

Simulation, Emulation et Virtualisation

Evaluation de performance et analyse des résultats

Naceur.Malouch@lip6.fr

LIP6, Université de Paris 6

Master 1^e : Réseaux Informatiques

Plan

- > Observation des événements avec OMNET++
- > Procédure globale d'évaluation
- > Récupération des résultats avec OMNET++
- > Intervalle de confiance
 - **Quel crédit accorder aux résultats ?**

Observation des événements

> Une fois le code de la simulation est terminé

On peut observer la simulation :

- Animation graphique montrant l'évolution dans le temps de certains objets de la simulation
 - Vérification visuelle que le code implante a priori le comportement souhaité du système
 - Compréhension et observation du système
 - Certains événements peuvent être mis en avant
- Le déroulement des événements et leur instant d'occurrence
 - Permet de déboguer les événements (plus précis, automatisation possible)

Observation des événements

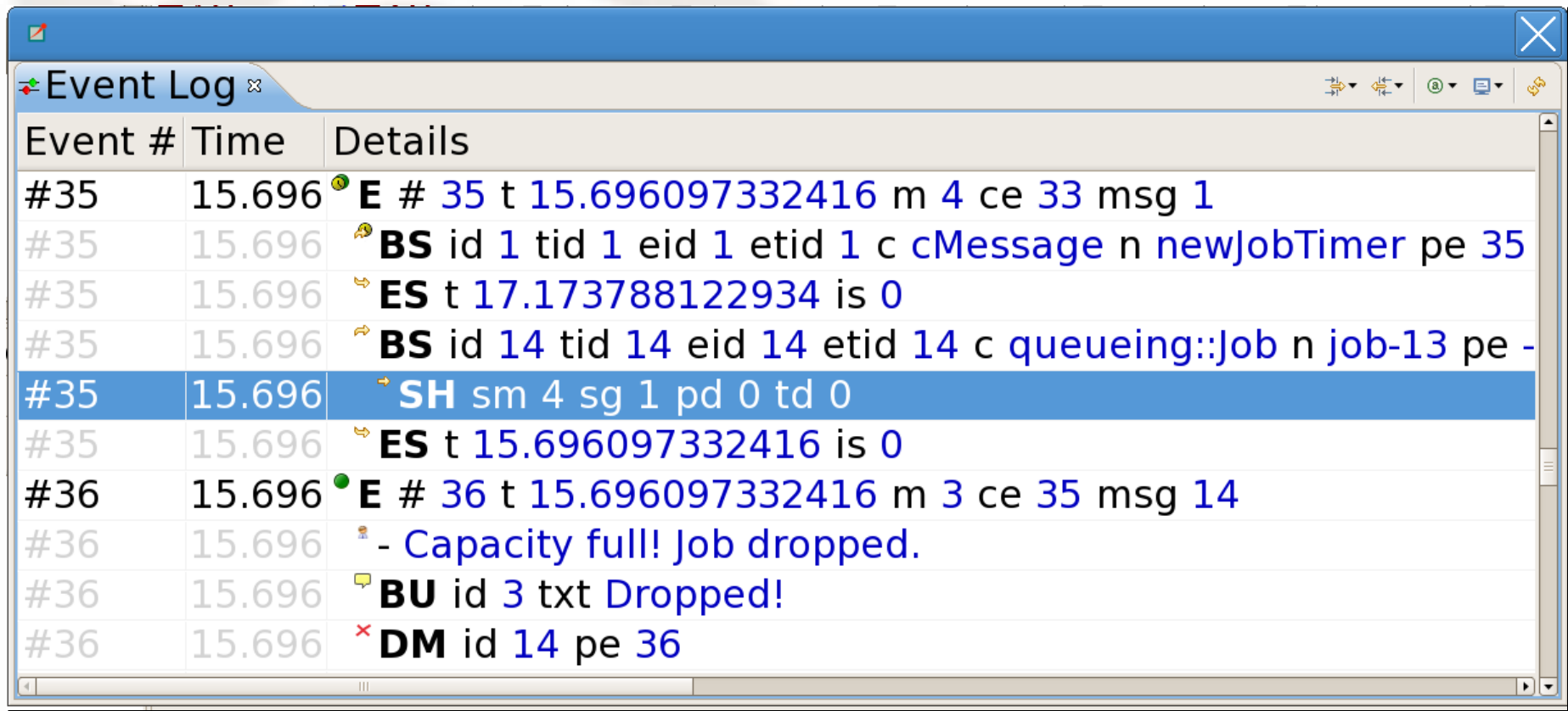
- > Log des événements et/ou le graphe de séquence (sequence chart)
 - Si dans le fichier .ini : **record-eventlog = true**
On obtient un fichier .elog avec tous les événements

```
E # 35 t 15.696097332416 m 4 ce 33 msg 1
BS id 1 tid 1 eid 1 etid 1 c cMessage n newJobTimer pe 35
ES t 17.173788122934
BS id 14 tid 14 eid 14 etid 14 c queueing::Job n job-13
SH sm 4 sg 1
ES t 15.696097332416











E # 36 t 15.696097332416 m 3 ce 35 msg 14
- Capacity full! Job dropped.
BU id 3 txt Dropped!
DM id 14 pe 36
```

Observation des événements

> EventLog Window : Raw format (IDE)

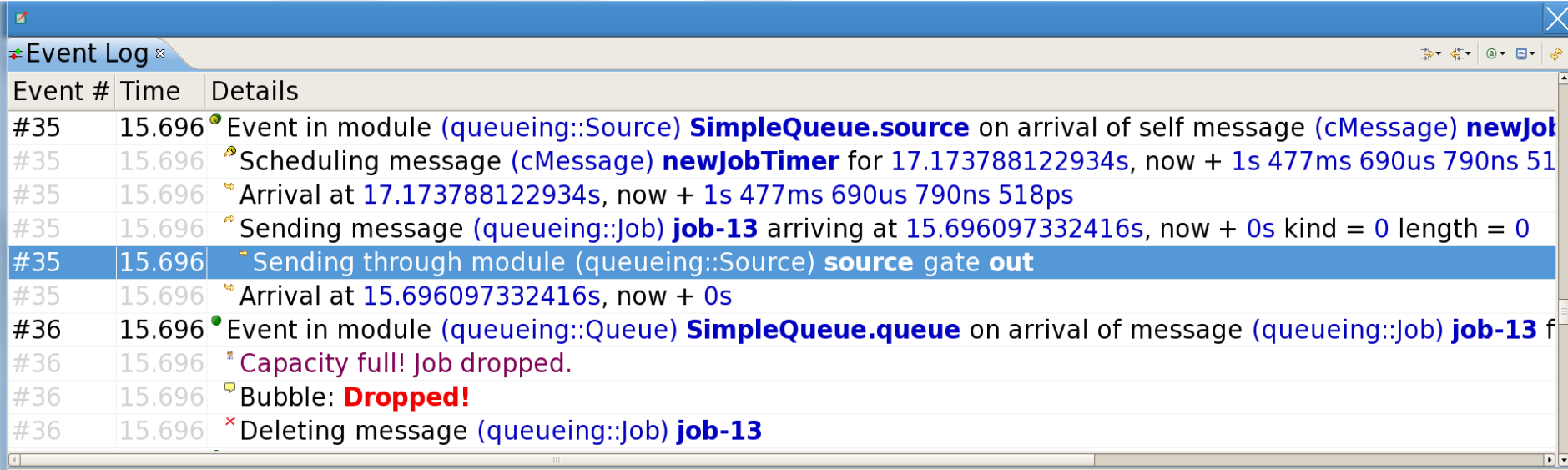


The screenshot shows a window titled "Event Log" with a tab icon and a close button. The window contains a table with three columns: "Event #", "Time", and "Details". The table lists several events, with the fifth event (Event #35) highlighted in blue. The events include various status codes like E, BS, ES, SH, and DM, along with timestamps and details such as message IDs, capacities, and job names.

Event #	Time	Details
#35	15.696	 E # 35 t 15.696097332416 m 4 ce 33 msg 1
#35	15.696	 BS id 1 tid 1 eid 1 etid 1 c cMessage n newJobTimer pe 35
#35	15.696	 ES t 17.173788122934 is 0
#35	15.696	 BS id 14 tid 14 eid 14 etid 14 c queueing::Job n job-13 pe -
#35	15.696	 SH sm 4 sg 1 pd 0 td 0
#35	15.696	 ES t 15.696097332416 is 0
#36	15.696	 E # 36 t 15.696097332416 m 3 ce 35 msg 14
#36	15.696	 - Capacity full! Job dropped.
#36	15.696	 BU id 3 txt Dropped!
#36	15.696	 DM id 14 pe 36

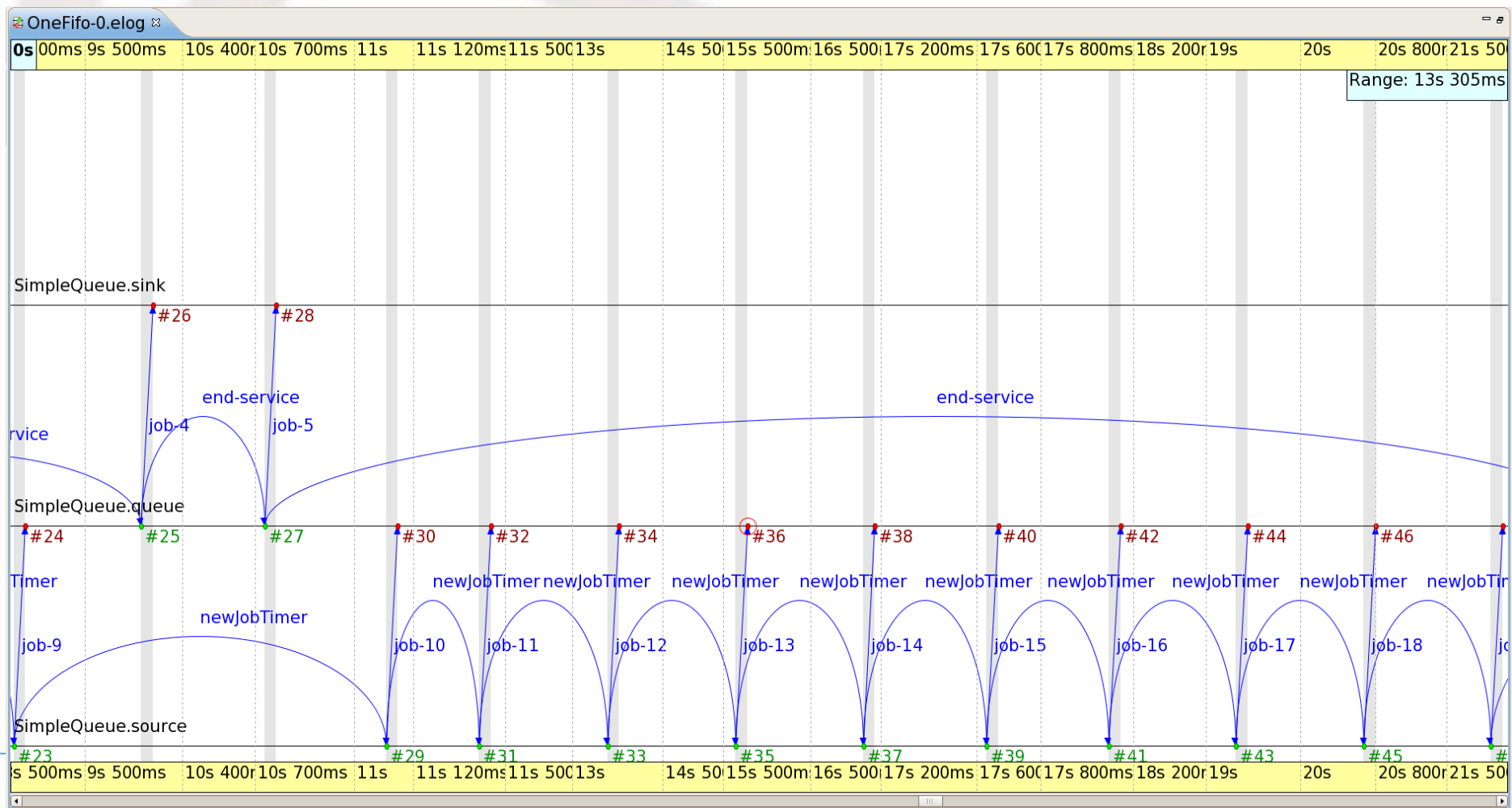
Observation des événements

> EventLog Window : Descriptive format (IDE)



Event #	Time	Details
#35	15.696	Event in module (queueing::Source) SimpleQueue.source on arrival of self message (cMessage) newJob
#35	15.696	Scheduling message (cMessage) newJobTimer for 17.173788122934s, now + 1s 477ms 690us 790ns 518ps
#35	15.696	Arrival at 17.173788122934s, now + 1s 477ms 690us 790ns 518ps
#35	15.696	Sending message (queueing::Job) job-13 arriving at 15.696097332416s, now + 0s kind = 0 length = 0
#35	15.696	Sending through module (queueing::Source) source gate out
#35	15.696	Arrival at 15.696097332416s, now + 0s
#36	15.696	Event in module (queueing::Queue) SimpleQueue.queue on arrival of message (queueing::Job) job-13 f
#36	15.696	Capacity full! Job dropped.
#36	15.696	Bubble: Dropped!
#36	15.696	Deleting message (queueing::Job) job-13

> Sequence Chart (IDE)



Observation des événements

> Une fois le code de la simulation est terminé

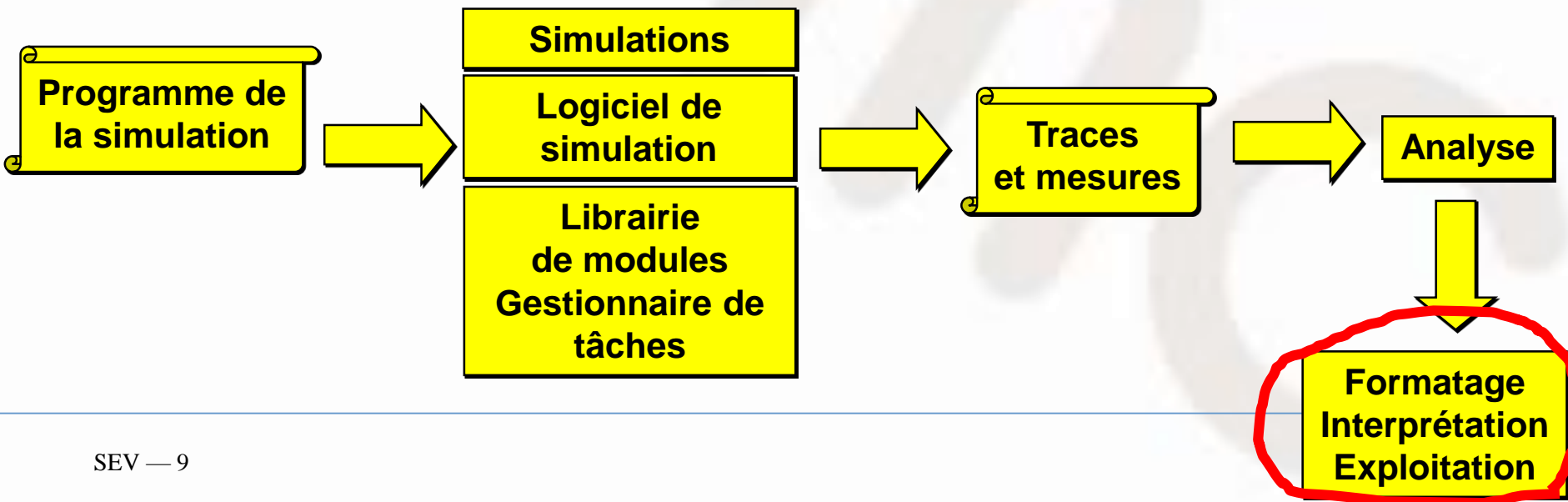
On peut observer la simulation :

- Animation graphique montrant l'évolution dans le temps de certains objets de la simulation
- Le déroulement des événements et leur instant d'occurrence

> **Insuffisant pour l'évaluation de performances**

Rappel : Procédure globale d'exécution et d'analyse

- > Développement du code qui simule le système
- > Configuration : modules, topologie, paramètres de simulation
- > Identification (Définition) des paramètres à mesurer pendant et/ou à la fin de la simulation
- > Lancement des simulations
- > Analyse de traces et récupération des résultats



Rappel : Procédure globale d'exécution et d'analyse

- > Procédure similaire pour l'émulation et pour l'expérimentation (testbed)

Obtention des mesures de performances

- > Plusieurs méthodes
- > Dans OMNET++ : Il y a notamment le collecteur de statistiques ou la récupération par signal.
 - → fichier .sca

```
scalar SimpleQueue.queue "min length" 0
scalar SimpleQueue.queue "max length" 5
scalar SimpleQueue.queue "avg length" 2.9
scalar SimpleQueue.queue utilization 0
scalar SimpleQueue.queue "dropped jobs" 8
scalar SimpleQueue.source "jobs created" 21
```

Obtention des mesures de performances

> → fichier .vec

- Plusieurs valeurs enregistrées durant la simulation pour chaque paramètre :

Vectors (10 / 10)

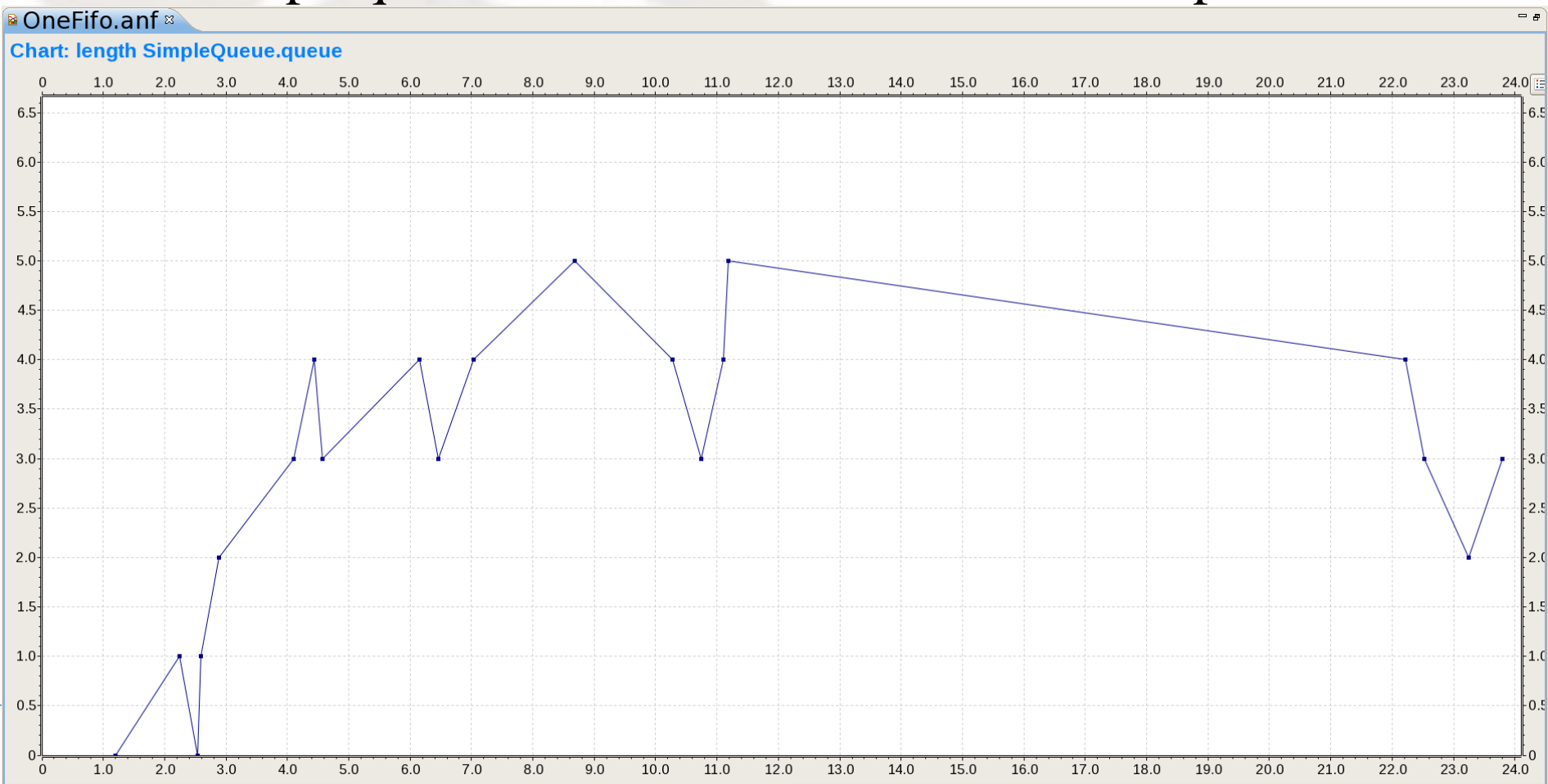
file(/resultfiles/OneFifo-0.vec)

Folk	File na	Conf	F	Run id	Module	Name	Count	Mean
/res	OneFif	OneF	C	OneFif	SimpleQue	queueing time	9	7.159673434609666
/res	OneFif	OneF	C	OneFif	SimpleQue	length	20	2.9
/res	OneFif	OneF	C	OneFif	SimpleQue	total lifetime	8	8.98980811960425
/res	OneFif	OneF	C	OneFif	SimpleQue	total queueing time	8	6.234554544809375
/res	OneFif	OneF	C	OneFif	SimpleQue	number of queue noc	8	0.0
/res	OneFif	OneF	C	OneFif	SimpleQue	total service time	8	2.755253574794875
/res	OneFif	OneF	C	OneFif	SimpleQue	total delay	8	0.0
/res	OneFif	OneF	C	OneFif	SimpleQue	number of delay nod	8	0.0
/res	OneFif	OneF	C	OneFif	SimpleQue	generation	8	0.0
/res	OneFif	OneF	C	OneFif	SimpleQue	dropped jobs	8	4.5

Obtention des mesures de performances

> .vec → .anf (IDE)

- Graphiques : Evolution en fonctions du temps



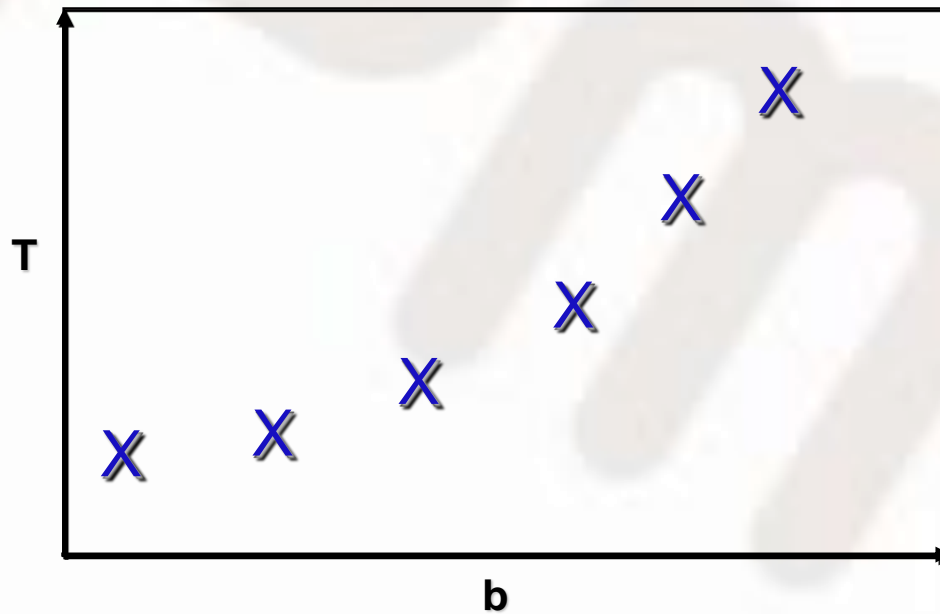
Obtention des mesures de performances

- > On peut exporter les données dans un autre format.
 - Utilisation d'autres logiciels
- > On s'intéresse à :
 - L'évolution en fonction du temps
 - La distribution (pdf ou cdf)
 - Le résultat final après chaque simulation en variant un ou plusieurs paramètres d'entrée du système, puis l'évolution des valeurs de performances en fonction des paramètres d'entrée

Exemple : Délai moyen d'attente en fonction du débit d'arrivée des paquets

Obtention des mesures de performances

- > Exemple : Délai moyen d'attente en fonction du débit d'arrivée des paquets



Outils de création de graphes

- > gnuplot : permet de tracer rapidement des graphes, plus d'options

```
% gnuplot
gnuplot> set output "image.jpg"
gnuplot> set terminal jpeg
gnuplot> plot "fichier" using 1:2 with lines
```

- > "fichier" contient plusieurs colonnes
- > Autre logiciel : xgraph, matlab, maple, R, octave, etc.

Paramètres de performance à mesurer

- > Les paramètres de performance à mesurer dépendent du système à étudier et notamment des services fournis par ce système.
- > Souvent en réseau :
 - Délai d'attente des paquets à travers un ou plusieurs équipements réseaux
 - Nombre de paquets dans les buffers associés aux interfaces de sortie
 - Taux de perte des paquets
 - Débit efficace (= “goodput” = débit de sortie = débit applicatif)
 - Taux d'utilisation de liens
 - La gigue (= “jitter” = taux de variation de délais)
 - BER (“Bit Error Rate”)
 - Taux d'interférence ou collisions (réseau sans fil)
 - Surcoût (“overhead”) engendré par un protocole
 - etc.

Intervalle de confiance : Très important pour l'analyse des simulations

Intervalle de confiance

Intervalle de confiance

- > Soient n observations (mesures) obtenues par simulations :
 - X_1, X_2, \dots, X_n du paramètre de performance X
 - Le paramètre mesuré peut être considéré comme une variable aléatoire
- > Soit μ la moyenne **réelle** de X
- > Soit σ^2 la variance **réelle** de X (σ est l'écart-type)
- > **Supposons que les X_i s sont des v.a. i.i.d, alors :**

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$$

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2 .$$

- > (\bar{X}_n, S_n) est un estimateur non-biaisé et consistant de (μ, σ)
- > Propriétés :

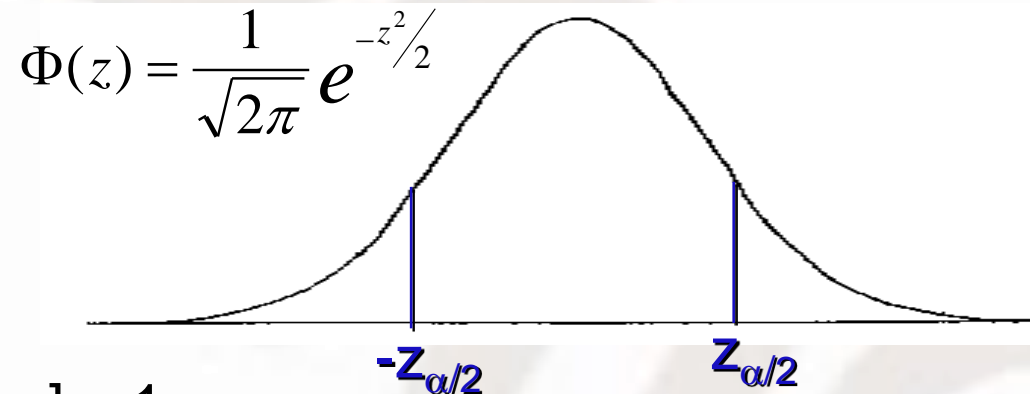
$$E[\bar{X}_n] = \mu, \quad E[S_n^2] = \sigma^2$$

$$Var(\bar{X}_n) = \frac{\sigma^2}{n}, \quad Var(S_n^2) = \frac{2\sigma^4}{n-1}$$

Intervalle de confiance

> Soit $Z = \frac{\bar{X}_n - \mu}{\sigma / \sqrt{n}}$

> D'après le théorème Centrale Limite, quand n tends vers l'infini alors Z tends vers une v.a. de distribution normale de densité :



> $P[-z_{\alpha/2} \leq Z \leq z_{\alpha/2}] = 1 - \alpha$

> $\Rightarrow P[\bar{X}_n - \beta \leq \mu \leq \bar{X}_n + \beta] = 1 - \alpha$, avec $\beta = \frac{z_{\alpha/2} \sigma}{\sqrt{n}}$

> La valeur moyenne réelle se trouve dans l'intervalle de confiance $[\bar{X}_n - \beta, \bar{X}_n + \beta]$ avec une probabilité $1 - \alpha$.

Intervalle de confiance

- > **Mais généralement on ne connaît pas la valeur réelle de σ !!**
 - Dans ce cas on utilise S_n au lieu de σ .
- > **Et/ou Le nombre de simulation n peut être faible**
- > **→ On ne peut plus multiplier par $z_{\alpha/2}$, car Z ne tends plus vers la distribution normale. On doit utiliser *student* appelée aussi t .**
- > La distribution *student* est définie par la probabilité α et un autre paramètre appelé “degré de liberté”.

...

> **Résultat** : Intervalle de confiance $\beta = t(\alpha, n-1) \times S_n / \sqrt{n}$

> **Exemple** :

t à 95% pour 50 mesures = 2.0096

→ Intervalle de confiance : $\beta = 2.0096 \times S_n / \sqrt{n}$

Intervalle de confiance

Un outil pour obtenir les $t(\alpha, n-1)$:
<http://statpages.org/pdfs.html>

n -1	90% ($\alpha=0.1$)	95% ($\alpha=0.05$)
4	2.13	2.78
5	2.01	2.57
7	1.89	2.36
8	1.86	2.31
10	1.81	2.23
15	1.75	2.13
20	1.72	2.09
25	1.71	2.06

Intervalle de confiance

- > Comment obtenir des mesures i.i.d :
- > Plusieurs méthodes existent :
 - Méthode par répliques indépendantes :
 - Effectuer n simulations indépendantes de m observations chacune
 - A chaque simulation, on change la graine (seed) du générateur de nombres aléatoires afin d'obtenir une série d'événements différentes.
 - La moyenne sur les m observations de chaque simulation joue le rôle d'un X_i
 - Méthode par lots
 - Découper la simulation en lots de manière arbitraire
 - Il faut vérifier que les lots sont indépendants, pas évident.

Intervalle de confiance

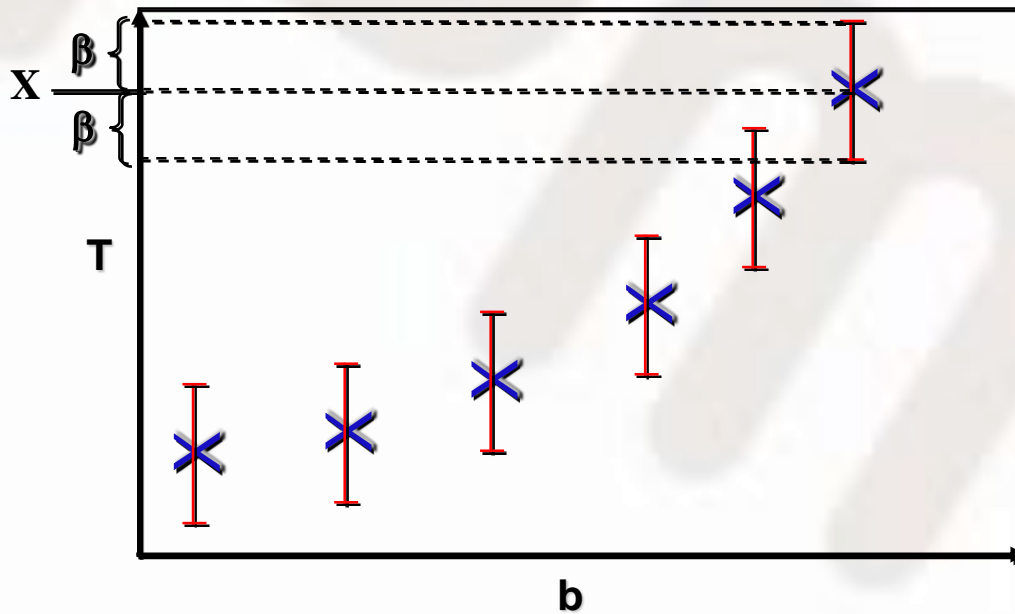
> En pratique :

- Chaque simulation donne lieu à une seule observation mesurée à la fin de la simulation, généralement c'est une moyenne
- La durée de simulation est assez longue afin de générer un nombre élevé d'événements pour la mesure
- On effectue n simulations avec n suffisamment grand pour avoir un intervalle de confiance assez étroit pour une valeur de α donnée.

Intervalle de confiance

> Représentation graphique :

Exemple : Figure représentant l'impact d'un paramètre de contrôle b , sur la valeur moyenne du paramètre T



Interprétation et exploitation des résultats

- > **C'est l'objectif ultime de la simulation (ou émulation, etc.)**
- > Quelles propriétés les résultats mettent en évidence ?
- > Les résultats montrent-ils un comportement adéquat du système ?
- > Quels problèmes les résultats mettent en évidence ?
- > Quels sont les causes des problèmes découverts ?
- > Quels sont les implications sur le système étudié ?
 - Faut-il modifier certains composants du système ?
 - Comment améliorer le fonctionnement du système ?
- > Le système répond-t-il aux besoins ? Fournit-il les services avec la qualité requise ?

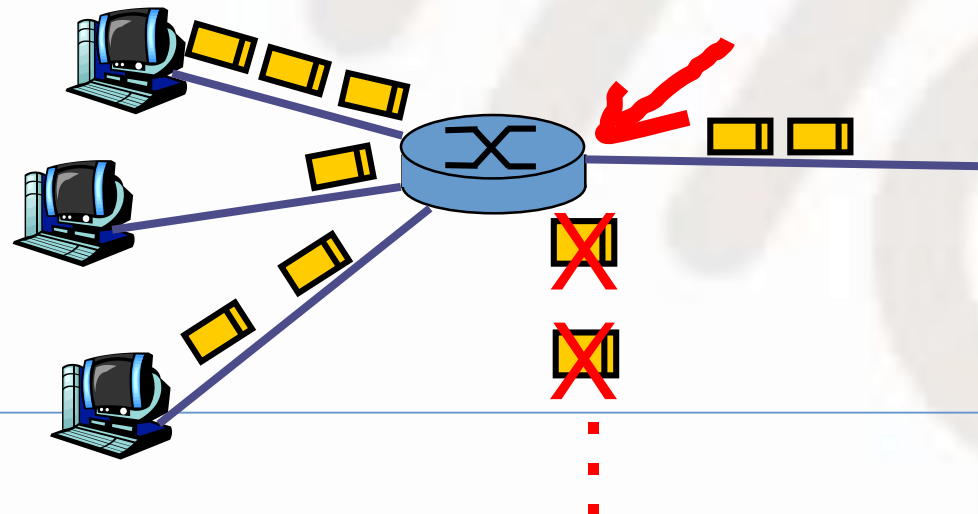
> . . .

Interprétation et exploitation des résultats

> Exemples :

> Taux de perte très élevé

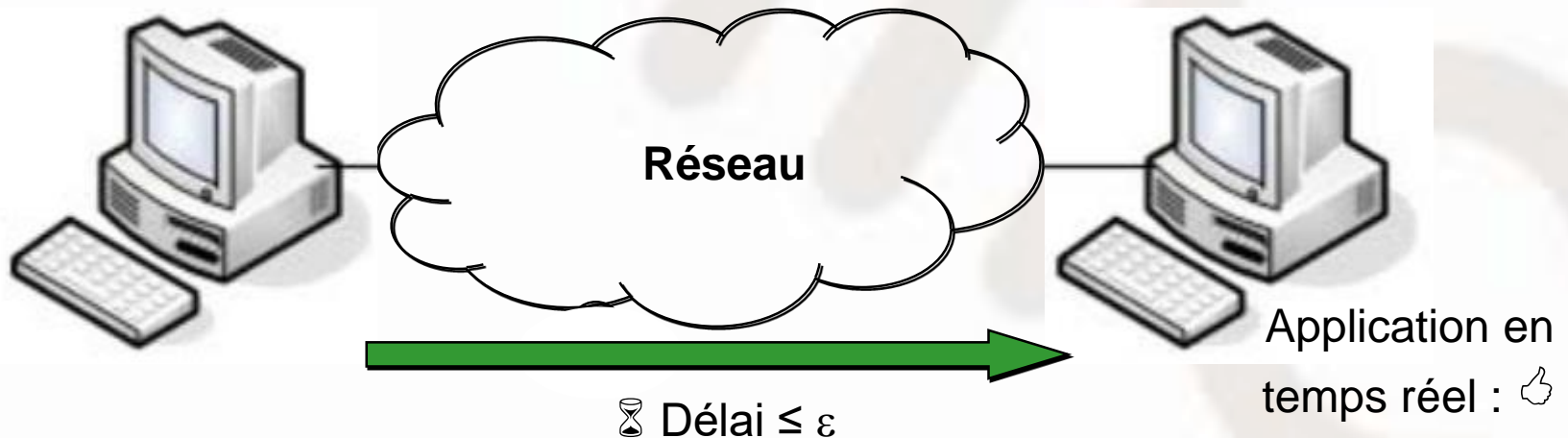
- Cause : Débit d'arrivée des paquets est largement supérieur à la capacité de transmission de ces paquets.
- Implication : Il faut augmenter la capacité du lien de transmission ou limiter le débit d'envoi de paquets



Interprétation et exploitation des résultats

> Exemples :

- > Délai moyen à travers le réseau est très proche de zéro
 - Comportement adéquat
 - Le réseau n'est pas surchargé
 - Les capacités de transmission sont bien dimensionnées



END

BACKUP

Intervalle de confiance

> En pratique :

- Chaque simulation donne lieu à une seule observation mesurée à la fin de la simulation, généralement c'est une moyenne
- La durée de simulation est assez longue afin de générer un nombre élevé d'événements pour la mesure
- On effectue n simulations avec n suffisamment grand pour avoir un intervalle de confiance assez étroit pour une valeur de α donnée.

> Problème : Si après n simulations, l'intervalle obtenu est grand, combien faut-il rajouter pour espérer avoir un intervalle de confiance plus petit

> Une solution :

– La formule approximative : $n_2 \approx n \times \left(\frac{\beta}{\beta_2} \right)^2$