

Description détaillée des projets

Sujet : Infonébulisation: Fog Computing

Etudiants affectés :

Mohamed Sofiene Barka	Rafrat Mohamed	
Nour Elhouda Mami	Adam Lahbib	Doukali Syrine

Objectif :

L'objectif de ce projet est de se familiariser aux nouvelles technologies émergentes de fog computing et de les différencier de Cloud Computing et simulation d'un exemple complet de tous les composant de l'infonébulisation.

1. Notion de base :

1.2 Présentation du Fog Computing

Fog Computing est une nouvelle architecture (cf. figure 1) qui crée une liaison entre le nuage et le monde de l'internet des objets (IoT), agissant comme une couche entre les deux afin de fournir des services directement à la périphérie du réseau (Edge network). L'informatique dans le brouillard (the fog) est un scénario dans lequel un grand nombre de dispositifs hétérogènes (sans fil et parfois autonomes), ubiquitaires et décentralisés communiquent et coopèrent potentiellement entre eux et avec le réseau pour effectuer des tâches de stockage et de traitement sans l'intervention de tierces parties. Il s'agit d'une approche prometteuse pour le traitement de l'énorme volume de données produites et consommées par les applications, les machines et leurs utilisateurs. Le traitement complexe et l'informatique distribuée sont placés respectivement dans les nuages (cloud) et sur les appareils qui nous entourent. Le brouillard (the fog) est une technologie alternative au nuage qui vise à répondre aux exigences des utilisateurs en matière de sécurité, de données, d'agilité, de latence et d'efficacité (SCALE-Security, Cognition, Agility, Latency, Efficiency).

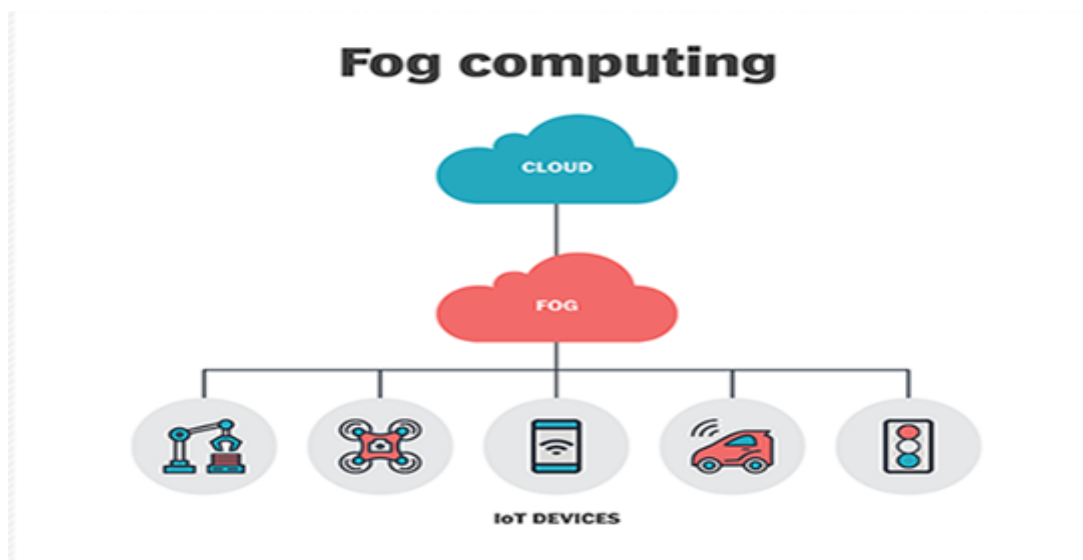


Figure1 : Présentation du Fog Computing

Il n'existe pas d'architecture standard pour Fog Computing. La figure 2 présente une vue d'ensemble de la structure de Fog Computing. Elle se compose de trois couches principales : le nuage, le brouillard et les dispositifs finaux.

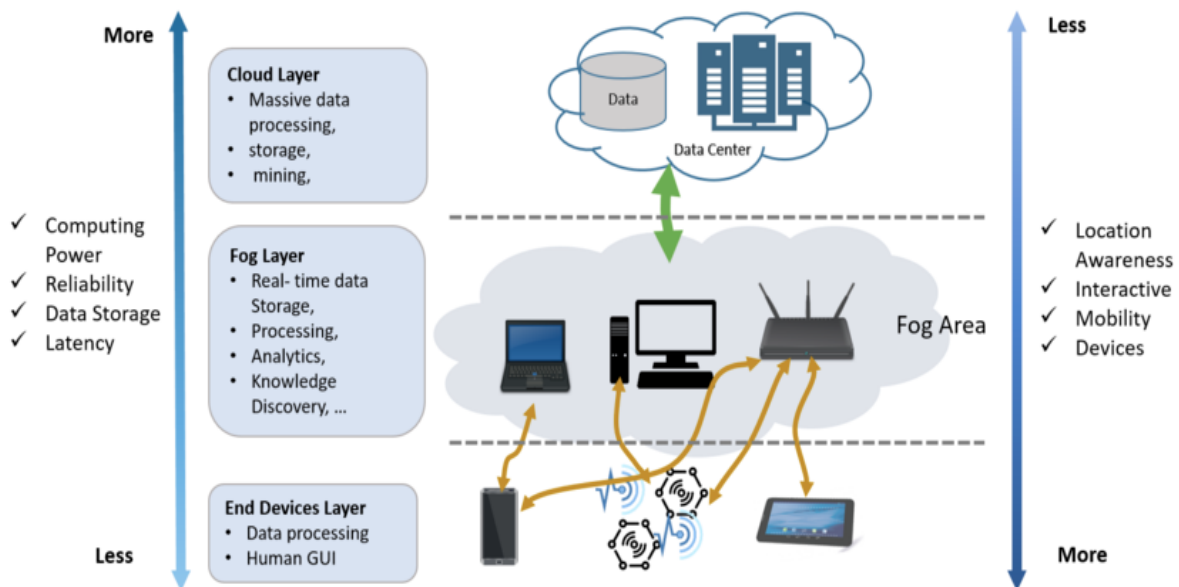


Figure2 : Architecture technique du Fog Computing

1.2 Différents simulateurs de Fog Computing

Selon l'article **“Modeling and Simulation Tools for Fog Computing—A Comprehensive Survey from a Cost Perspective”**, on trouve un ensemble d’outils différents illustré par le tableau1

Year	Simulator	Implementation Technologies	Metrics	Objective	Citations
2016	Edge-Fog [23]	Python		Distribute task processing on the participating cloud resources	73
2017	FogTorch [64]	Java	QoS, reliability of links and nodes, power consumption, security, monetary costs	Find eligible deployments of an application over a fog infrastructure	173
2017	FogTorchII [65]	Extension of FogTorch	Resource utilization and QoS accuracy	Same as FogTorch and many QoS profile according to a probability distribution	51
2017	iFogSim [35]	Extension of CloudSim, Java, JSON	Energy consumption, network congestion, and operational costs	Performance of resource management policies	509
2017	MyiFogSim [34]	Extension of iFogSim	Latency	Resource allocation	38
2018	FogDirSim [70]	RESTful API, Python	Performce in terms of uptime, energy consumption, resource usage	Compare application management and infrastructure management policies	4
2018	FogNetSim++ [63]	based on OMNeT++	Energy module, scheduling algorithms, pricing model	General simulation of fog environments	25
2019	FogWorkFlowSim [72]	Java	Performance (time, energy and processing cost	Evaluate resource and task management strategies	
2019	OPNET [73]	Visual studio, based on OPNET	Processing time	Cope with the massive amount of confidential and security-sensitive data	3
2019	YAFS [68]	Python, JSON	Energy models, network utilization, response time, network delay	Analyze the design and deployment of applications	7
2019	FogDirMime [69]	Python		Support FogDirector [77]	9
2019	Fogbus [71]		Latency, energy, network and CPU usage		27
2020	MobFogSim [5]	Extends iFogSim	Same as iFogSim	Evaluate application behaviour and performance	0

Tableau1 : Outils de simulation pour Fog Computing

1.3 Les challenges rencontrés par Fog Computing

La figure 3 illustre les différents challenges que Fog Computing rencontre vis-à-vis ces trois composantes.

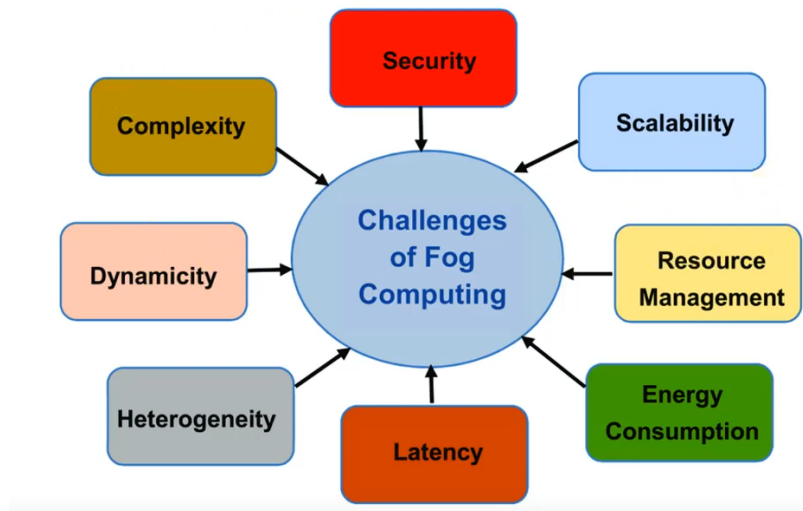


Figure 3: Les challenges pour Fog Computing

2. Travail Demandé

2.1 Première manipulation

Nous utiliserons pour notre cas d'utilisation le simulateur FogNetSim++.

Nous prendrons l'architecture de la figure 4 comme un plan de travail où on trouve 4 centres de Fog reliés par « base broker » afin d'assurer la communication entre eux

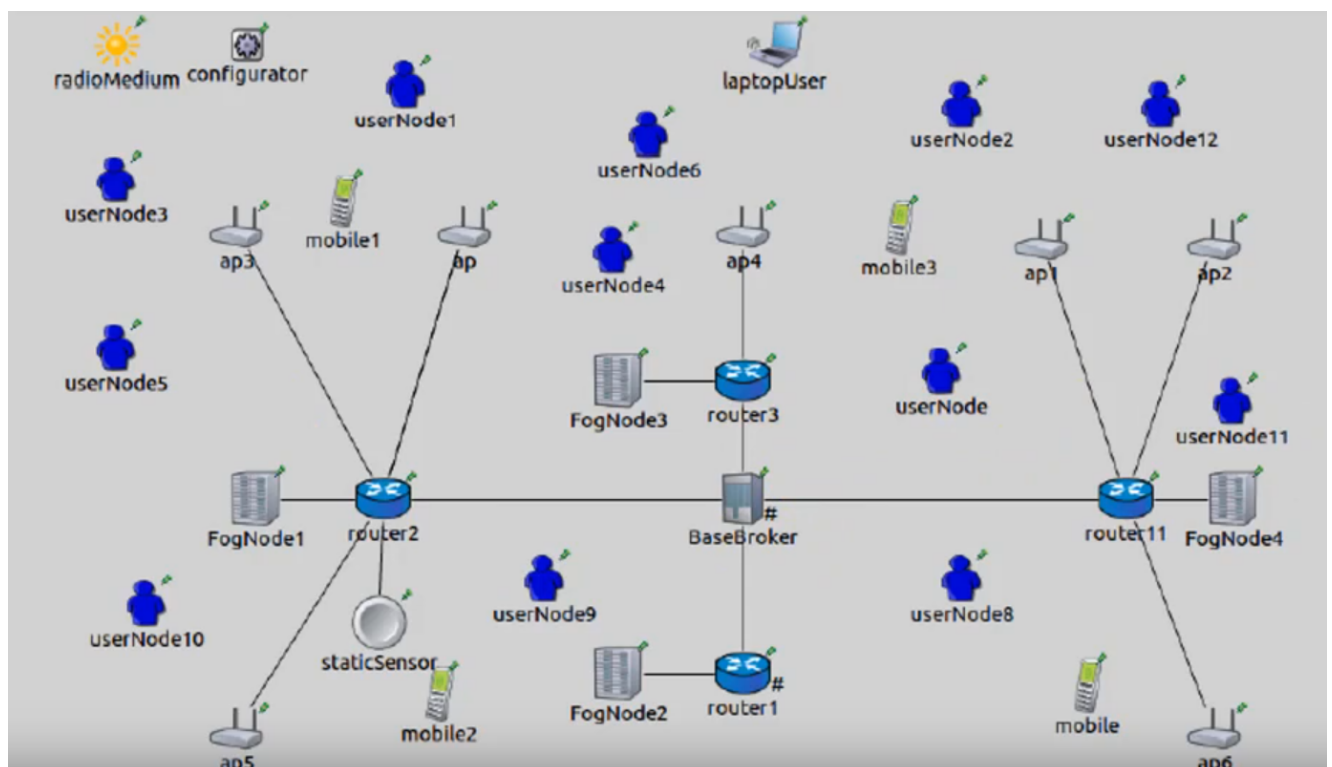


Figure4 : Architecture à simuler

Nous distinguons 2 types d'utilisateurs : UserNode et Mobile Node qui accède au réseau via des points d'accès. Le fog computing propose une multitude de services. Le choix des services à utiliser est laissé ouvert.

La figure5 présente une visibilité par rapport au composant du noeud broker.

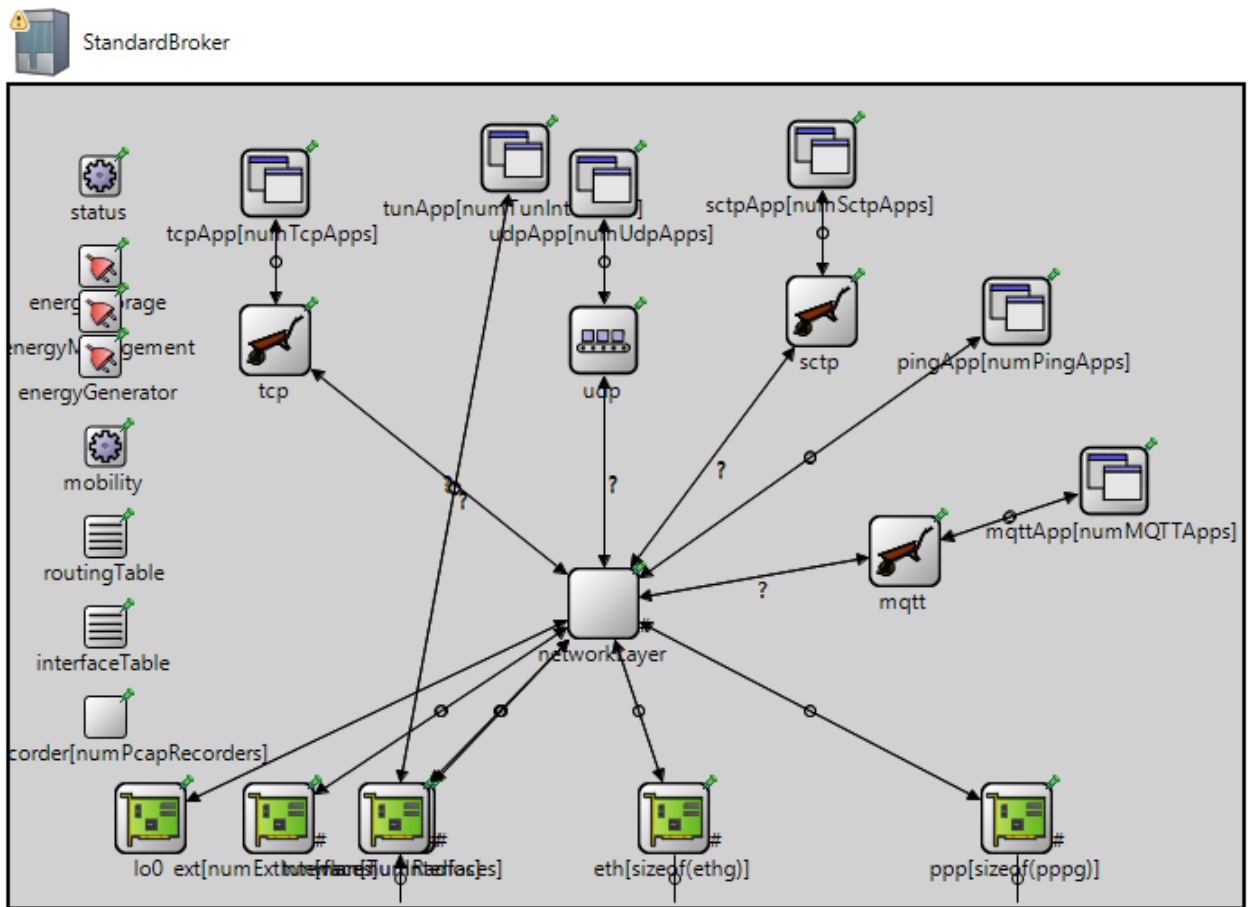


Figure 5 : Vue modulaire interne du nœud de Broker

On vous demande de :

- Choisir les services à simuler
- Simuler l'architecture de la figure 4 tout en prenant en considération que la communication capteur-Fog est unidirectionnel alors que les mobiles doivent avoir des communication half-duplex ou full-duplex
- Tracer des courbes montrant la performance de l'architecture réseau simulée (task computation dans le temps, latence moyenne de point de l'utilisateur et l'opérateur, consommation moyenne de l'énergie ...)

2.2 Second manipulation

Deux cas de figure peuvent avoir lieu pour engendrer un Handover

- Si un nœud mobile ayant un modèle de mobilité donné change de point d'accès mais reste sous le contrôle du même routeur.
- Si un nœud mobile ayant un modèle de mobilité donné change de point d'accès mais change de routeur de contrôle.

On vous demande de :

- 1) Simuler les deux cas d'utilisation et étudier l'impact de ce type de handover sur un grand nombre de nœuds mobiles (taux d'erreur de point de vue utilisateur, les latences...)

2) FogNetSim++ supporte plusieurs types de modèles de mobilités illustré par la figure6

- a random waypoint,
- Mass mobility,
- Gauss Markov,
- Chaing mobility,
- CircleMobility,
- LinearMobility and
- vehicle mobility.

Figure6 : Les mobilités supportés par FogNetSim++

Tester les différents modèles et étudier l'impact de chacun d'entre eux sur les performances de réseaux.

3) FogNetSim++ suggère différents modèles de pricing. Le broker central supervise les différentes ressources et ce dernier est responsable de gérer les SLA(Service Level Agreement) des utilisateurs (voir exemple tableau 2).

No.	Pricing Model	Features/ Description	Price
1.	Pay-as-you-go	Networking, Storage Compute	\$.0004/Mb
2.	Subscription	Monthly Task size 10Gb Monthly Task size 100Gb	\$50/User \$450/User
3.	Pay-for-resources	Storage Compute	\$0.0048/GB \$.0002/Mb
4.	Hybrid Model	A dynamic model changes according to the queue size of broker	\$.0004-\$.0008

Tableau2 : Les modèles de Pricing suggérés par FogNetSim++

On vous demande d'intégrer les différents contrôles de SLA dans votre simulation.

3. Références

https://www.researchgate.net/publication/328484070_FogNetSim_A_Toolkit_for_Modeling_and_Simulation_of_Distributed_Fog_Environment
<https://github.com/rtqayyum/fognetsimpp>