Università degli Studi Roma Tre
Dipartimento di Informatica e Automazione
Computer Networks Research Group

netkit lab(s)

spanning-tree

Version	2.0
Author(s)	G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini
E-mail	contact@netkit.org
Web	http://www.netkit.org/
Description	experiences with the spanning tree protocol

identificador de puente

- el *bridge-id* se forma concatenando:
  - la prioridad
    - es local para cada puente
    - se puede configurar
    - por defecto: 80-00
  - la mac del puente
    - mac del “primer” puerto del puente
    - usualmente el que tiene la mac más baja
- ejemplo: 80-00-23-ef-...

los dos tipos de Bridge PDUs

- BPDU de configuración (CBPDU)
  - contiene toda la información necesaria para el algoritmo STP, incluyendo:
    - *root-bridge-identifier* (puente raíz actual)
    - *root-path-cost* (coste del camino al puente raíz)
    - *bridge-identifier* (id. del que envía el BPDU)
    - *port-identifier* (puerto por el que se envía el BPDU)
  - ejemplo:

root-bridge-id	80-00-23-ef-...
root-path-cost	100
bridge-id	80-00-2d-12-...
port-id	6

- BPDU de cambio de topología
  - sólo contiene los datos que se necesitan para identificar el paquete como un BPDU de cambio de topología

las 4 operaciones del algoritmo STP

- elección del *root-bridge*
  - un único puente se elige raíz del *spanning tree*
- identificación del *root-port* en cada puente
  - cada puente no raíz selecciona uno de sus puertos como el más cercano al puente raíz
- determinación de los *designated-bridges*
  - por cada LAN se elige un puente como el que conecta la LAN al *spanning tree*
  - el puerto del puente designado que conecta la LAN al *spanning tree* es el puerto designado
- bloqueo de puertos redundantes
  - los que no son *root-ports* ni *designated-ports*

1) elección del *root bridge*

- cada puente crea un BPDU de configuración usando su propio *bridge-id* como *root-identifier*
- cuando recibe un BPDU de configuración con un *bridge-id* más bajo:
  - deja de enviar BPDUs de configuración
  - reenvía el nuevo BPDU de configuración por todos los puertos
- el *root-bridge* es el único que continúa enviando BPDUs de configuración con su propio *bridge-id* en el campo *root-identifier*

reenvío de bpdu de configuración

- cuando un CBPDU es creado por un puente su campo *root-path-cost* se pone a cero
- cuando un CBPDU es reenviado por un puente que no es el *root-bridge* sus campos se actualizan así:
  - el *root-path-cost* se le suma el coste del puerto que recibió el CBPDU
  - el *bridge-identifier* se pone al *bridge-id* del puente actual
  - el *port-identifier* se pone al número del puerto por el que el CBPDU es enviado

root-bridge-id	00-00-23-ef-...
root-path-cost	100
bridge-id	10-00-2d-12-...
port-id	6

root-bridge-id	00-00-23-ef-...
root-path-cost	200
bridge-id	10-00-43-2f-...
port-id	5

2) identificación del *root port*

- cada puente que no es el *root-bridge* identifica el puerto a través del cual se alcanza el *root-bridge* con el mínimo coste
- el *root-port* se elige para que sea el que recibe CBPDUs con
  - 1. más bajo *root-path-cost* (tras añadir el *port-cost*)
  - 2. más bajo *bridge-identifier*
  - 3. más bajo *port-identifier*
  - 4. o, a iguales valores, el puerto propio con más bajo *port-identifier*

3) determinación de los puentes designados

- de entre los puentes que conectan un segmento a la LAN se elige el puerto que tiene menor *root-path-cost* como *designated-port* basándose en los CBPDUs que son reenviados por ese puerto
- el puente que tiene el *designated-port* se llama *designated-bridge*
- como *designated-port* se elige el que envía CBPDUs con el:
  - 1. más bajo *root-path-cost*
  - 2. más bajo *bridge-identifier*
  - 3. más bajo *port-identifier*

4) estado de bloqueo

- todos los puertos que no son *root-ports* o *designated-ports* se colocan en estado de bloqueo (*blocking state*)
- todos los *root-ports* y *designated-ports* se colocan en estado de reenvío (*forwarding state*)

estados de los puertos

- durante el cálculo del árbol de expansión (*spanning tree*), el estado de un puerto puede ser:

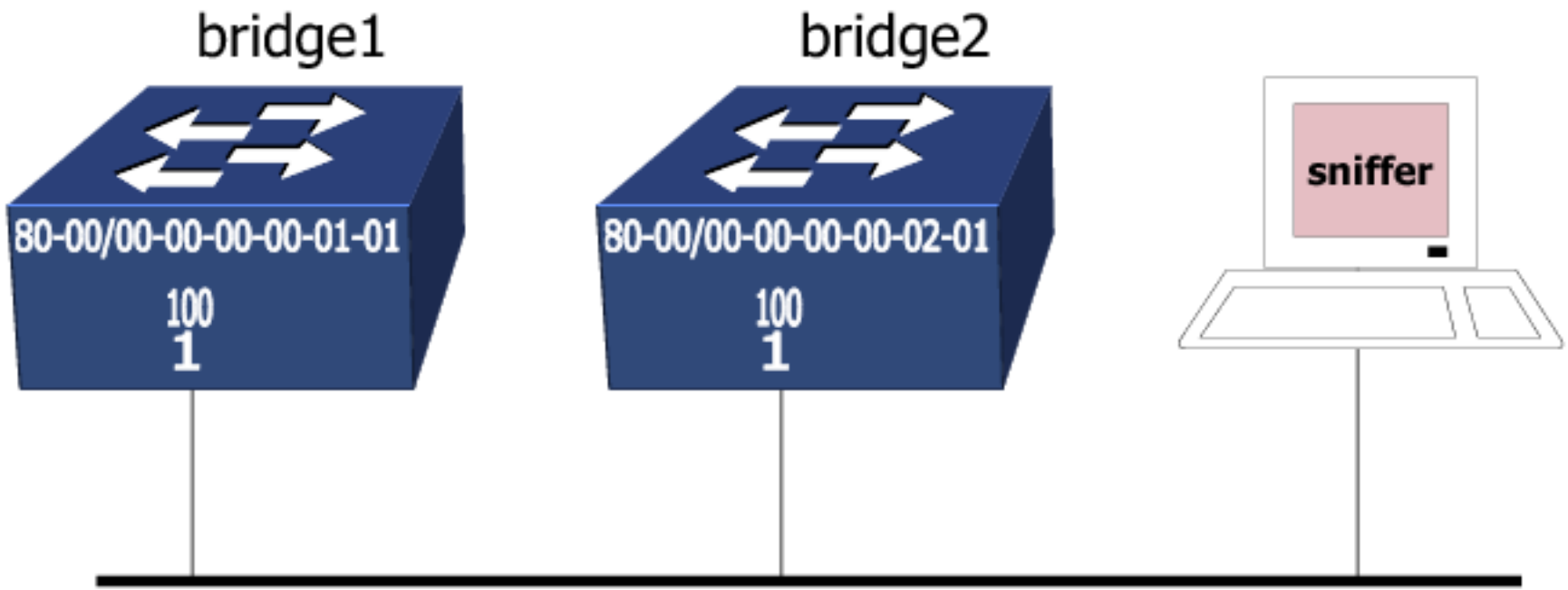
	receives frames	forwards frames	updates database	receives bpdus	transmits bpdus
blocking	✗	✗	✗	✓	✗
listening	✗	✗	✗	✓	✓
learning	✗	✗	✓	✓	✓
forwarding	✓	✓	✓	✓	✓
disabled	✗	✗	✗	✗	✗

temporizadores

- hello time [2]
  - intervalo de tiempo entre la generación de BPDU por un puente
- max age [20]
  - cantidad de tiempo que un puente esperará a recibir un BPDU. Una vez que expira, se transmite una notificación de cambio de topología (TCN-BPDU)
- forward delay [15]
  - cantidad de tiempo que un puente mantendrá un puerto en estado (1º) de escucha (*listening*) y (2º) de aprendizaje (*learning*)

[x] = el valor por defecto es de x segundos

lab1: root bridge election



lab1: root bridge election

host machine

```
user@localhost:~$ cd netkit-lab_stp-root-election
user@localhost:~/netkit-lab_stp-root-election$ lstart
```

- the lab is configured to
  - start the two bridges
  - start a virtual machine with a sniffer that listens to the traffic generated for the computation of the spanning tree
    - after 20 packets have been captured, the **`sniffer`** virtual machine is automatically halted
    - a file "**`sniffer.cap`**" is created in the lab directory on the host, for later investigation (e.g., with wireshark, tshark)

lab1: root bridge election

- the elected root bridge can be checked as follows:

bridge2

```
bridge2:~# brctl showstp br0
br0
 bridge id              8000.000000000201
 designated root        8000.000000000101
 root port                 1                    path cost                  100
 max age                  20.00                 bridge max age              20.00
 hello time                2.00                 bridge hello time            2.00
 forward delay            15.00                 bridge forward delay        15.00
 ageing time             300.00
 hello timer               0.00                 tcn timer                    0.00
 topology change timer     0.00                 gc timer                   249.75
 flags                  TOPOLOGY_CHANGE
.....
```

lab1: root bridge election

- a detailed dump of the file **`sniffer.cap`** can be obtained as follows:

host machine

```
user@localhost:~/netkit-lab_stp-root-election$ tshark -V -r sniffer.cap
Frame 1 (60 bytes on wire, 60 bytes captured)
    Arrival Time: May  8, 2007 14:48:25.602819000
    [Time delta from previous packet: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Packet Length: 60 bytes
    Capture Length: 60 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:llc:stp]
IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
.....
```

lab1: root bridge election

- a detailed dump of the file **`sniffer.cap`** can be obtained as follows:

host machine

```
user@localhost:~/netkit-lab_stp-root-election$ tshark -V -r sniffer.cap
Frame 1 (60 bytes on wire, 60 bytes captured)
    Arrival Time: May  8, 2007 14:48:25.602819000
    [Time delta from previous packet: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Packet Length: 60 bytes
    Capture Length: 60 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:llc:stp]
IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:
.....
```

display packet details

dissect saved capture data from file instead of performing a live capture

lab1: root bridge election

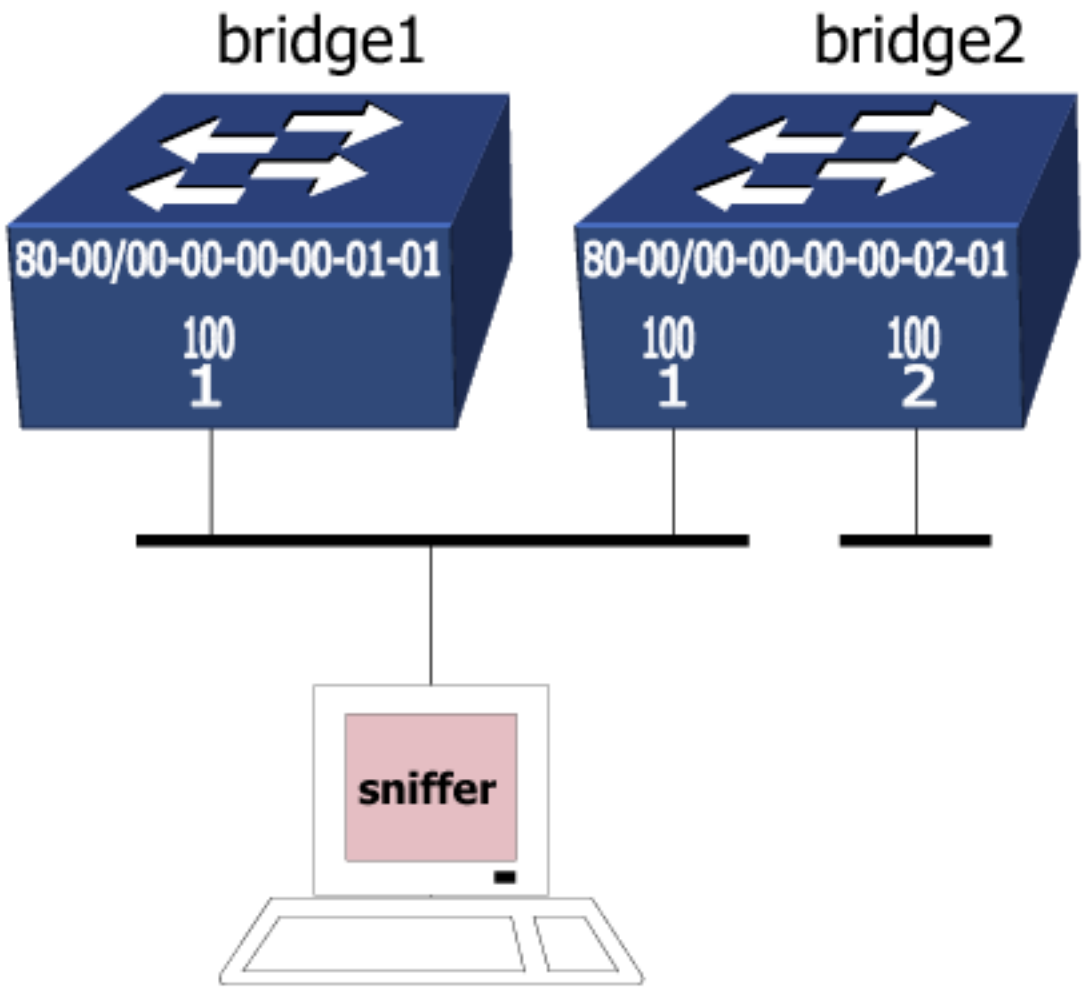
```
⊞ IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
    Source: 00:00:00_00:02:01 (00:00:00:00:02:01)
    Length: 38
⊞ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    BPDU flags: 0x00
        0... .... = Topology Change Acknowledgment: No
        .... ...0 = Topology Change: No
    Root Identifier: 32768 / 00:00:00:00:02:01
    Root Path Cost: 0
    Bridge Identifier: 32768 / 00:00:00:00:02:01
    Port identifier: 0x8001
    Message Age: 0
    Max Age: 20
    Hello Time: 2
    Forward Delay: 15
```

multicast address (all the bridges on the lan)

this is a configuration bpdu

at the beginning the bridge claims to be the root bridge

lab2: topology change



lab2: topology change

host machine

```
user@localhost:~$ cd netkit-lab_stp-topology-change
user@localhost:~/netkit-lab_stp-topology-change$ lstart
```

- in a way similar to lab1, the lab is configured to
  - start the two bridges
  - start a virtual machine with a sniffer
    - the **`sniffer`** virtual machine is **not** automatically halted
    - a file "**`sniffer.cap`**" is created in the lab directory on the host, for later investigation (e.g., with wireshark, tshark)

lab2: cambio de topología

- cuando un puente detecta un cambio de topología envía TCN BPDUs por su *root-port*
- continúa enviando notificaciones hasta que el *designated-bridge* en el segmento conectado al *root-port* lo confirma
- el *designated-bridge*, a su vez, enviará TCN-BPDU por su *root-port*, hasta que es recibido por el *root-bridge*
- el *root-bridge* comienza a colocar la bandera de cambio de topología en sus CBPDUs
- todos los puentes, al oír que un cambio de topología está teniendo lugar, usará el tiempo de retardo de reenvío (*forward delay time*) (usualmente 15 segundos) en vez del tiempo de envejecimiento (*ageing time*) de filtrado de la base de datos (usualmente 5 minutos) para jubilar las entradas de la tabla de direcciones de origen

lab2: topology change

- running this command in the lab forces the generation of topology change notification bpdus:

bridge2

```
bridge2:~# brctl addif br0 eth1
```

- the notification is correctly captured by the sniffer after observing at least these lines:

sniffer

```
.....
12:22:24.362414 802.1d tcn
12:22:24.685188 802.1d config TOP_CHANGE TOP_CHANGE_ACK
8000.00:00:00:00:01:01.8001 root 8000.00:00:00:00:01:01 pathcost 0 age 0
max 20 hello 2 fdelay 15
.....
```

lab2: topology change

```
⊞ IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
    Source: 00:00:00_00:02:01 (00:00:00:00:02:01)
    Length: 7
⊞ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Topology Change Notification (0x80)
```

bridge2 generates a topology change notification

this is a topology change notification (tcn)

lab2: topology change

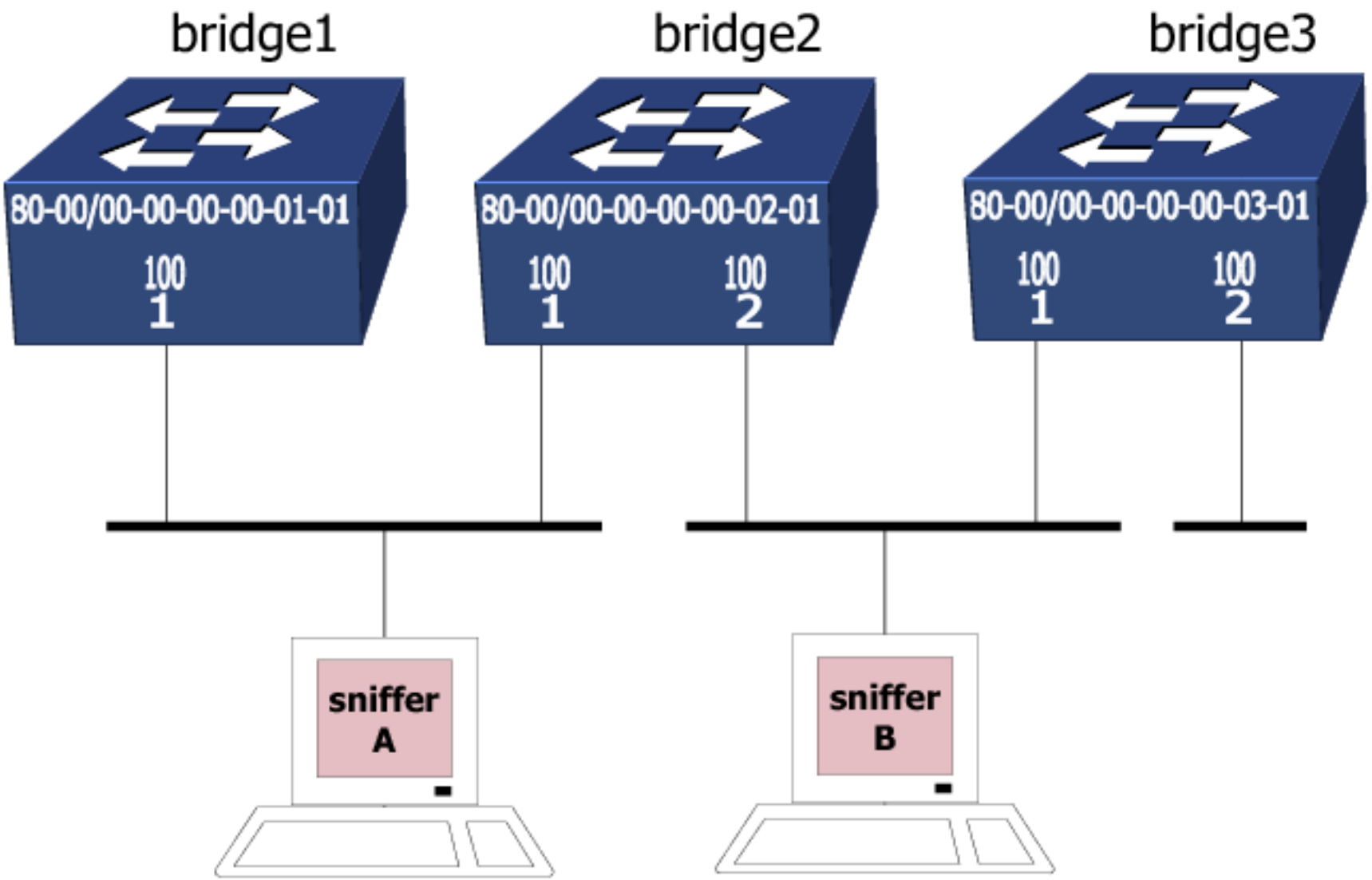
```
⊞ IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
    Source: 00:00:00_00:01:01 (00:00:00:00:01:01)
    Length: 38
⊞ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    BPDU flags: 0x81 (Topology Change Acknowledgment, Topology Change)
        1... .... = Topology Change Acknowledgment: Yes
        .... ...1 = Topology Change: Yes
    Root Identifier: 32768 / 00:00:00:00:01:01
    Root Path Cost: 0
    Bridge Identifier: 32768 / 00:00:00:00:01:01
    Port identifier: 0x8001
    Message Age: 0
    Max Age: 20
    Hello Time: 2
    Forward Delay: 15
```

the root bridge acknowledges the notification

this configuration bpdu carries a topology change notification acknowledgment

the topology change is further propagated

lab3: remote topology change



lab3: remote topology change

host machine

```
user@localhost:~$ cd netkit-lab_stp-remote-change
user@localhost:~/netkit-lab_stp-remote-change$ lstart
```

- the lab is configured to start the whole network and create the following capture files on the host:
  - **`sniffer_A.cap`**
  - **`sniffer_B.cap`**

lab3: remote topology change

- the topology change notification can be triggered by using the following command:

bridge3

```
bridge3:~# brctl addif br0 eth1
```

- the notifications are correctly captured once at least these lines appear in the console of the sniffer virtual machines:

sniffer_A

```
.....
13:01:22.366772 802.1d tcn
13:01:22.367400 802.1d config TOP_CHANGE
TOP_CHANGE_ACK 8000.00:00:00:00:01:01.8001
root 8000.00:00:00:00:01:01 pathcost 0 age
0 max 20 hello 2 fdelay 15
.....
```

sniffer_B

```
.....
13:01:22.365668 802.1d tcn
13:01:22.366464 802.1d config
TOP_CHANGE_ACK 8000.00:00:00:00:02:01.8002
root 8000.00:00:00:00:01:01 pathcost 100
age 1 max 20 hello 2 fdelay 15
.....
```

lab3: remote topology change

```
⊞ IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
    Source: 00:00:00_00:03:01 (00:00:00:00:03:01)
    Length: 7
⊞ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Topology Change Notification (0x80)
```

bridge3 port 1

step 1: **bridge3** generates a topology change notification

lab3: remote topology change

```
⊞ IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
    Source: 00:00:00_00:02:02 (00:00:00:00:02:02)
    Length: 38
⊞ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    BPDU flags: 0x80 (Topology Change Acknowledgment)
        1... .... = Topology Change Acknowledgment: Yes
        .... ...0 = Topology Change: No
    Root Identifier: 32768 / 00:00:00:00:01:01
    Root Path Cost: 100
    Bridge Ide
    Port ident
    Message Ac
    Max Age: 2
    Hello Time
    Forward Delay: 15
```

bridge2 port 2

this is just an acknowledgment

step 2: **bridge2** acknowledges (hence, **bridge3** stops sending notifications)

lab3: remote topology change

```
⊞ IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
    Source: 00:00:00_00:02:01 (00:00:00:00:02:01)
    Length: 7
⊞ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
     Protocol Identifier: Spanning Tree Protocol (0x0000)
     Protocol Version Identifier: Spanning Tree (0)
     BPDU Type: Topology Change Notification (0x80)
```

bridge2 port 1

step 3: **bridge2** propagates the topology change notification through its root port

lab3: remote topology change

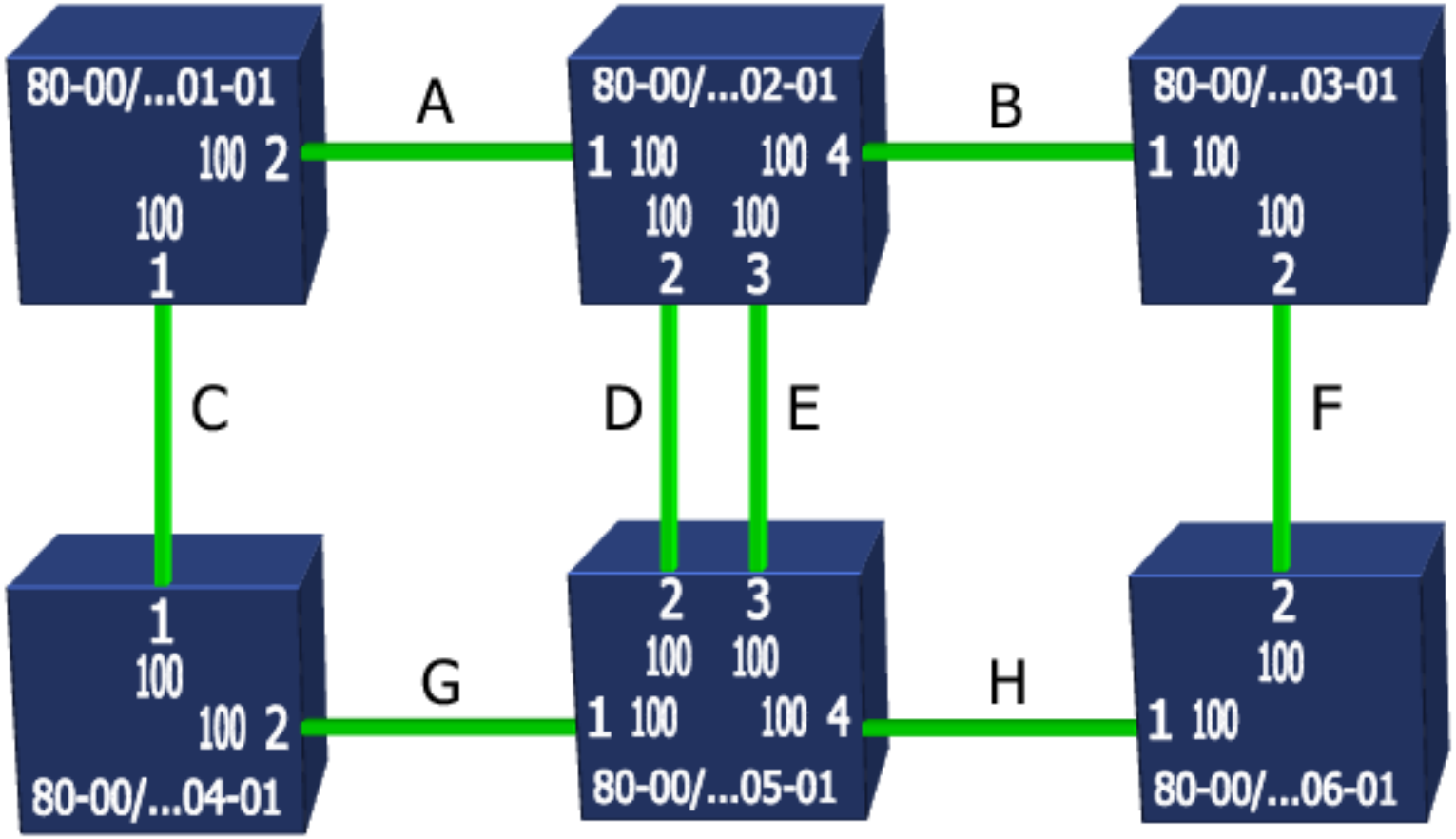
```
⊞ IEEE 802.3 Ethernet
    Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
    Source: 00:00:00_00:01:01 (00:00:00:00:01:01)
    Length: 38
⊞ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    BPDU flags: 0x81 (Topology Change Acknowledgment, Topology Change)
        1... .... = Topology Change Acknowledgment: Yes
        .... ...1 = Topology Change: Yes
    Root Identifier: 32768 / 00:00:00:00:01:01
    Root Path Cost: 0
    Bridge
    Port i
    Messag
    Max Ag
    Hello
    Forward Delay: 15
```

bridge1 port 1

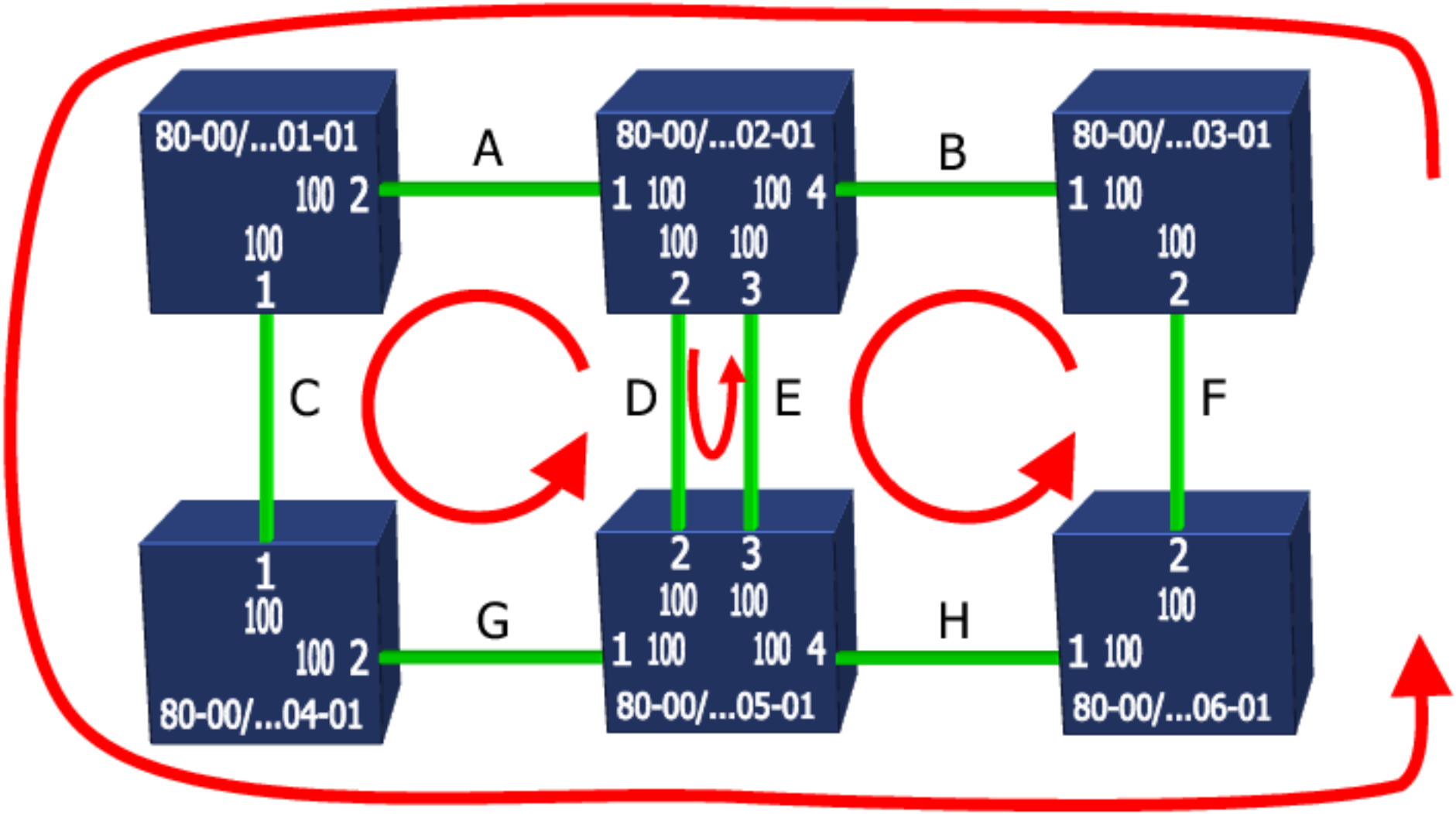
the topology change flag is set

step 4: **bridge1** acknowlegdes (hence, **bridge2** stops sending notifications) and sets the topology change flag

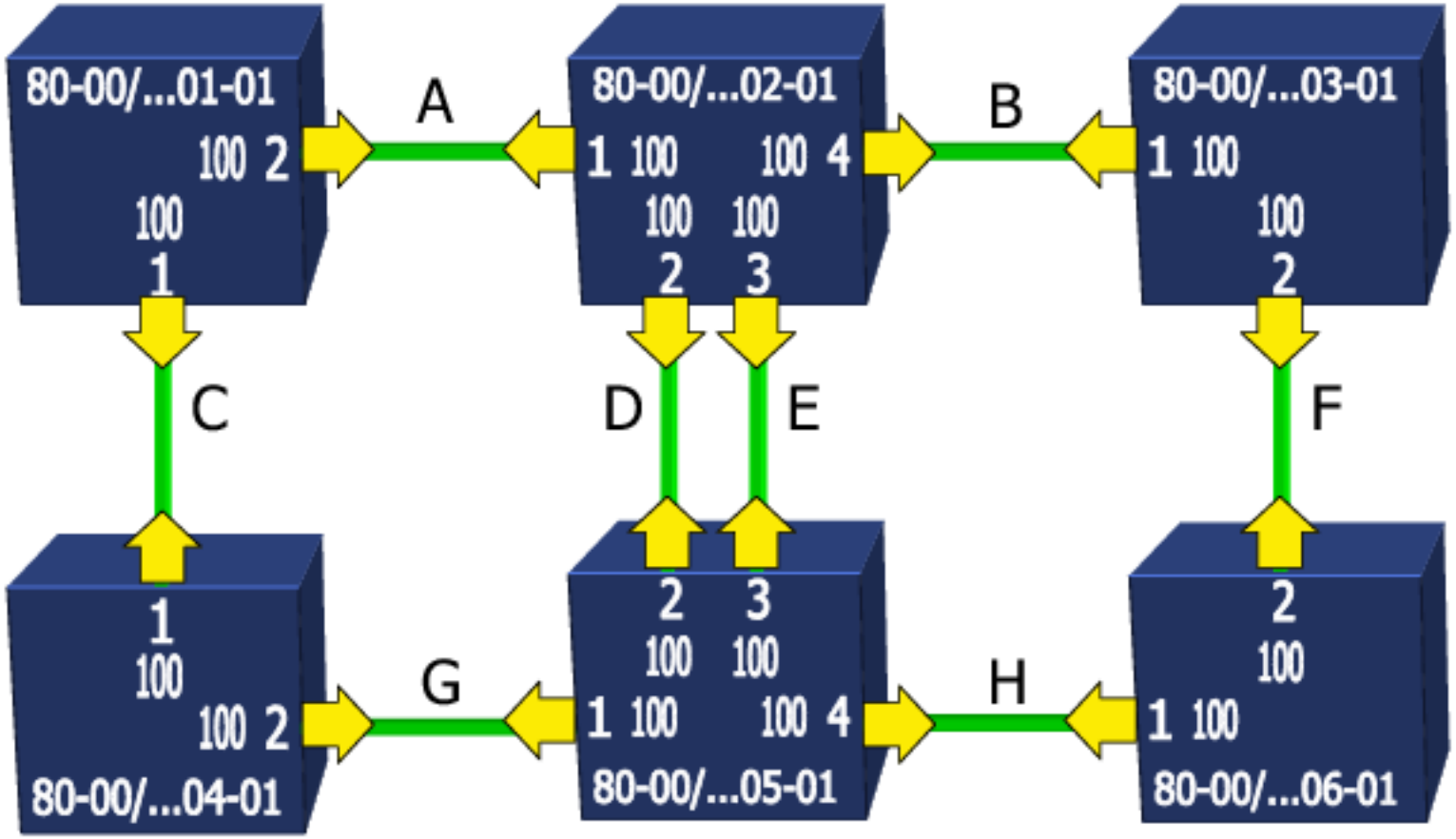
lab4: a more complex scenario



lab4: a more complex scenario



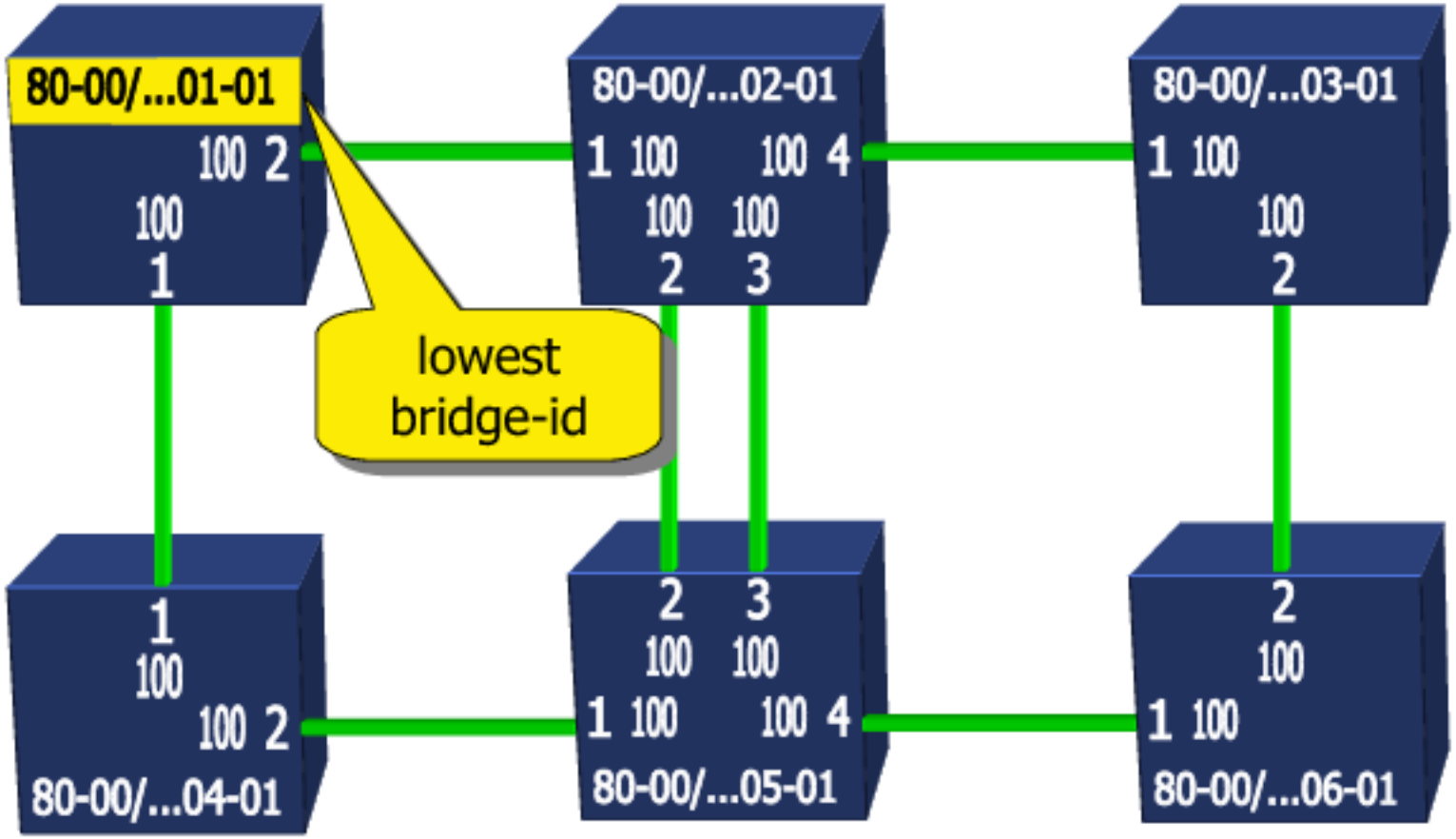
root bridge election



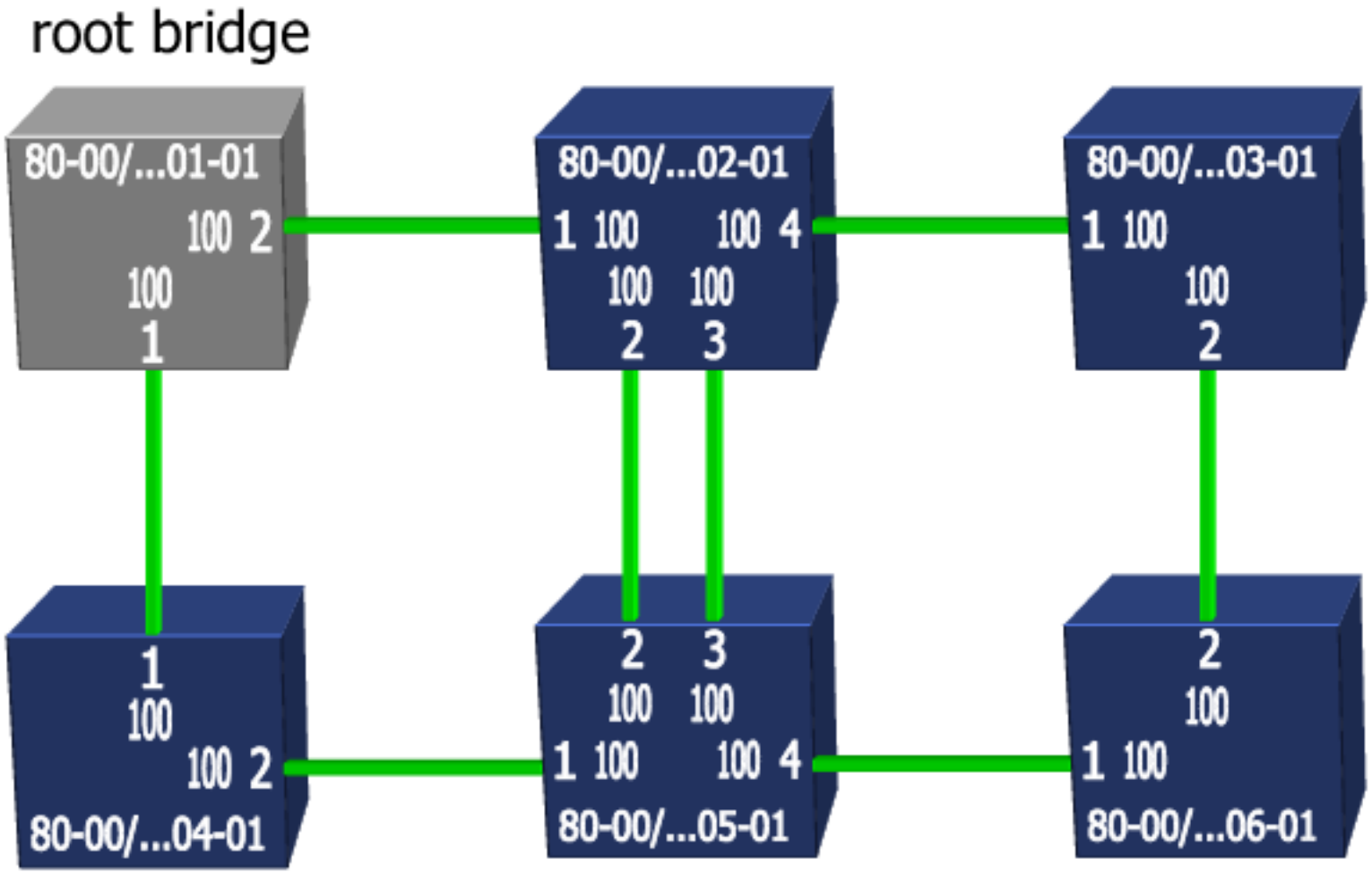
= "I am the root bridge"

last update: May 2007

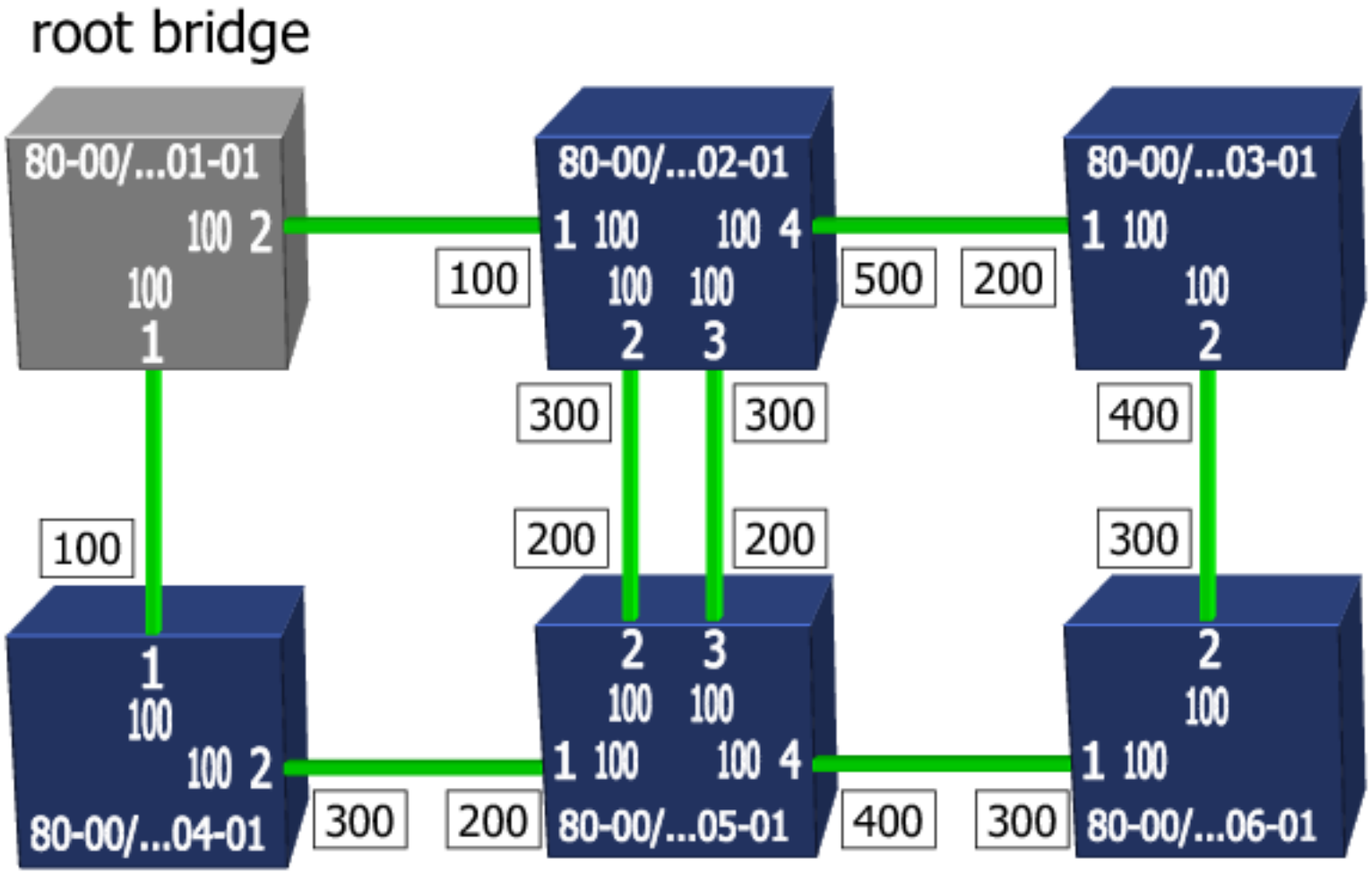
root bridge election



root bridge election



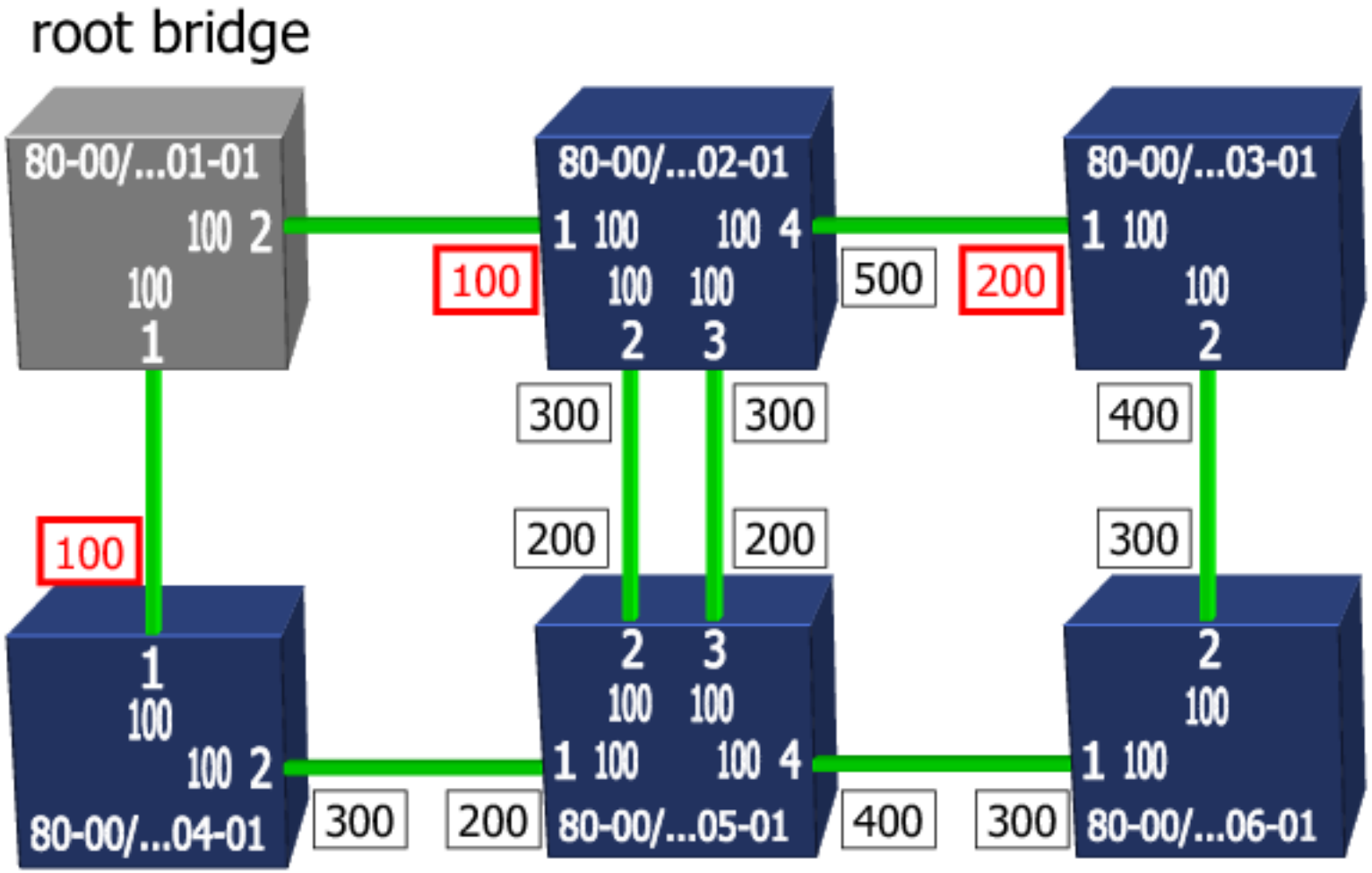
root ports identification



xxx = minimum root-path-cost of bpdus received through the port

last update: May 2007

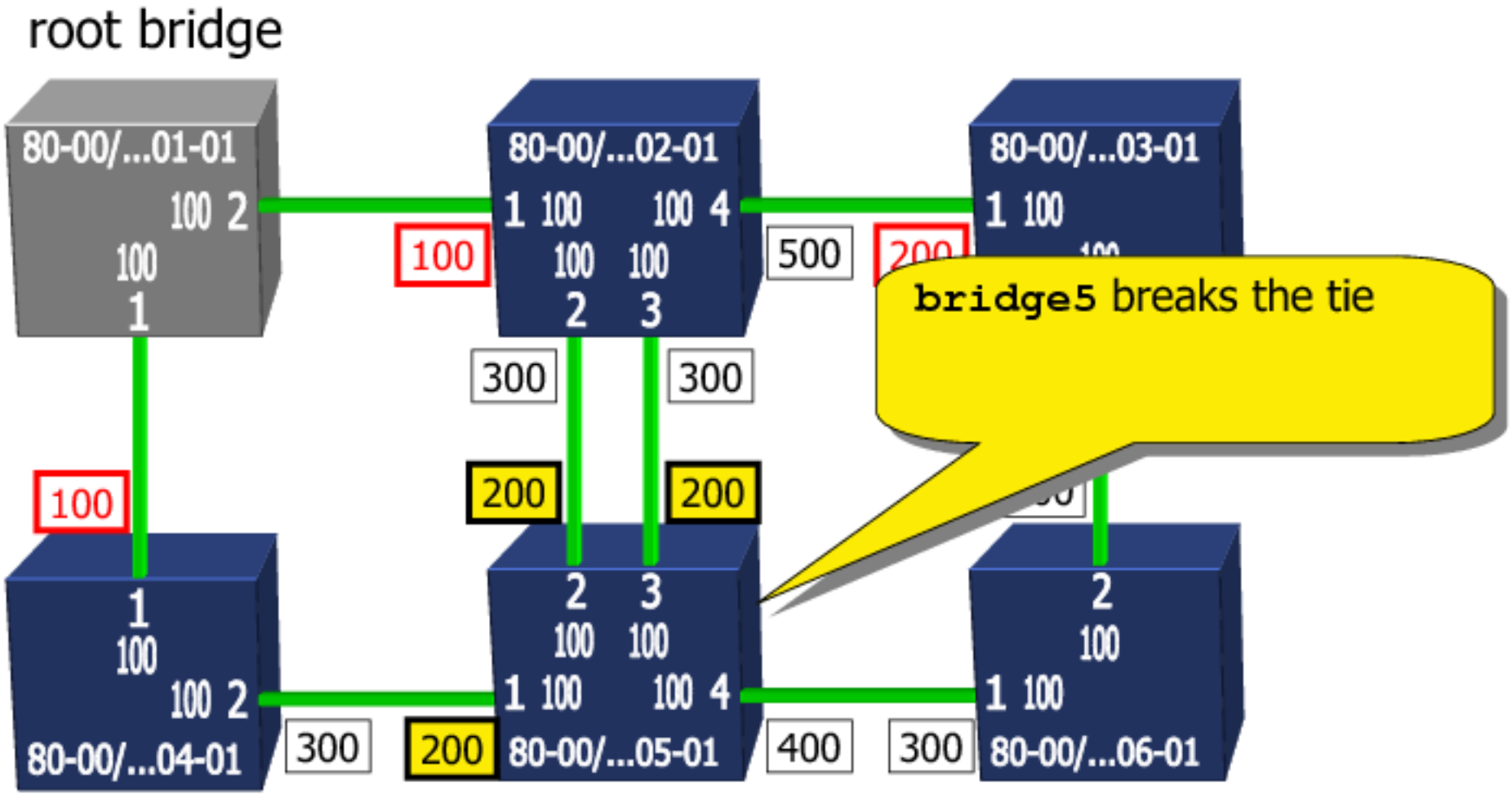
root ports identification



xxx = root port (xxx is the root path cost)

last update: May 2007

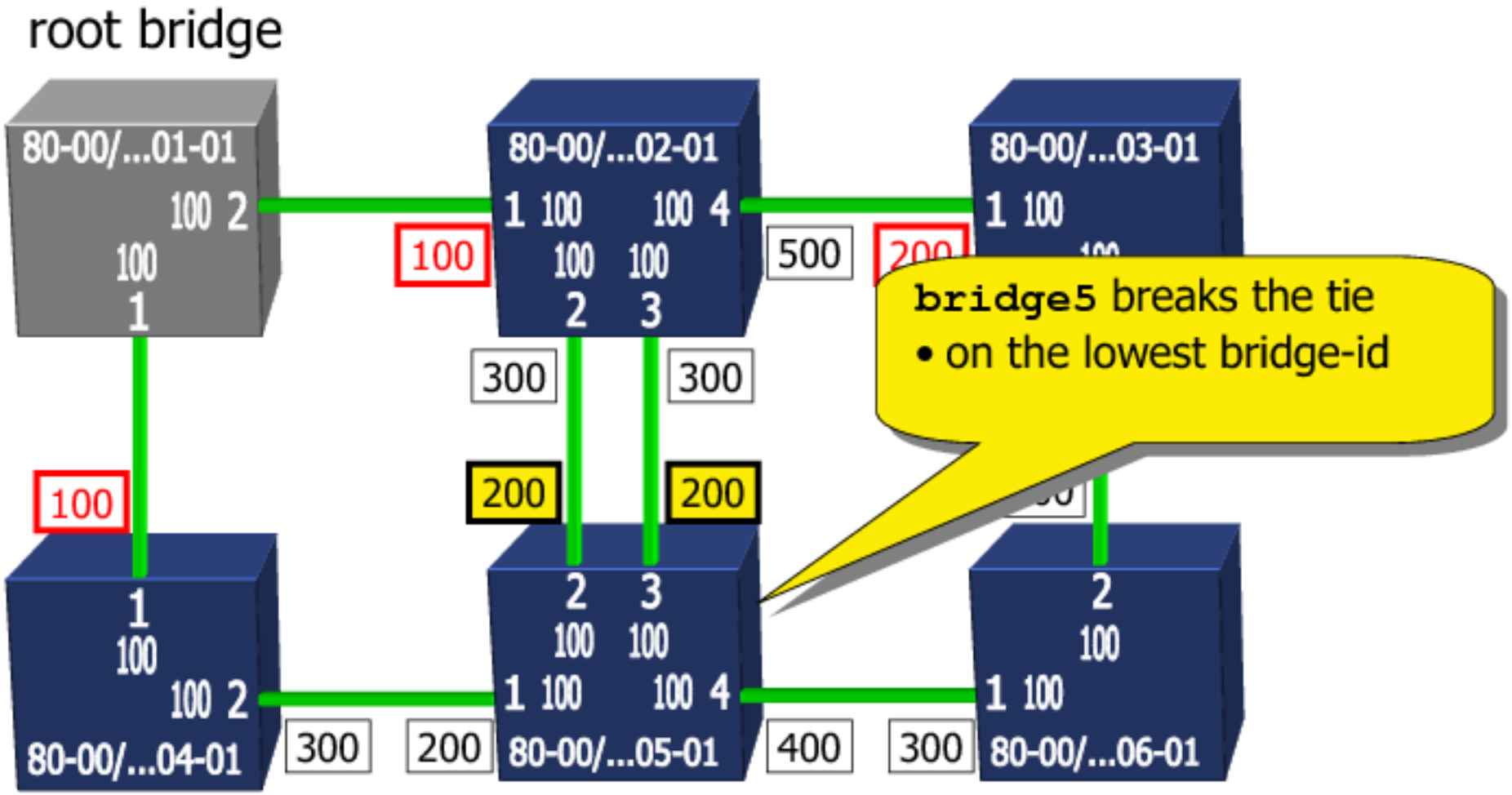
root ports identification



xxx = root port (xxx is the root path cost)

last update: May 2007

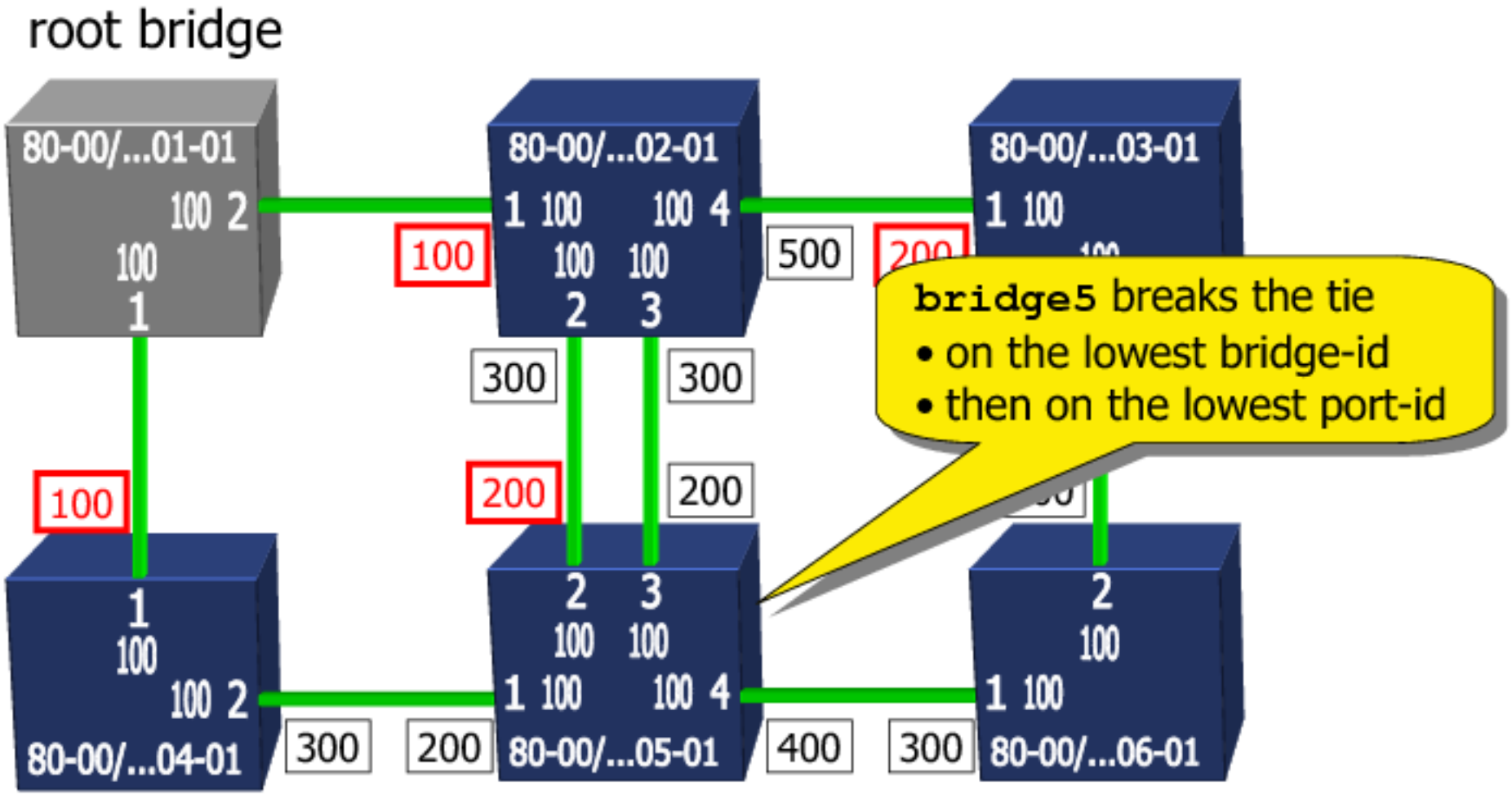
root ports identification



xxx = root port (xxx is the root path cost)

last update: May 2007

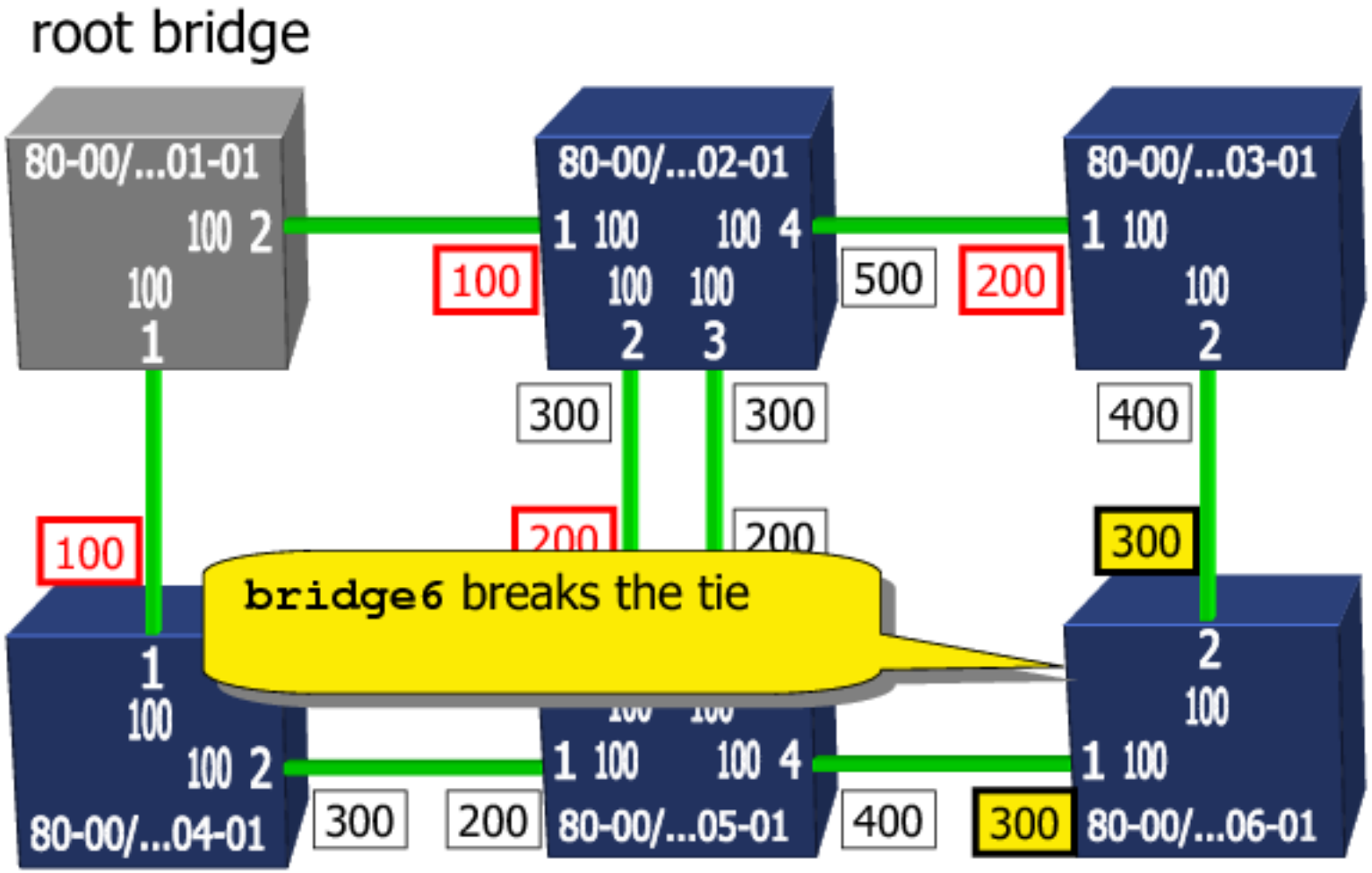
root ports identification



xxx = root port (xxx is the root path cost)

last update: May 2007

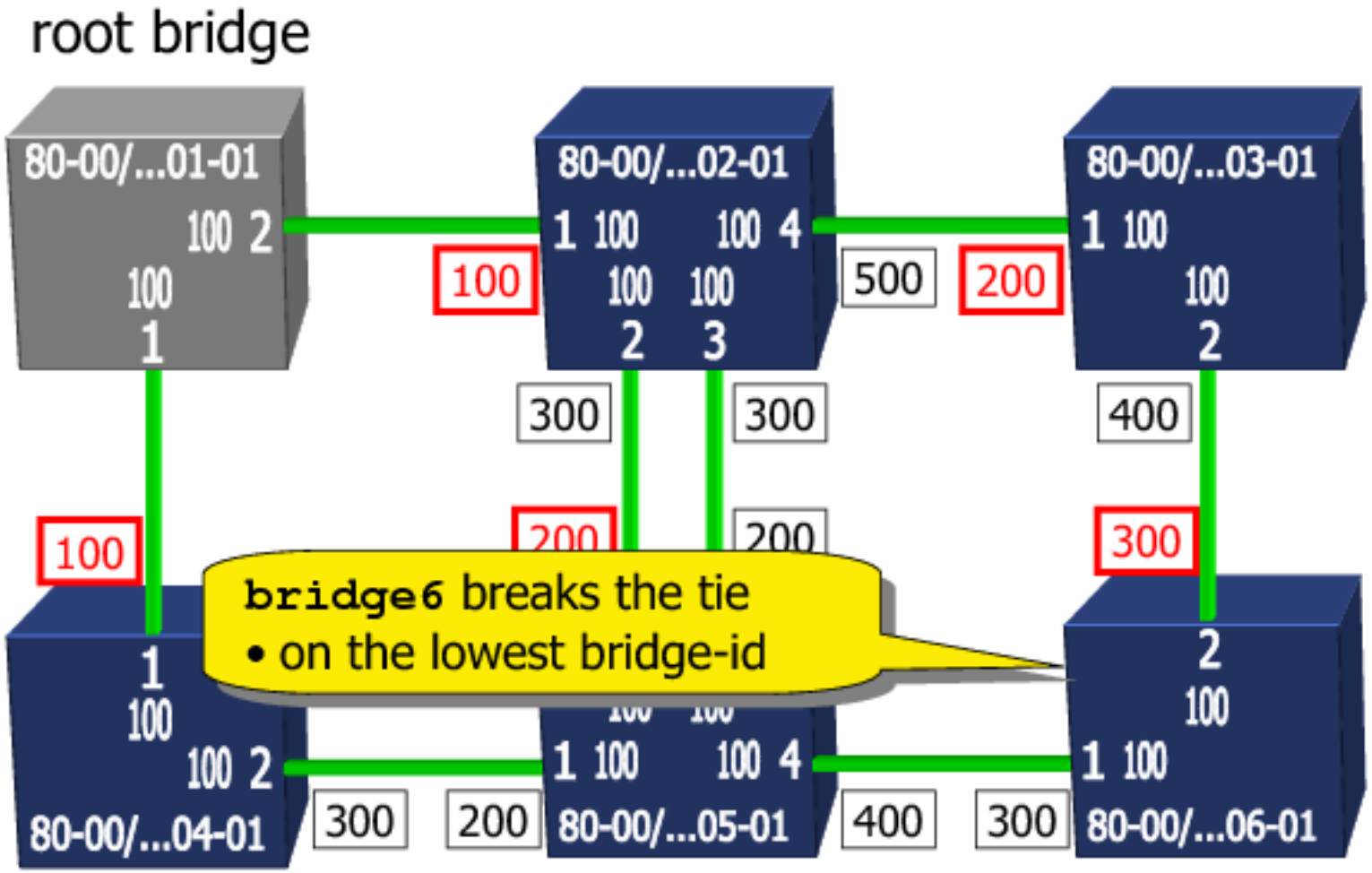
root ports identification



xxx = root port (xxx is the root path cost)

last update: May 2007

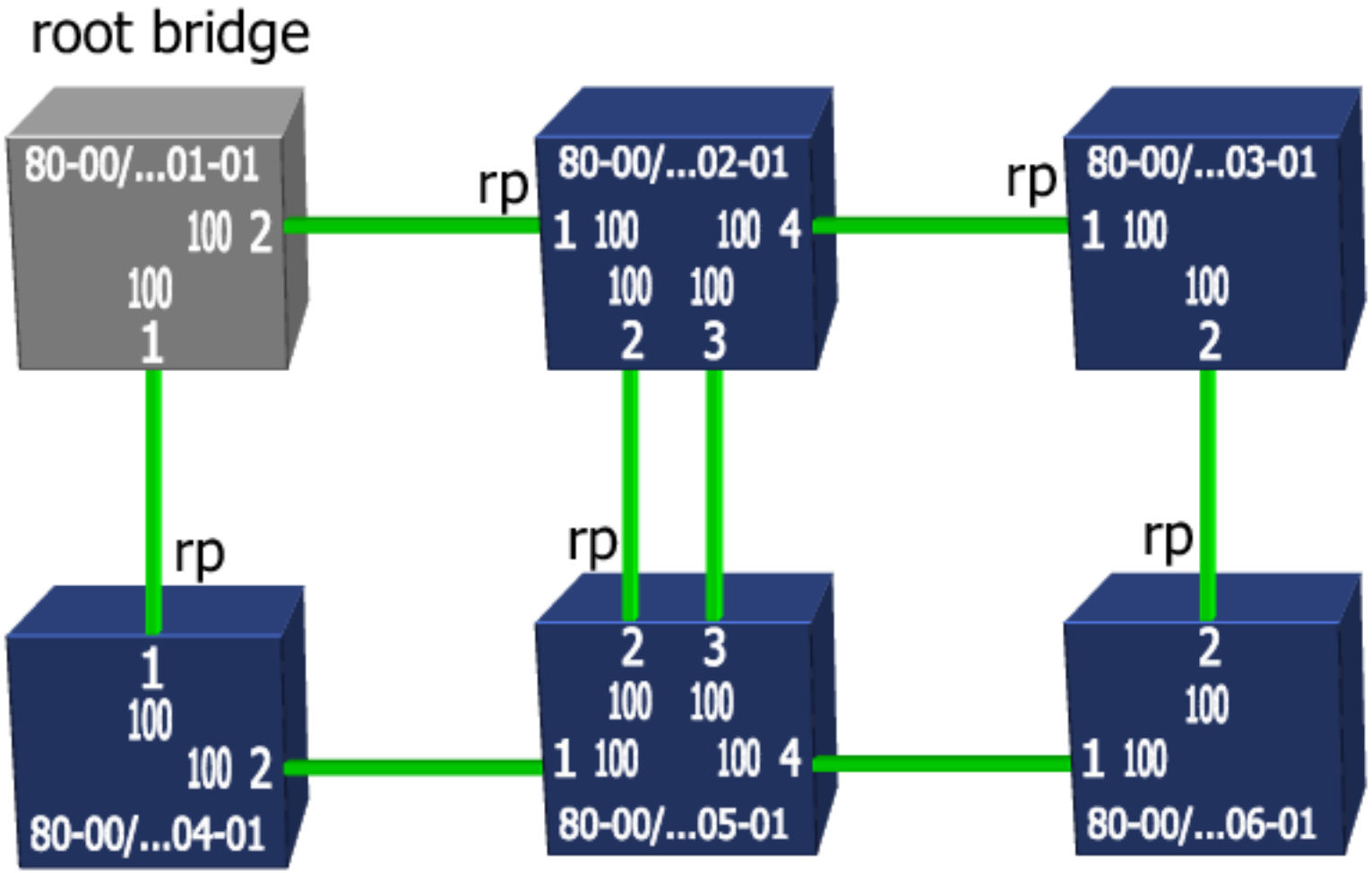
root ports identification



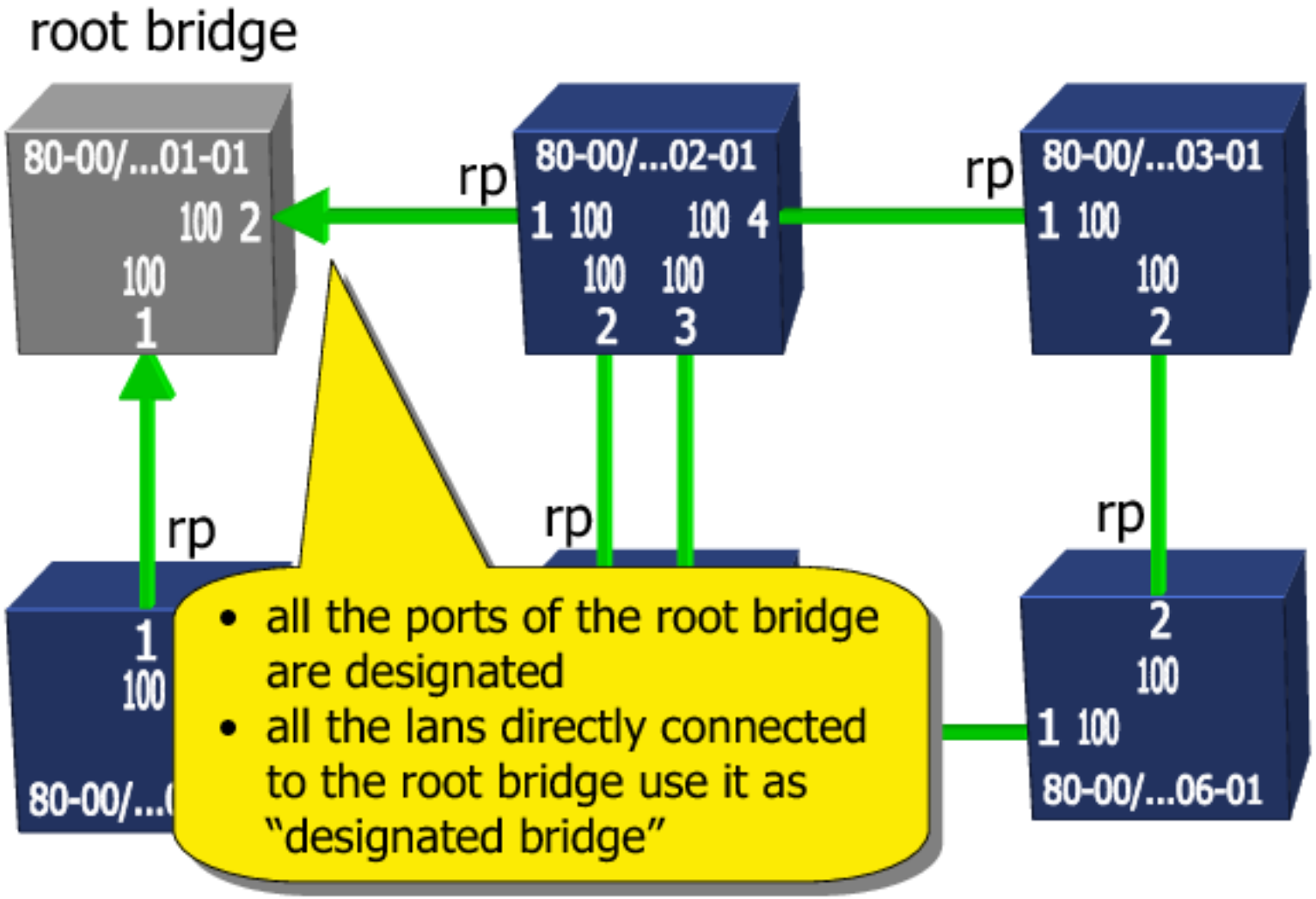
xxx = root port (xxx is the root path cost)

last update: May 2007

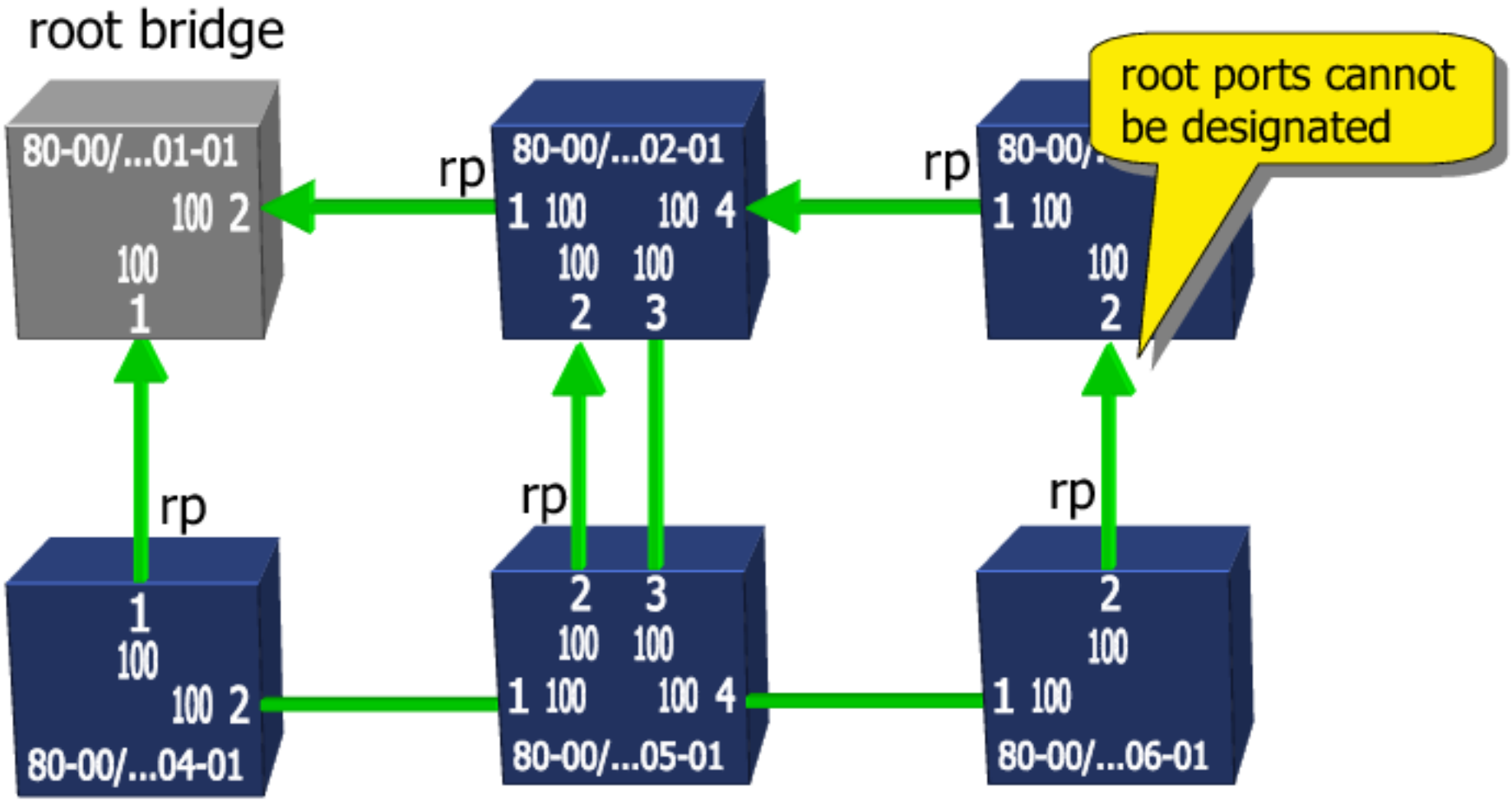
root ports identification



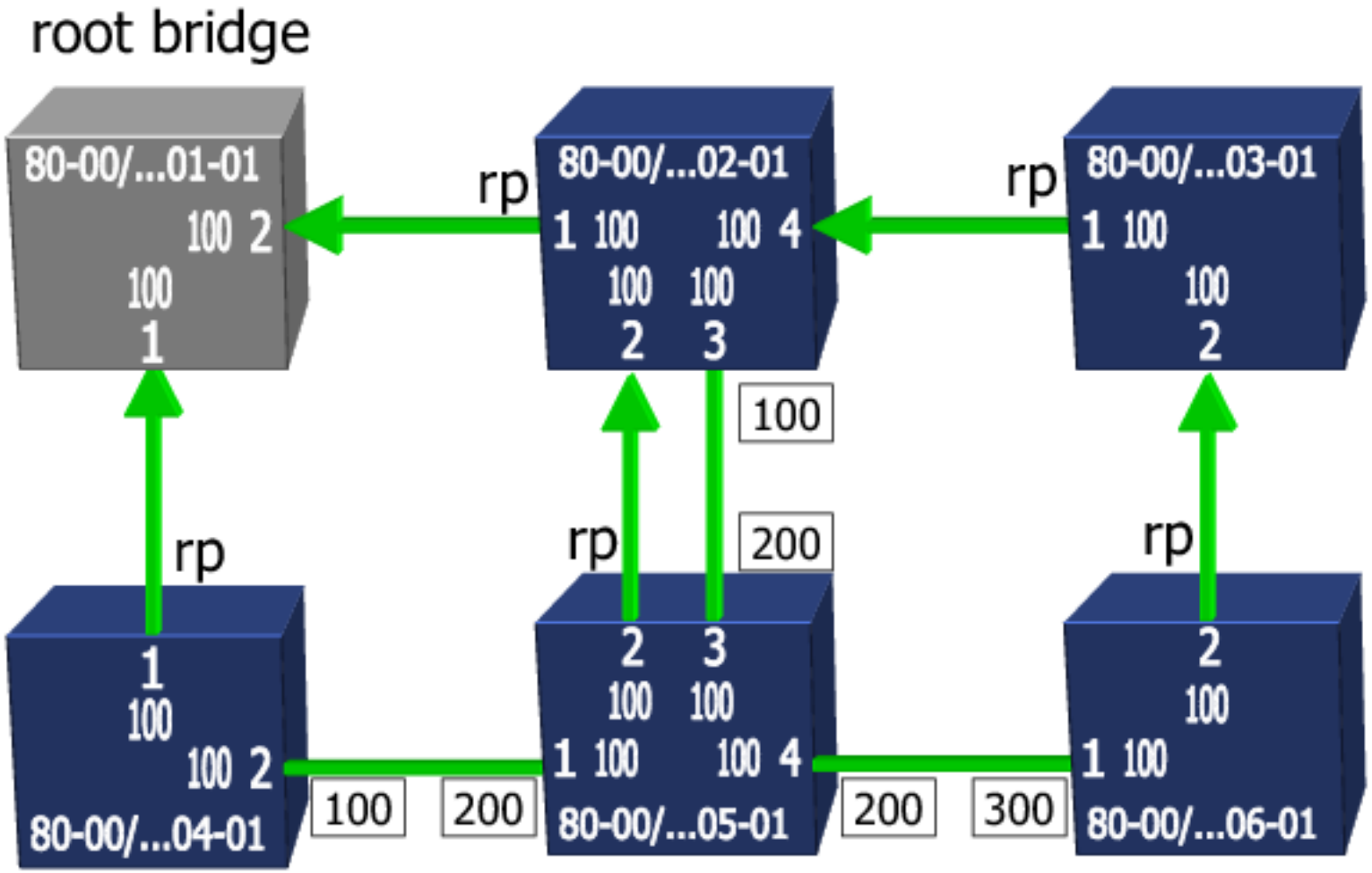
designated ports/bridges identification



designated ports/bridges identification



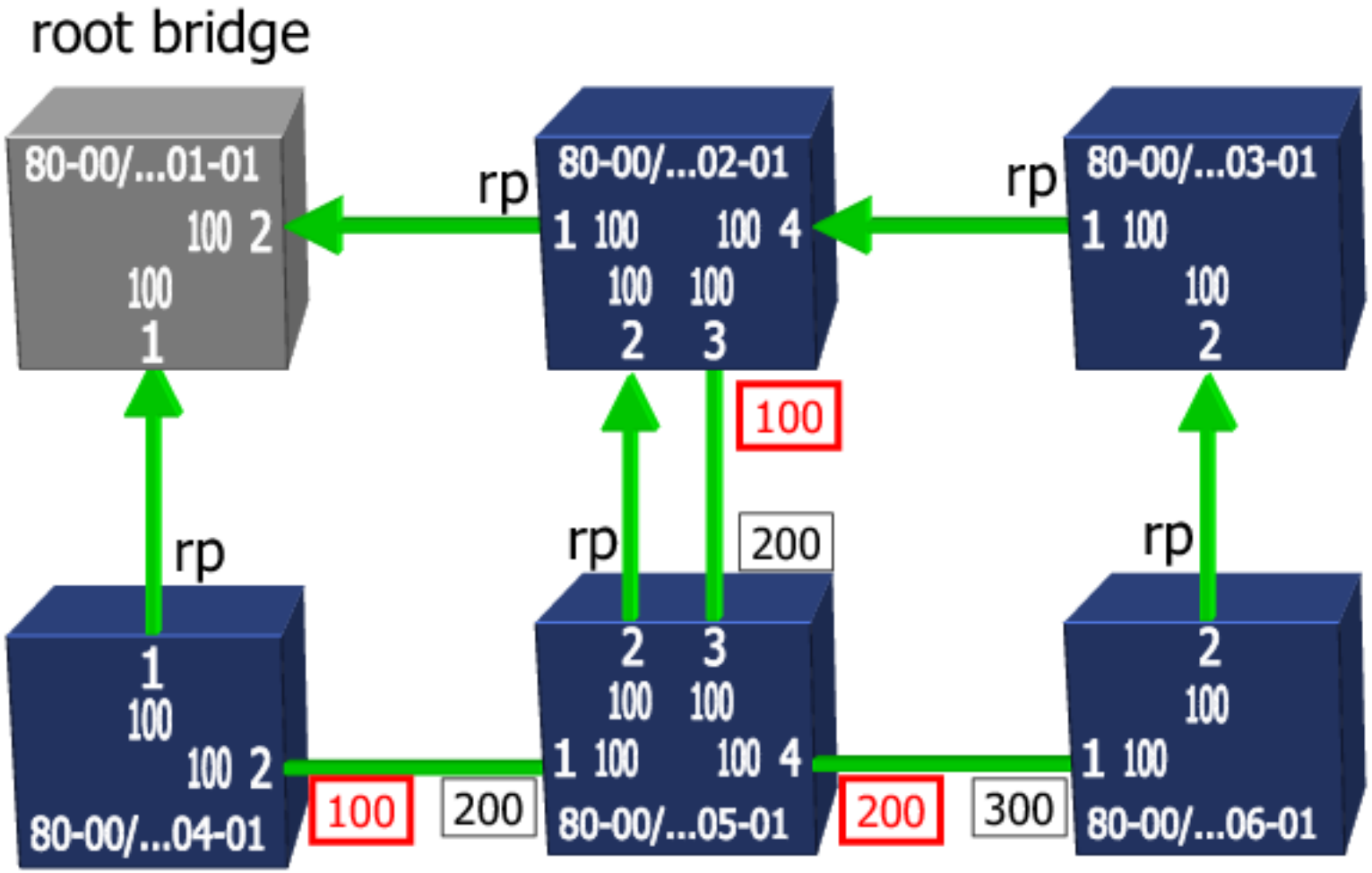
designated ports/bridges identification



xxx = min root-path-cost of bpdus received *by the lan* through a given port

last update: May 2007

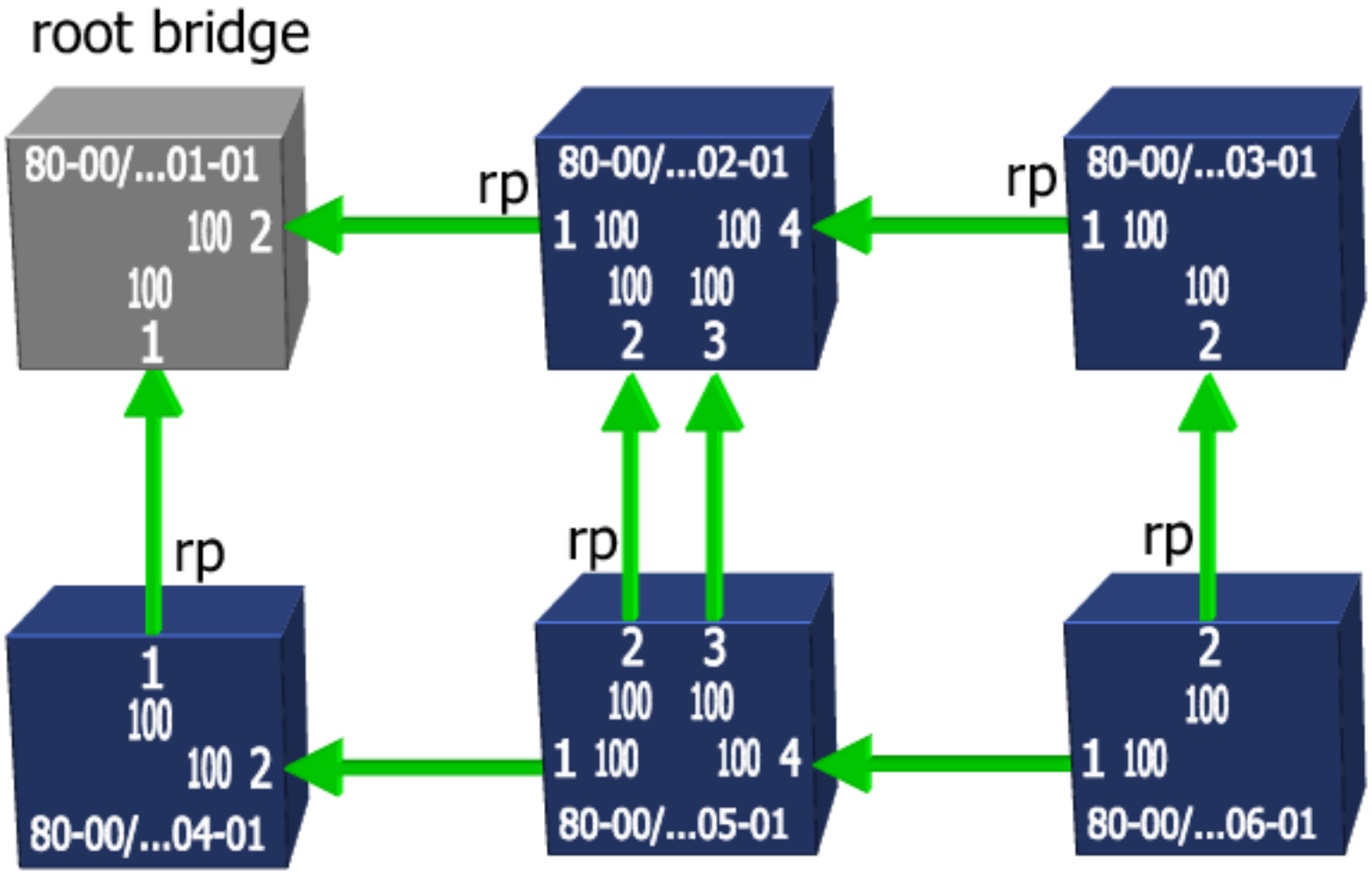
designated ports/bridges identification



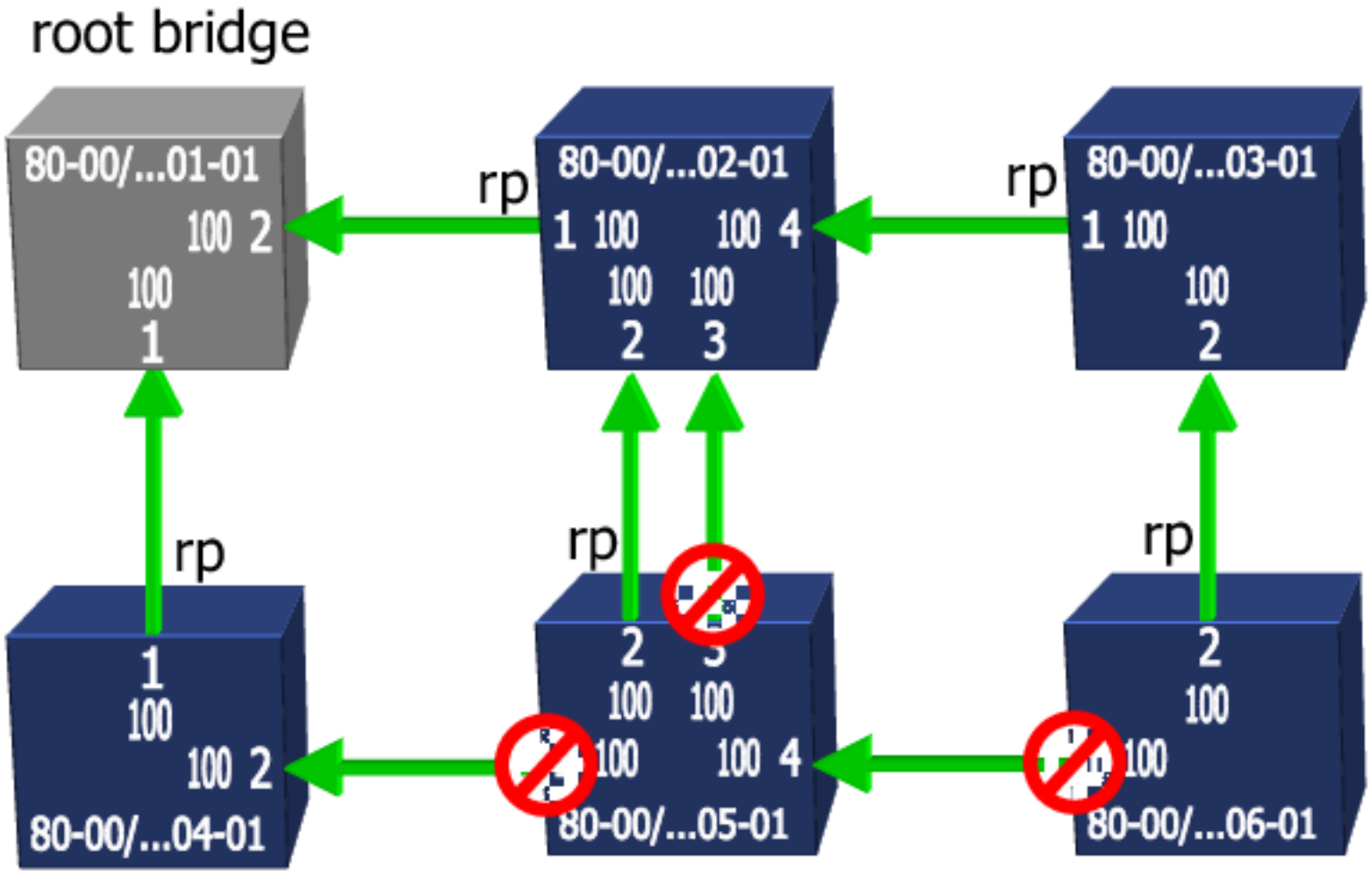
xxx = designated port (xxx is the root path cost)

last update: May 2007

designated ports/bridges identification



blocking


root bridge
80-00/...01-01
100 2
100
1
rp
80-00/...02-01
1 100
100 4
100 100
2 3
rp
80-00/...03-01
1 100
100
2
rp
1
100
100 2
80-00/...04-01
rp
2
100 100
100
100 4
80-00/...05-01
rp
2
100
100
80-00/...06-01

lab4: a more complex scenario

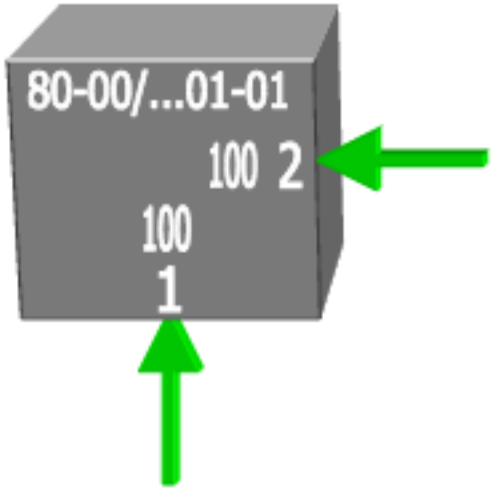
host machine

```
user@localhost:~$ cd netkit-lab_stp-complex
user@localhost:~/netkit-lab_stp-complex$ lstart
```

- the lab is configured to start the 6 bridges and to run the spanning tree protocol (stp) on all of them

stp status

root bridge

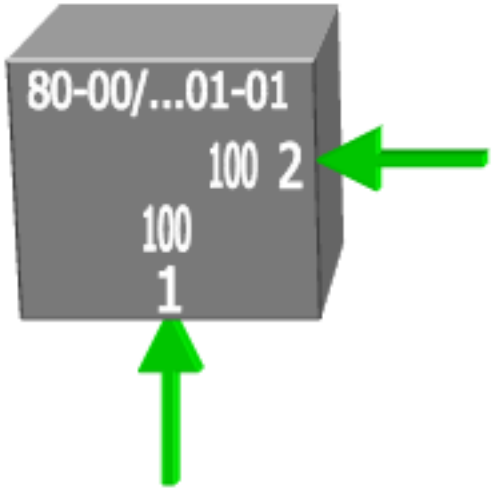


bridge1

```
bridge1:~# brctl showstp br0
br0
 bridge id              8000.000000000101
 designated root        8000.000000000101
 root port                 0                    path cost                  0
 max age                  20.00                 bridge max age            20.00
 hello time                2.00                 bridge hello time          2.00
 forward delay            15.00                 bridge forward delay      15.00
 ageing time             300.00
 hello timer               1.57                 tcn timer                  0.00
 topology change timer 0.00                     gc timer                  91.36
 flags

eth0 (1)
 port id                8001                    state               forwarding
 designated root        8000.000000000101       path cost                 100
 designated bridge      8000.000000000101       message age timer          0.00
 designated port        8001                    forward delay timer        0.00
 designated cost           0                    hold timer                 0.57
 flags

eth1 (2)
 port id                8002                    state               forwarding
 designated root        8000.000000000101       path cost                 100
 designated bridge      8000.000000000101       message age timer          0.00
 designated port        8002                    forward delay timer        0.00
 designated cost           0                    hold timer                 0.57
 flags
```

stp status

root bridge



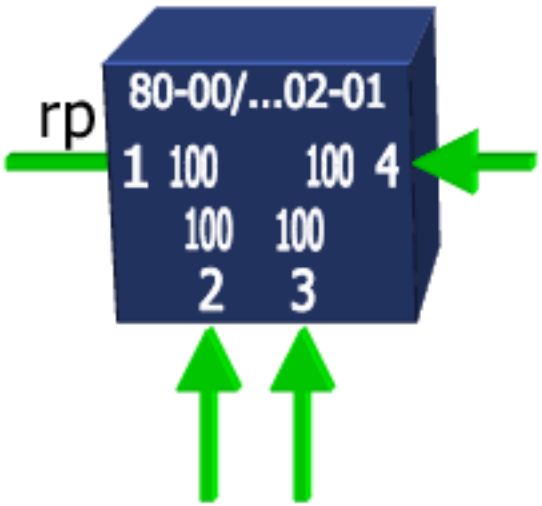
bridge1

```
bridge1:~# brctl showstp br0
br0
 bridge id              8000.000000000101
 designated root        8000.000000000101
 root port                 0                    path cost                  0
 max age                                        bridge max age            20.00
 hello time                                     bridge hello time          2.00
 forward delay                                  bridge forward delay      15.00
 ageing time
 hello timer                                    tcn timer                  0.00
 topology change timer 0.00                     gc timer                  91.36
 flags

eth0 (1)
 port id              8001                      state                forwarding
 designated root      8000.000000000101         path cost                  100
 designated bridge    8000.000000000101         message age timer          0.00
 designated port      8001                      forward delay timer        0.00
 designated cost         0                      hold timer                 0.57
 flags

eth1 (2)
 port id              8002                      state                forwarding
 designated root      8000.000000000101         path cost                  100
 designated bridge    8000.000000000101         message age timer          0.00
 designated port      8002                      forward delay timer        0.00
 designated cost         0                      hold timer                 0.57
 flags
```

root bridge

root path cost

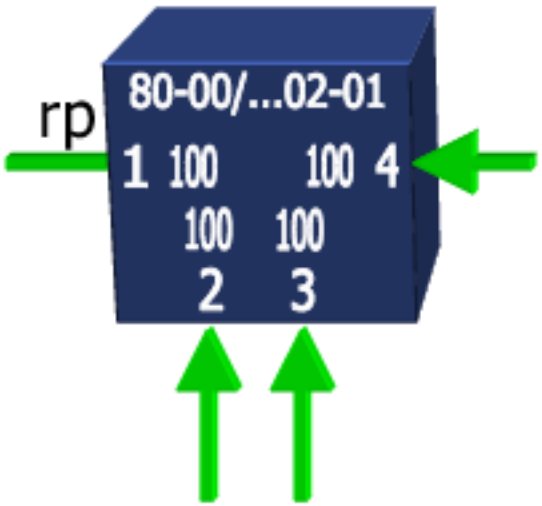
stp status



bridge2

```
bridge2:~# brctl showstp br0
br0
 bridge id              8000.000000000201
 designated root        8000.000000000101
 root port                 1                    path cost                  100
.....
eth0 (1)
 port id                8001                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000101       message age timer        19.67
 designated port        8002                    forward delay timer       0.00
.....
eth1 (2)
 port id                8002                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer         0.00
 designated port        8002                    forward delay timer       0.00
.....
eth2 (3)
 port id                8003                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer         0.00
 designated port        8003                    forward delay timer       0.00
.....
eth3 (4)
 port id                8004                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer         0.00
 designated port        8004                    forward delay timer       0.00
.....
```

stp status

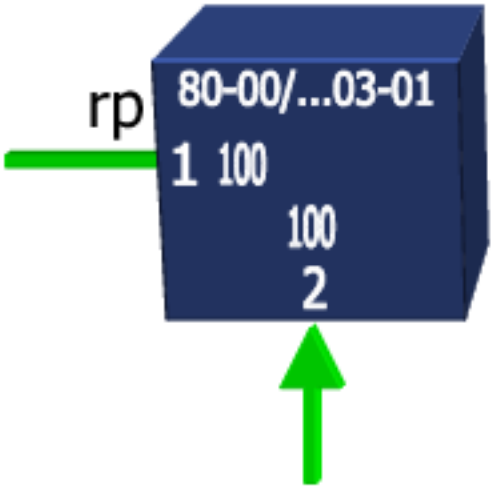


bridge2

```
bridge2:~# brctl showstp br0
br0
 bridge id              8000.000000000
 designated root        8000.000000000101
 root port                 1                    path cost                  100
.....
eth0 (1)
 port id                8001                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000101       message age timer        19.67
 designated port        8002                    forward delay timer       0.00
.....
eth1 (2)
 port id                8002                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer         0.00
 designated port        8002                    forward delay timer       0.00
.....
eth2 (3)
 port id                8003                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer         0.00
 designated port        8003                    forward delay timer       0.00
.....
eth3 (4)
 port id                8004                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer         0.00
 designated port        8004                    forward delay timer       0.00
.....
```

designated bridge for the lan connected to eth*x* (*x*+1)

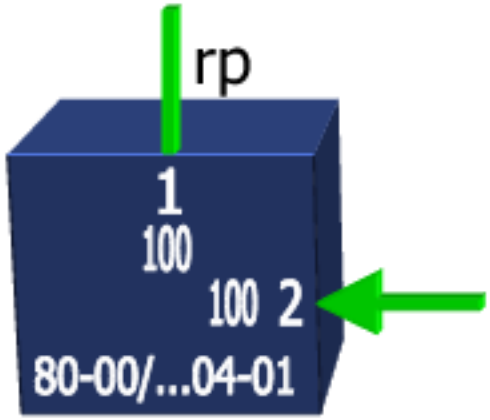
stp status



bridge3

```
bridge3:~# brctl showstp br0
br0
 bridge id              8000.000000000301
 designated root        8000.000000000101
 root port                 1                    path cost                  200
 max age                  20.00                 bridge max age            20.00
 hello time                2.00                 bridge hello time          2.00
 forward delay            15.00                 bridge forward delay      15.00
 ageing time             300.00
 hello timer               0.00                 tcn timer                  0.00
 topology change timer 0.00                     gc timer                 189.82
 flags


eth0 (1)
 port id                8001                    state                forforwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer         19.91
 designated port        8004                    forward delay timer        0.00
 designated cost         100                    hold timer                 0.00
 flags

eth1 (2)
 port id                8002                    state                forforwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000301       message age timer          0.00
 designated port        8002                    forward delay timer        0.00
```

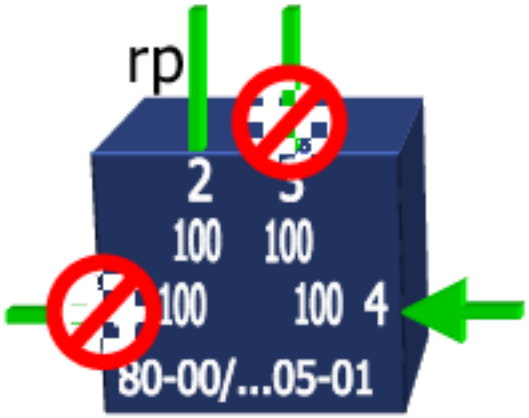
stp status



bridge4

```
bridge4:~# brctl showstp br0
br0
 bridge id              8000.000000000401
 designated root        8000.000000000101
 root port                 1                    path cost                  100
 max age                  20.00                 bridge max age            20.00
 hello time                2.00                 bridge hello time          2.00
 forward delay            15.00                 bridge forward delay      15.00
 ageing time             300.00
 hello timer               0.00                 tcn timer                  0.00
 topology change timer 0.00                     gc timer                 289.91
 flags


eth0 (1)
 port id                8001                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000101       message age timer         19.91
 designated port        8001                    forward delay timer        0.00
 designated cost           0                    hold timer                 0.00
 flags

eth1 (2)
 port id                8002                    state                forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000401       message age timer          0.00
 designated port        8002                    forward delay timer        0.00
 designated cost         100                    hold timer                 0.91
 flags
```

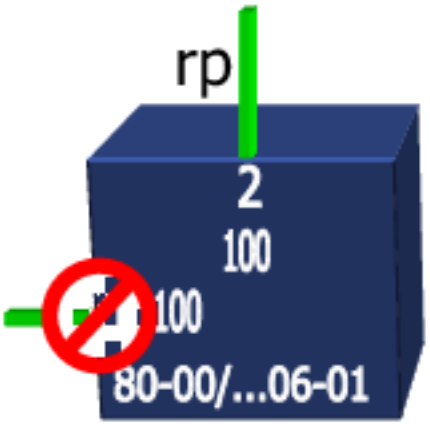
stp status



bridge5

```
bridge5:~# brctl showstp br0
br0
 bridge id              8000.000000000501
 designated root        8000.000000000101
 root port                 2                    path cost                  200
.....
eth0 (1)
 port id                8001                    state                 blocking
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000401       message age timer        18.24
 designated port        8002                    forward delay timer       0.00
.....
eth1 (2)
 port id                8002                    state               forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer        18.24
 designated port        8002                    forward delay timer       0.00
.....
eth2 (3)
 port id                8003                    state                 blocking
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000201       message age timer        18.24
 designated port        8003                    forward delay timer       0.00
.....
eth3 (4)
 port id                8004                    state               forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000501       message age timer         0.00
 designated port        8004                    forward delay timer       0.00
.....
```

stp status



bridge6

```
bridge6:~# brctl showstp br0
br0
 bridge id              8000.000000000601
 designated root        8000.000000000101
 root port                 2                    path cost                  300
 max age                  20.00                 bridge max age            20.00
 hello time                2.00                 bridge hello time          2.00
 forward delay            15.00                 bridge forward delay      15.00
 ageing time             300.00
 hello timer               0.00                 tcn timer                  0.00
 topology change timer 0.00                     gc timer                 133.65
 flags

eth0 (1)
 port id                8001                    state                 blocking
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000501       message age timer        19.82
 designated port        8004                    forward delay timer       0.00
 designated cost         200                    hold timer                0.00
 flags

eth1 (2)
 port id                8002                    state               forwarding
 designated root        8000.000000000101       path cost                  100
 designated bridge      8000.000000000301       message age timer        19.71
 designated port        8002                    forward delay timer       0.00
 designated cost         200                    hold timer                0.00
 flags
```

further experiments

- try changing the root bridge by setting the bridge priorities
- try using both ports of a specified link by using port priority
  - why is this difficult?

bridge

```
bridge:~# brctl
commands:
.....
        setbridgeprio   <bridge> <prio>          set bridge priority
.....
        setportprio     <bridge> <port> <prio>   set port priority
```