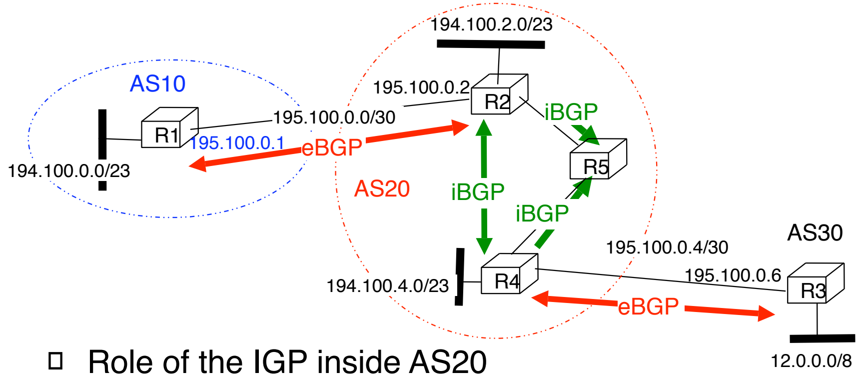
Using non-BGP routers (4)



- Third solution
  - Run BGP on internal backbone routers
  - Internal backbone routers need to participate in iBGP full mesh
    - Internal backbone routers receive BGP routes via iBGP but never advertise any routes
      - Remember : a route learned over an iBGP session is never advertised over another iBGP session

© O. Bonaventure, 2003

The roles of IGP and BGP

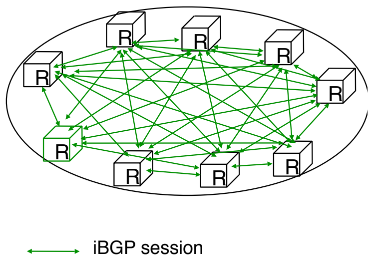


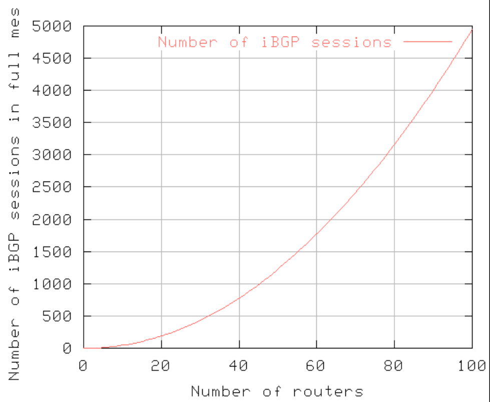
- Role of the IGP inside AS20
  - Distribute internal topology and internal addresses R2-R4-R5)
- Role of BGP inside AS20
  - Distribute the routes towards external destinations
  - IGP must run to allow BGP routers to establish iBGP sessions

© O. Bonaventure, 2003

The iBGP full mesh

- Drawback
  - N*(N-1)/2 iBGP sessions for N routers

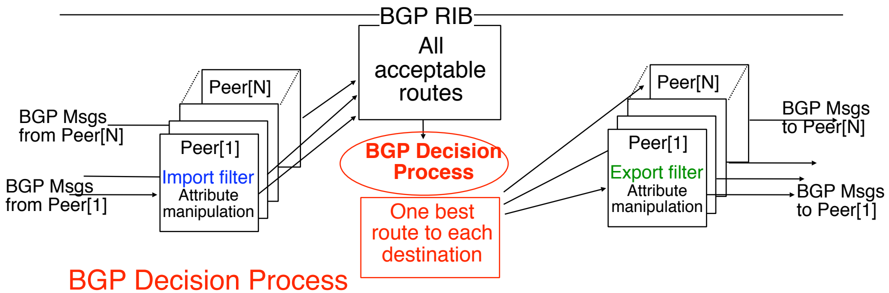




CNPP/2008.4.

© O. Bonaventure, 2003

The BGP decision process



BGP Decision Process

- *Ignore routes with unreachable nexthop*
- Prefer routes with highest local-pref
- Prefer routes with shortest ASPath
- Prefer routes with smallest MED
- Prefer routes learned via eBGP over routes learned via iBGP
- Prefer routes with closest next-hop
- Tie breaking rules
  - Prefer Routes learned from router with lowest router id

© O. Bonaventure, 2003

The BGP decision process also contains a additional step after the ASPath step where the routes with the lowest ORIGIN attribute are preferred. We ignore this step and this attribute in this tutorial.

The BGP decision process used by router vendors may change compared to this theoretical description. For real BGP decision processes, see :

http://www.cisco.com/en/US/tech/tk826/tk365/technologies_tech_note09186a0080094431.shtml

http://www.riverstonenet.com/support/bgp/routing-model/index.htm#_Route_Selection_Process

http://www.juniper.net/techpubs/software/junos53/swconfig53-ipv6/html/routing-overview-ipv69.html

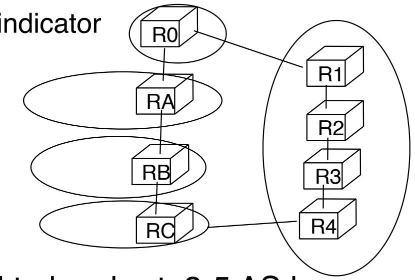
http://www.foundrynet.com/services/documentation/ecmg/BGP4.html

There have been some proposals to allow ISPs to change the BGP decision process on their routers to have a better control on the selected routes.
A. Retana, R. White, BGP Custom Decision Process, Internet draft, draft-retana-bgp-custom-decision-00.txt, work in progress, 2003

One usage of this decision process may be found in
http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a008022ab06.html

The shortest AS-Path step in the BGP decision process

- Motivation
  - BGP does not contain a real " metric"
  - Use length of AS-Path as an indication of the quality of routes
    - Not always a good indicator



- Consequence
  - Internet paths tend to be short, 3-5 AS hops
  - Many paths converge at Tier-1 ISPs and those ISPs carry lots of traffic

CNPP/2008.4.

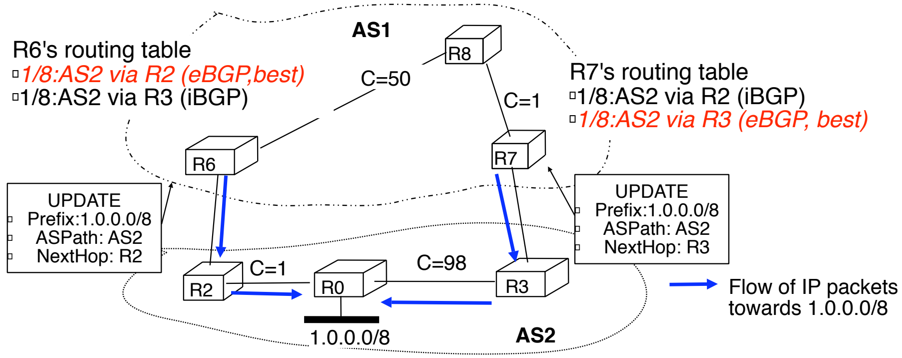
© O. Bonaventure, 2003

A recent study of the quality of the AS Path as a performance indicator compared the round trip time with the length of the AS Path and has shown that the length of the AS Path was only a good indicator for 50% of the considered paths. See :

Bradley Huffaker, Marina Fomenkov, Daniel J. Plummer, David Moore and k claffy, Distance Metrics in the Internet, Presented at the IEEE International Telecommunications Symposium (ITS) in 2002.
http://www.caida.org/outreach/papers/2002/Distance/

The prefer eBGP over iBGP step in the BGP decision process

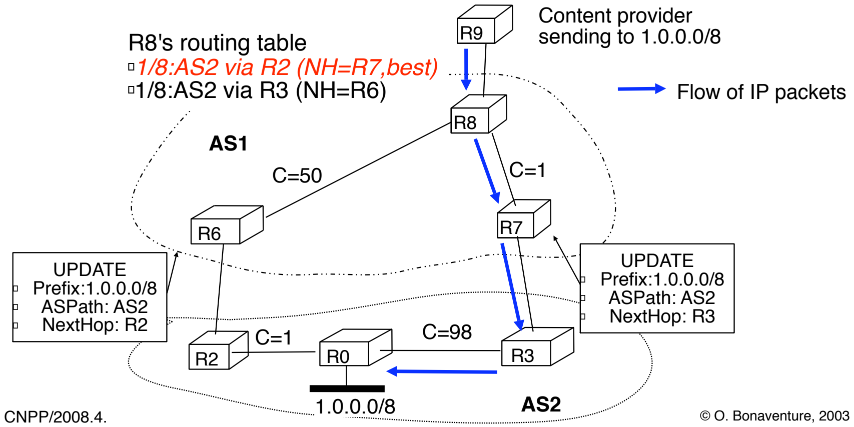
- Motivation : hot potato routing
  - A router should try to get rid of packets sent to external domains as soon as possible



© O. Bonaventure, 2003

The closest `nexthop` step in the BGP decision process

- Motivation : hot potato routing
  - A router should try to get rid of packets sent to external domains as soon as possible

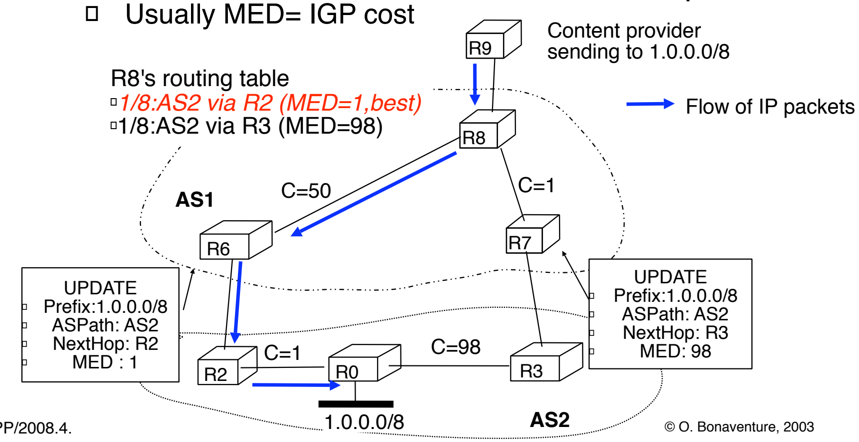


CNPP/2008.4.

© O. Bonaventure, 2003

The lowest MED step in the BGP decision process

- Motivation : cold potato routing
  - In a multi-connected AS, indicate which entry border router is closest to the advertised prefix
    - Usually MED= IGP cost

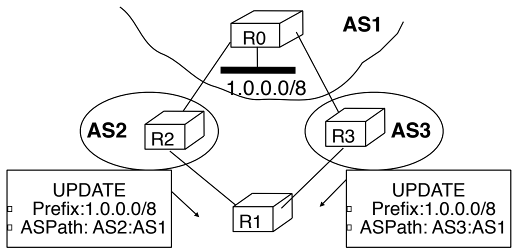


CNPP/2008.4.

© O. Bonaventure, 2003

The lowest router id step in the BGP decision process

- Motivation
  - A router must be able to determine *one* best route towards each destination prefix
    - A router may receive several routes with comparable attributes towards one destination



- Consequence
  - A router with a low IP address will be preferred

© O. Bonaventure, 2003

Note that on some router implementations, the lowest router id step in the BGP decision process is replaced by the selection of the oldest route. See e.g. : http://www.cisco.com/warp/public/459/25.shtml
Preferring the oldest route when breaking times is used to prefer stable paths over unstable paths, however, a drawback of this approach is that the selection of the BGP routes will depend on the arrival times of the corresponding messages. This makes the BGP selection process non-deterministic and can lead to problems that are difficult to debug.

Allocation of IP addresses

- How to allocate IP addresses
- First solution
  - Objective : Ensure that IP addresses are unique
  - Rule used by registries
    - Any organisation can be allocated a unique IP subnet on a FCFS basis
  - Size of the allocated subnet : three classes
    - Class A : subnet with 8 bits mask
    - Class B : subnet with 16 bits mask
    - Class C : subnet with 24 bits mask
  - Drawbacks
    - Too rigid
      - Class A is too large for most networks and Class C too small
        - address waste !
    - Difficult to aggregate prefixes

© O. Bonaventure, 2007

A titre d'exemple concernant l'allocation des adresses IP avec la première solution, on peut citer les universités belges. Actuellement ces universités se connectent à l'Internet à travers le réseau Belnet, mais elles utilisent, pour des raisons historiques, des identificateurs de sous-réseaux fort différents :

- 138.48.0.0/16 pour FUNDP
- 139.165.0.0/16 pour Ulg
- 130.104.0.0/16 pour l'UCL

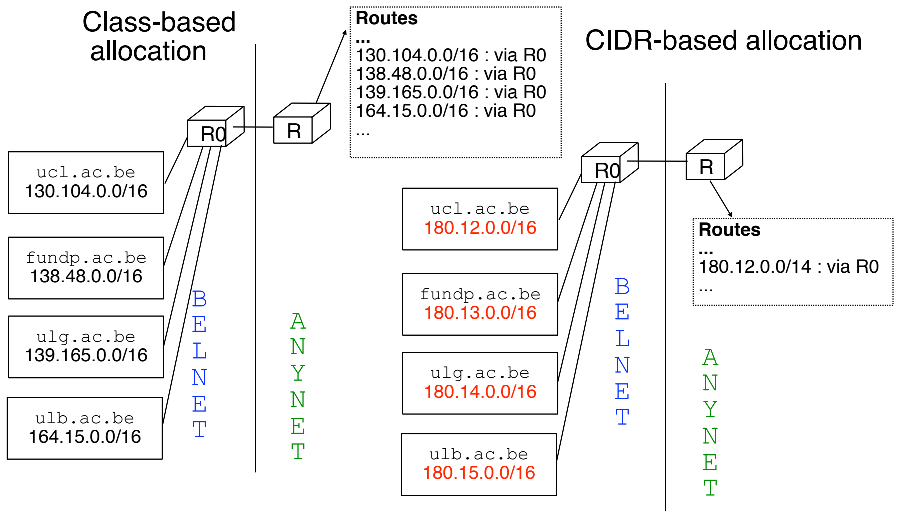
Cela implique que le réseau Belne doit annoncer à l'ensemble de l'Internet une route pour chaque université belge plutôt que d'annoncer une seule route pour l'ensemble des universités.

Allocation of IP addresses (2)

- CIDR
  - Goals
    1. Ensure that IP addresses are unique
    2. Allow BGP routers to advertise aggregated prefixes
  - Rules used by registries
    - Only Internet Service Providers (and large companies) can obtain IP subnets
      - Size of allocated subnet is function of current and expected number of customers
    - An organisation willing to be connected to the Internet must obtain IP addresses from its ISP
  - Advantage
    - Improved aggregation of addresses
  - Drawback
    - If a company switches from one provider to another, it will need to renumber its IP network - a real pain !

© O. Bonaventure, 2007

Allocation of IP addresses (3)

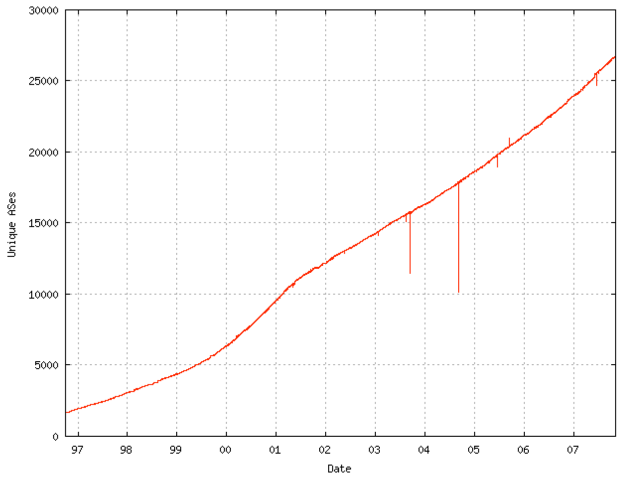


CNPP/2008.4.

© O. Bonaventure, 2007

Internet evolution

- Number of Autonomous Systems



CNPP/2008.4.

Source : http://bgp.potaroo.net

© O. Bonaventure, 2007

Pour plus d'informations, voir
http://bgp.potaroo.net

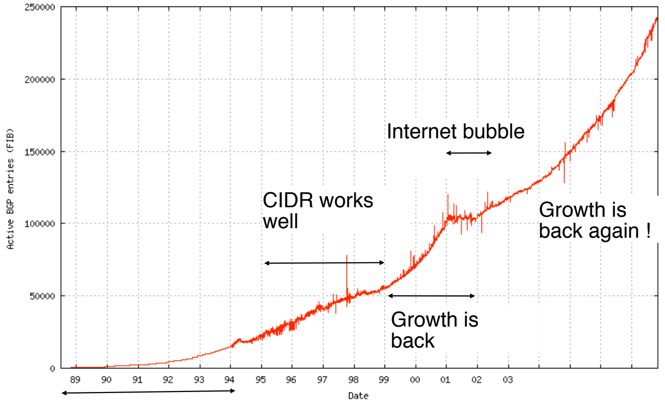
D'autres sources de données utiles sur l'état des tables BGP sont :
http://www.netlantis.org/

http://www.route-views.org

http://www.ripe.net/ris/

Internet evolution (2)

- Size of the BGP routing tables
  - Number of IPv4 prefixes in default-free routers



CNPP/2008.4.

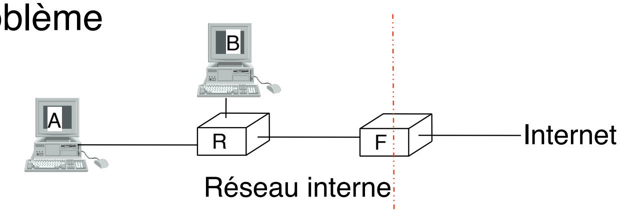
Source : http://bgp.potaroo.net

© O. Bonaventure, 2007

Source : http://bgp.potaroo.net

Firewalls

- Problème



- Comment pour des raisons de sécurité contrôler le traffic à l'entrée d'un réseau de façon à
  - autoriser uniquement l'accès d'adresses IP connues ?
  - autoriser l'accès de l'extérieur à n'importe quelle machine interne via `telnet ?`
  - n'autoriser la réception d'email que par les serveurs SMTP officiels du réseau interne
  - autoriser n'importe quel échange TCP initié par une machine interne

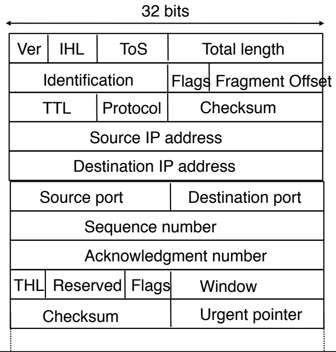
© O. Bonaventure, 2007

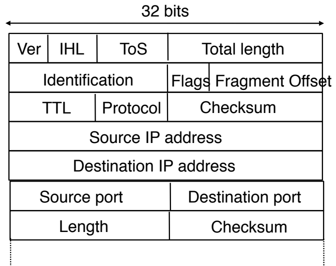
Pour une description détaillée des firewalls, voir par exemple :
Cheswick, William R., Bellovin, Steven M., Rubin, Aviel D. Firewalls and internet security - Second edition - Repelling the Wily Hacker, Addison-Wesley 2003

Firewalls (2)

- Principe de fonctionnement
  - Inspecter tous les paquets IP qui transitent par le firewall
  - Définir des règles permettant d'accepter ou de refuser les paquets en fonction de leurs entêtes

TCP





UDP

© O. Bonaventure, 2007

Firewalls : exemples de règles

- Accepter les paquets venant de 12.0.0.0/8
  - paquet acceptable si adresse source East dans le sous-réseau 12.0.0.0/8
- Permettre de contacter le serveur web sur toute machine interne
  - paquet acceptable si
    - protocole de transport = TCP
    - port destination = 80
    - attention aux paquets IP fragmentés !
- Permettre de contacter le serveur SMTP sur la machine 1.2.34
  - paquet acceptable si
    - adresse destination = 1234
    - protocole de transport = TCP et port destination = 25

© O. Bonaventure, 2007

Firewalls : exemples de règles (2)

- Comment permettre les connexions TCP initiées par une machine interne ?
  - exemple : accès web
  - Quand un paquet arrive de l'extérieur, il faut pouvoir déterminer si il appartient à une connexion qui a été ouverte depuis l'intérieur
    - tous les paquets d'une même connexion TCP contiennent l'identification de la connexion :
      - adresse IP source
      - adresse destination
      - port TCP source
      - port TCP destination
      - (champ protocol=TCP)

© O. Bonaventure, 2007

Firewalls : exemples de règles (3)

- Autorisation des connexions TCP ouvertes depuis les machines internes
- Principe
  - maintenir une liste des connexions TCP ouvertes actuellement à travers le firewall
    - un segment de données sera accepté par le firewall si il appartient à une connexion se trouvant dans la liste
  - implémenter une machine à états finis pour chaque connexion TCP passant à travers le firewall
    - arrivée d'un segment SYN de l'intérieur
      - insérer l'id de la connexion dans la liste, attendre SYN+ACK
    - arrivée de SYN+ACK de l'extérieur
      - connexion East ouverte, on peut accepter des segments de données
    - fermeture de la connexion TCP avec segment RST ou fermeture normale (FIN/ACK)
      - supprimer la connexion de la liste des connexions

CNPP/2008.4.

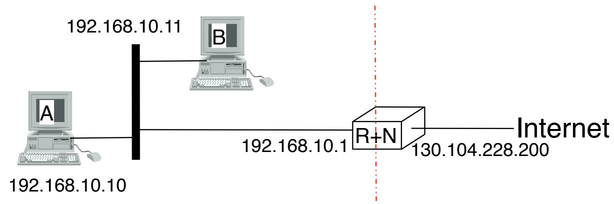
© O. Bonaventure, 2007

Lors d'une fermeture de connexion TCP par arrivée d'un segment RST de l'intérieur ou de l'extérieur, il East possible de supprimer immédiatement la connexion de la liste des connexions ouvertes.
Lors d'une fermeture de connexion TCP par arrivée d'un segment FIN, il ne faudra supprimer la connexion de la liste des connexions ouvertes qu'après l'échange FIN/ACK dans les deux sens. Afin de prendre en compte la perte éventuelle du dernier ACK, il peut être utile de ne pas suppprimer la connexion de la liste immédiatement, mais après expiration d'un timer fixé à 2*MSL.

Lorsque le firewall implémente une machine à états finis pour suivre l'état des connexions TCP qui passent à travers lui, on parle de firewall *stateful* . Un firewall qui se content d'analyser les entêtes sans mémoriser d'état East dit de type *stateless*.

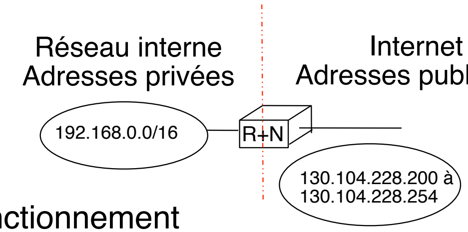
Network Address Translator

- Problème
  - les adresses IP disponibles sont en nombre limité
- Solution
  - économiser le plus possible les adresses IP
  - utiliser des adresses "privées" dans de petits réseaux et traduire dynamiquement les paquets envoyés sur Internet



© O. Bonaventure, 2007

Un NAT simple



- Fonctionnement
  - Arrivée d'un paquet non fragmenté du réseau interne
    - Vérifier si cette adresse East déjà associée à une adresse publique du pool du NAT
      - si oui, traduire le paquet avant de l'envoyer
      - modifier adresse source en changeant les checksums IP et UDP/TCP et envoyer le paquet vers Internet
      - réinitialiser le temporisateur
      - si non, associer une adresse libre du pool à l'adresse privée et armer un temporisateur
      - modifier adresse source en changeant les checksums IP et UDP/TCP et envoyer le paquet vers Internet
      - réinitialiser le temporisateur

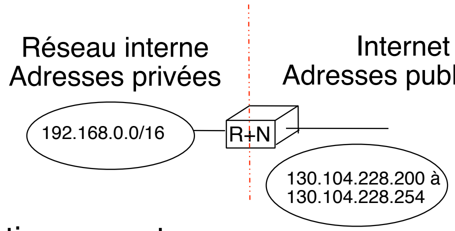
CNPP/2008.4.

© O. Bonaventure, 2007

Dans ce cas, le NAT devra modifier l'adresse source des paquets allant du réseau interne vers le réseau externes et l'adresse destination des paquets IP allant dans le sens inverse.

Le temporisateur mentionné dans le transparent East une des solutions possibles pour éviter qu'une adresse publique ne soit associée éternellement à une adresse privée. Une autre solution East d'analyser les paquets échangés et par exemple si seul du trafic TCP East échangé de libérer l'adresse IP publique lorsqu'aucun paquet n'a été reçu durant une période de x secondes ou minutes.

Un NAT simple (2)



- Fonctionnement
  - Arrivée d'un paquet non-fragmenté du réseau Internet
    - Vérifier si cette adresse East déjà associée à une adresse privée du réseau interne
      - si oui, traduire le paquet avant de l'envoyer
      - modifier adresse destination en changeant les checksums IP et UDP/TCP et envoyer le paquet vers le réseau interne
      - réinitialiser le temporisateur
      - si non, jeter le paquet reçu

CNPP/2008.4. © O. Bonaventure, 2007

Le NAT simple tel qu'il East décrit suppose que tous les paquets reçus par le NAT le sont en réponse à l'envoi préalable de paquets IP par le réseau interne. C'East typiquement le cas pour les connexions TCP qui sont établies par les machines du réseau interne.

Si des serveurs du réseau interne doivent être accessibles depuis le réseau Internet, une possibilité East crééer une association statique entre l'adresse interne du serveur et une adresse IP publique du pool.

Lorsque des paquets sont fragmentés, cela peut poser des problèmes au NAT car le si paquet IP contient un segment TCP ou UDP, alors en changeant l'adresse IP source du paquet, le NAT devra mettre à jour le checksum IP mais aussi le checksum TCP/UDP. En effet, les checksums de la couche transport sont calculés en considérant le segment plus un pseudo-header contenant notamment les adresses IP source et destination. Si ce checksum transport se trouve dans le premier fragment du paquet IP, il peut être mis à jour dans façon incrémentale. Si il se trouve dans le deuxième fragment, le NAT devra prendre en compte des informations du premier fragment pour calculer le checksum transport se trouvant dans le second.

Le support d'ICMP au niveau du NAT oblige en pratique le NAT à analyser complètement le contenu du message ICMP reçu pour traduire les bonnes adresses IP.

Un NAT plus complexe

- Principe
  - Le NAT traduit les adresses IP et/ou les ports TCP/UDP des paquets qui le traversent
    - table de traduction des adresse et port maintenue dynamiquement par le NAT
      - traduction de l'adresse source *et* du port source pour les paquets sortants en mettant à jour les checksums
      - traduction de l'adresse destination *et* du port destination pour les paquets entrants en mettant à jour les checksums

Addresse interne	Protocol	Port Interne	Adresse externe	Port externe
192.168.10.10	UDP	2340	130.104.228.200	4567
192.168.10.10	TCP	512	130.104.228.200	520
192.168.10.11	TCP	1024	130.104.228.200	2048

    - Difficultés
      - applications qui encodent des adresses IP/port dans le contenu des paquets (exemple : ftp)
      - applications UDP où la réponse East envoyée sur un autre port
      - certains paquets fragmentés

CNPP/2008.4. © O. Bonaventure, 2007

Le NAT décrit dans le transparent ci-dessus doit donc :
- modifier les champs adresse source (de l'entête IP) et port source (de l'entête TCP ou UDP) dans le sens réseau interne -> Internet
- modifier les champs adresse destination (de l'entête IP) et port destination (de l'entête TCP ou UDP) dans le sens Internet -> réseau internet

Lorsqu'il modifie ces champs, le NAT East obligé de recalculer le checksum se trouvant dans l'entête IP et celui se trouvant dans l'entête transport.