Examen Administration Réseaux

M. Heusse, O. Richard, J.-L. Richier

9 avril 2010

Calculatrice et tous documents autorisés

1 SNMP (30mn)

Note : On trouvera en annexe les extraits utiles de la MIB-II ; on a supprimé certaines variables pour simplifier, considérez que seules les variables indiquées existent. Pour les Objet Identifier (OID), donner le nom simple et aussi la forme numérique, complète à partir de la racine.

1. On veut regarder par SNMP si une machine *machine* est un routeur ou non. Quelle variable SNMP consulter ? Donner le nom, l'OID. Indiquer une requête SNMP pour lire cette variable ; quels sont les arguments de cette requête et le résultat ?
2. Donner l'appel SNMP (avec ses arguments) modifiant *machine* pour qu'elle devienne un routeur.

- On considère pour *machine* la table de routes suivante (cas d'une machine d'interface Ethernet d'adresse 210.1.1.10, netmask 255.255.255.0) :

Destination/prefixe	Passerelle	Type	Index interface
195.1.1.0/24	210.1.1.1	Gateway	1
default	210.1.1.3	Gateway	1
210.1.1.0/24	210.1.1.10	Direct	1

3. Quelles sont les variables SNMP associées à la route 195.1.1.0/24 (donner les OID et les valeurs) ?
4. Quel est le nom, et l'OID numérique de la variable donnant la "passerelle" pour la route par défaut ?
5. Donner l'appel SNMP (avec ses arguments) mettant hors service l'interface de la route 195.1.1.0/24

2 Performance (30mn)

2.1 α-Quantile

Donnez un algorithme en pseudo langage qui, pour une série de **100** observations d'une variable aléatoire X et un paramètre α compris entre 0 et 1, donne x_α tel que : $\mathbb{P}(X \leq x_\alpha) = \alpha$. Vous n'utiliserez que des fonctions de contrôles et des structures de données élémentaires (boucles, tests contionnels, variables simple et tableaux).

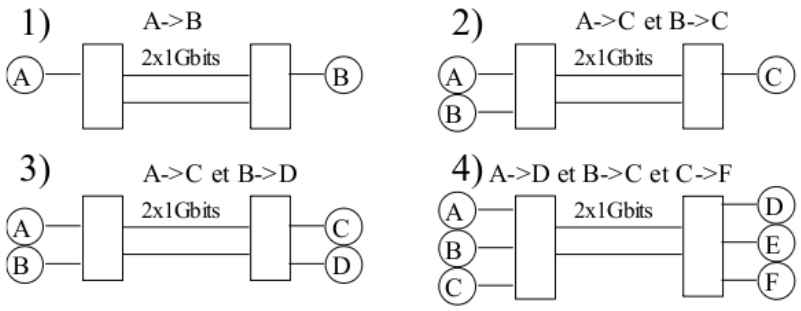

FIGURE 1 – Commutateurs et débits

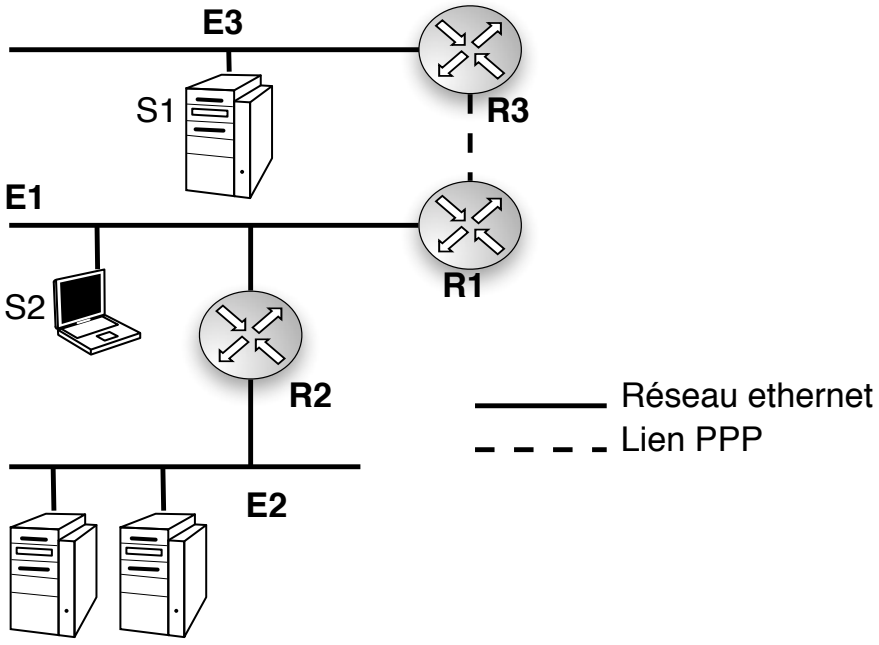

FIGURE 2 – Réseau – OSPF

2.2 Commutateurs et débits

Le dessin figure 1 représente différents réseaux où un ensemble de stations (les cercles) sont interconnectées par des commutateurs (rectangles). Tout les liens sont à 1Gbits/seconde. Les commutateurs sont reliés entre eux par 2 liens. Le débit réellement obtenu entre deux stations directement reliées est de 100Mo/s (on ne s'intéresse pas au protocole réseau utilisé). Dans chacun des 4 cas un certain nombre de transfert de fichiers sont effectués (1 de A $\rightarrow$ B pour le cas 1, 2 de A $\rightarrow$ C et de B $\rightarrow$ C pour le cas 2, 2 de A $\rightarrow$ et B $\rightarrow$ D pour le cas 3 et 3 de A $\rightarrow$ D, B $\rightarrow$ C et C $\rightarrow$ F pour le cas 4). Les transferts sont initiés simultanément. On suppose que les commutateurs exploite au mieux les liens qui sont *full-duplex*. La taille de chaque fichier transféré est de 10 Go. Pour chaque cas quel est la durée totale des transferts dans **le meilleur des cas** ?

3 Administration de réseau (1h)

3.1 DHCP

1. En DHCP, dans quel cas la présence d'un relai est-elle nécessaire ? Cette tâche échoit-elle nécessairement à un routeur (pourquoi) ?

2. Dans quel cas (non exceptionnel) un serveur peut-il ne pas recevoir (ou, en tout cas, ne pas prendre en compte) de DHCPREQUEST à la suite d'un DHCPOFFER ? Quel problème cela peut-il poser dans le cas de l'utilisation d'adresses allouées dynamiquement ? Est-ce problématique dans le cas d'allocations permanentes (non-automatiques) ?
3. Quelle solution envisageriez-vous pour permettre l'utilisation de plusieurs serveurs sur le même LAN *avec des adresses allouées automatiquement* ?

3.2 OSPF

1. On considère le réseau de la figure 2. Proposez un plan d'adressage dans la plage d'adresses 172.16.0.0/16.
2. Quelle est la topologie OSPF correspondante ? (Seuls les routeurs utilisent OSPF. Détailler le contenu de chacun des LSAs)
3. Pour un bon fonctionnement de S2, quel doit y être la table de routage ? En quoi cela sera-t-il plus simple si, par exemple, R2 fait de la translation d'adresses depuis et vers le réseau E2 ?
4. Si un routeur additionnel R4 est placé sur E1 et qu'il est le routeur désigné, quels seront les échanges de paquets qu'on pourra observer sur ce réseau au cas où l'interface vers E2 de R2 est déconnectée ?

RFC 1155 SMI May 1990

```
       RFC1155-SMI DEFINITIONS ::= BEGIN
       EXPORTS -- EVERYTHING
           internet, directory, mgmt, experimental, private, enterprises,
           OBJECT-TYPE, ObjectName, ObjectSyntax, SimpleSyntax, ApplicationSyntax,
           NetworkAddress, IpAddress, Counter, Gauge, TimeTicks, Opaque;
       -- the path to the root
          internet    OBJECT IDENTIFIER ::= { iso(1) org(3) dod(6) 1 }
          directory   OBJECT IDENTIFIER ::= { internet 1 }
          mgmt        OBJECT IDENTIFIER ::= { internet 2 }
          experimental        OBJECT IDENTIFIER ::= { internet 3 }
          private     OBJECT IDENTIFIER ::= { internet 4 }
          enterprises OBJECT IDENTIFIER ::= { private 1 }
       -- names of objects in the MIB
          ObjectName ::= OBJECT IDENTIFIER
       -- syntax of objects in the MIB
          ObjectSyntax ::= CHOICE {
                simple           SimpleSyntax,      application-wide ApplicationSyntax
          }
          SimpleSyntax ::= CHOICE {
                number        INTEGER,              string        OCTET STRING,
                object        OBJECT IDENTIFIER,    empty         NULL
          }
          ApplicationSyntax ::= CHOICE {
                address       NetworkAddress,       counter       Counter,
                gauge         Gauge,                ticks         TimeTicks,
                arbitrary     Opaque
          -- other application-wide types, as they are defined, will be added here
          }
       -- application-wide types
          NetworkAddress ::= CHOICE {  internet IpAddress  }
          IpAddress ::= -- in network-byte order
                    [APPLICATION 0] IMPLICIT OCTET STRING (SIZE (4))
          Counter ::=   [APPLICATION 1] IMPLICIT INTEGER (0..4294967295)
          Gauge ::=     [APPLICATION 2] IMPLICIT INTEGER (0..4294967295)
          TimeTicks ::= [APPLICATION 3] IMPLICIT INTEGER (0..4294967295)
          Opaque ::=    [APPLICATION 4]         -- arbitrary ASN.1 value,
                          IMPLICIT OCTET STRING   --   "double-wrapped"
```

RFC 1213 MIB-II March 1991

```
  RFC1213-MIB DEFINITIONS ::= BEGIN
      IMPORTS
          mgmt, NetworkAddress, IpAddress, Counter, Gauge, TimeTicks FROM RFC1155-SMI
          OBJECT-TYPE FROM RFC-1212;
  --  MIB-II (same prefix as MIB-I)
      mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }
  -- textual conventions
      DisplayString ::= OCTET STRING
  -- This data type is used to model textual information taken from the NVT ASCII character set.
  -- By convention, objects with this syntax are declared as having SIZE (0..255)

  -- groups in MIB-II
      system        OBJECT IDENTIFIER ::= { mib-2 1 }
      interfaces    OBJECT IDENTIFIER ::= { mib-2 2 }
      ip            OBJECT IDENTIFIER ::= { mib-2 4 }
.................
```

```
  -- the Interfaces group
      ifNumber OBJECT-TYPE
          SYNTAX  INTEGER
          ACCESS  read-only
          STATUS  mandatory
          DESCRIPTION   "The number of network interfaces (regardless of
                 their current state) present on this system."
          ::= { interfaces 1 }
  -- the Interfaces table
  -- The Interfaces table contains information on the entity's interfaces.
      ifTable OBJECT-TYPE
          SYNTAX  SEQUENCE OF IfEntry
          ACCESS  not-accessible
          STATUS  mandatory
          DESCRIPTION   "A list of interface entries."
          ::= { interfaces 2 }
      ifEntry OBJECT-TYPE
          SYNTAX  IfEntry
          ACCESS  not-accessible
          STATUS  mandatory
          DESCRIPTION   "An interface entry containing objects at the
                 subnetwork layer and below for a particular interface."
          INDEX   { ifIndex }
          ::= { ifTable 1 }
      IfEntry ::= SEQUENCE {
-- NOTE: plusieurs champs ont été supprimés pour simplifier le texte
              ifIndex INTEGER,
              ifDescr DisplayString,
              ifAdminStatus INTEGER,
              ifOperStatus INTEGER,
      }
      ifIndex OBJECT-TYPE
          SYNTAX  INTEGER
          ACCESS  read-only
          STATUS  mandatory
          DESCRIPTION   "A unique value for each interface.  Its value ranges between
                 1 and the value of ifNumber. The value for each interface must remain
                 constant at least from one re-initialization of the entity's network
                 management system to the next re-initialization."
          ::= { ifEntry 1 }
      ifDescr OBJECT-TYPE
          SYNTAX  DisplayString (SIZE (0..255))
          ACCESS  read-only
          STATUS  mandatory
          DESCRIPTION   "A textual string containing information about the interface.
                 This string should include the name of the manufacturer, the product
                 name and the version of the hardware interface."
          ::= { ifEntry 2 }
      ifAdminStatus OBJECT-TYPE
          SYNTAX  INTEGER {
                      up(1),       -- ready to pass packets
                      down(2),
                      testing(3)   -- in some test mode
                  }
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION   "The desired state of the interface. The testing(3) state
                 indicates that no operational packets can be passed."
          ::= { ifEntry 7 }
```

```
      ifOperStatus OBJECT-TYPE
          SYNTAX  INTEGER {
                      up(1),       -- ready to pass packets
                      down(2),
                      testing(3)   -- in some test mode
                  }
          ACCESS  read-only
          STATUS  mandatory
          DESCRIPTION   "The current operational state of the interface. The testing(3)
         state indicates that no operational packets can be passed."
          ::= { ifEntry 8 }
.......
  -- the IP group
  -- Implementation of the IP group is mandatory for all systems.
      ipForwarding OBJECT-TYPE
          SYNTAX  INTEGER {
                  forwarding(1),        -- acting as a gateway
                  not-forwarding(2)     -- NOT acting as a gateway
          }
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION   "The indication of whether this entity is acting as an IP gateway
                 in respect to the forwarding of datagrams received by, but not addressed
                 to, this entity. IP gateways forward datagrams. IP hosts do not (except
                 those source-routed via the host)."
          ::= { ip 1 }
      ipDefaultTTL OBJECT-TYPE
          SYNTAX  INTEGER
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION   "The default value inserted into the Time-To-Live field
                 of the IP header of datagrams by the transport layer protocol."
          ::= { ip 2 }
      ipInReceives OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          DESCRIPTION   "The total number of input datagrams received
                 from interfaces, including those received in error."
          ::= { ip 3 }
.......................
  -- the IP routing table
  -- The IP routing table contains an entry for each route presently known to this entity.
-- NOTE: plusieurs champs ont été supprimés pour simplifier le texte
      ipRouteTable OBJECT-TYPE
          SYNTAX  SEQUENCE OF IpRouteEntry
          ACCESS  not-accessible
          STATUS  mandatory
          DESCRIPTION   "This entity's IP Routing table."
          ::= { ip 21 }
      ipRouteEntry OBJECT-TYPE
          SYNTAX  IpRouteEntry
          ACCESS  not-accessible
          STATUS  mandatory
          DESCRIPTION   "A route to a particular destination."
          INDEX   { ipRouteDest }
          ::= { ipRouteTable 1 }
```

```
      IpRouteEntry ::= SEQUENCE {
             ipRouteDest           IpAddress,
             ipRouteIfIndex        INTEGER,
             ipRouteNextHop        IpAddress,
             ipRouteType           INTEGER,
             ipRouteMask           IpAddress,
      }
      ipRouteDest OBJECT-TYPE
          SYNTAX  IpAddress
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION   "The destination IP address of this route.
                 An entry with a value of 0.0.0.0 is considered a default route."
          ::= { ipRouteEntry 1 }
      ipRouteIfIndex OBJECT-TYPE
          SYNTAX  INTEGER
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION    The index value which uniquely identifies the local interface
                 through which the next hop of this route should be reached.
                 The interface identified by a particular value of this index is the
                 one identified by the same value of ifIndex."
          ::= { ipRouteEntry 2 }
      ipRouteNextHop OBJECT-TYPE
          SYNTAX  IpAddress
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION   "The IP address of the next hop of this route.  (In the case of
                 a route bound to an interface which is realized via a broadcast media,
                 the value of this field is the agent's IP address on that interface.)"
                  ::= { ipRouteEntry 7 }
      ipRouteType OBJECT-TYPE
          SYNTAX  INTEGER {
                  other(1),     -- none of the following
                  invalid(2),   -- an invalidated route
                  direct(3),    -- route to directly connected (sub-)network
                  indirect(4)   -- route to a non-local host/network/sub-network
          }
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION  "The type of route.  Note that the values direct(3) and indirect(4)
                 refer to the notion of direct and indirect routing in the IP architecture."
          ::= { ipRouteEntry 8 }
      ipRouteMask OBJECT-TYPE
          SYNTAX  IpAddress
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION   "Indicate the mask to be logical-ANDed with the destination
                 address before being compared to the value in the ipRouteDest field.
                 If the value of the ipRouteDest is 0.0.0.0 (a default route), then
                 the mask value is also 0.0.0.0. It should be noted that all IP routing
                 subsystems implicitly use this mechanism."
          ::= { ipRouteEntry 11 }
```