DELIVRABLE DU PROJET VTHD++

DEPLOIEMENT ET TESTS DU MULTICAST FIABLE ACTIF SUR VTHD++

Auteurs :

Faycal Bouhafs

Ingénieur INRIA

Congduc Pham

Maître de conférences, HDR

RESO/LIP/UCB Lyon

Introduction

Presentation du multicast

Principes de base

Un service de communication point à multipoint offre un moyen efficace de diffuser des unités de données à un groupe de récepteurs, en ce sens qu'une seule copie de chacune de ces unités est envoyée par la source (s'il n'y pas de pertes). Le multicast IP (RFC 1112) fournit au niveau réseau un support efficace pour la diffusion non fiabilisé des paquets pour un grand nombre d'applications: visio-conférence, ftp multidestinataire, mise à jour d'informations dupliquées réparties, applications coopératives et tableau blanc, simulations distribuées,…Certaines de ces applications prévoient la mise en rapport d'un nombre de participants de l'ordre de plusieurs milliers et peuvent aussi, en plus de l'efficacité du routage, nécessiter une grande fiabilité dans la délivrance des données.

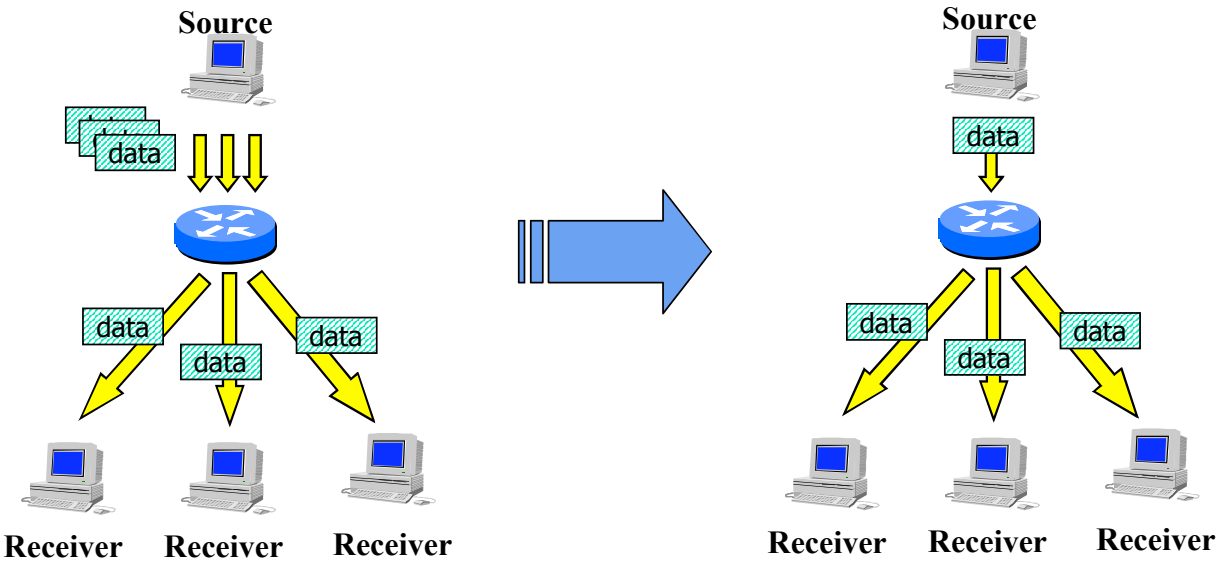


figure 1.

Le problème de la fiabilité en point à point est bien maîtrisé et des solutions satisfaisantes (du moins pour TCP) ont été déployées. Par contre l'assurance de la fiabilité dans le contexte du multicast est un problème plus ardu et les solutions sont moins évidentes, surtout sur des réseaux étendus et hétérogènes. La conception de protocoles de diffusion fiable efficaces est plus difficile et doit tenir compte des contraintes imposées par la résistance au facteur d'échelle. L'implosion et la surcharge de la source par les messages de contrôle (ACK/NAK), l'exposition des récepteurs à des transmissions dupliquées ou le contrôle de congestion en sont des exemples (voir figure). Malgré plusieurs années d'efforts consacrées à la conception de protocoles de diffusion fiable au dessus d'IP multicast, il est toujours aussi difficile de fournir une solution de bout-en-bout capable de satisfaire à l'ensemble des contraintes liées au facteur d'échelle.

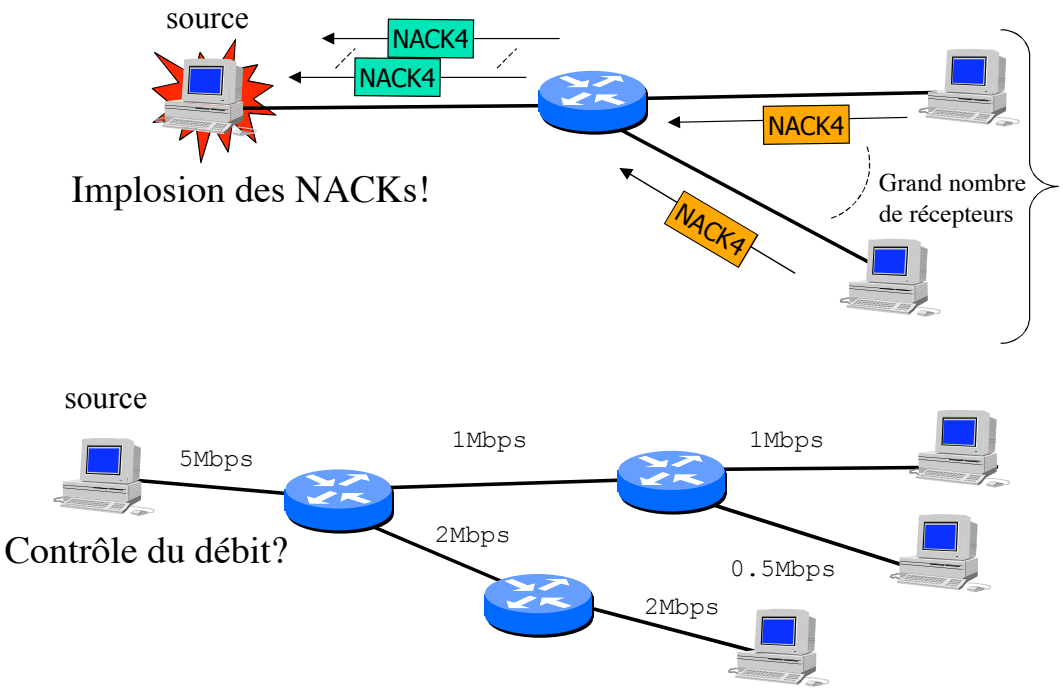


figure 2.

La plupart des propositions récentes pour assurer la fiabilité utilisent du recouvrement local pour considérablement diminuer le temps de recouvrement. Par exemple, un récepteur peut recevoir un paquet retransmis d'un autre récepteur dans le voisinage (SRM [SRM]), d'un récepteur désigné (RMTP [RMTP], TMTP [TMTP], LMS [LMS], PGM [PGM]) ou d'un serveur dédié (LBRM [LBRM]). Parmi ceux-ci, PGM et LMS utilisent les routeurs du réseau pour mettre en oeuvre une structure hiérarchique. LMS exploite quelques informations sur la topologie disponible au niveau des routeurs. Avec des fonctions d'aiguillage spécifiques au niveau des routeurs, le protocole fait l'élection d'un récepteur comme *leader* pour chaque sous arbre pour satisfaire les requêtes de ses fils. PGM de son côté utilise une assistance de routeurs PGM pour choisir un récepteur (*Designed Local Retransmitter,* DLR) qui doit cependant être sur le chemin vers la source. Des approches alternatives basées sur des codages FEC évitent, ou diminuent considérablement, les retransmissions à partir de la source ou des entités réparatrices (Asynchronous Layered Coding, RFC 3450-3453, en est un des représentants). Un des problèmes de ces approches est néanmoins la production des paquets FEC qui nécessitent souvent de garder (à un instant donné) en mémoire l'intégralité des données (tout le fichier de 10Go par exemple). Ces techniques FEC peuvent aussi être utilisées de manière conjointe à une approche conventionnelle (avec feedback). Dans ce cas, il est possible de voir le fichier comme plusieurs morceaux plus petits, et les ACK permettent de savoir si un morceau a été correctement reçu.

Une autre propriété souhaitable, et non des moindres, est la co-habitation des flux multicast avec les autres flux unicast gérés essentiellement par TCP. C'est ce que l'on nomme communément le contrôle de congestion. Dans les versions actuellement déployées de TCP par exemple ce contrôle est une combinaison de la phase de *slow-start* avec une phase de *congestion avoidance*. Ce mécanisme permet aux entités sources des connexions TCP de partager relativement équitablement la bande passante

disponible. Dans le cas du multicast, il est ardu de concevoir des mécanismes de contrôle de congestion qui offrent à la fois une bonne utilisation du réseau pour tous les récepteurs, et une compatibilité avec les flux TCP. Lorsque les données peuvent être décomposées en plusieurs couches (comme c'est assez facilement le cas pour des données vidéo), des approches utilisant des souscriptions à plusieurs groupes, de manière cumulative, sont possibles.

De manière générale, la fiabilité et le contrôle de congestion en utilisant l'Internet tel qu'il est actuellement, c'est-à-dire avec un service IP non intelligent, est très difficile (manque d'information sur la topologie, sur le nombre de récepteurs, sur leur capacité…). Certaines approches qui ont été proposées (comme PGM [PGM] ou LMS [LMS] par exemple) ajoutent de nouvelles fonctionnalités dans les routeurs (et nécessitent donc des routeurs spécifiques). Cette démarche montre la nécessité de pouvoir adapter et ajouter dynamiquement des fonctionnalités de haut-niveau dans l'Internet pour offrir un support performant pour les communications de groupe.

Le multicast actifs (avec assistance des routeurs)

Les réseaux actifs

Voir la documentation sur les réseaux actifs

Le multicast actif

Le multicast actif ou à assistance des routeurs utilisent les routeurs dans l'infrastructure réseau pour effectuer des traitements sur les flux (que nous appelerons service actif). Par exemple, des services d'aggrégation des messages de contrôle peuvent être mis en place pour éviter l'implosion des messages de contrôle au niveau de la source. La figure suivante illustre ce cas.

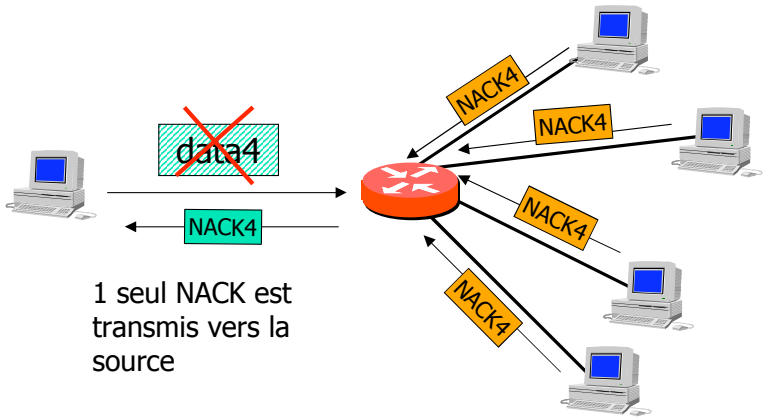


Figure 3.

L'implémentation d'un tel service peut se faire très simplement, et le plus important est de maintenir le coût d'un service actif à un niveau très bas. La figure suivante montre schématiquement comment un tel service peut-être implémenté avec des structures de données simples.

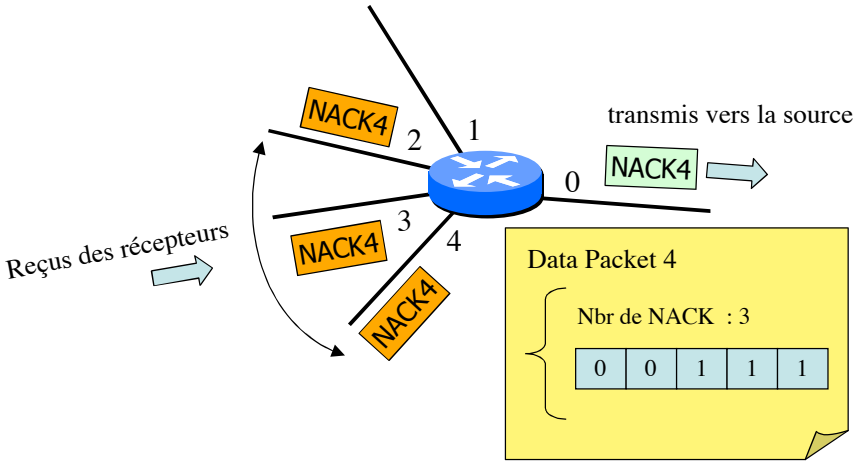


Figure 4.

Les services actifs que nous avons introduits sont :

- l'élection dynamique d'un retransmetteur: un routeur actif d'assistance peut élire un récepteur pour retransmettre un paquet perdu par un de ces voisins.
- la détection rapide des pertes dans les routeurs: un routeur actif d'assistance peut générer un NAK vers la source s'il détecte une perte de séquence dans le flux des paquets.
- l'aggrégation des RTTs: les routeurs actifs d'assistance participent à l'aggrégation des RTTs par segment afin de générer une valeur de RTT plus précise permettant une régulation du débit plus fluide par la source.
- le partionnement des récepteurs en sous-groupes pour améliorer la gestion des groupes hétérogènes.

L'élection dynamique d'un retransmetteur est le principal apport de la solution proposée. La figure suivante illustre comment ce mécanisme peut permettre à un récepteur voisin de retransmettre un paquet perdu. Les récepteurs ayant perdus des paquets envoient des NAK vers leur routeur père qui va aggréger les NAK et garder trace de l'adresse des récepteurs ayant envoyer un NAK. Ce routeur va ensuite déterminer un lien menant vers un récepteur ayant correctement reçu le paquet.

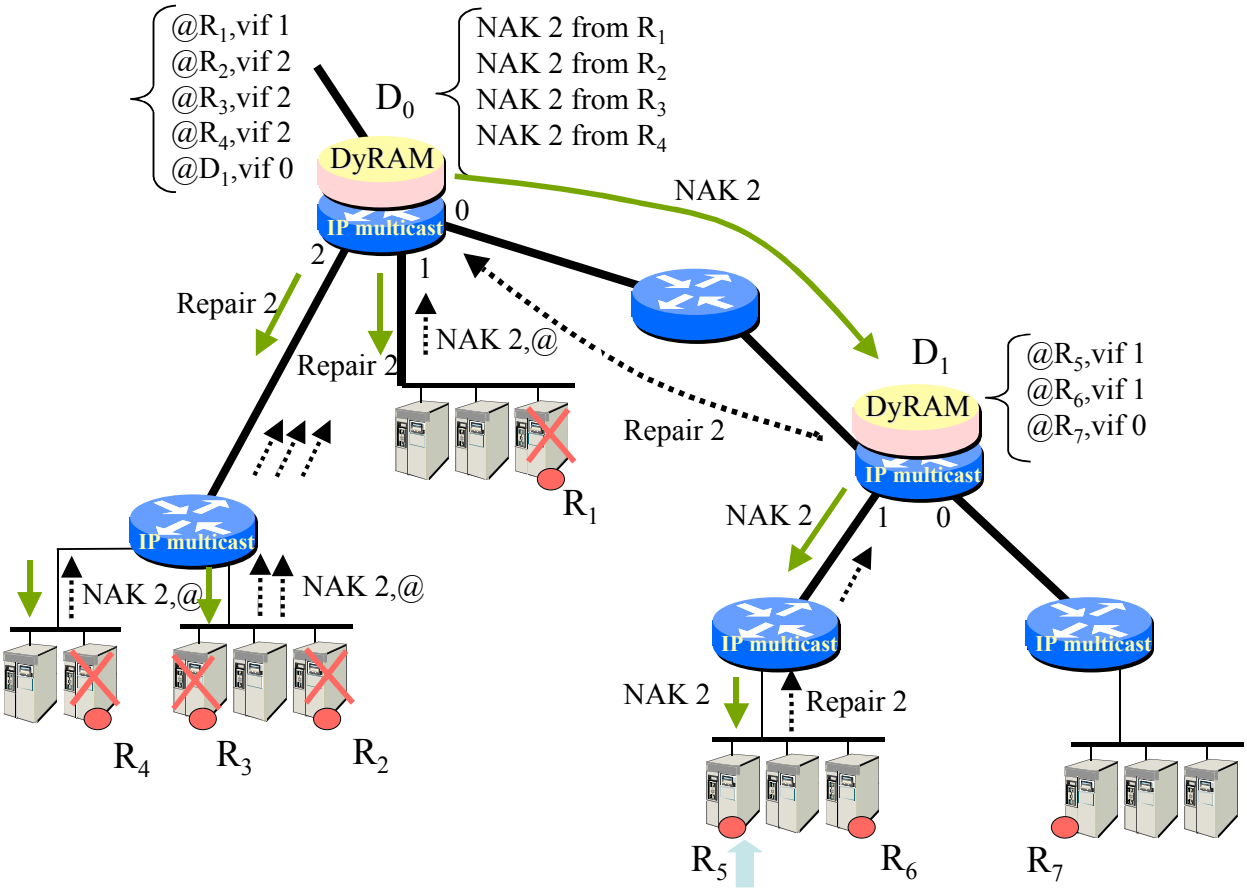


Figure 5.

Ce récepteur sera ensuite élu par son routeur père pour retransmettre le paquet perdu. Dans l'exemple, c'est R5 qui sera élu.

Scénario de déploiement sur une grille de calcul en utilisant VTHD

Une telle solution active peut être déployée une infrastructure de grille comme illustrée par la figure suivante où les routeurs d'assistance peuvent exécuter des fonctions différentes en fonction de l'endroit où ils se trouvent.

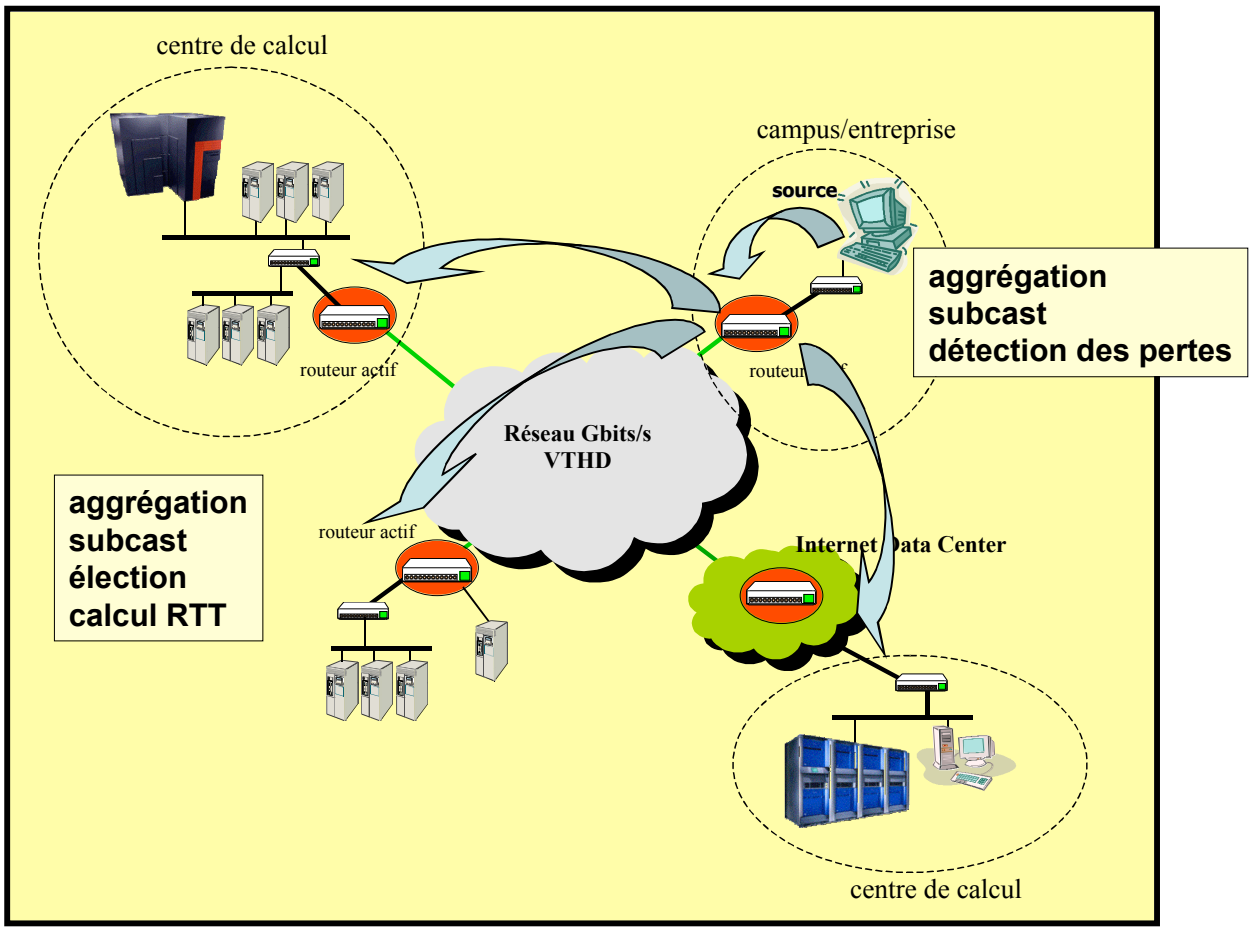

centre de calcul
campus/entreprise
source
aggrégation
subcast
détection des pertes
routeur actif
Réseau Gbits/s
VTHD
aggrégation
subcast
élection
calcul RTT
routeur actif
Internet
centre de calcul

Figure 6.

Le protocole DyRAM et MFTP

Nous avons intégré dans un protocole que nous avons appelé DyRAM, pour *Dynamic Replier Active reliable Multicast*, les nouveaux services que nous avons proposé, ainsi que les services d'aggrégation globale et de subcast. L'un des objectifs majeurs de DyRAM est de se passer du cache dans les routeurs et d'obtenir une latence de recouvrement faible. En ce sens, il est bien différent de ARM [ARM] ou AER [AER].

Il faut néanmoins préciser que le cas du multicast fiable que nous étudions est de pouvoir supporter dans une session multicast de plusieurs centaines à quelques milliers de récepteurs (ce qui est déjà un défi pour le multicast fiable). Les applications types sont par exemple la mise à jour de programmes, l'échanges de bases de données et de fichiers de grande taille, la soumission de travaux sur une grille de calcul,… Contrairement au multicast non fiable plutôt utilisé pour la diffusion temps-réel de contenus multimédia, nous n'envisageons pas d'avoir des groupes de plusieurs millions de récepteurs pour l'envoi fiable de fichiers. Les mécanismes qui sont donc proposés dans DyRAM ne sont donc pas fait pour ce scénario d'utilisation.

Presentation de MFTP

MFTP est une API en JAVA permettant de faire du transfert de fichiers en mode multicast en se basant sur les spécifications du protocole Dyram. L'environnement TAMANOIR [TAN] est utilisé pour le déploiement des services actifs propres à DyRAM.
MFTP a été testé sous Linux. Il n'y a pas eu de tests sous Windows.
`m_ftp` est un petit programme utilisant la librairie MFTP qui sert a envoyer des fichiers.

Pré-requis pour MFTP

Voici les différents pré-requis pour une bonne installation de MFTP et une bonne exécution de `m_ftp`

Les pré-requis pour l'émission/reception avec MFTP:

Noyau 2.4.18 ou supérieur;
Support du multicast;
JDK1.4;

Configuration de l'émetteur/récepteur MFTP:

Rajouter le chemin de `m_ftp` et de `Tamanoir` dans la variable `CLASSPATH`.
Rajouter le chemin de la librairie active dans la variable `LD_LIBRARY_PATH`.

Pré-requis pour le noeud actif:

Noyau 2.4.18 ou supérieur;
Support de `netfilter`;

Support du multicast;
Support du routage multicast et du tunneling;
Configurer la machine comme routeur;
Avoir un daemon de routage multicast installé sur la machine.
JDK1.4.

Configuration du routeur Actif

Rajouter le chemin de `m_ftp` et de `Tamanoir` dans la variable `CLASSPATH`.
Rajouter le chemin de librairie Active dans la variable `LD_LIBRARY_PATH`.

Construction d'un objet MFTP

Pour créer un objet MFTP il suffit d'appeler la méthode suivante :

```
Public M_FTP ()
```

L'émission d'un fichier avec MFTP

L'émission d'un fichier sur une adresse multicast avec MFTP se fait avec la méthode `mSend` :

```
public void mSend (String  @IP_Multicast, String Nom_Fichier)
Exception : UnknownHostException, FileNotFoundException,
IOException
```

En réalité cette méthode permet juste d'envoyer des message de demande d'initialisation vers les récepteurs potentiels sur la session multicast, l'émission ne commencera que lorsque au moins un récepteur répondra à cette demande.

La partie Récepteur

La réception d'un fichier sur une adresse multicast avec MFTP se fait avec la methode `mReceive` :

```
public void mReceive (String  @IP_Multicast, String
Nom_Fichier)
```

Exception : `UnknownHostException,FileNotFoundException,IOException, BadANEPversionException, SocketException.`

Cette méthode permet au récepteur de rester en attente d'un message de demande d'initialisation de l'émetteur, de répondre à cette requête, et de commencer à recevoir le fichier. Cette méthode reste dans un état bloquant jusqu'à la réception de l'intégralité du fichier.

Les services actifs

Comme il a été décrit précédemment, l'environnement TAMANOIR est utilisé pour gérer les services actifs multicast (aggrégation des NAKs, gestion des paquets CR...), et il doit être invoqué par l'administrateur du routeur actif.

Interception des paquets actifs

Un élément important dans le système MFTP est l'interception des paquets échangés entre l'émetteur et le(s) récepteur(s). Ces paquets sont appelés Paquets Actifs et le module « `mitm` » permet d'intercepter tous ces paquets lors de leur passage par le routeur et de les renvoyer à la couche supérieure c'est à dire au service actif. La figure suivante illustre ce comportement ou Tamanoir est chargé avec 2 services actifs.

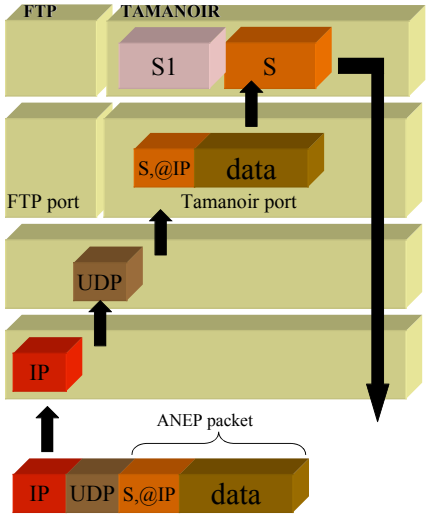


figure 7.

Pour charger ce module en mémoire il faut taper la commande suivante :

```
insmod mitm.o local= "@IP de la machine local"
```

Lancement des services actifs

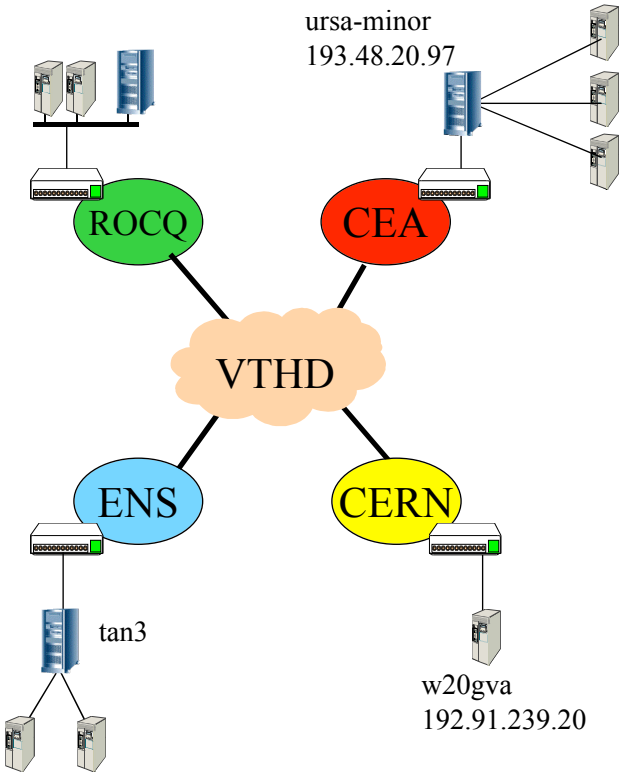
Pour lancer l'environnement actif Tamanoir il suffit de taper la ligne de commande suivante :

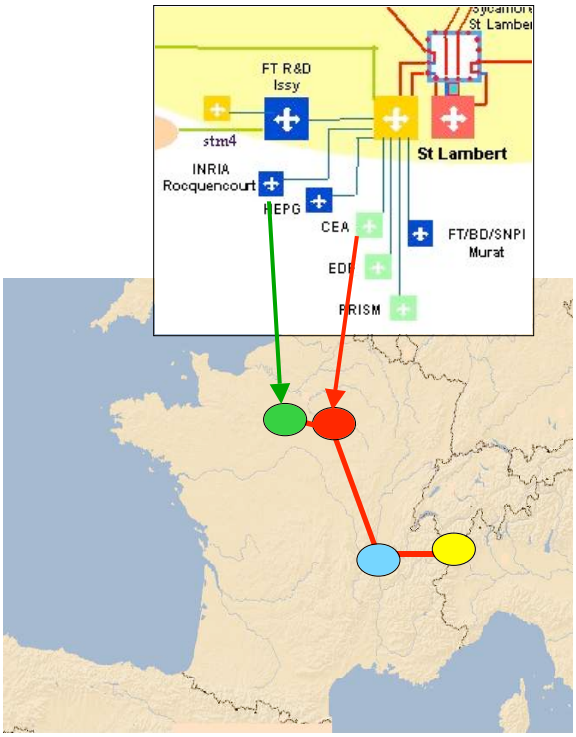
```
java TAMANOIRd
```

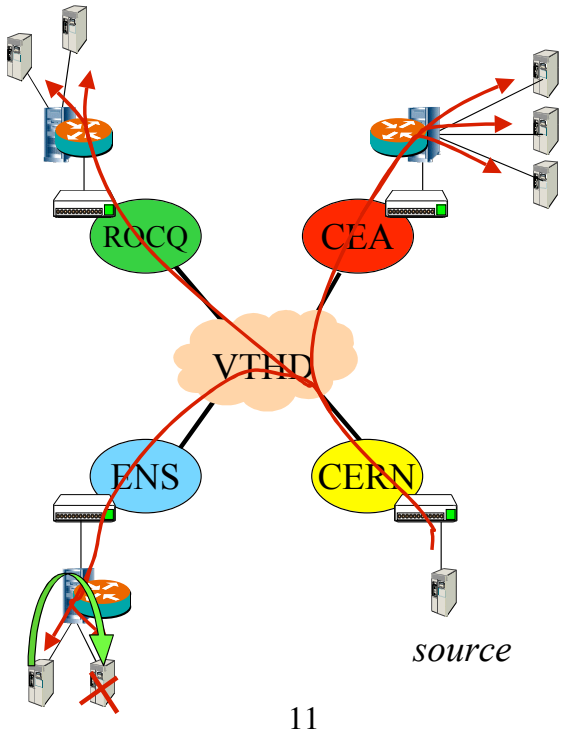
L'environnement TAMANOIR va se charger de lancer le service actif multicast dès la détection du premier message d'initialisation.

Déploiement dans le cadre du projet e-Toile sur VTHD

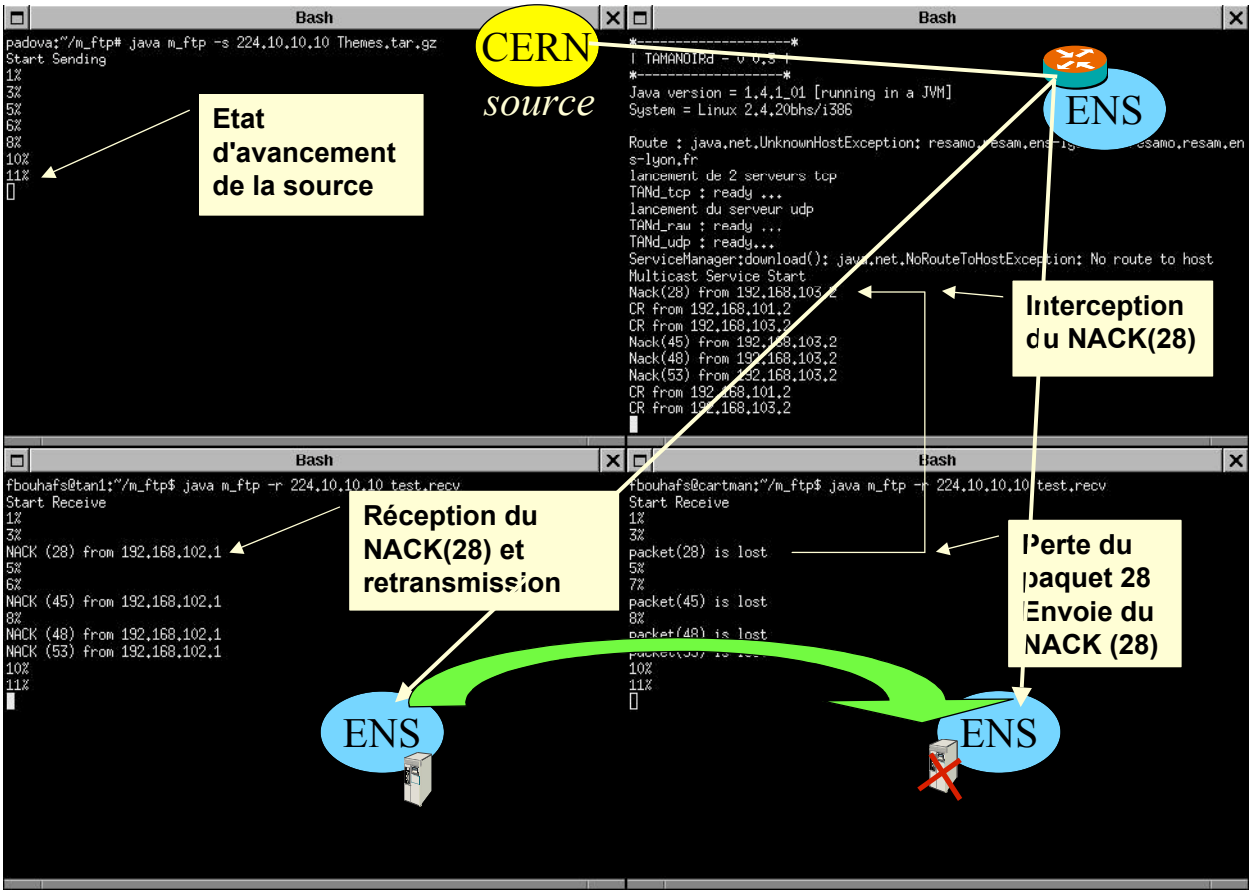
La figure suivante montre la topologie de test effectué le 5 juin 2003 pour la démonstration e-Toile. Le test consiste à valider la récupération des erreurs localement sur le site de l'ENS.







source



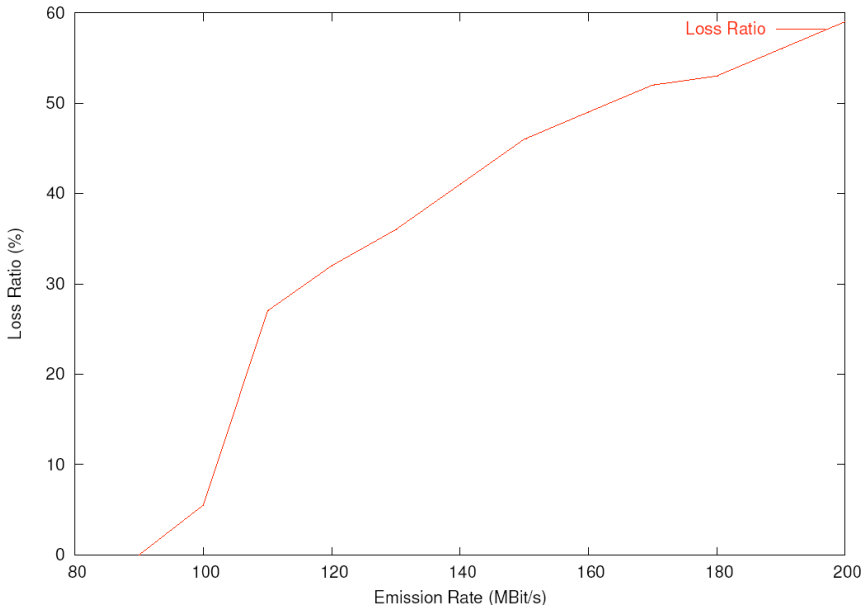
Problème de déploiement sur le site du PRISM

Test du mtrace vers une machine du PRISM

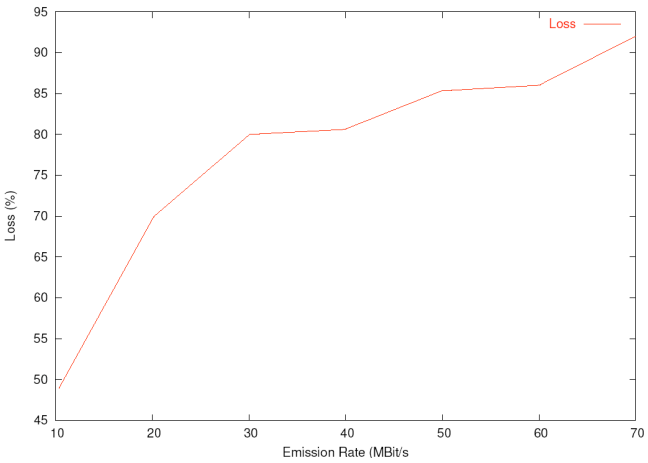
```
./mtrace 193.253.175.170
mtrace: WARNING: no multicast group specified, so no statistics printed
Mtrace from 193.253.175.170 to 193.51.24.93 via group 0.0.0.0
Querying full reverse path... * switching to hop-by-hop:
  0  ? (193.51.24.93)
 -1  r-vthd.reseau.uvsq.fr (193.51.24.94)  PIM  thresh^ 0   [default]
 -2  ? (193.252.226.221)  PIM  thresh^ 1
 -3  * * ? (193.252.113.22)  PIM  thresh^ 1
 -4  * * ? (193.252.113.26)  PIM  thresh^ 0
 -5  ? (193.253.175.170)
Round trip time 13 ms; total ttl of 4 required
```

Etude des performances sur VTHD

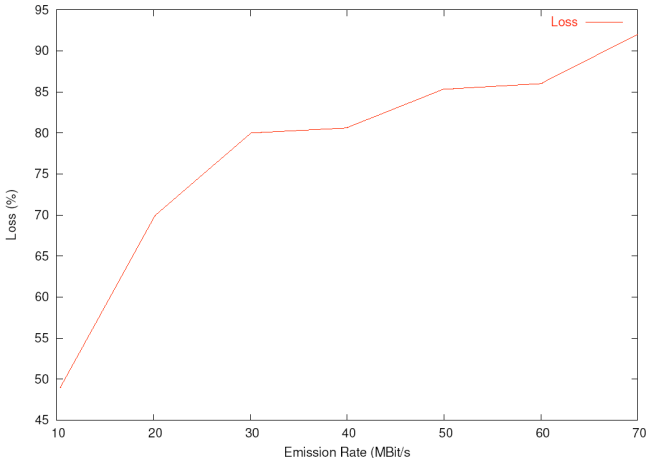
Taux de pertes en fonction du débit d'émission mesurés avec iperf sur VTHD (UDP multicast, 8000 octets/paquets). Mesures effectuées entre le PRISM et l'IMAG



Il n'y a pas de pertes constatées en UDP unicast, c'est un phénomène que nous n'expliquons pas encore. Il semble que le taux de pertes augmente considérablement lorsque l'on dépasse le seuil des 100Mbits/s.



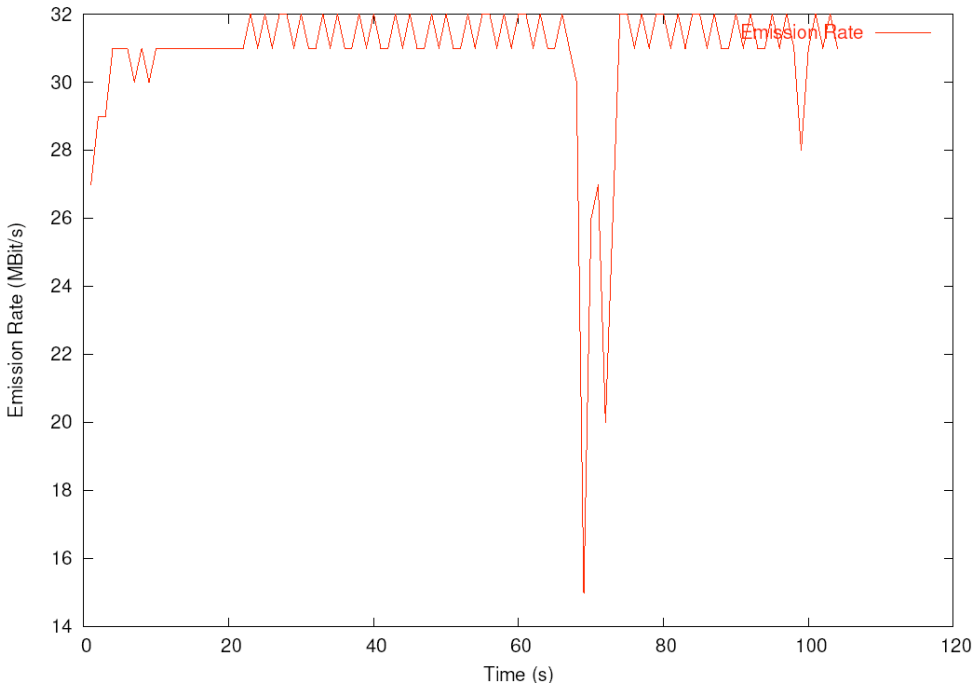
6000 octets/paquets



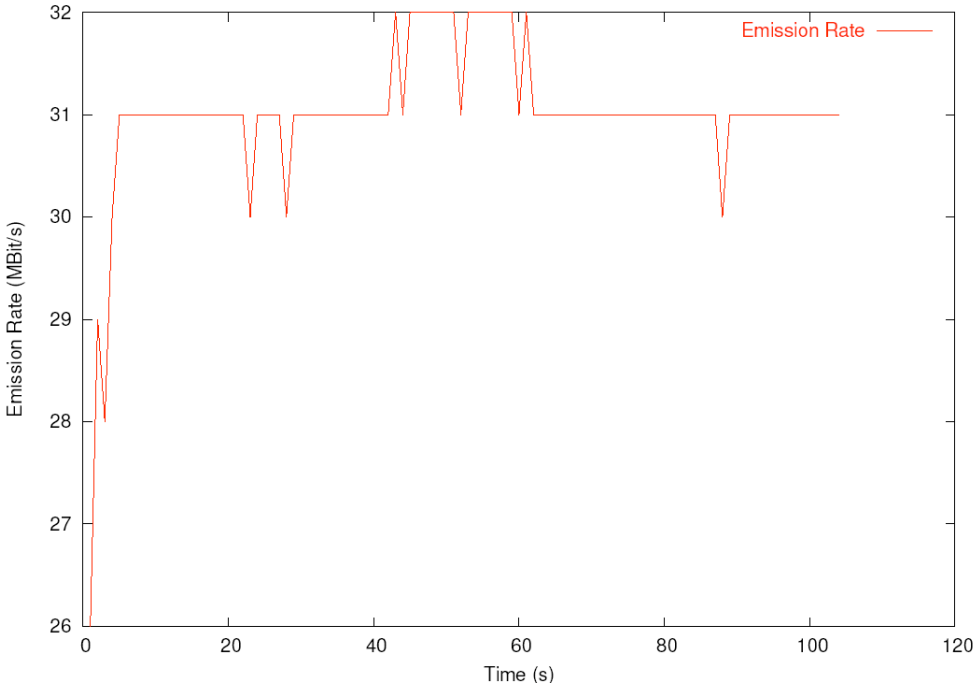
1000 octets/paquets

Les figures ci-dessous montre le taux de pertes avec MFTP sans encapsulation ANEP.

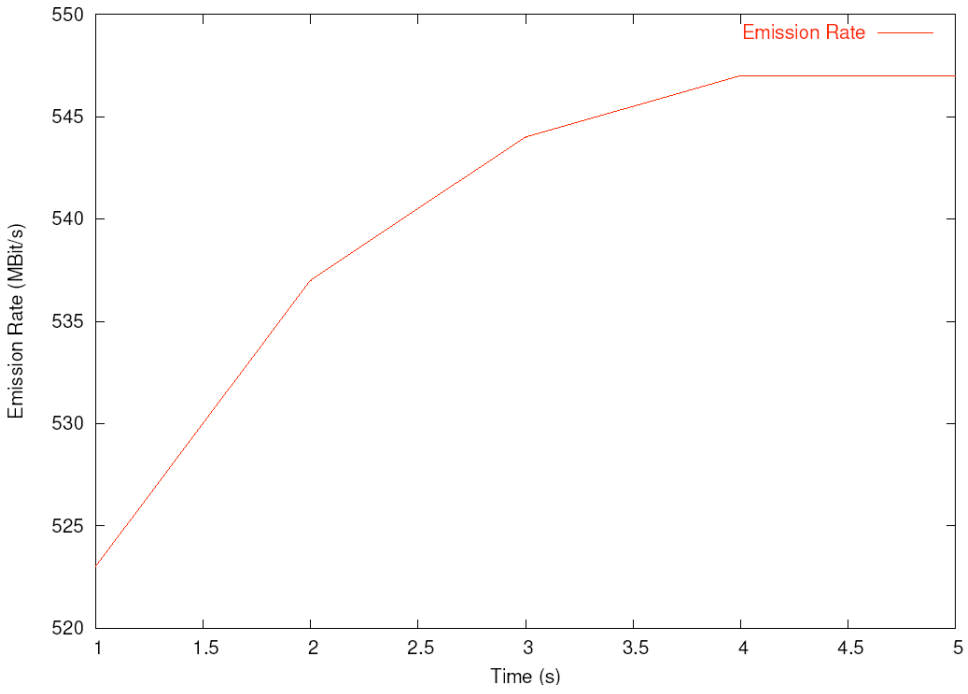
Débit mesuré à l'émission avec MFTP, sans contrôle de flux, sans retransmissions (fichier de 400Mo). Encapsulation DyRAM/ANEP/IP



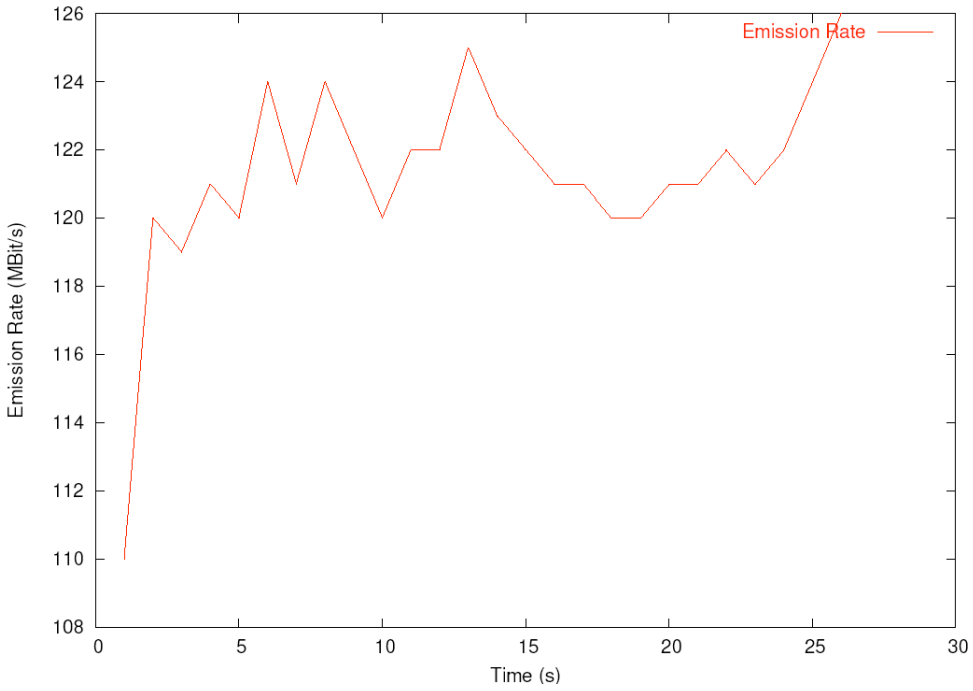
Débit mesuré à l'émission avec un FTP écrit en java pour comparer avec MFTP (fichier de 400Mo). Encapsulation DyRAM/ANEP/IP



Débit mesuré à l'émission avec MFTP sans encapsulation DYRAM/ANEP (fichier de 400Mo). Sans contrôle de flux, ni retransmission



Débit mesuré à l'émission avec MFTP sans encapsulation ANEP (fichier de 400Mo). Sans contrôle de flux, ni retransmission



ANNEXE A : Exemple de simple programme

```
import java.lang.*;
import java.net.*;
import java.io.*;
import java.util.*;


public class M_FTP   {

public static void main (String args[]) throws
UnknownHostException,FileNotFoundException, IOException,
BadANEPversionException
{
     String address, filename;
      m_ftp s = new m_ftp ();

      if (args[0].charAt(0)!= '-')
     {
         System.out.println("Erreur de parametrage \n");
         System.exit(-1);
     }
     else
      {
         char command = args[0].charAt(1);
         address = args[1];
         filename = args[2];

         switch (command){

         case 's':
         {  s.mSend (address,filename);
             break;}
           case 'r':
         {  s.mReceive (address,filename);
             break;}
         default :

         System.out.println("Erreur de parametrage \n");
         System.exit(-1);
         }
     }
}}
```

ANNEXE B : références

[AER] S. Kasera and S. Bhattacharya. Scalabe fair reliable multicast using active services. IEEE Network Magazine's Special Issue on Multicast, 2000.

[ARM] L. Lehman, S. Garland, and D. Tennehouse. Active reliable multicast. Proc. of the IEEE INFOCOM, San Francisco, CA, March 1998.

[LBRM] Hugh W. Holbrook, Sandeep K. Singhal, and David R. Cheriton. Log-based receiver-reliable multicast for distributed interactive simulation. SIGCOMM'95, Oct. 1995.

[LMS] Christos Papadopoulos, Guru~M. Parulkar, and George Varghese. An error control scheme for large-scale multicast applications. Proc. of the IEEE INFOCOM}, March 1998.

[PGM] Jim Gemmell and al. The pgm reliable multicast protocol. IEEE Networks, special issue on Multicasting: An Enabling Technology, January 2003.

[RMTP] S. Paul and K. Sabnani. Reliable multicast transport protocol (RMTP). IEEE JSAC, Spec. Issue on Network Support for Multipoint Communications, 15(3), April 1997.

[SRM] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. IEEE ACM Transactions on Networking, 5(6), 1997.

[TAN] J.P. Gelas and L. Lefèvre. TAMANOIR : A High Performance Active Network Framework. Workshop on Active Middleware Services 2000, 9th IEEE International HPDC, Pittsburgh.