Découverte et analyse de dépendances dans des réseaux d'entreprise

Samer MERHI

Sous la direction de:
Olivier FESTOR

LORIA - Equipe MADYNES - Université Henri Poincaré

June 24, 2008

Plan

1. Introduction
2. Approches existantes
3. Notre contribution
4. Evaluation et analyse
5. Conclusion et perspectives

Introduction

L'étude des dépendances est indispensable pour:

- analyser les causes de fautes
- améliorer les performances.

Problématique

Les techniques proposées:

- problème de débit
- problème de légalité

Notre objectif

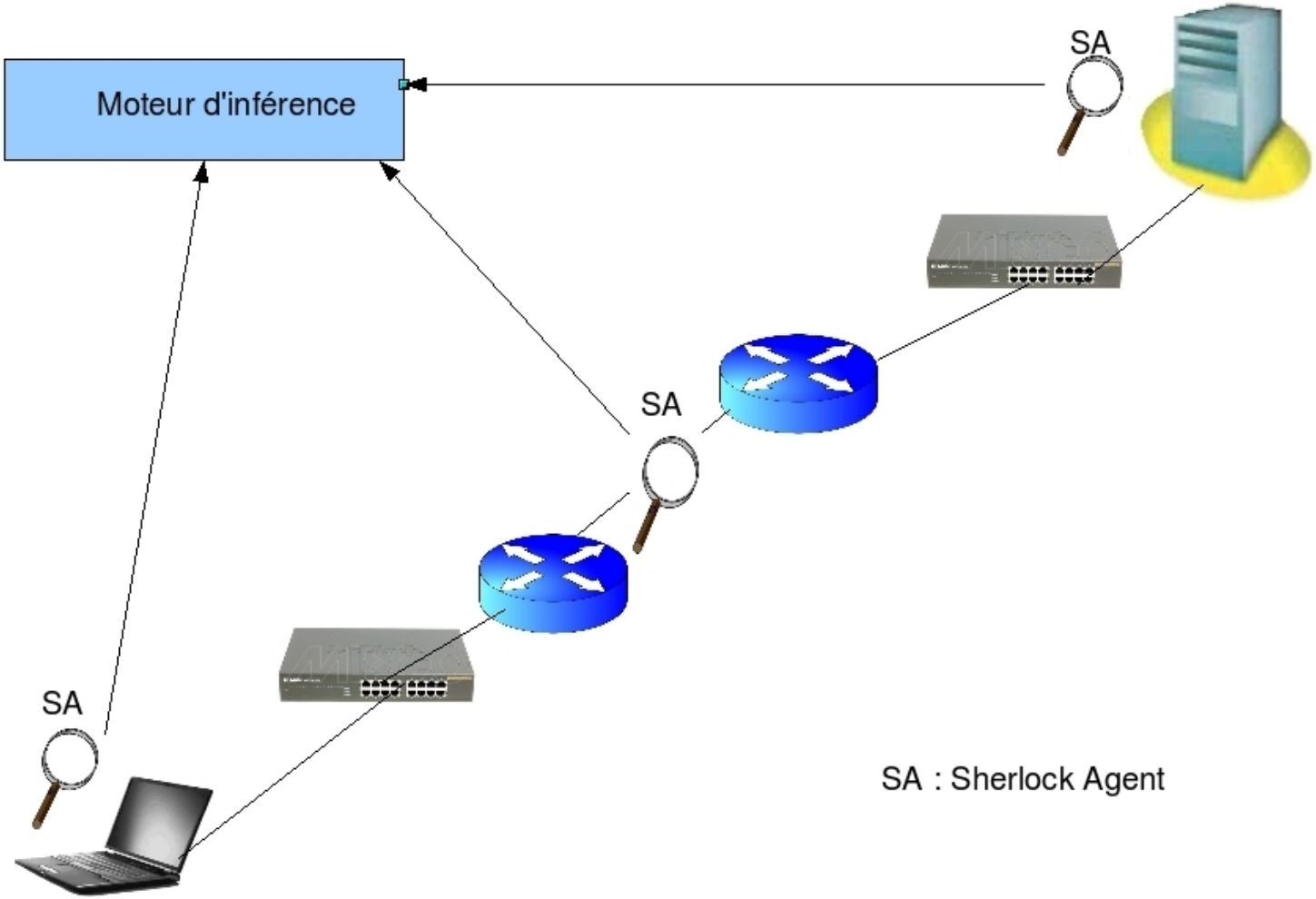
Aadaptation des méthodes existantes en exploitant une donnée facilement disponible (Flux Netflow).

Approches existantes

- Sherlock
- Kachima et al.
- Active Dynamic Discovery (ADD)

Sherlock

- Towards Highly Reliable Enterprise Network Services Via Inference of Multi-level Dependencies.
- Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula,David A. Maltz, Ming Zhang.
- SIGCOMM'2007, Aout 27 -31, 2007

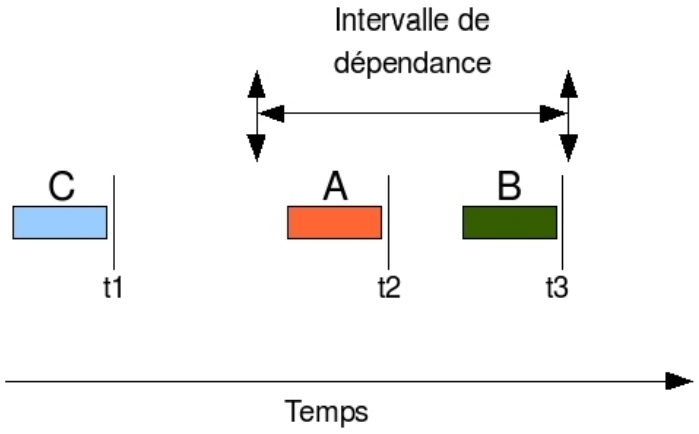

Moteur d'inférence
SA
SA
SA
SA : Sherlock Agent

Découverte des services

- (IPdest,IPSource,Portdest,PortSource)

Découverte des dépendances:

- Un service B est considéré comme dépendant de A si au moins un paquet de A apparaît avant B avec une probabilité forte pendant un intervalle de temps dit Intervalle de dépendances.

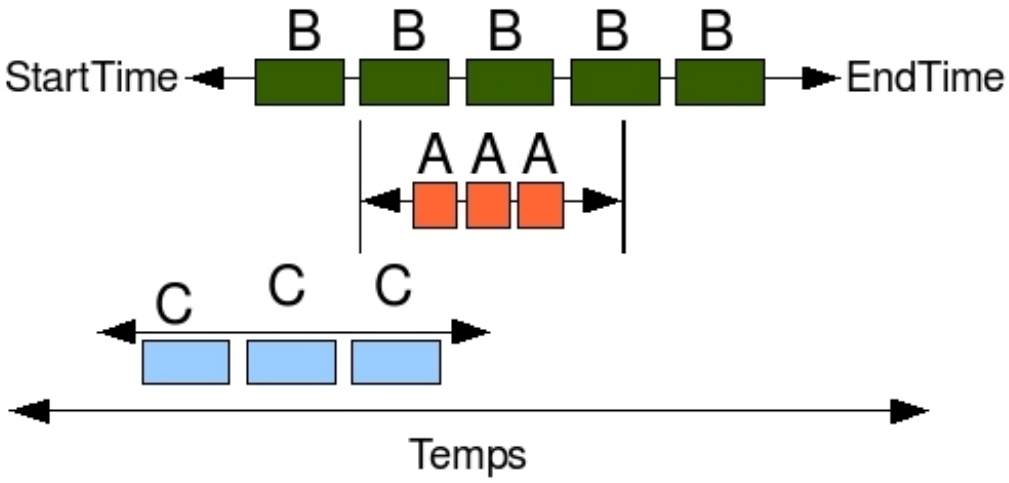


B dépend de A

Kachima et al.

- Network-based Problem Detection for Distributed Systems
- Hisashi Kashima, Tadashi Tsumura, Tsuyoshi Idé, Takahide Nogayama, Ryo Hirade, Hiroaki Etoh Takeshi Fukuda
- 21st International Conference, Japan , 5-8 Avril 2005

- Une technique basée sur l'analyse de l'historique D des temps de début et de fin d'exécution des services.



B dépend de A

- Nécessite une inspection profonde

ADD

- Approche active
- L'injection d'une perturbation à une entité cause l'interruption d'une ou de plusieurs applications $- - -\triangleright$ dépendance entre l'entité perturbée et l'application affectée par la perturbation.

Table de Comparaison

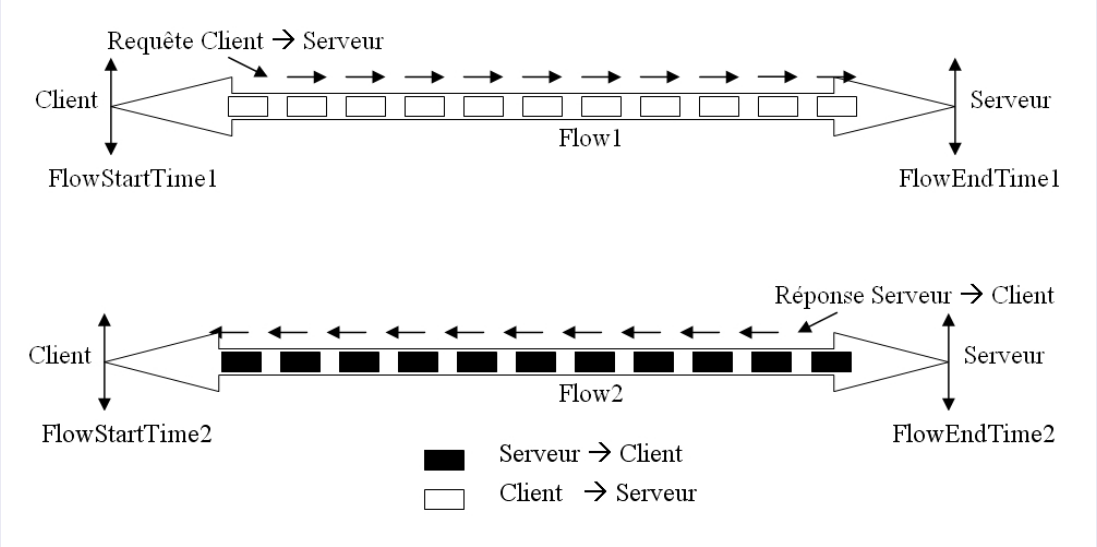
Technique	Graphe de dépendance	Données en entrée	Architecture	Actif/Passif
Sherlock	(couche physique + services)	Entêtes des paquets IP	centralisée + distribué	Passif
Kachima et al.	(services)	Paquets IP Complets	distribuée	Passif
ADD	Couche Physique + Services	Résultats des perturbations	distribuée	Actif

Notre Contribution

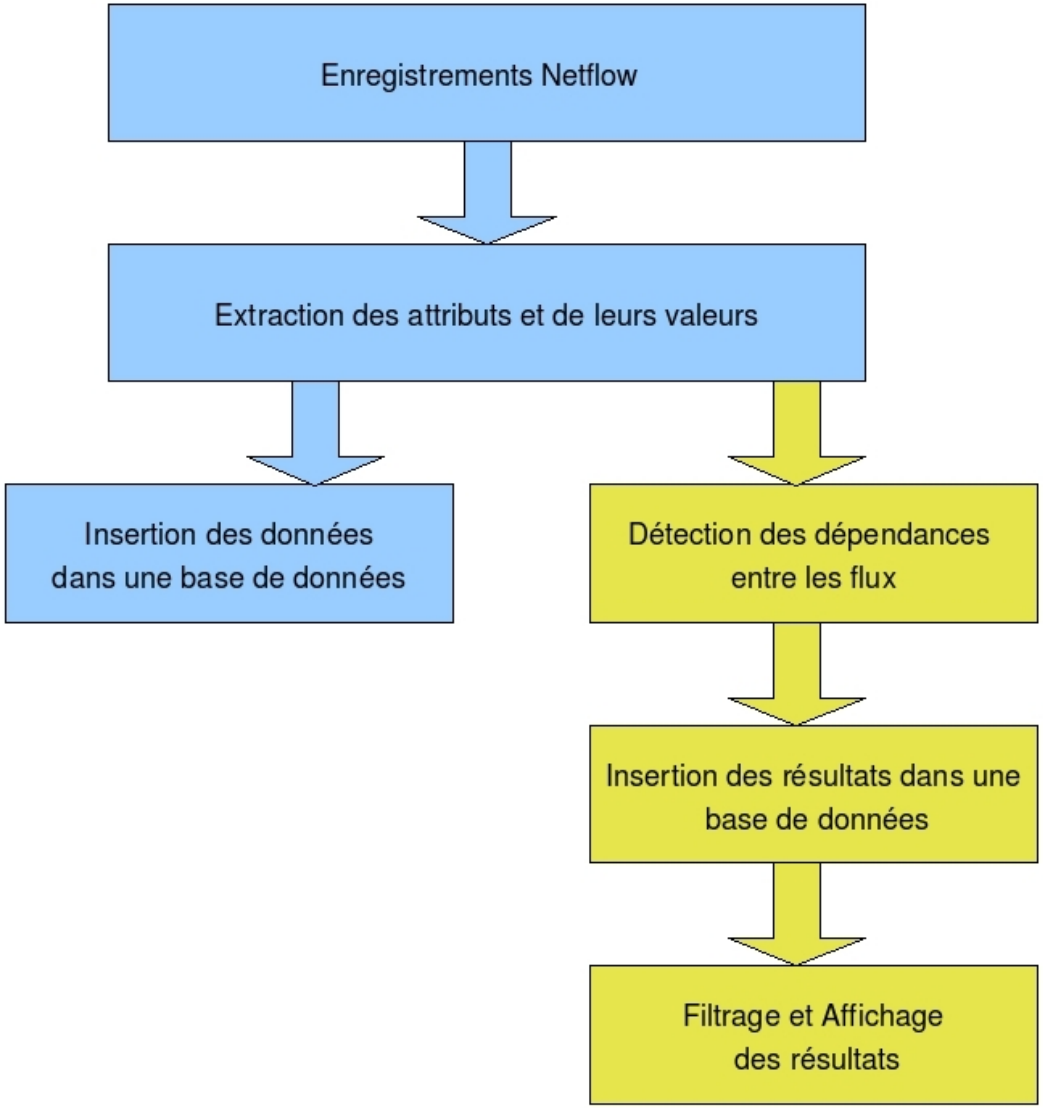
Une technique de découverte des dépendances basée sur l'analyse des enregistrements des flux Netflow

Champs clé d'un flux Netflow

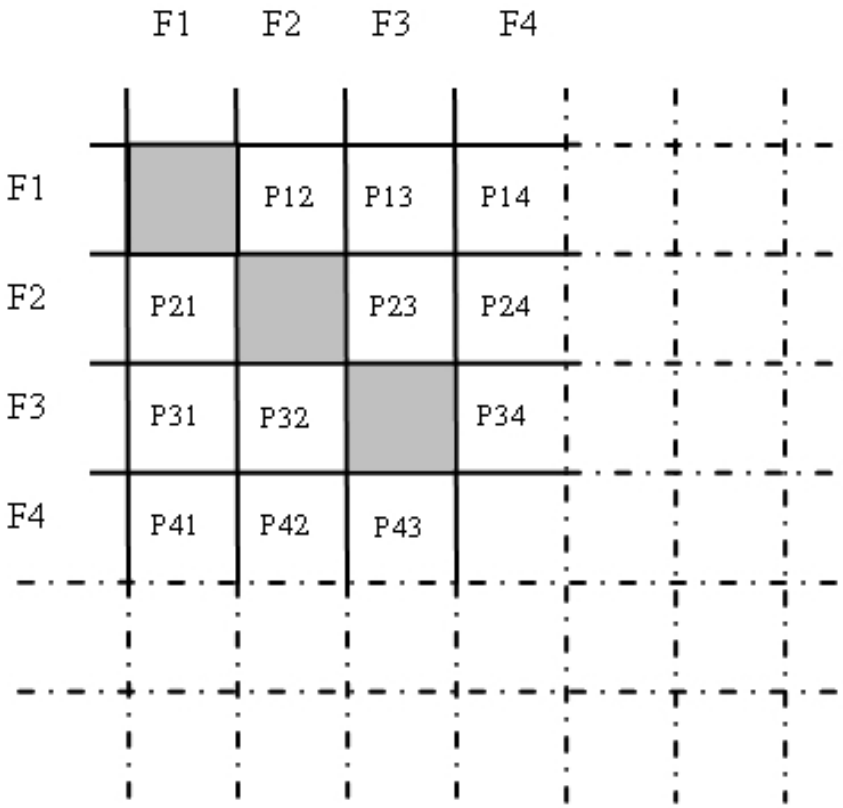
FlowStartTime
FlowEndTime
SourceIP
DestinationIP
SourcePort
DestinationPort
Protocol



Besoins fonctionnels



la matrice des facteurs de dépendances:

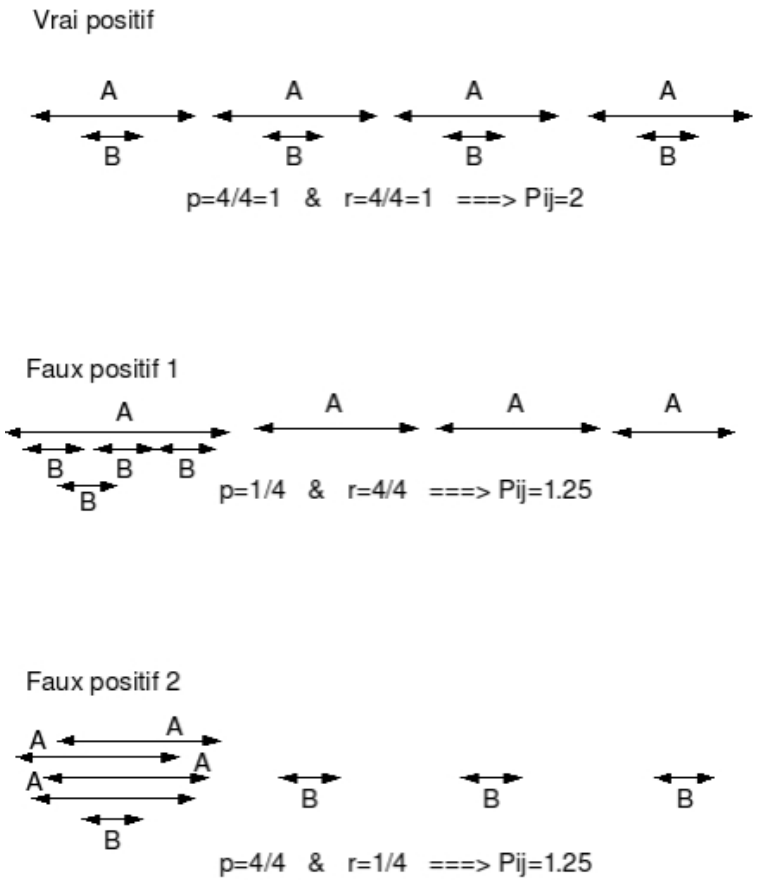


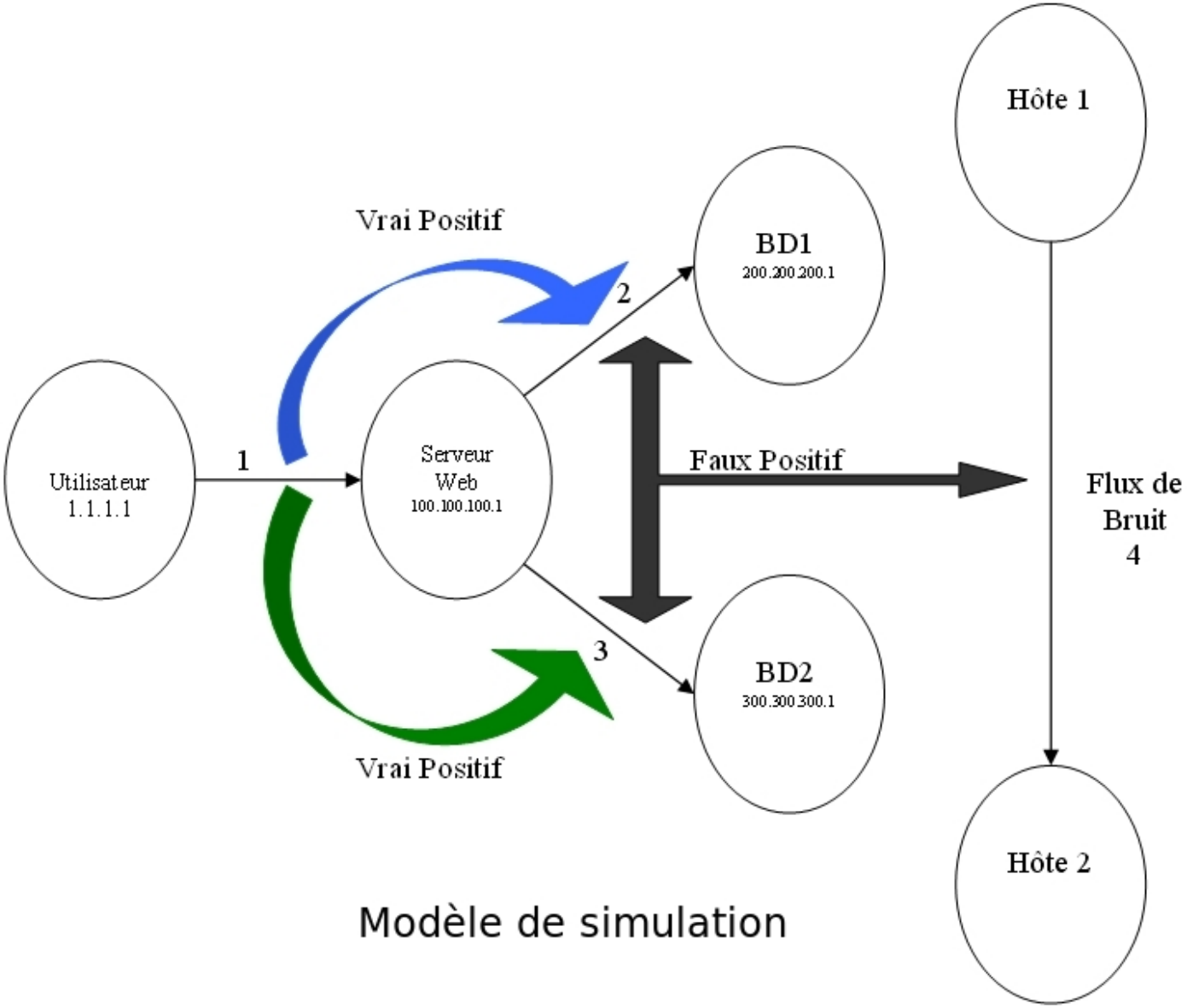
$P_{ij}/2$ correspond à la probabilité que le flux Fi dépend du flux Fj.

$P_{ij} = max(p, r) + p * r$ avec p=$\frac{\#(F_i/F_i>F_j)}{\#F_i}$ et r=$\frac{\#(F_j/F_i>F_j)}{\#F_j}$.

$$P_{ij} = max(p, r) + p * r \text{ avec p}=\frac{\#(F_i/F_i>F_j)}{\#F_i} \text{ et r}=\frac{\#(F_j/F_i>F_j)}{\#F_j}.$$

Vraix positifs vs Faux positifs





Modèle de simulation

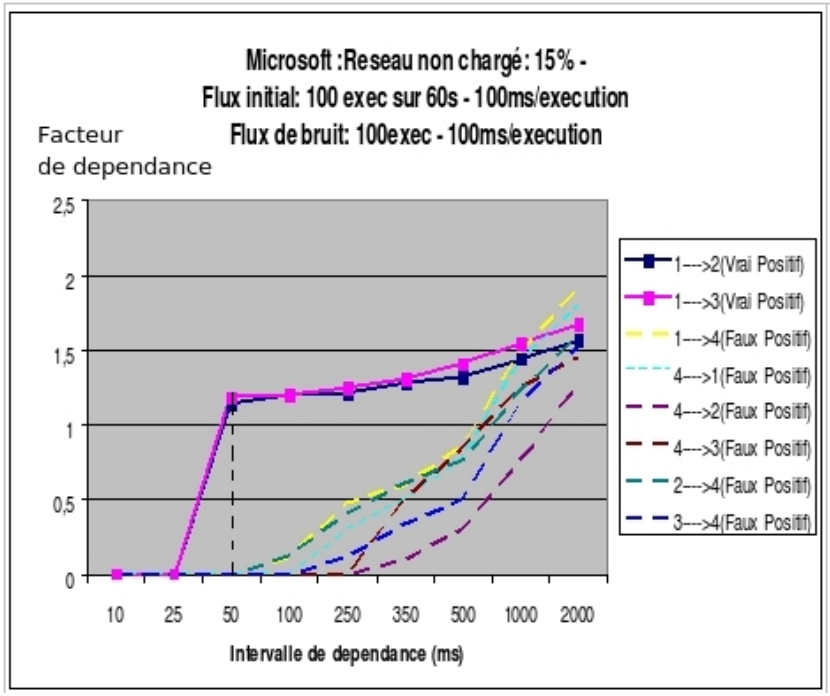
Paramètres de simulation

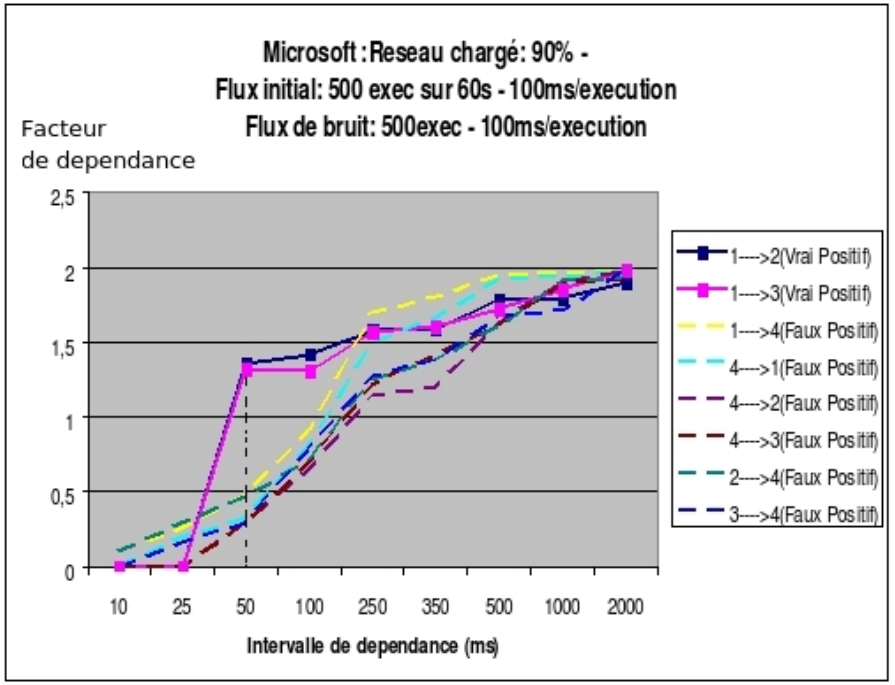
Paramètres	Description
ExecTime	Durée d'exécution d'un flux.
NbFlux	Nombre des flux générés ou des instances exécutées.
NbHôtes	Nombre d'hôtes générant du trafic en parallèle.
%BD1	Probabilité avec laquelle une requête est envoyée vers la première base de données.
%BD2	Probabilité avec laquelle une requête est envoyée vers la deuxième base de données.
(SST) et (SET)	SimulationStartTime et SimulationEndTime.

Deux environnements

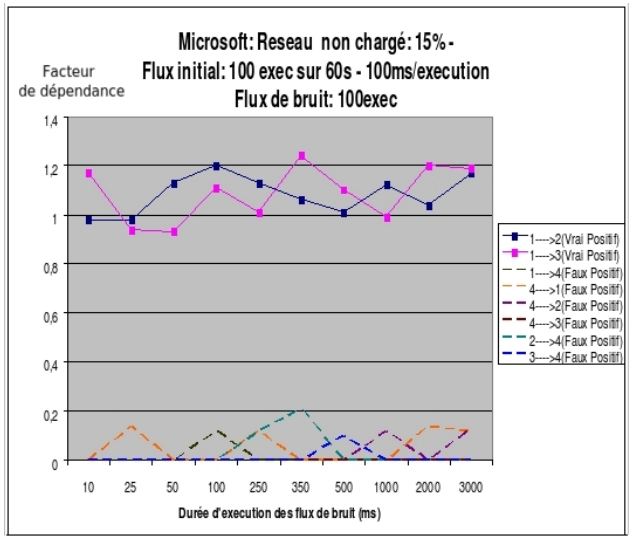
1. Réseau chargé: les flux de dépendance occupent 90% du temps de simulation.
   - NbFlux généré égal à 500 flux
   - Durée d'exécution de chaque flux suit une loi N(100ms,10)
   - Durée de simulation égale à 60s
2. Réseau non chargé: les flux de dépendance occupent 15% du temps de simulation.
   - NbFlux généré égal à 100 flux
   - Durée d'exécution de chaque flux suit une loi N(100ms,10)
   - Durée de simulation égale à 60s

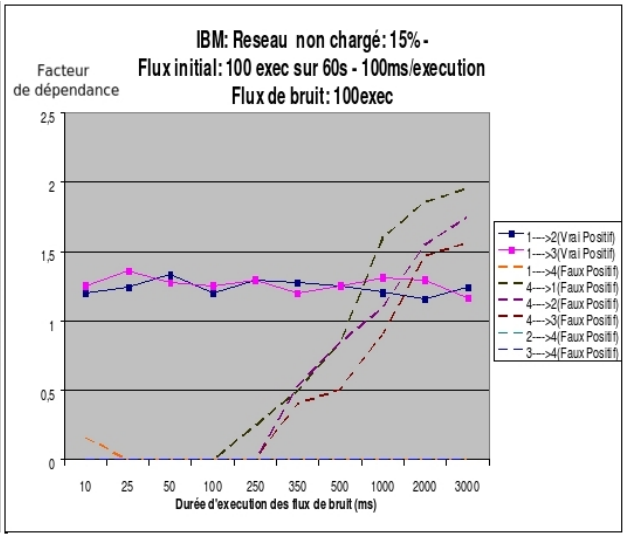
Intervalle de dépendance optimal (50ms)

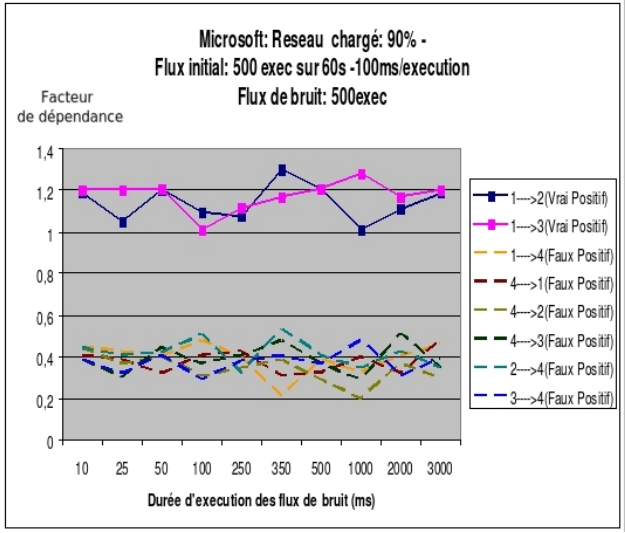


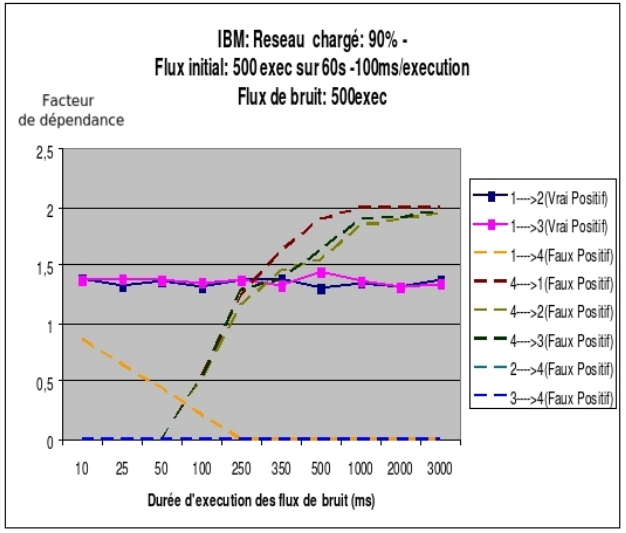


L'effet de la durée d'exécution des flux de bruit

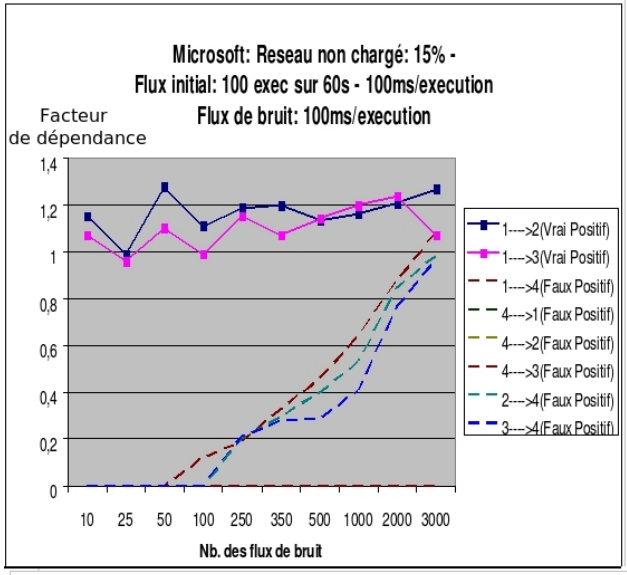


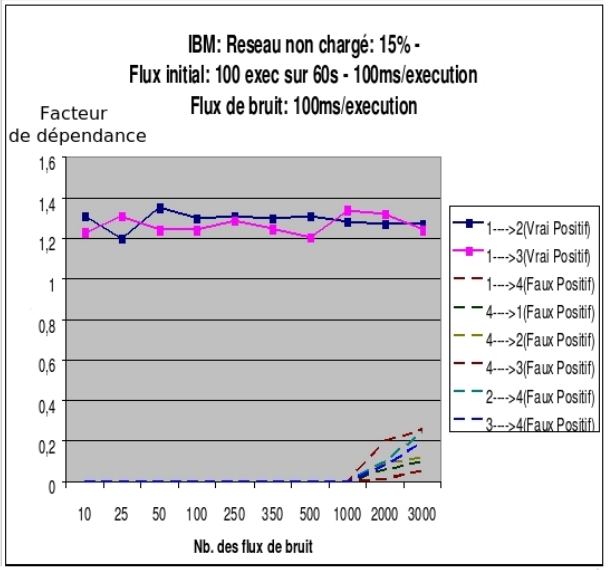


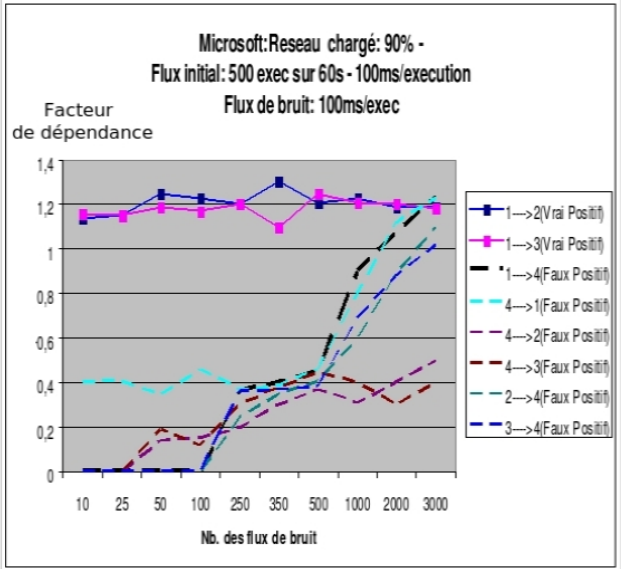


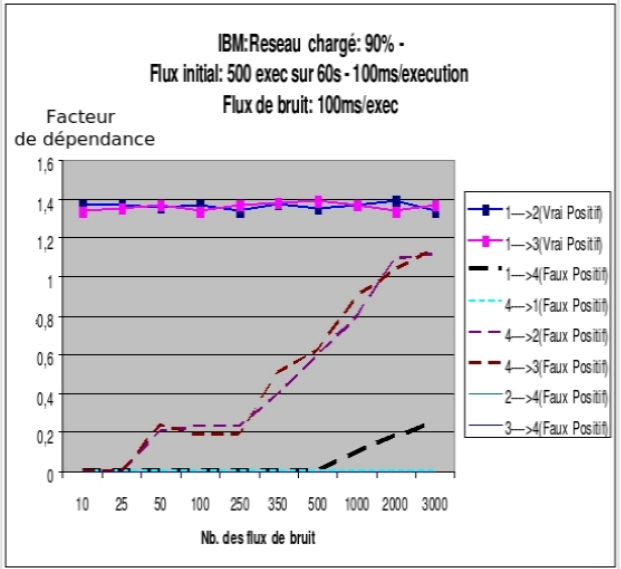


L'effet du nombre des flux de bruit









Résumé des résultats

Technique	Durée d'exécution	Nb de flux de bruit
Algorithme 1	OK	X
Algorithme 2	X	OK

Algorithme 1 : L'algorithme respectant la définition des chercheurs de Microsoft.

Algorithme 2 : L'algorithme respectant la définition des chercheurs de IBM.

Conclusions

- Analyse et comparaison des différentes techniques de détection des dépendances
- Conception d'une nouvelle méthode basée sur l'analyse des enregistrements Netflow
- Implémentation et évaluation de la méthode

Perspectives

- Evaluation à grande échelle de notre approche
- Etude d'une approche hybride $\longrightarrow$ Modèle de séléction permettant de basculer d'une approche à une autre
- Simulation sur des autres modèles de trafic.

Merci Pour Votre Attention