ROMA TRE
UNIVERSITÀ DEGLI STUDI

INGEGNERIA
CIVILE, INFORMATICA E DELLE TECNOLOGIE AERONAUTICHE

Internet and Data Centers

un algoritmo per il calcolo dello spanning tree nelle reti locali

G. Di Battista, M. Patrignani

copyright notice

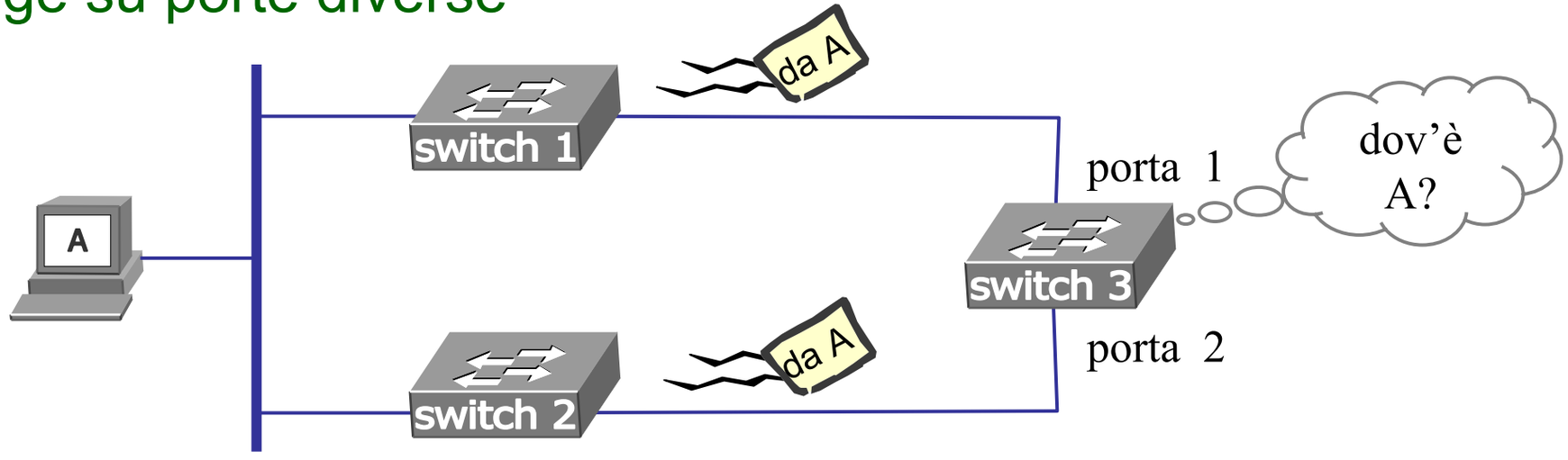
- all the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright
- this material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide
- this material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes
- any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement
- this copyright notice must always be redistributed together with the material, or its portions

l'algoritmo sta e le reti locali

- lo *spanning tree algorithm* (*sta*) è pensato per essere utilizzato in una rete locale nella quale siano presenti switch (bridge)
- si assume che gli switch compilino automaticamente le proprie tabelle d'instradamento facendo backward learning
  - l'interfaccia su cui instradare un pacchetto viene inferita in base agli indirizzi mittente dei pacchetti in ingresso

le reti locali e i cicli

- in una rete locale costituita da bridge e domini di collisione la presenza di cicli renderebbe impossibile l'operazione di backward learning
  - i pacchetti provenienti dallo stesso mittente potrebbero giungere a un bridge su porte diverse



- rinunciare del tutto alla presenza di cicli comporterebbe la rinuncia alla ridondanza offerta da questi

scopo dello spanning tree algorithm

- lo *spanning tree algorithm* (*sta*) è un algoritmo distribuito preposto al calcolo di un albero ricoprente (spanning tree) in una rete *magliata* (cioè con cicli)
- grazie allo spanning tree algorithm è possibile:
  - garantire il funzionamento del backward learning
    - utilizzando per l'inoltro dei pacchetti esclusivamente l'albero computato
  - conservare la ridondanza offerta dai cicli
    - nel caso di guasti l'albero viene ricalcolato

lo spanning tree algorithm in sintesi

- è descritto nello standard IEEE 802.1D
- è un algoritmo distribuito che calcola un albero ricoprente della rete
  - l'albero calcolato è *radicato* a un bridge che viene scelto come radice: il *root bridge*
  - al termine della computazione alcuni bridge mettono alcune porte in "blocking state"
    - le porte in blocking non vengono utilizzate
- l'amministratore può influenzare l'esito del calcolo modificando opportuni parametri di configurazione

requisiti dell'algoritmo

- robustezza
  - l'albero ricoprente deve essere computato in maniera distribuita
- efficacia
  - la formazione di cicli deve essere esclusa sia in condizioni stazionarie che transitorie
- efficienza
  - lo spanning tree calcolato deve corrispondere ad un uso efficiente delle risorse di rete disponibili
- bandwidth
  - pacchetti scambiati dai bridge (*bridge pdu*, *bpdu*) devono comportare un limitato consumo di banda

requisiti dell'algoritmo

- tempo
  - la computazione dell'albero deve essere più veloce possibile per limitare il disservizio
    - tipicamente 30-40 secondi sono sufficienti allo sta per convergere
- flessibilità
  - l'amministratore deve poter assegnare priorità a ciascun bridge e a ciascuna porta per influire sull'output dell'algoritmo
- facilità d'uso
  - l'algoritmo deve essere in grado di funzionare correttamente anche in assenza di configurazione dell'amministratore

input dello spanning tree algorithm

- una topologia costituita da:
  - un insieme di LAN (domini di collisione)
  - un insieme di bridge le cui porte afferiscono alle LAN
- un identificatore (detto *bridge-id*) distinto per ogni bridge della rete
  - un id più basso indica un bridge preferibile
- un identificatore (detto *port-id*) per ogni porta di ogni bridge
  - unico all'interno del bridge
  - un id più basso indica una porta preferibile
- un valore (detto *costo*) per ogni LAN
  - il costo viene effettivamente associato alle porte che afferiscono alla LAN e convenzionalmente *sommato in ingresso*

output dello spanning tree algorithm

- un insieme di porte che, una volta messe in blocking, garantiscano:
  - che ci sia piena connettività
    - esista un cammino tra ogni coppia di LAN
  - che non ci siano cicli
    - il cammino tra ogni coppia di LAN sia unico
- caratteristiche desiderabili
  - il costo dei cammini sia contenuto
    - la minimalità non è garantita
    - si minimizza il costo dei soli cammini tra la radice dell'albero ricoprente e qualsiasi altro bridge

identificatori dei bridge e delle loro porte

- il *bridge-id* è costituito dalla concatenazione di due parti:
  - 2 byte di “priorità” scelti dall’amministratore (il default è spesso: 80-00)
  - 6 byte corrispondenti all’indirizzo mac della prima porta del bridge (per esempio: 23-ef-c0-4b-93-a0)
  - è dunque lungo 8 byte (per esempio: 80-00/23-ef-c0-4b-93-a0)
- anche la *port-id* è costituita dalla concatenazione di due parti:
  - un byte di “priorità” scelto dall’amministratore (il default è spesso: 80)
  - un byte corrispondente al numero della porta sul bridge (per esempio: 0a per la decima porta)
  - è dunque lunga 2 byte (per esempio: 80/0a)

costi delle LAN

- il *costo* di attraversamento delle LAN (domini di collisione) è specificato sulle porte afferenti alle LAN stesse
- può essere modificato dall'amministratore
- il valore di default è inversamente proporzionale alla banda della tecnologia utilizzata:
  - costo 100 per una LAN a 10 Mb/s
  - costo 10 per una LAN a 100 Mb/s
  - costo 1 per una LAN a 1 Gb/s

fasi dello spanning tree algorithm

1. elezione del *root bridge*
   - un singolo bridge è selezionato per essere la radice dell'albero
2. identificazione della *root port* su ogni bridge
   - ogni bridge diverso dal root bridge seleziona una delle sue porte come quella "più conveniente" per connettersi al root bridge
3. determinazione delle *designated ports*
   - per ogni LAN una porta di un bridge è scelta come quella che connetterà la LAN allo spanning tree
4. blocco di porte ridondanti
   - le porte inutilizzate (non root ports e non designated ports) sono messe in *blocking*

pacchetti utilizzati

- i pacchetti scambiati dai bridge vengono chiamati *bpdu* (bridge protocol data unit)
- le bpdu viaggiano a bordo di
  - pacchetti LLC con dsap = ssap = 0x42, che a loro volta sono incapsulati in...
  - pacchetti MAC con sorgente l'indirizzo della porta mittente e destinazione multicast 01:80:c2:00:00:00
- sono previsti due tipi di bpdu:
  - la *configuration bpdu* contiene tutte le informazioni necessarie per lo spanning tree algorithm
  - la *topology change bpdu* non contiene nessun dato
    - non la vedremo in dettaglio, serve solo a segnalare che un cambio di topologia è in atto e a gestire i timer del protocollo

configuration bpdu

- la *configuration bpdu* contiene in particolare le seguenti informazioni:
  - *root-bridge-identifier*
    - l'attuale root dello spanning tree
  - *root-path-cost*
    - il costo del cammino verso il root bridge
  - *bridge-identifier*
    - l'id del bridge che invia questa bpdu
  - *port-identifier*
    - la porta da cui è uscita questa bpdu
- esempio:

root-bridge-identifier	**80-00/23-ef-00-a1-32-4d**
root-path-cost	**310**
bridge-identifier	**80-00/2d-12-d4-23-8e-5f**
port-identifier	**80/06**

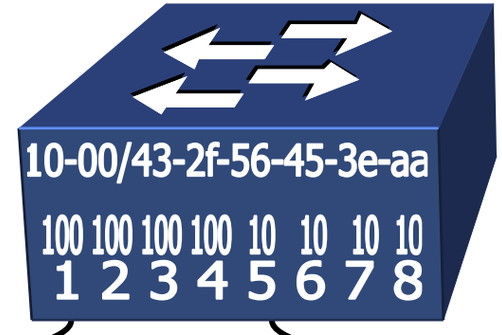
fase 1: elezione del root bridge

- ogni bridge invia una configuration bpdu nella quale specifica il proprio bridge-id come root-identifier
- quando un bridge riceve una configuration bpdu con un valore più basso di bridge-id
  - smette di produrre configuration bpdu con il suo bridge-id come root-identifier
  - propaga la nuova configuration bpdu su tutte le porte
- il root bridge è quello che continuerà a produrre configuration bpdu con il suo bridge-id nel campo root-identifier

inoltro delle configuration bpdu

- quando una configuration bpdu è prodotta dal root bridge il suo campo root-path-cost è posto a zero
- quando una configuration bpdu è inoltrata da un bridge che non è il root bridge, i suoi campi sono aggiornati come segue:
  - il root-path-cost è incrementato con il costo della porta del bridge che riceve la configuration bpdu
  - il bridge-identifier è sostituito con il bridge-id del bridge corrente
  - la port-identifier è sostituita con l'id della porta da cui la bpdu sarà inviata

root-bridge-id	**00-00/23-ef-…**
root-path-cost	**100**
bridge-id	**10-00/2d-12-…**
port-id	**00/06**



root-bridge-id	**00-00/23-ef-…**
root-path-cost	**200**
bridge-id	**10-00/43-2f-…**
port-id	**00/05**

fase 2: identificazione delle root port

- ogni bridge diverso dal root bridge identifica la porta attraverso la quale il root bridge è più facilmente raggiungibile
- la *root port* è quella che riceve le configuration bpdu tali che (in ordine di priorità):
  1. il root-path-cost della bpdu sommato al costo della porta ricevente è il più basso
  2. il bridge-identifier specificato nella bpdu è il più basso
  3. la port-identifier specificata nella bpdu è la più bassa
  4. la port-identifier della porta ricevente è la più bassa

fase 3: determinazione delle designated port

- per ogni LAN una porta di un bridge è scelta come *designated port* in base alle configuration bpdu che sono inviate nella LAN da quella porta
- la *designated port* è quella che invia le configuration bpdu con (in ordine di priorità):
  1. root-path-cost più basso
  2. bridge-identifier più basso
  3. port-identifier più basso

fase 4: blocking

- tutte le porte che non sono root-ports o designated-ports sono poste in blocco
- tutte le root-ports e designated-ports sono poste in stato di forwarding

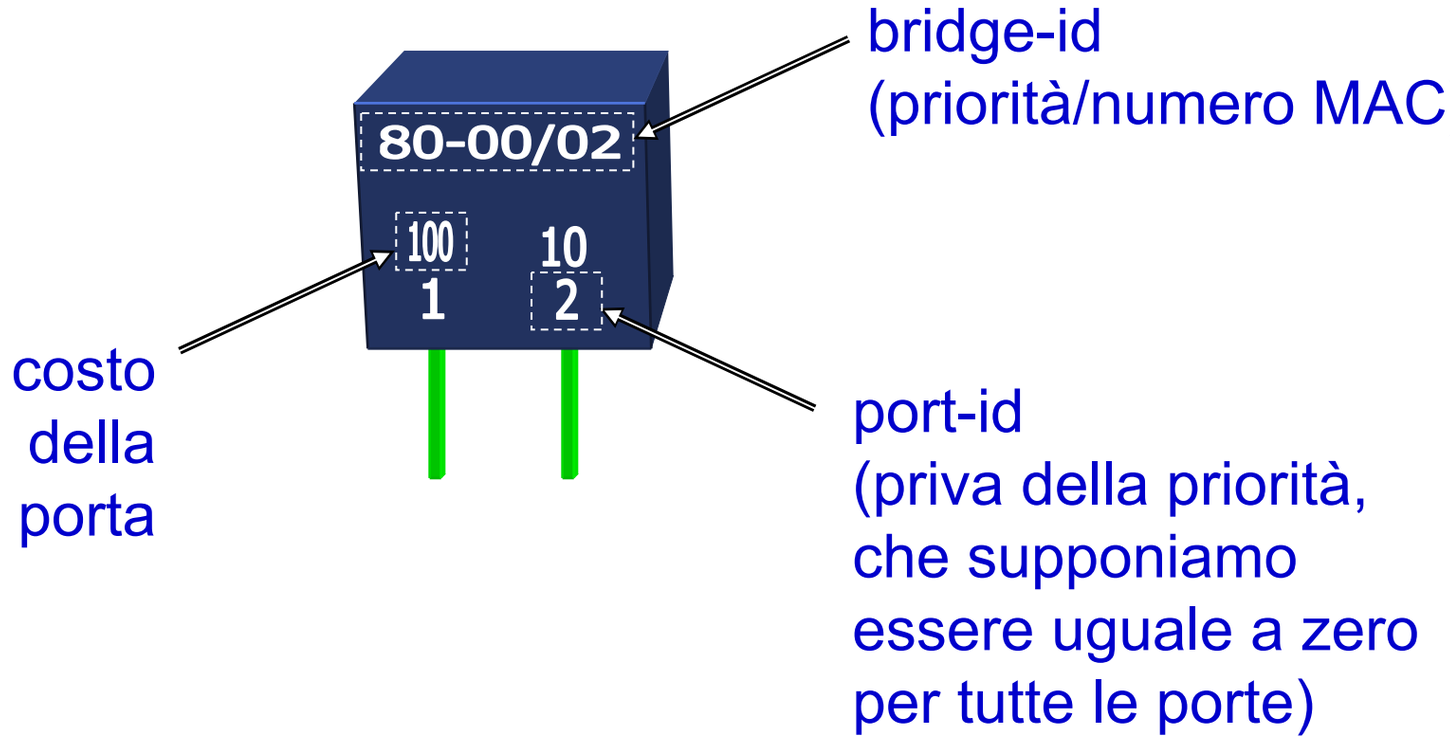
topology change notification bpdu

- un bridge può rilevare un cambiamento di topologia nella rete
  - un cambiamento di topologia può generare cicli, questi a loro volta possono portare a pacchetti che girano a vuoto e/o a saturazione della rete
- il bridge che rileva un cambiamento di topologia lo comunica al root bridge con una *topology change notification bpdu*
  - la topology change notification bpdu raggiunge il root bridge attraverso le root port
  - il root bridge comunica il cambiamento con delle configuration bpdu con topology change flag ad 1
  - il pacchetto topology change notification viene trasmesso dal bridge fino a quando non riceve una configuration bpdu con acknowledgement
  - tutti i bridge abbassano i valori dei timer del protocollo in risposta alla temporanea instabilità della rete

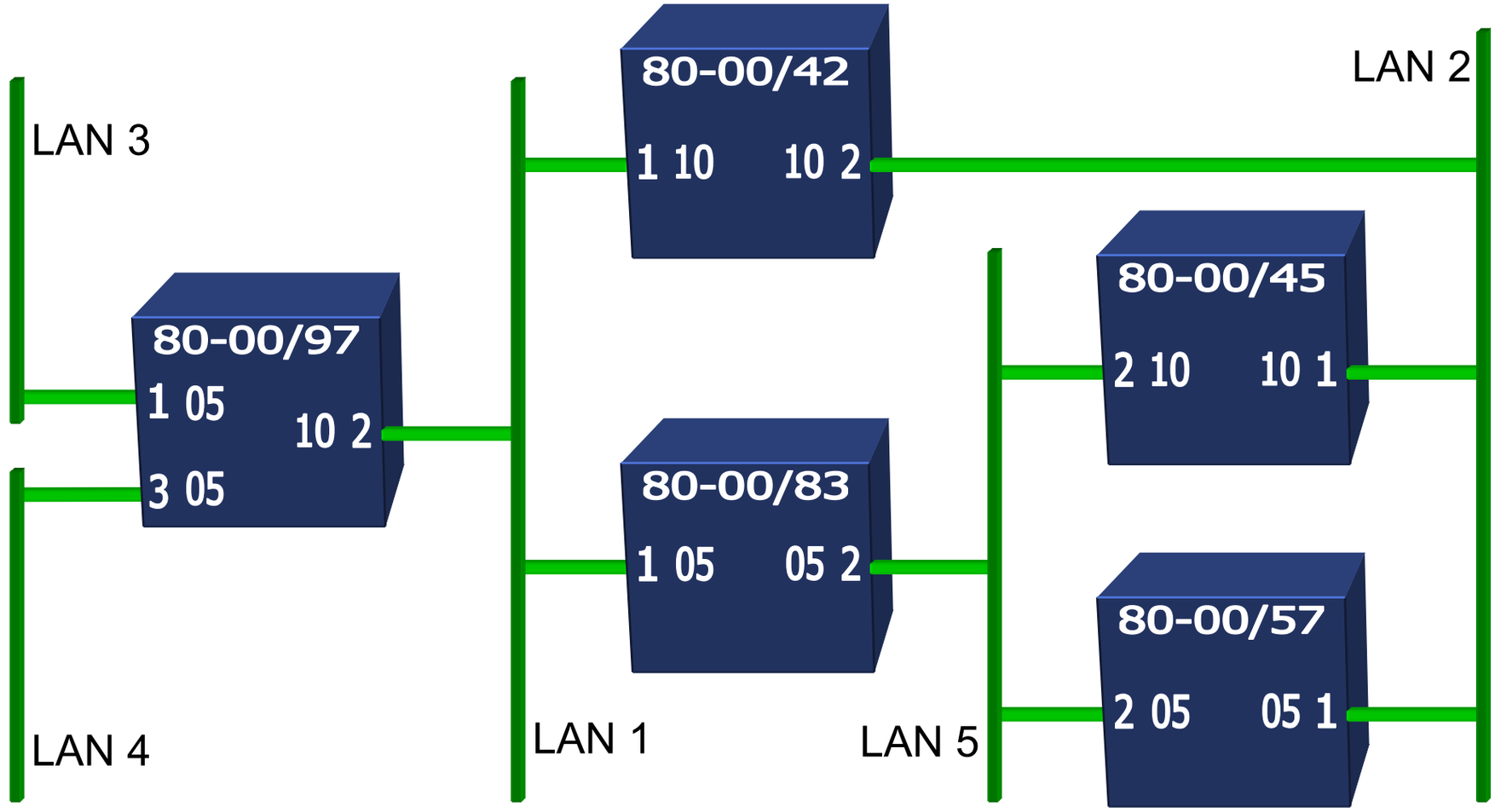
valori contenuti nelle configuration bpdu

- flag di topology change
- flag di topology change acknowledgement
- root-bridge-id (identificatore del root bridge)
- root-path-cost (aggiornato ad ogni attraversamento di un bridge)
- sender bridge-id (id del bridge che ha trasmesso la bpdu)
- sender port-id (id della porta che ha trasmesso la bpdu)
- tempo stimato da quando il root-bridge ha emesso la bpdu
- tempo in cui scartare la bpdu
- tempo tra due bpdu successive
- tempo da attendere per effettuare le transizioni delle porte *listening-learning-forwarding*

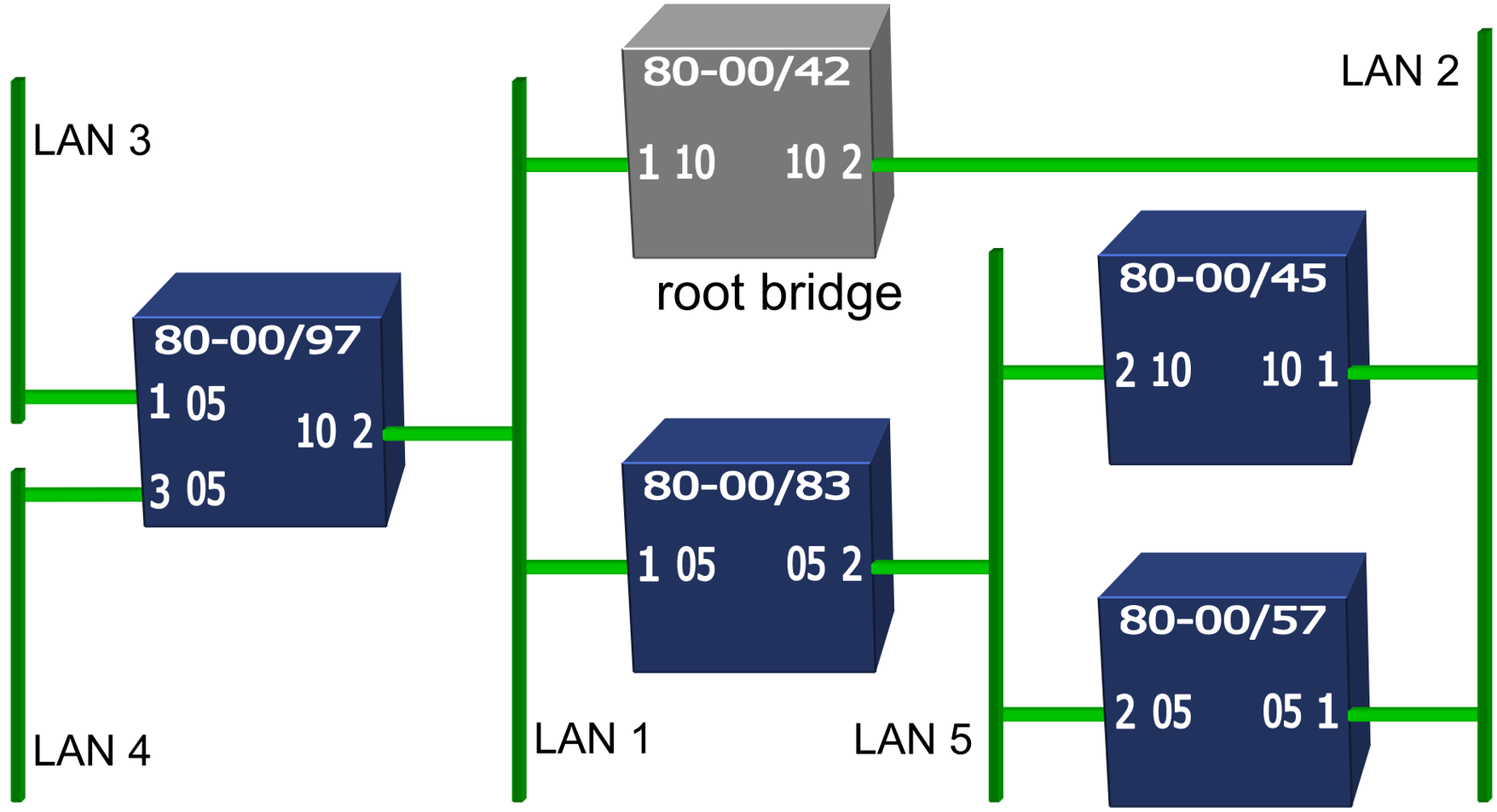
notazione utilizzata negli esempi ed esercizi



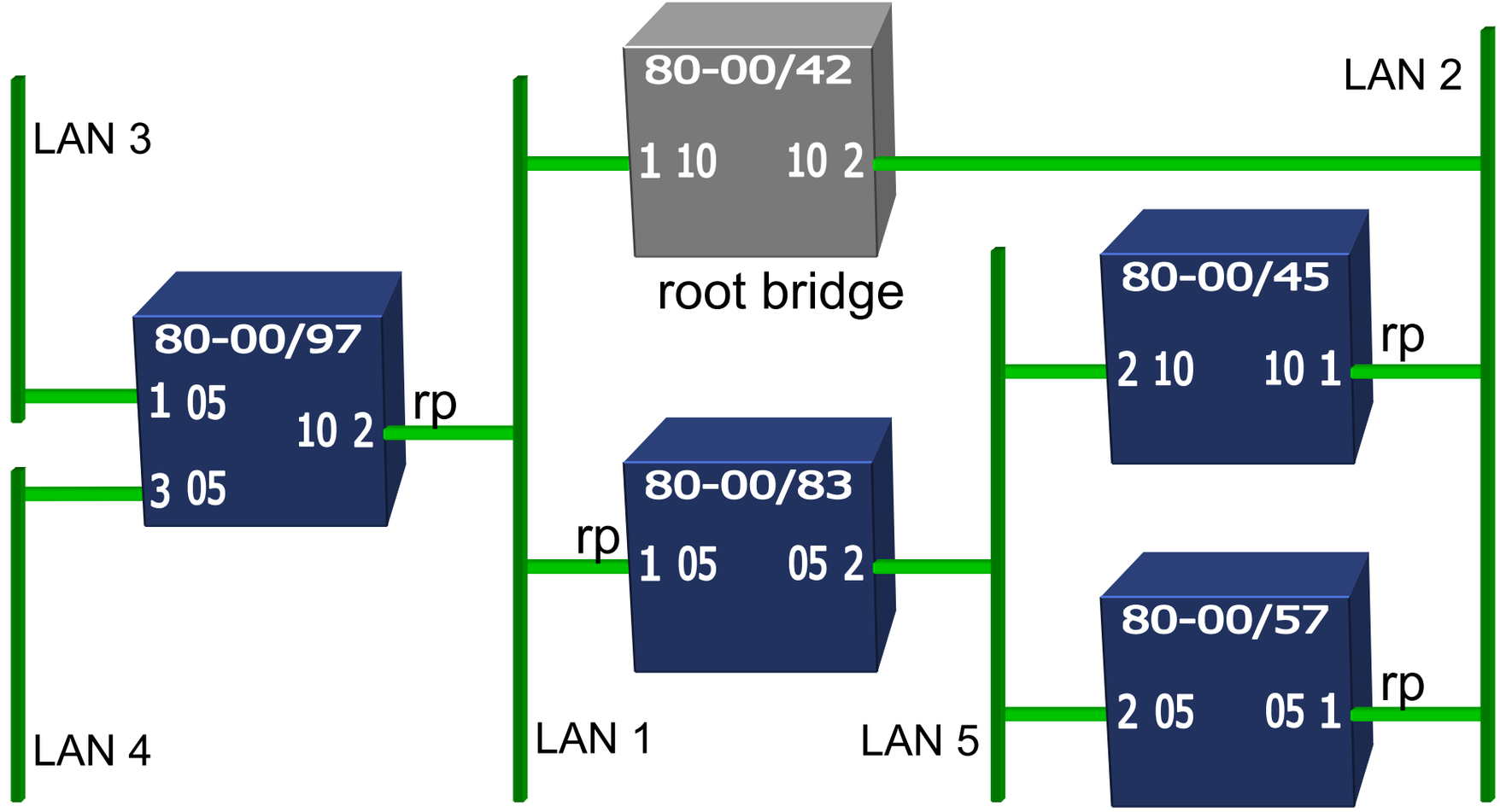
esempio (dallo standard IEEE 802.1D)



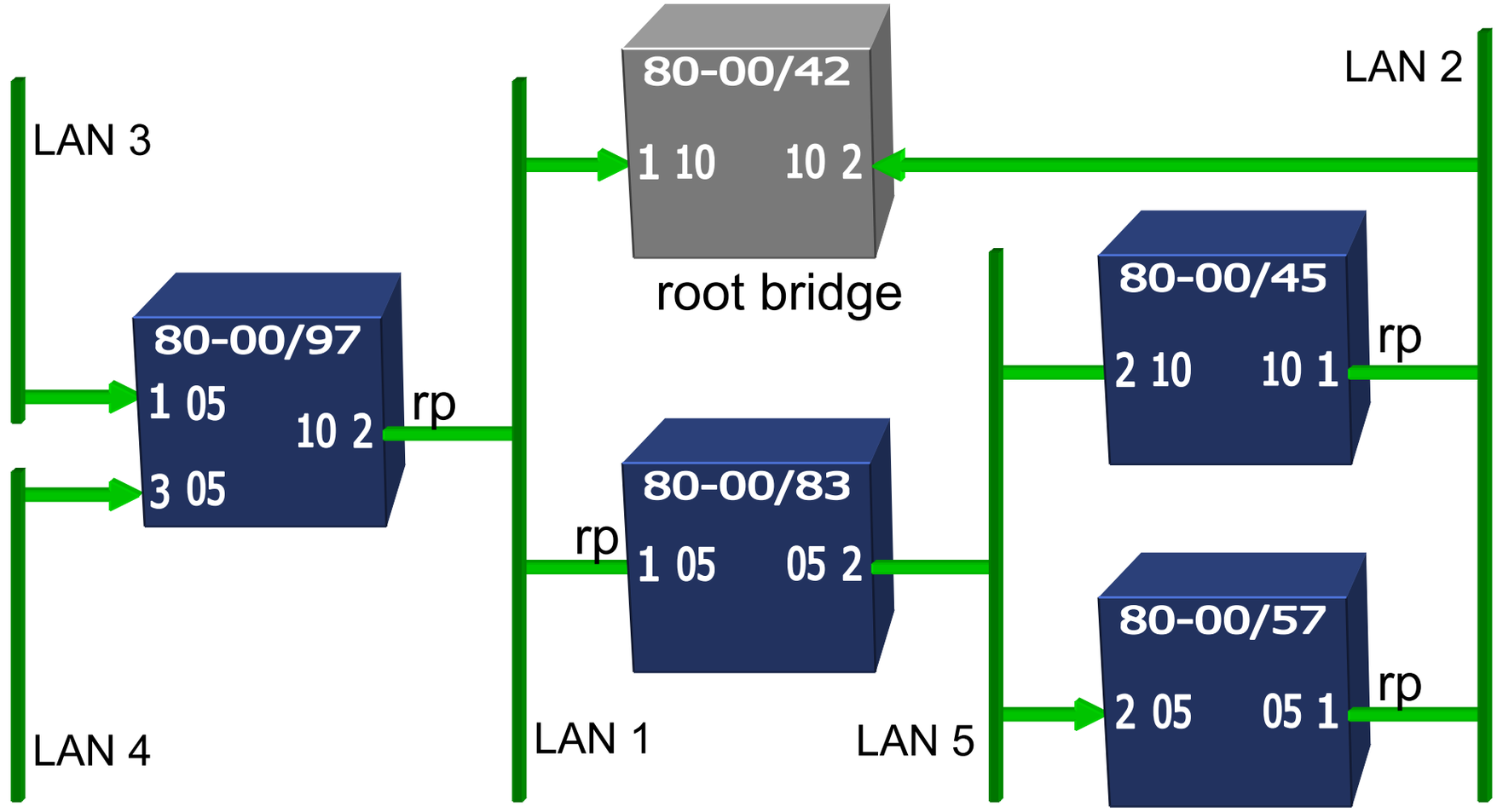
fase 1: elezione del root bridge



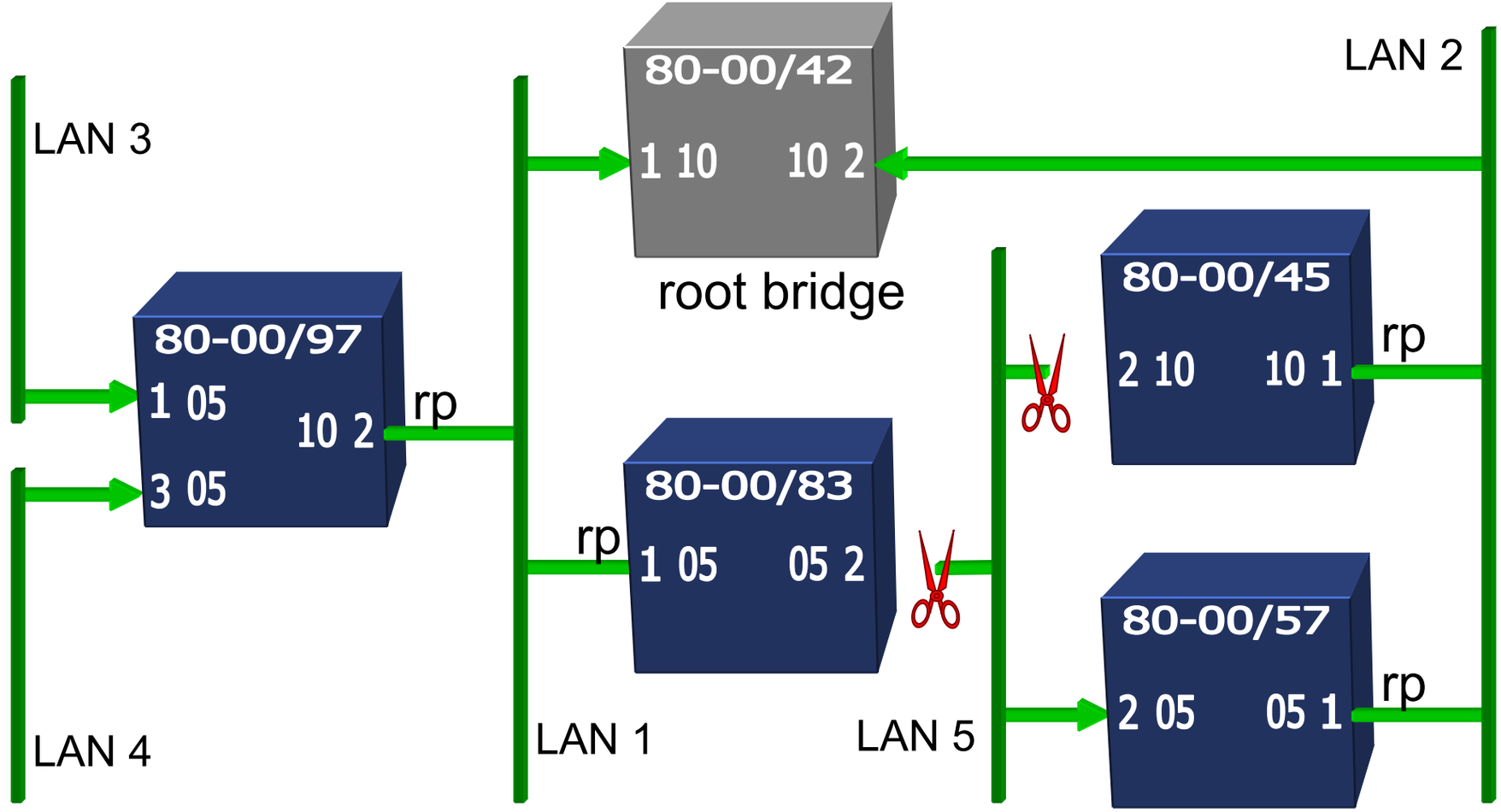
fase 2: identificazione delle root port



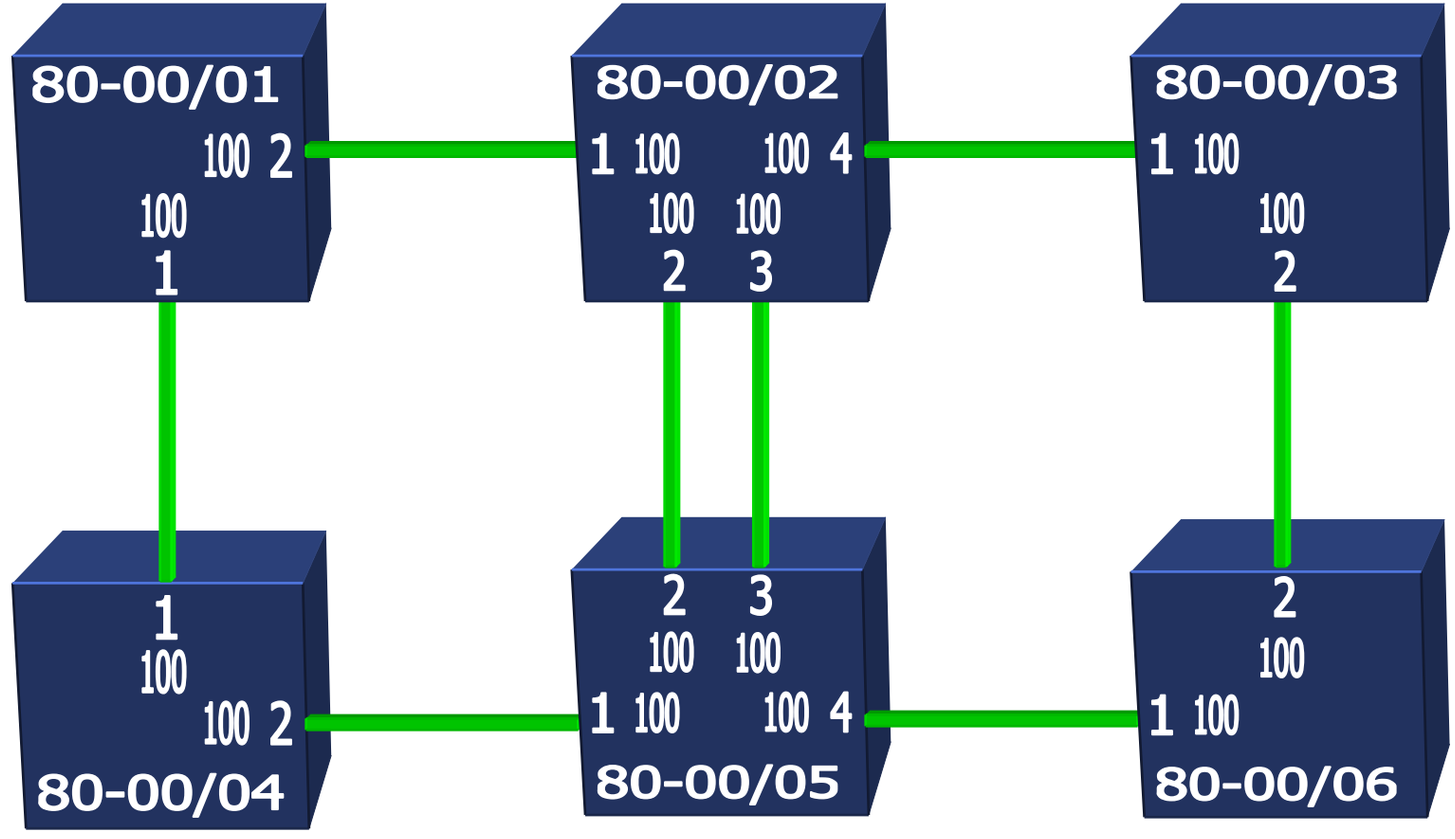
fase 3: determinazione delle designated port



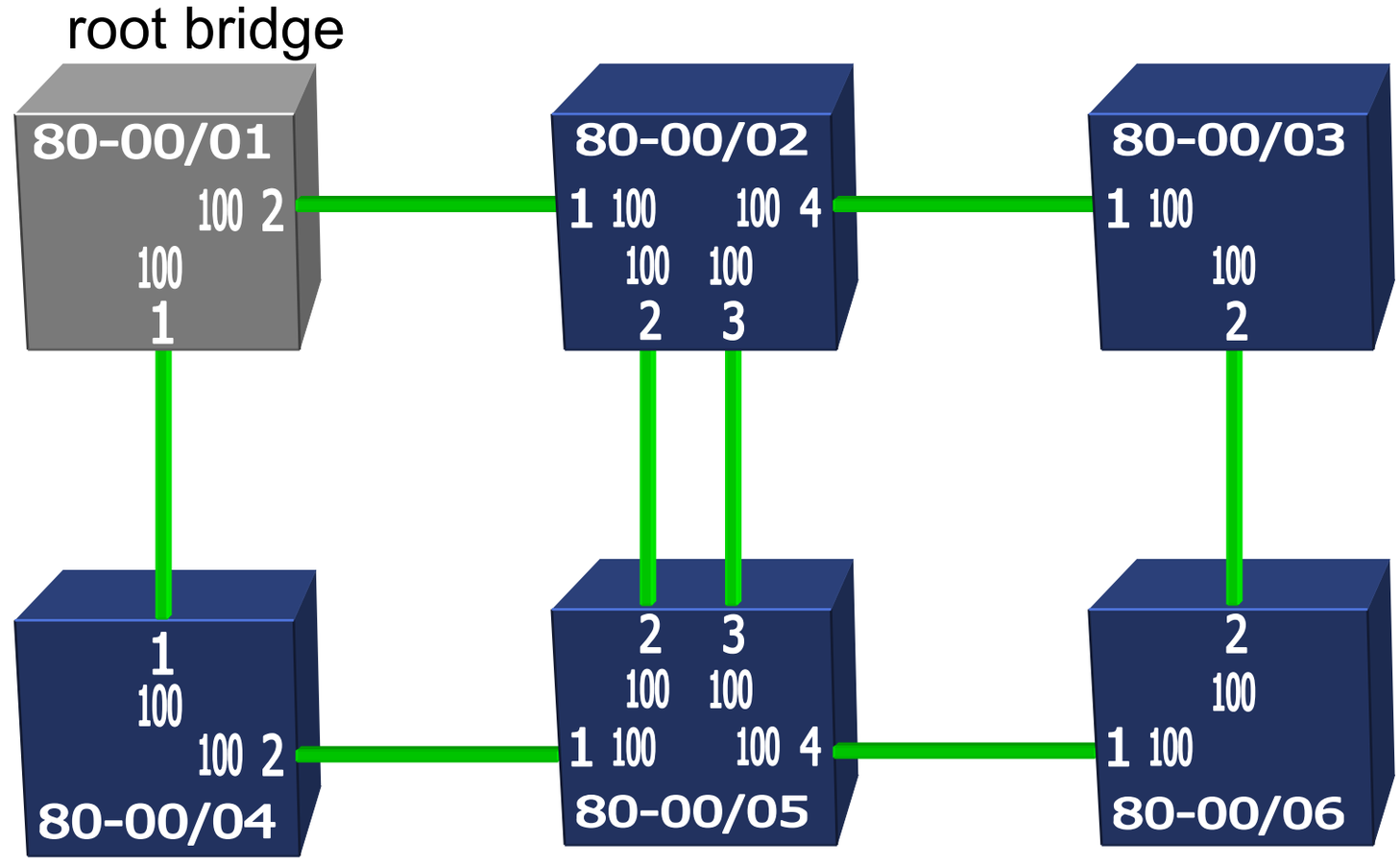
fase 4: blocking



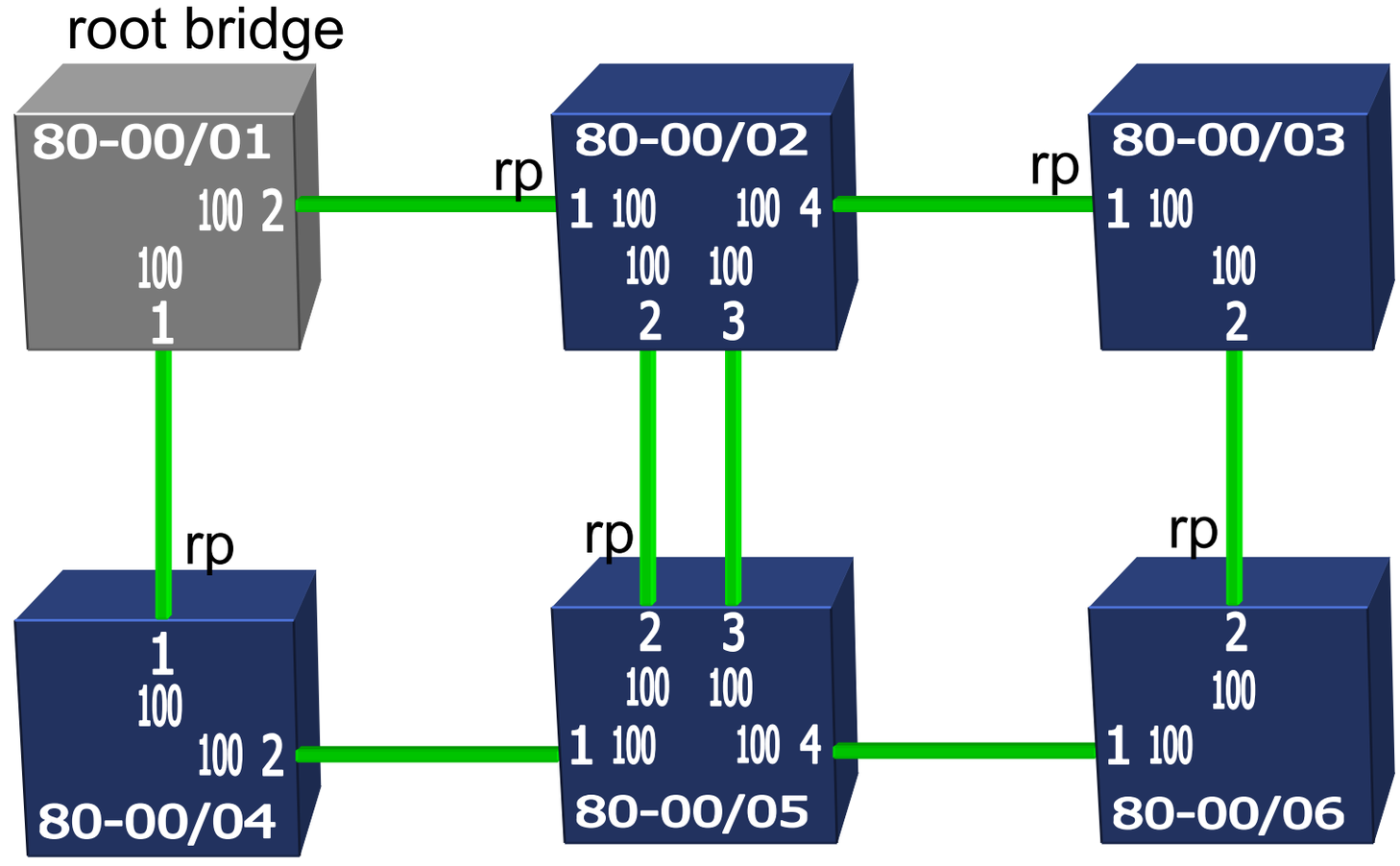
altro esempio



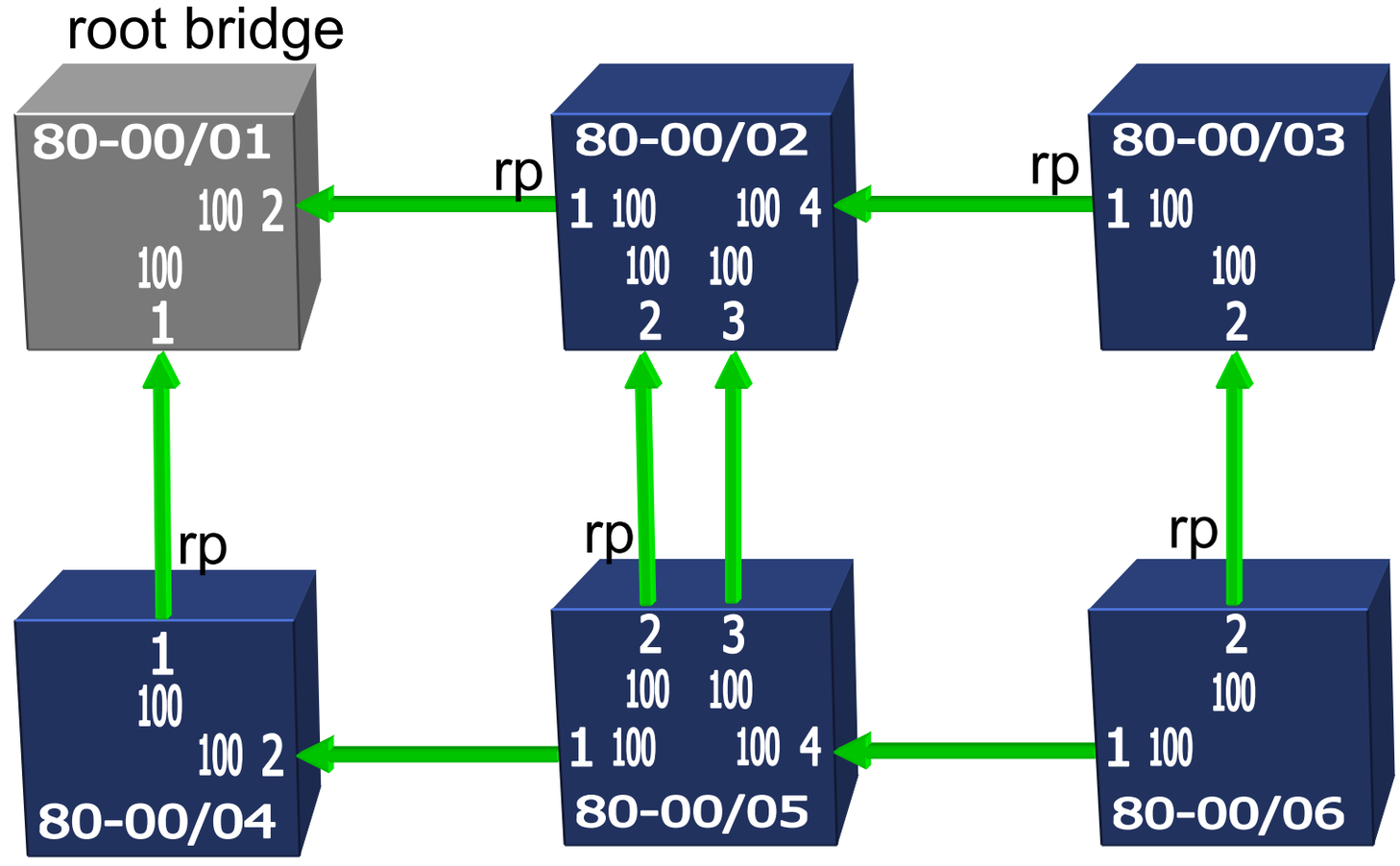
fase 1: elezione del root bridge



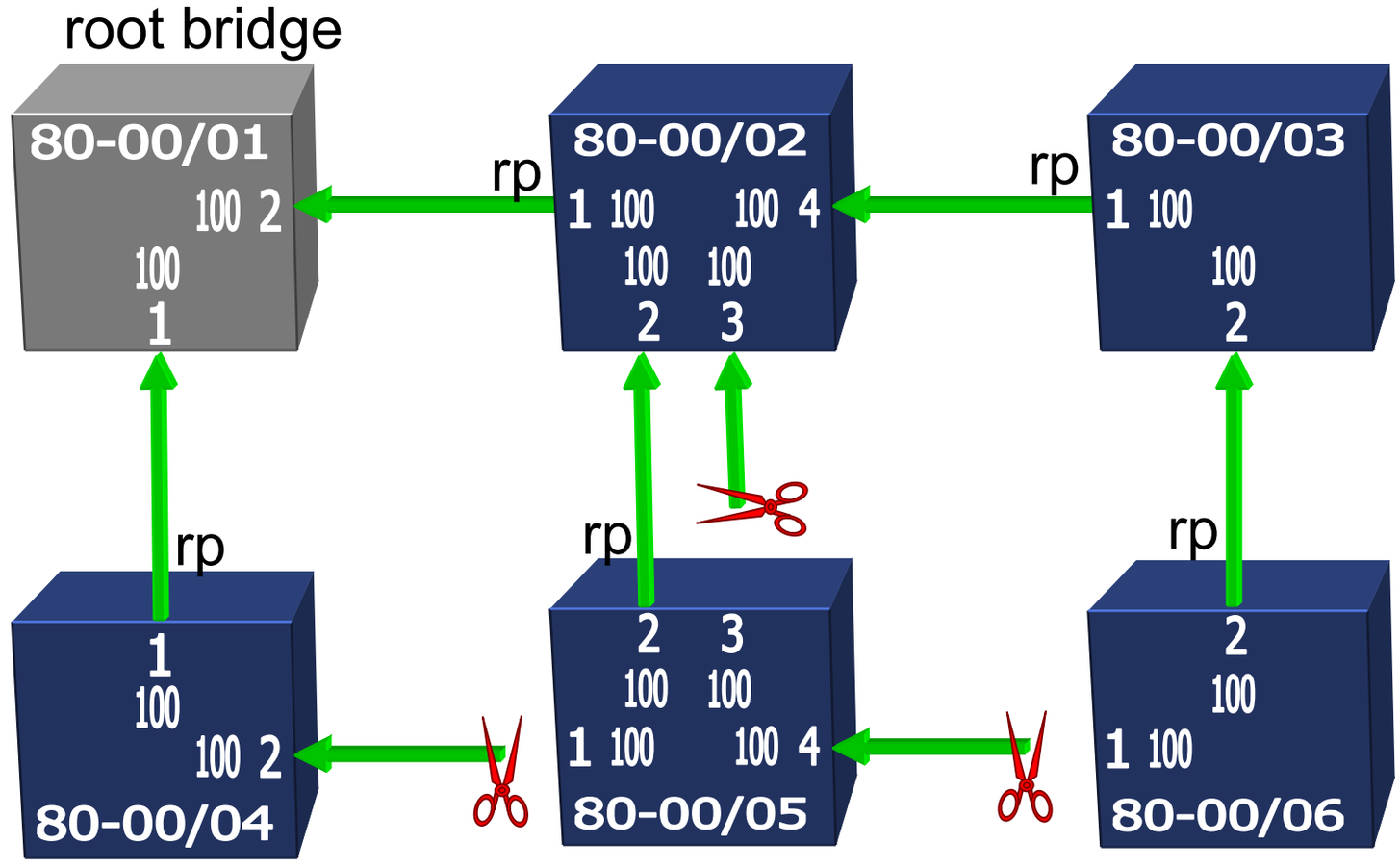
fase 2: identificazione delle root port



fase 3: determinazione delle designated port



fase 4: blocking



esercizio più complesso

- tutti i costi e le priorità delle porte sono posti amministrativamente a zero