PRISM

ARFANET : Une nouvelle approche pour les Réseaux Actifs

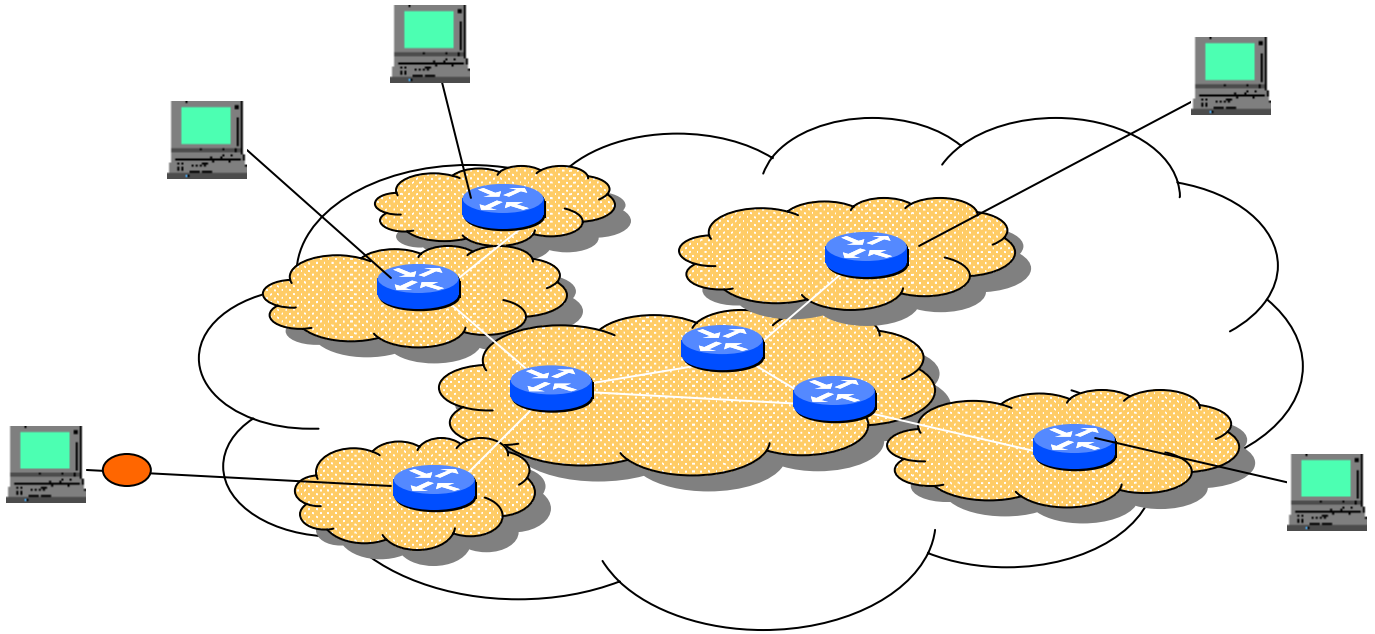
Amdjed Mokhtari

30 septembre 2005

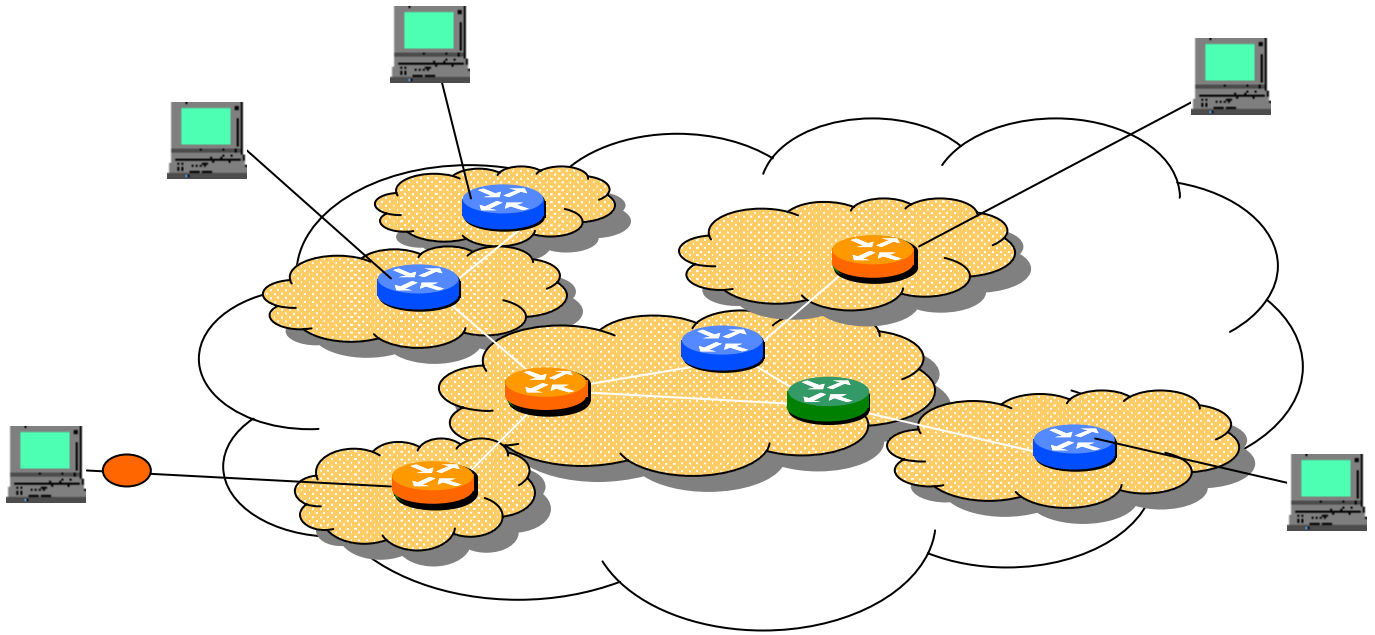
PLAN

- Introduction & Motivation
- ARFANet : une approche événementielle pour les réseaux actifs
- Analyse de performances d'un routeur ARFANet
- Mécanismes de distribution du code
- Mécanismes de sécurité
- Conclusions et perspectives

Introduction : Réseaux actifs



Introduction : Réseaux actifs

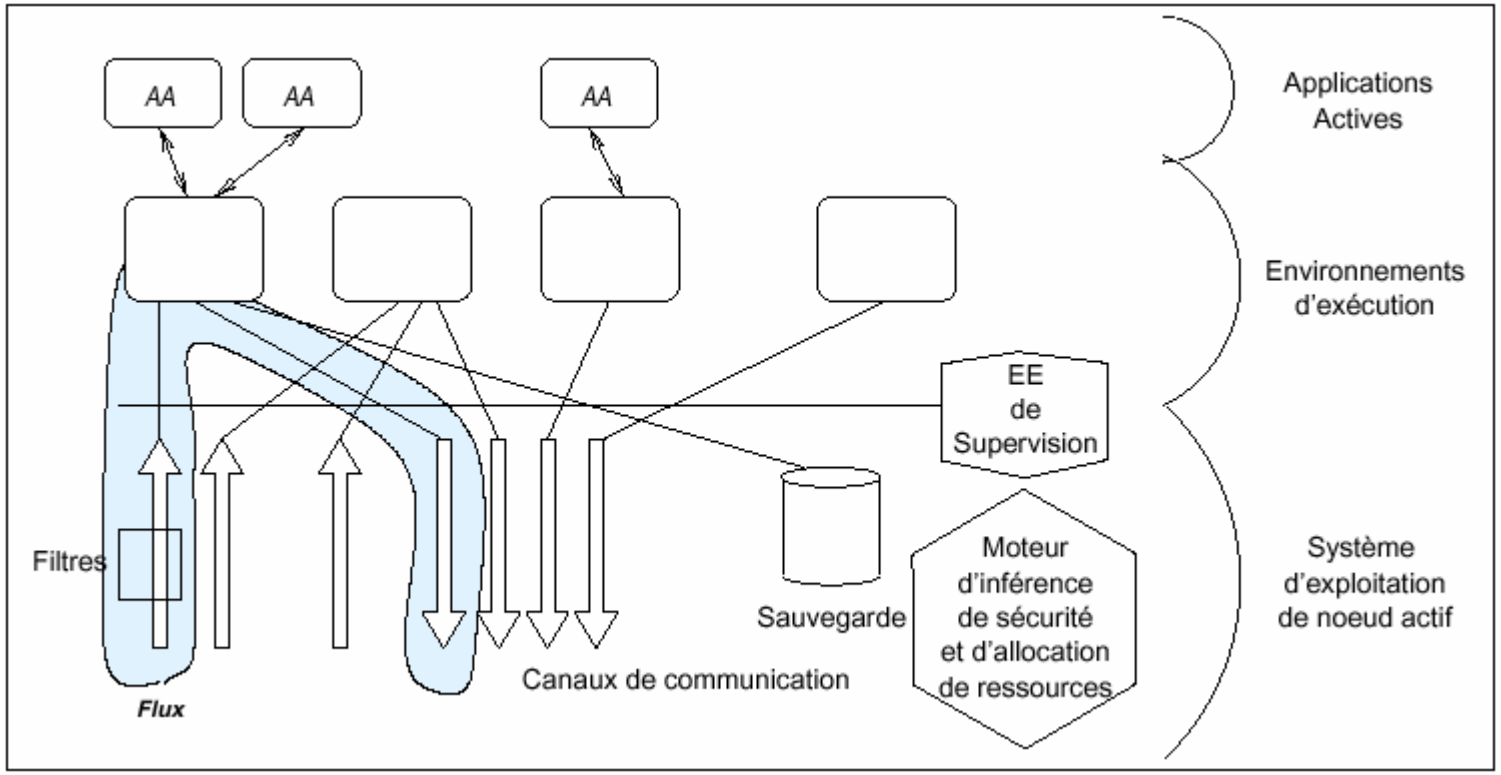


Introduction : Réseaux actifs

- Objectifs
  - Ouverture et flexibilité du réseau
  - Intégrer de nouveaux services sous forme de programme
  - Réduire le temps de déploiement des services

Introduction : Réseaux actifs

- Architecture d'un routeur actif



Introduction : Réseaux actifs

- Problèmes
  - Composition
  - Distribution & Gestion du code
  - Performances
  - Sécurité

Ces problèmes sont liés à la forme compacte du code

Introduction : Réseaux actifs

- Inconvénients de la forme compacte du Code
  - Difficile à composer avec d'autres codes dans le même langage ou des langages différents
  - Ne permet pas d'optimisation des codes
  - Des contrôles de sécurité coûteux
  - Non adapté à l'aspect événementiel du réseau

Formalisme de spécification adapté aux événements

ARFANet

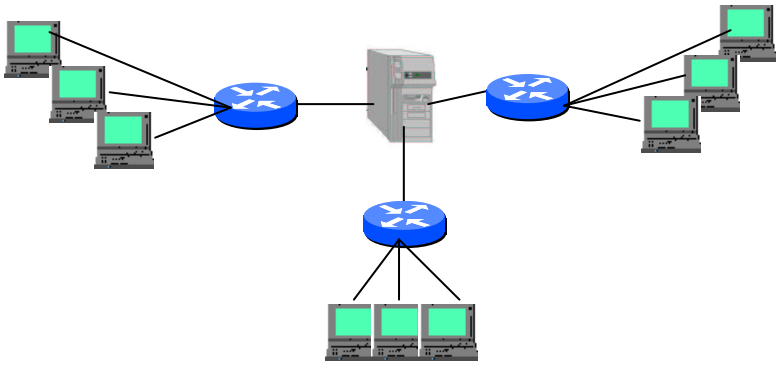
- Règles actives
- Forme : ON Événement IF Condition THEN Action
- Les composantes
  - Événement : signal discret (*l'arrivée d'un paquet, etc)*
  - Condition : formule logique ou booléenne (*si un paquet est dans le cache, etc)*
  - Action : traitement autorisé sur le nœud ou les données (*envoi d'un paquet, etc)*

ARFANet

- Exemple d'application
- soulagement d'un serveur Web en cas de surcharge

Degré :

- 0 : tout client
- 1 : en navigation
- 2 : valider un produit
- 3 : en transaction
- 4 : en confirmation



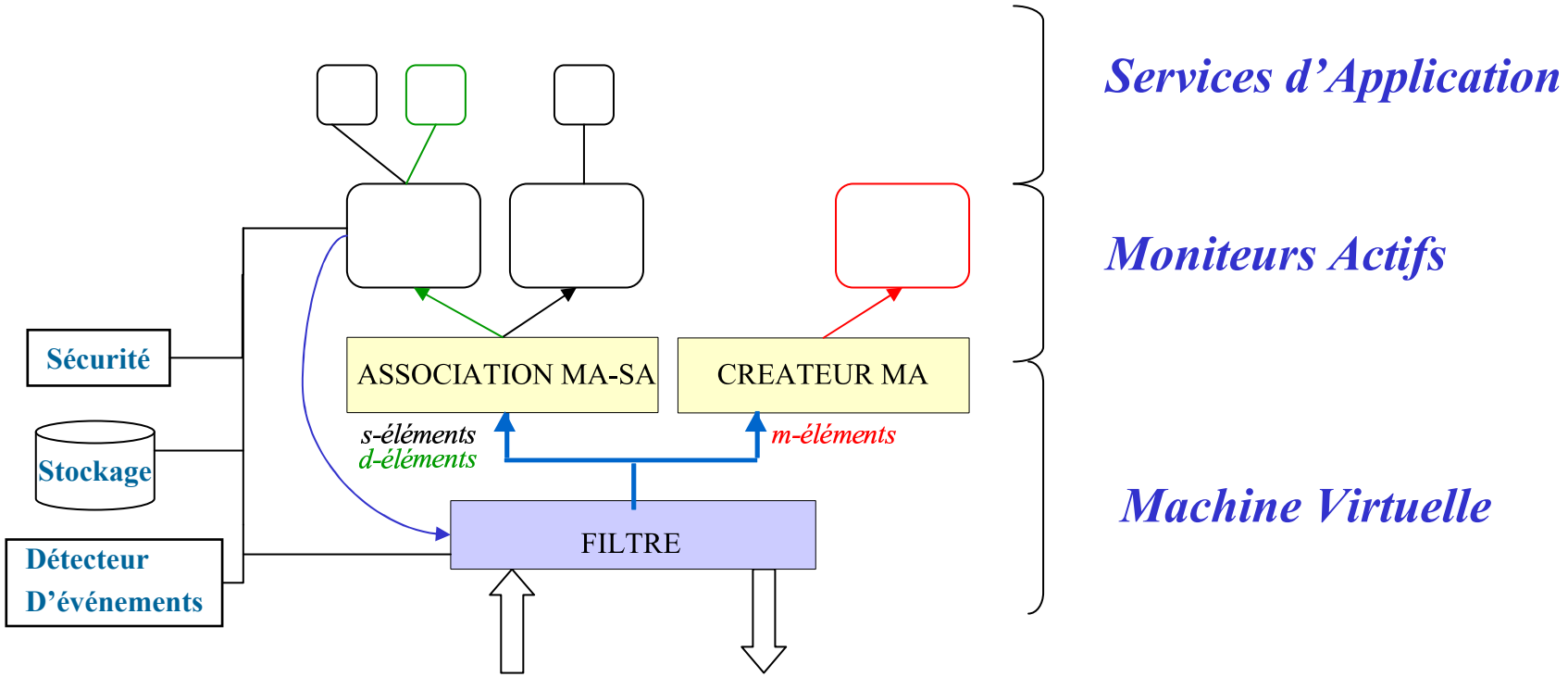
R1 : ON *Arrivée d'un client* IF *Catégorie du client* < *Degré* THEN *Rejeter le client*

R2 : ON *Arrivée d'un client* IF *catégorie du client* ≥ *Degré* THEN *Envoyer la requête au serveur*

R3: ON *Arrivée du degré* IF *true* THEN *Mise à jour de la valeur du Degré local*

ARFANet

- Architecture d'un nœud ARFANet
  - Les règles actives requièrent une architecture à trois niveaux



ARFANet

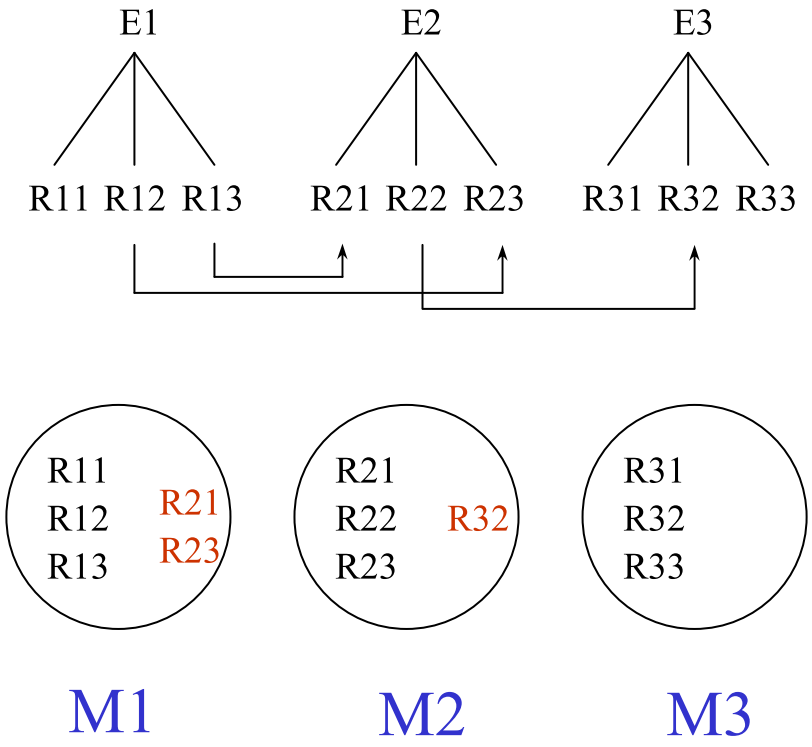
Couche services d'application

- Une application est un ensemble de règles actives regroupées en *modules*
- Module
  - la *sémantique d'exécution*
  - des *règles actives*

Module M1
Sémantique d'Exécution Consommation d'Évènement = toujours Mode de couplage E-C = immédiate Mode de couplage C-A = immédiate Priorité des Règles = aucune Exécution en Cascade = itérative
Règles Actives R1: On E1 If C1 Then A1() R2: On E1 If C2 Then A2() ... Rn: On Ep If Cn Then An()

ARFANet

- Modularité et Composition
  - Une règle ne peut déclencher qu'une règle du même module

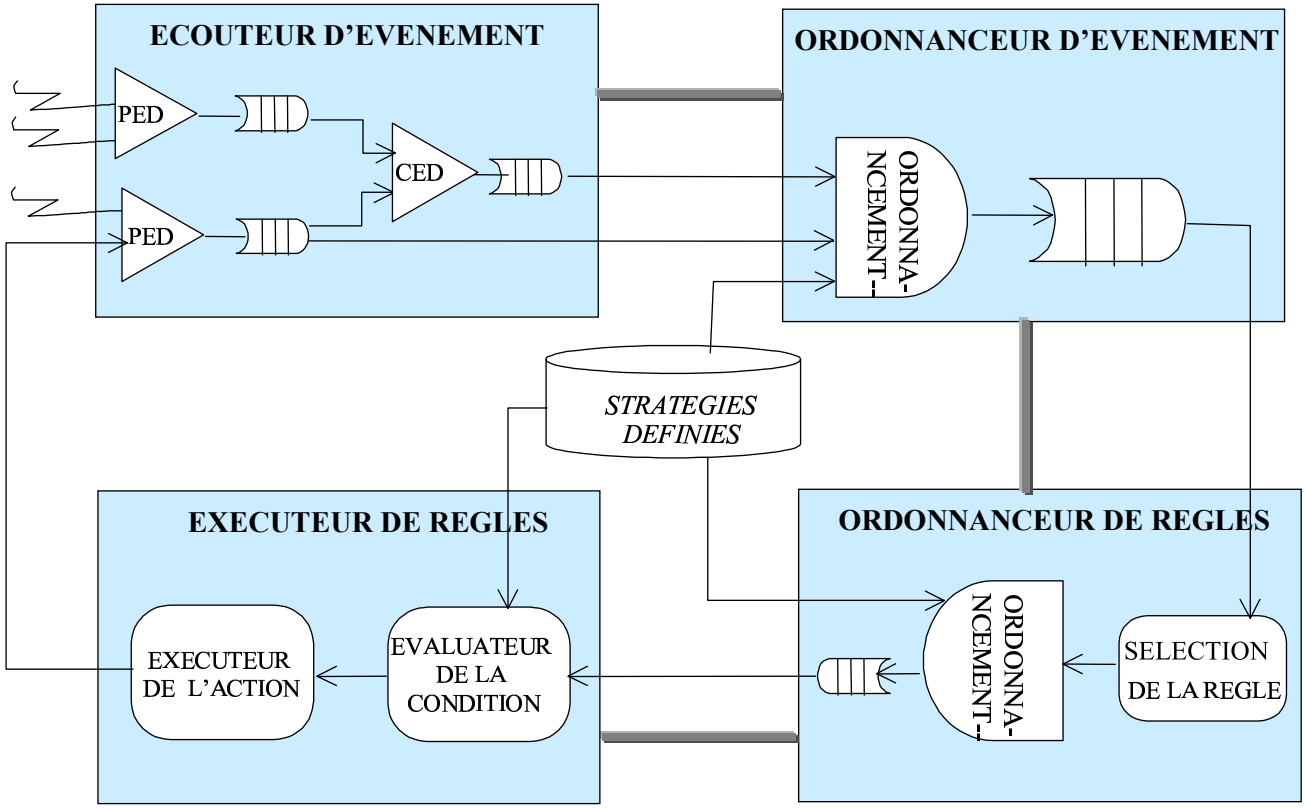


ARFANet

- Modularité et Composition
  - Aucune composition en amont
    - Pas de pré-compilation du code en intégrant les composantes
  - Aucune composition en aval
    - Pas de traitement par le nœud avant l'exécution, ce que fait FAIN
  - Mise en référence du module par un de ses événements

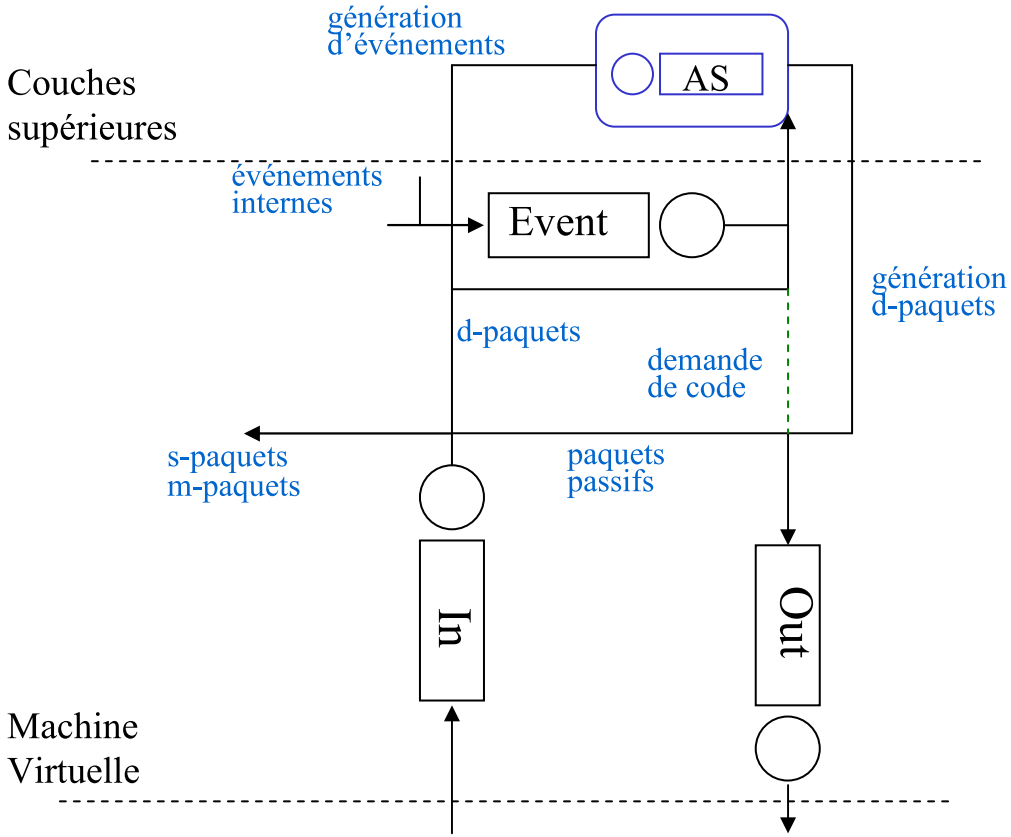
ARFANet

- Couche moniteur actif



Performances d'un nœud ARFANet

- Modélisation d'un nœud ARFANet



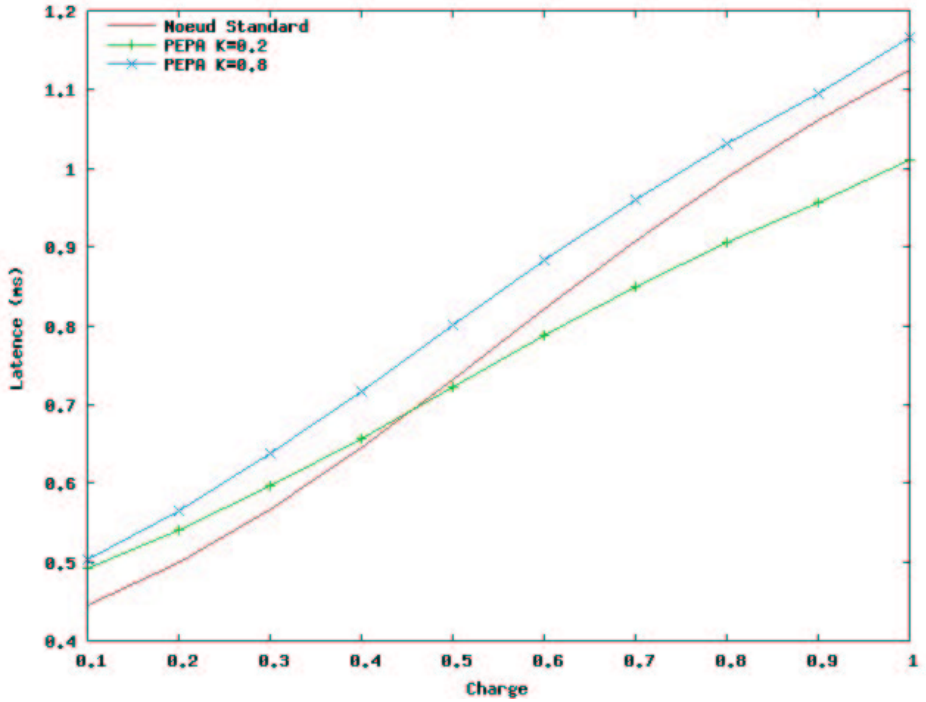
Performances d'un nœud ARFANet

- PEPA : Performance Evaluation Process Algebra [Hillston 94]
  - Approche compositionnelle
  - Systèmes concurrentiels
  - Chaque composantes comporte des activités
    - Type action
    - Taux exponentiel

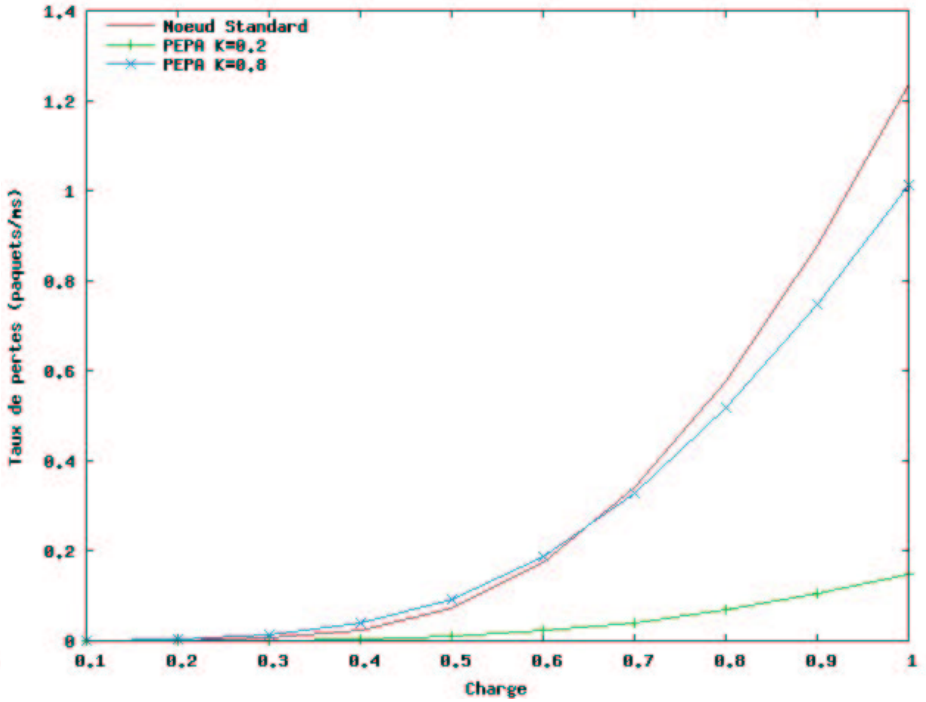
Performances d'un nœud ARFANet

- Résultats numériques : impact de la proportion des paquets actifs sur le débit, les pertes et la latence des paquets passifs

Latence



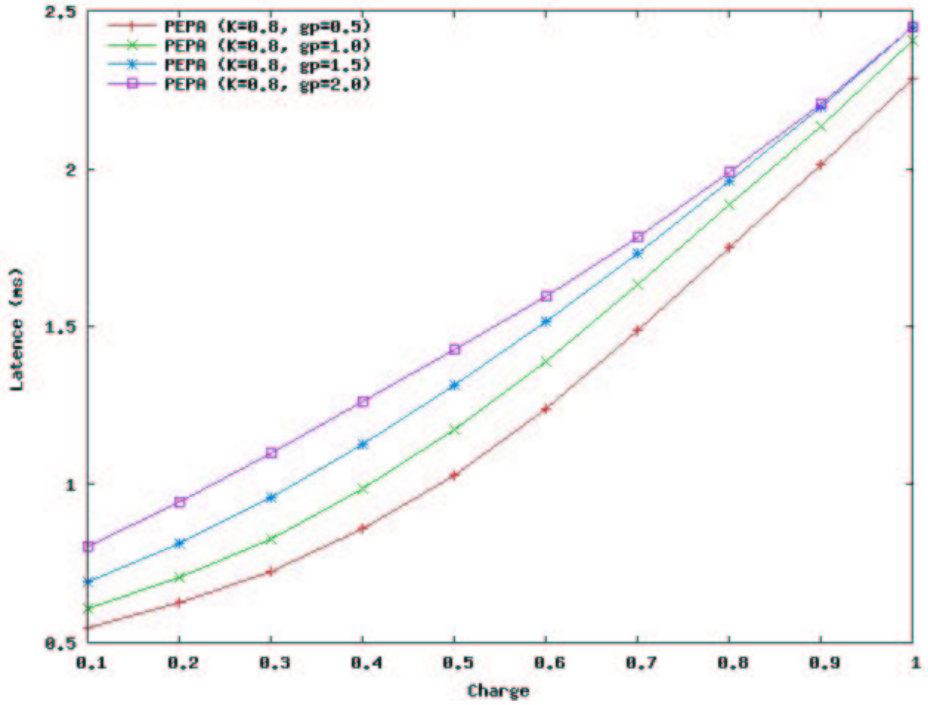
Taux de pertes



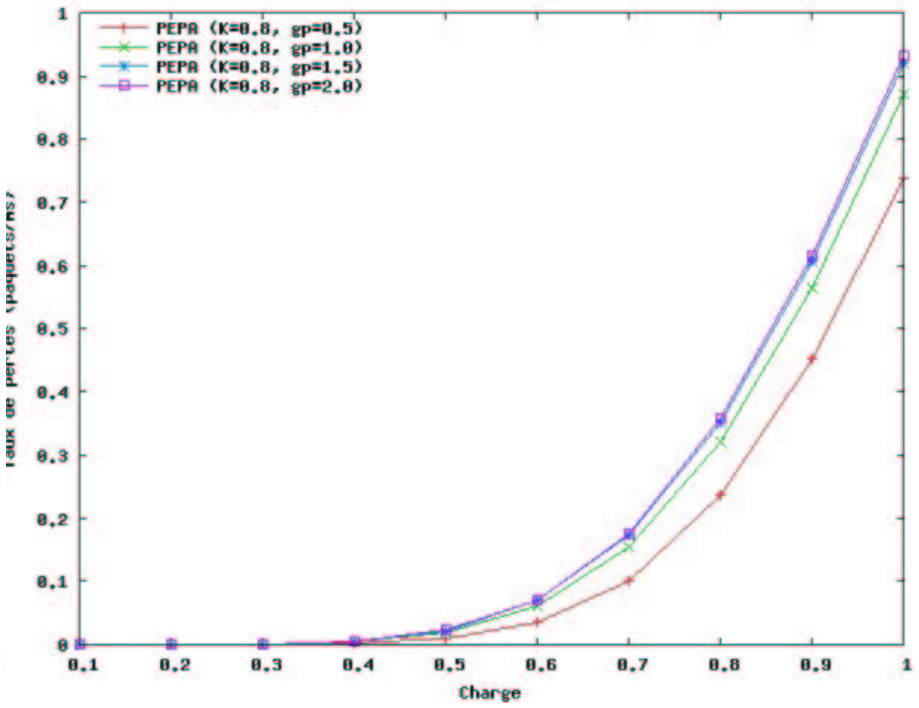
Performances d'un nœud ARFANet

- Résultats numériques : impact de la génération des paquets sur le débit, les pertes et la latence des paquets passifs

Latence



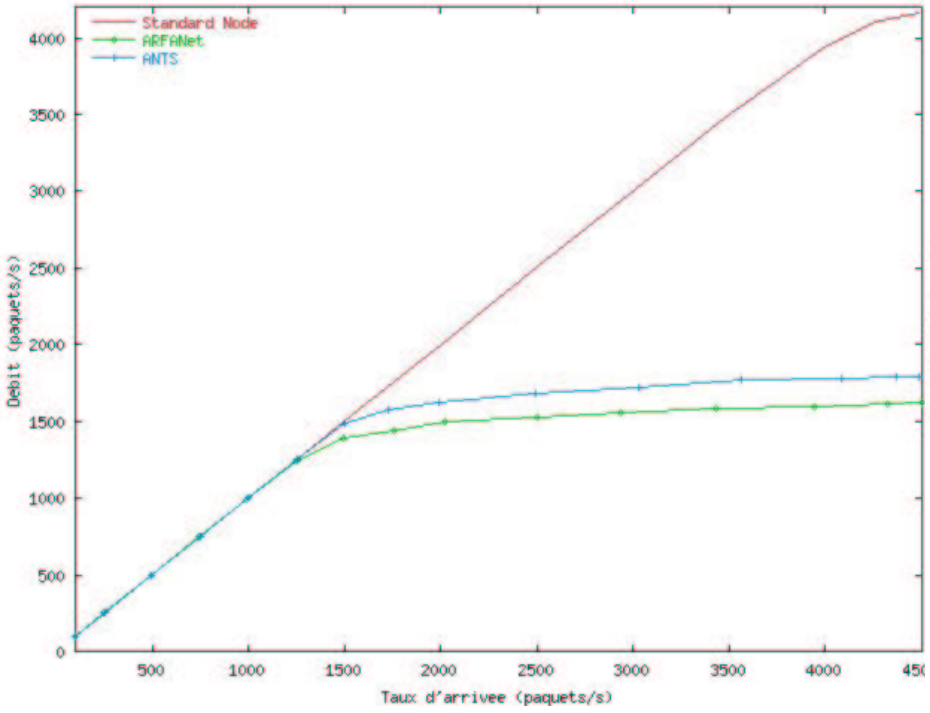
Taux de pertes



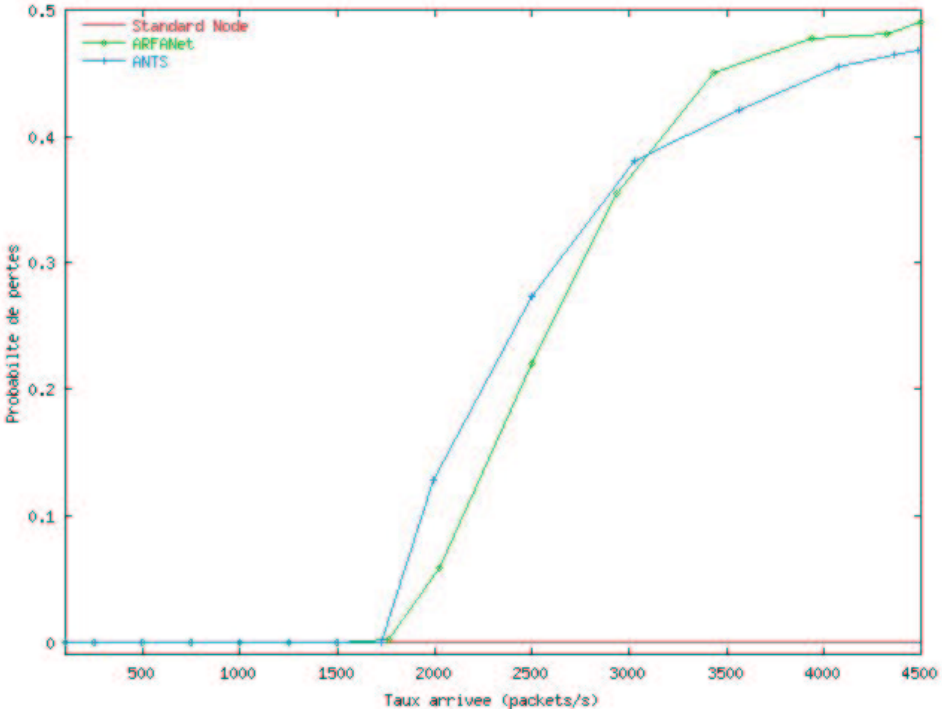
Performances d'un nœud ARFANet

- **Résultats numériques** : ARFANet v.s. ANTS (Active Node Transfer System)

Débit



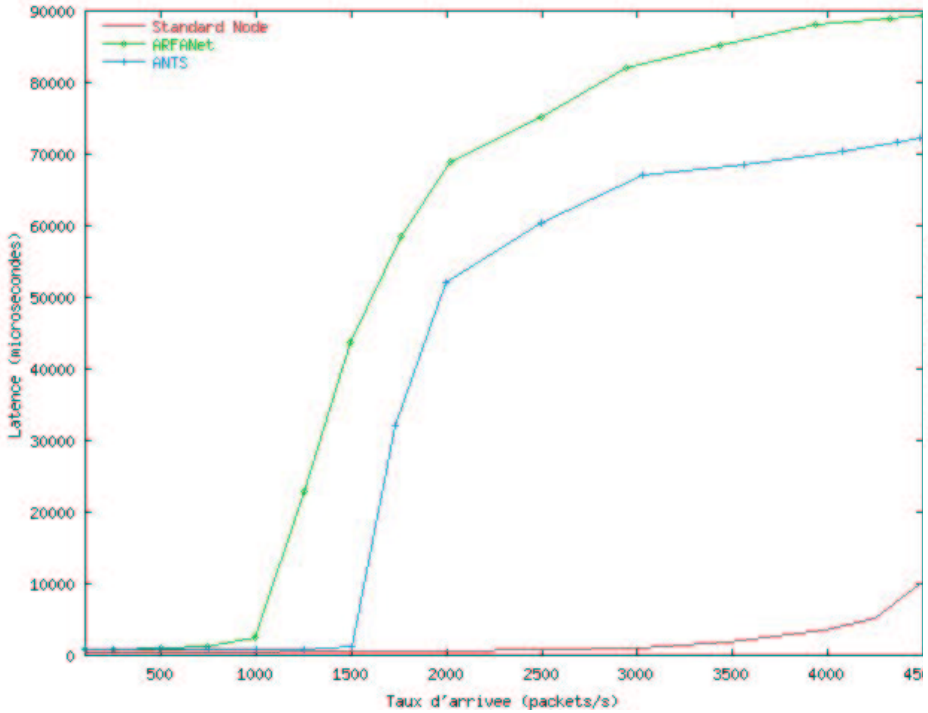
Taux de pertes



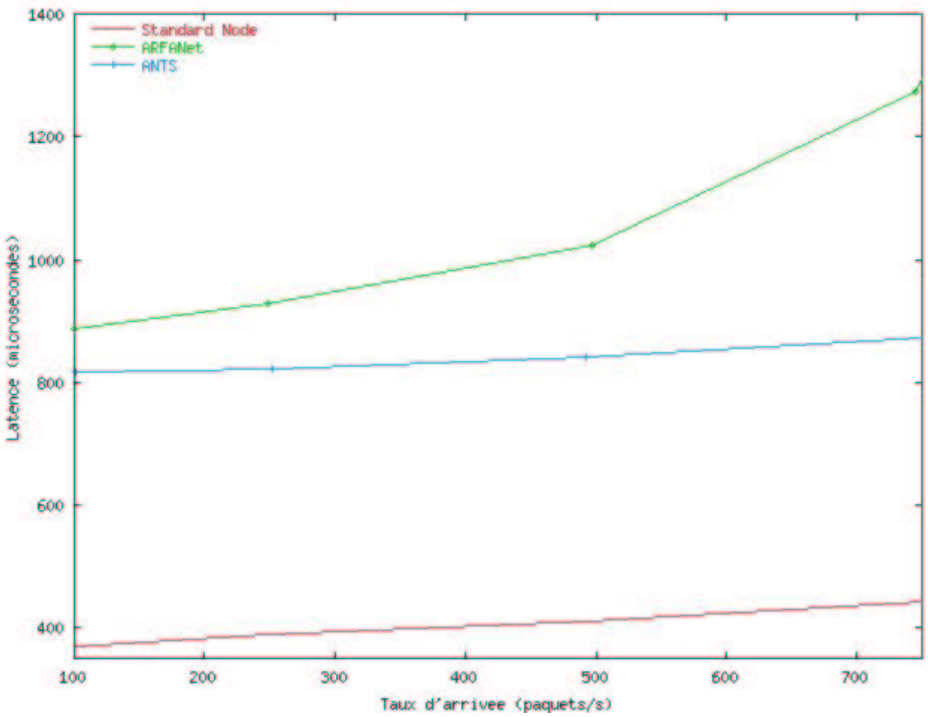
Performances d'un nœud ARFANet

- Résultats numériques : ARFANet v.s. ANTS (Active Node Transfer System)

Latence



Latence

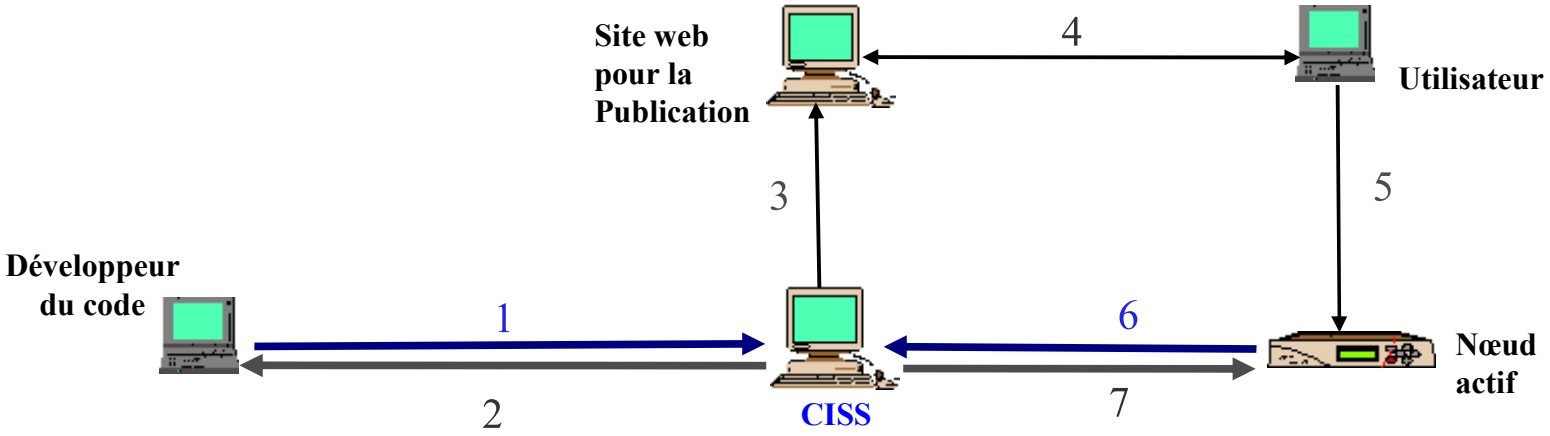


Distribution du code

- Identification du code
  - Filtre d'adresse (@source, …) et du type (TCP, …)
    - *limité à une classe d'utilisateurs*
  - Identificateur : signature numérique (MD5, …)
    - *lie l'identificateur à son développeur*
- Déploiement du code
  - In band
    - *non persistance et partage des codes, sécurité*
  - Out band
    - *présélection des nœuds, chemins multiples*

Distribution du code

- Approche CISS (Code Identification and Storage Server)



Phase de publication

1 – Envoi du code actif

2- Envoi de l'identifiant

3- Publication sur le site web

Phase de référencement

4-Consultation du service d'application et récupération de l'identifiant

5- Envoi des paquets de données actives avec référence

Phase de Déploiement

6- Demande du code référencé

7– Envoi du code actif

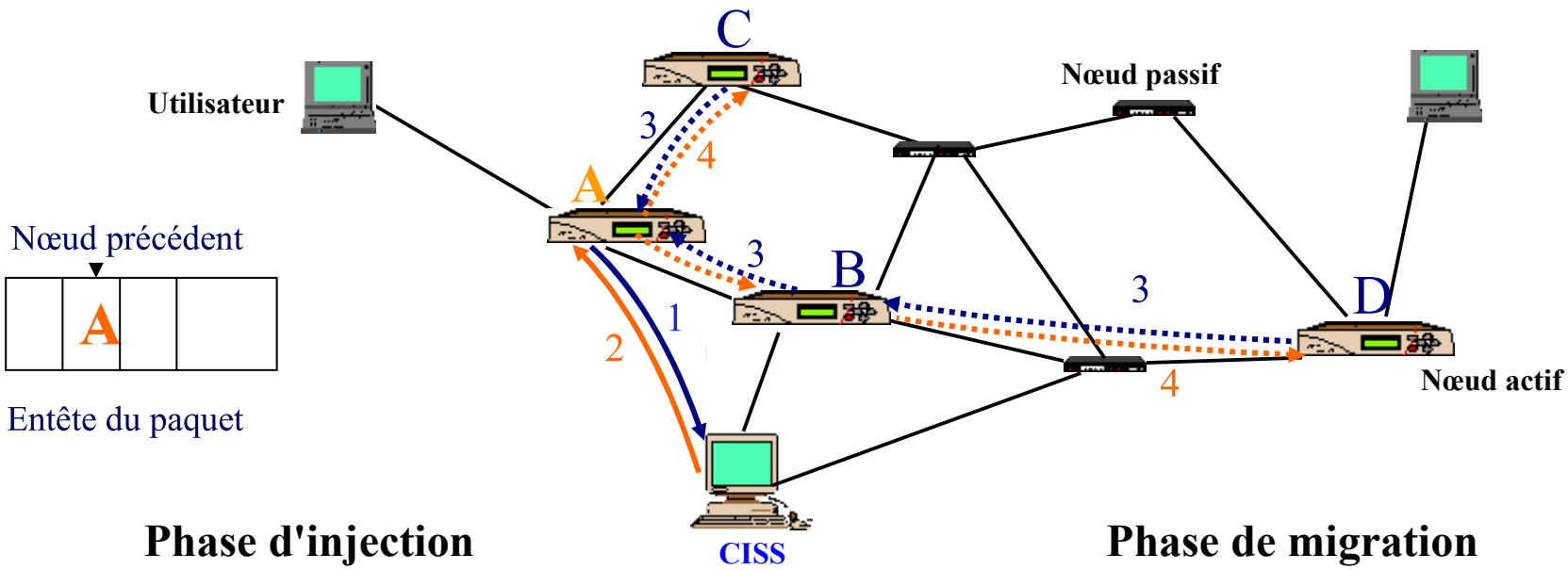
Distribution du code

- Approche multi CISS
  - Répartition des CISS
    - Placés à la périphérie du réseau
  - Gestion de la base de code
    - Bases de code réparties
    - Bases de code répliquées
  - Garantir l'unicité de l'identifiant

ARFAnet : gestion de la base de code répartie

Distribution du code

- **Approche Mixte** : combine l'approche CISS et l'approche Hop by Hop (Nœud par Nœud défini dans ANTS)



Phase d'injection

1- Demande de code

2 - Envoi du code actif

Phase de migration

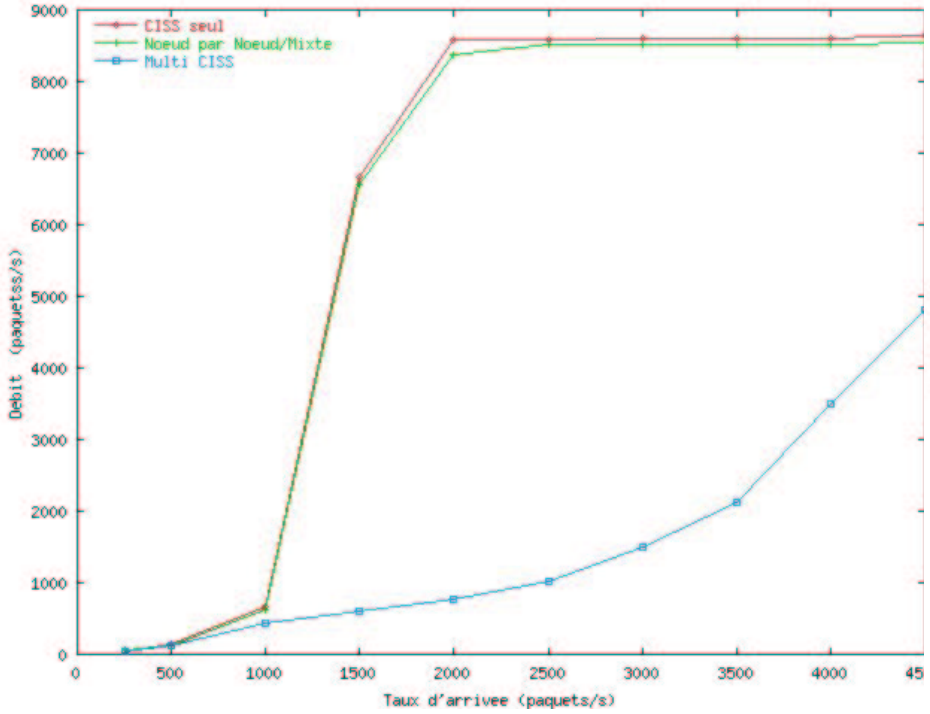
3- Demande de code

4 - Envoi du code actif

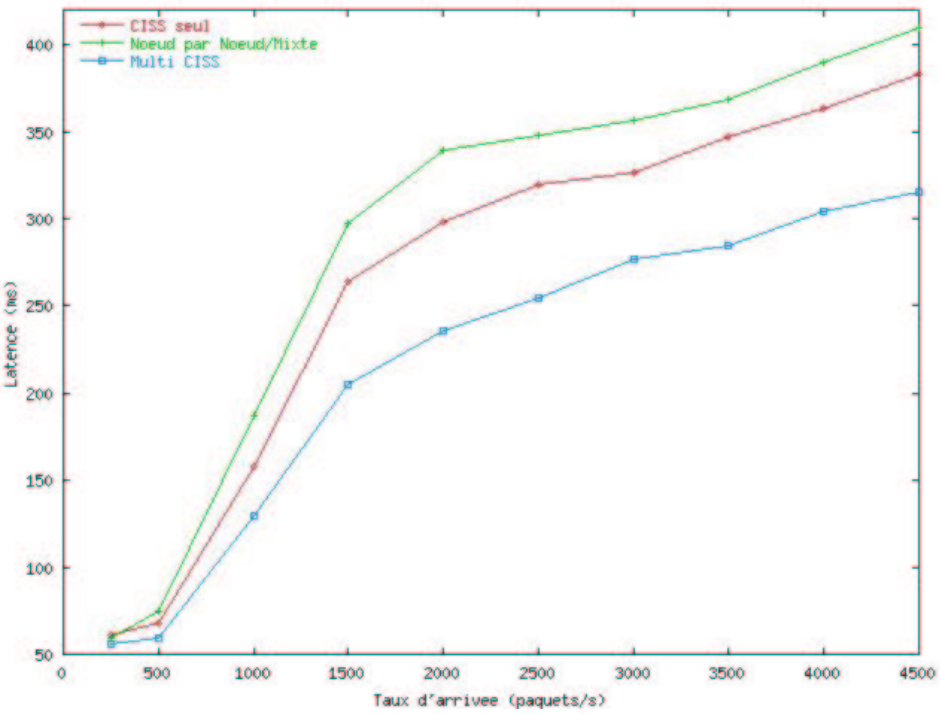
Distribution du code

- Résultats numériques : comparaison des différentes approches en termes de latence et de débit

Débit



Latence

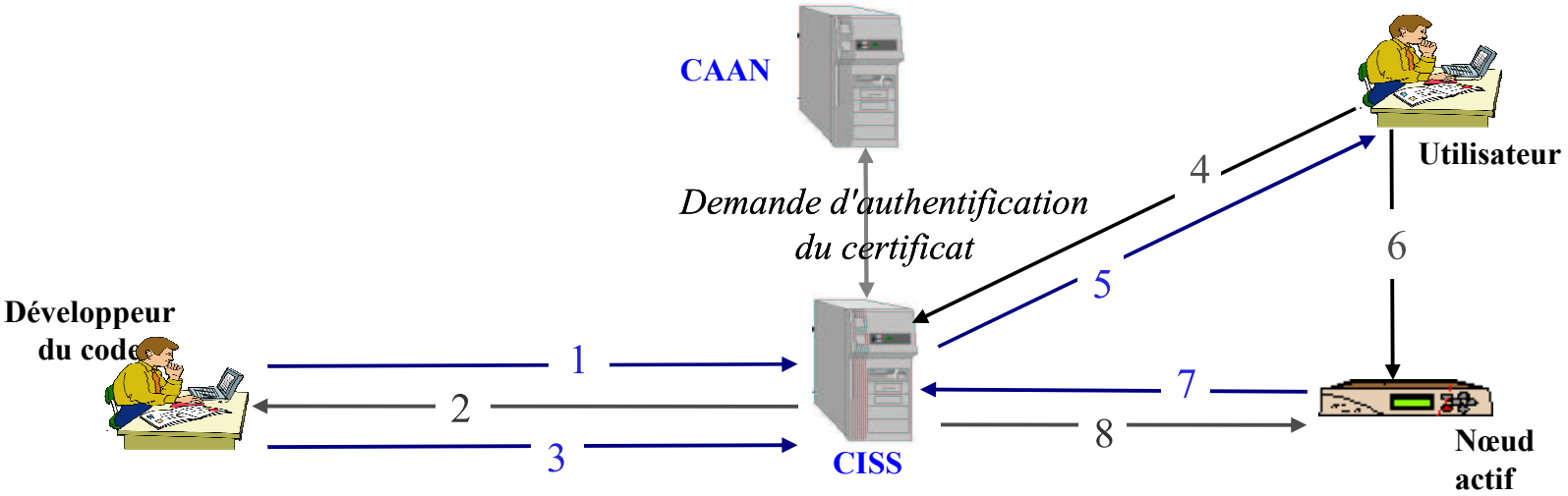


Mécanismes de sécurité

- Sécurité dans la distribution du code
  - Authentification
    - CAAN (Certificate Authority for Active Network)
    - Clef pour chaque entité : CISS, nœuds, développeurs, utilisateurs et aussi le code
  - Autorisation d'exécution
    - Utilisation des clefs temporaires
    - Amélioration de la technique ROSA [BAGNULO et al 02]
- Sécurité dans l'exécution du code

Mécanismes de sécurité

Sécurité dans la distribution du code



Phase de publication

1 – Envoi de certificat avec demande de publication de code

2- Acceptation d'envoi du code

3 – Envoi du code actif

Phase de référencement

4- Envoi du certificat avec Demande d'une clef temporaire

5- Envoi de la clef temporaire après vérification

6- Envoi des paquets de données actives avec référence et la clef temporaire

Phase de Déploiement

7- Demande du code avec la clef

8– Envoi du code avec sa clef

Conclusions

- ARFANet
  - Utilisation des règles actives
  - Modularité et composition
- Étude de performances
  - Impact de l'introduction des paquets actifs sur les paquets passifs
  - ANTS vs. ARFANet
- Distribution du code & sécurité
  - Schéma global pour la distribution du code basé sur
    - Un serveur d'identification et de sauvegarde (CISS)
    - Un site web pour la publication de la base de code du CISS
  - Schéma global pour la sécurité basé sur
    - Utilisation des clefs temporaires pour le déploiement du code
    - Une autorité de certification (CAAN)

Perspectives

- ARFANet
  - Optimisation de la machine virtuelle
  - Intégration d'autres langages
  - Composition de modules dont les langages sont différents
  - Implémentation d'autres applications réseau
- Distribution du code & sécurité
  - Étude de performances des techniques élaborées dans un réseau de grande échelle
  - Réaliser des mesures de performances pour étudier le coût en des mécanismes de sécurité développés