



heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Mesure de trafic, systèmes à attentes et à pertes

Trafic

Laboratoire de télécommunications et de téléinformatique pour les étudiants
des filières ET (Réseaux et Services).



Auteurs : Markus Jatton, Christian Roubaty

1.0 Buts du laboratoire

Ce laboratoire a pour objectif d'illustrer la théorie du télétrafic vue au cours de télécommunications par des exemples tirés de la pratique d'un grand réseau de télécommunications public d'une part, et de la simulation d'autre part.

2.0 Introduction

2.1 Rappels

La planification des ressources dans un réseau, téléphonique ou téléinformatique, se base sur la théorie du télétrafic, due principalement au mathématicien danois Erlang, sur la simulation par ordinateur, ainsi que sur l'analyse et l'interprétation de données de mesure de trafic. Ce sont ces trois techniques, alliées aussi à des considérations sociologiques et économiques, qui mises en commun, permettent une prévision réaliste des besoins futurs des usagers. Cette prévision permet à son tour de prédire les investissements nécessaires, ainsi que la mise sur pied d'un plan financier assurant (dans le meilleur des cas) la rentabilité du réseau de télécommunications, tout en fournissant une qualité de service satisfaisante pour les utilisateurs.

Le présent laboratoire propose l'utilisation de ces trois techniques dans divers cas de figure directement tirés de la réalité d'un grand réseau de télécommunications public.

On se référera au cours de télécommunications, partie “Réseaux” pour les notions de base, ainsi qu’aux quelques notions rappelées ou introduites dans la donnée de ce laboratoire.

2.2 Quelques notions

Une “valeur de trafic”, en télécommunications, se compose généralement de deux composantes : Valeur de trafic et instant d’apparition. La première composante est la plus intéressante, mais la seconde est parfois utilisée dans des applications spéciales, telle que l’identification de crêtes de trafic spécifiques à une certaine occupation (jeu radiophonique, par exemple).

La valeur de trafic est elle-même le produit de deux quantités : le taux de sollicitation (en s^{-1}), et la durée d’occupation (en s). Le trafic est donc sans unité. Toutefois, pour des raisons de commodité, on lui donne le nom d’Erlang (E).

Le trafic mesure l’occupation d’un média. Ainsi, une personne téléphonant de manière ininterrompue au cours de l’intervalle de temps d’observation (qui est théoriquement infini, mais bon l’infini, hein...) génère un trafic de 1 Erlang. En pratique, une personne “normale” génère un trafic de 0.05 Erlang au cours de l’“heure chargée”, qui se situe quelque part entre 9 et 11 heures dans nos pays occidentaux.

2.2.1 Valeur instantanée

On appelle valeur instantanée la valeur du trafic mesurée (moyennée) sur un intervalle de temps prédéfini. Dans le cadre de ce laboratoire, on utilisera en général comme intervalle de temps 15 minutes (900 secondes). A noter qu’il s’agit là d’une valeur communément utilisée dans les réseaux publics, pour les systèmes à pertes. Pour des systèmes à attente, comme c’est le temps que l’on mesure, l’intervalle de temps considéré est forcément beaucoup plus petit.

A noter que le trafic, dans les réseaux publics de télécommunications, est mesuré de diverses manières. Certains centraux mesurent non pas le trafic, mais la quantité de trafic (*Traffic quantity*, *Verkehrsmenge*). L’unité en est alors des Erlang * secondes. Pour retrouver des Erlang, il suffit bien sûr de diviser cette quantité par l’intervalle d’observation en secondes.

2.2.2 Valeur horaire

On définit une valeur horaire comme la moyenne sur quatre intervalles consécutifs d'un quart d'heure de la valeur du trafic en Erlang. Ainsi, si l'on dispose des valeurs instantanées suivantes :

1h00 : 10 E

1h15 : 12 E

1h30 : 14 E

1h45 : 12 E

La valeur horaire pour 01h00 sera $(10 + 12 + 14 + 12) / 4 = 12$ E. Il faut noter, même si cela peut paraître paradoxal, qu'il y a également une valeur horaire définie pour 1h15, 1h30, etc... Il y a donc quatre valeurs horaires par heure. (Dans quel but, au fait ?).

2.2.3 Heure chargée (Busy Hour, Hauptverkehrsstunde)

On appelle heure chargée l'heure de la journée pour laquelle on a la plus grande valeur horaire. Cette valeur horaire est appelée "Valeur du trafic à l'heure chargée".

2.2.4 Valeur journalière (Daily value, Tageswert)

On appelle valeur journalière la plus grande valeur de trafic à l'heure chargée constatée sur une semaine de cinq jours (week-end exclus).

2.2.5 Valeur hebdomadaire (Weekly value, Wochenwert)

La valeur hebdomadaire se calcule en faisant préalablement la moyenne de toutes les valeurs instantanées sur une semaine de cinq jours, de manière à obtenir une sorte de "jour moyen", puis en calculant l'heure chargée de ce jour moyen. La valeur obtenue correspond à la valeur hebdomadaire pour la semaine considérée. Si on ne dispose pas de cinq jours, on peut se contenter de 4, voire de trois jours, mais en aucun cas de moins.

2.2.6 Valeur mensuelle (Monthly value, Monatswert)

La valeur mensuelle est définie comme la moyenne des deux plus grandes valeurs hebdomadaires sur un mois.

2.2.7 Diagramme journalier (Daily traffic chart, Tagesverlaufsdigramm)

Le diagramme journalier consiste à représenter, pour chaque instant du jour, la valeur horaire correspondante sur un diagramme Trafic = fonction(Temps).

2.2.8 Autres valeurs

On définit également la valeur annuelle (*yearly value*, *Jahreswert*) comme la plus grande valeur moyenne de trois mois (pas forcément consécutifs) dans le courant de l'année. Dans les cas de forte dispersion de trafic (trafic saisonnier) on parle volontiers de valeur saisonnière (*season traffic*, *Saisonverkehr*), qui est la plus grande valeur mensuelle constatée. Dans le cas de zones touristiques, on définit également des valeurs de fin de semaine, mesurées sur le samedi et le dimanche, alors que ces données ne sont généralement pas mesurées dans les autres cas.

On définit encore d'autres grandeurs dans le cas de serveurs internationaux, ou dans des réseaux à routage dynamique, comme c'est le cas du réseau téléphonique aux Etats-Unis, mais ces grandeurs ne nous sont pas utiles dans le cadre de cette manipulation.

3.0 Questions préalables

3.1 Technique de mesure

Imaginer une méthode permettant de mesurer la valeur instantanée du trafic sur un ensemble de serveurs pendant un intervalle de temps déterminé. Proposer l'application pratique de cette méthode dans l'un ou l'autre des cas suivants:

- Faisceau de N lignes
- Routine d'interruption d'un processus informatique
- Groupe de M abonnés
- L'ensemble des guichets d'une grande banque

3.2 Pondérations

Pourquoi utilise-t-on de préférence la valeur horaire plutôt que la valeur instantanée ? Donner un exemple pratique où cette pondération se justifie.

3.3 Relation mesure-théorie

Lorsque l'on évalue le trafic et les pertes par des moyens mathématiques, on suppose le trafic stationnaire, ce qu'il n'est visiblement pas, comme le montrent les mesures de trafic sur une journée. Quelle valeur de trafic mesurée doit-on mettre en relation avec la valeur utilisée dans les calculs ?

4.0 Outils à disposition pour la manipulation

4.1 La classe Erlang

La classe Erlang est une classe Java très simple permettant de calculer la relation d'Erlang $E_n(A)$. Pour ceux qui ont besoin de cet outil, on se référera rapidement au paragraphe (Section 6.1, page 19) pour le paramétrage éventuel de la machine virtuelle Java.

Ce paramétrage ayant été effectué (si nécessaire), on utilise l'outil Erlang comme indiqué à la Figure 1, page 6. Le répertoire où se trouvent les classes Java ne correspond probablement pas au répertoire indiqué dans l'exemple !

On remarquera que la ligne de commande comprend l'invocation préalable de la machine virtuelle java, suivi du nom de la classe (sans l'extension `.class`), puis des paramètres, à savoir, dans l'ordre, la valeur du trafic (réel) et le nombre de serveurs (entier). Le résultat fourni correspond aux pertes. Cette syntaxe est sans doute familière aux habitués des lignes de commande style UNIX.

FIGURE 1.

L'outil Erlang

```

MS-DOS Command Prompt
04/16/99 08:11a          1,568 ta2.los
04/16/99 08:11a          3,407 ta2.tra
08/18/98 01:59p       3,633,152 Trafic.backup.fm
07/12/99 10:09a       3,632,128 Trafic.fm
07/12/99 11:58a       4,494,336 Trafic99.backup.fm
07/23/99 07:20p       4,495,360 Trafic99.fm
          15 File(s)          16,829,864 bytes
                        996,769,792 bytes free

D:\LABORATOIRE\Trafic>cd Java Simulator
D:\LABORATOIRE\Trafic\Java Simulator>dir
Volume in drive D is Fichiers
Volume Serial Number is 4CF6-1170

Directory of D:\LABORATOIRE\Trafic\Java Simulator

07/23/99 07:23p          <DIR>          .
07/23/99 07:23p          <DIR>          ..
06/24/98 11:18a          1,088 Erlang.class
06/23/97 02:02p          2,233 Excellio.class
08/09/97 02:13p           589 PulseSrc.class
08/09/97 02:13p          4,351 QModels.class
08/09/97 02:13p          1,290 QueueElement.class
08/09/97 02:13p           914 Server.class
08/09/97 02:13p          522 Source.class
01/14/99 10:21p          <DIR>          Sources
08/09/97 02:13p          1,354 SourceSet.class
08/09/97 02:13p          3,902 TestQ.class
08/09/97 02:13p          2,921 Waiting.class
          13 File(s)          19,164 bytes
                        996,769,792 bytes free

D:\LABORATOIRE\Trafic\Java Simulator>java Erlang
Usage : java Erlang <Trafic> <Nb serveurs>

D:\LABORATOIRE\Trafic\Java Simulator>java Erlang 25 30
Erlang(25.0, 30) = 0.052603172228140614

```

4.2 Le programme de simulation "Markov"

Ce programme est le fruit d'un travail de diplôme (brillant) mené à terme par Mr. Laurent Bussard, ingénieur diplômé en télécommunications (1996). Ce programme permet de simuler un ensemble de sources relié à un ensemble plus petit de serveurs, et de comparer les résultats de la simulation à la théorie (Erlang ou Engset).

A noter que la simulation ne se fait pas en temps réel, mais en temps simulé. En principe, on peut choisir librement ce que représente le temps simulé (Dt dans les dialogues), mais ce choix doit impérativement être utilisé de manière cohérente. Si on se fixe Dt = 10 secondes, les probabilités de nouvelle sollicitation et de fin de sollicitation seront à adapter relativement à un choix Dt = 1 seconde ! Le pro-

gramme effectue en principe cette adaptation automatiquement, mais ne contrôle pas la cohérence de Dt relativement aux deux probabilités qui régissent le comportement de la chaîne de Markov. Un Dt petit offrira une bonne précision de la simulation, mais demandera un temps calcul très long pour obtenir une statistique de quelque importance.

4.2.1 Localisation et lancement du programme

Ce programme est en libre copie pour les intéressés. Il se trouve sur le serveur du laboratoire, dans le volume “\\eint20\IICT\Laboratoires\Telecom\Reseaux\Trafic\Markov”.

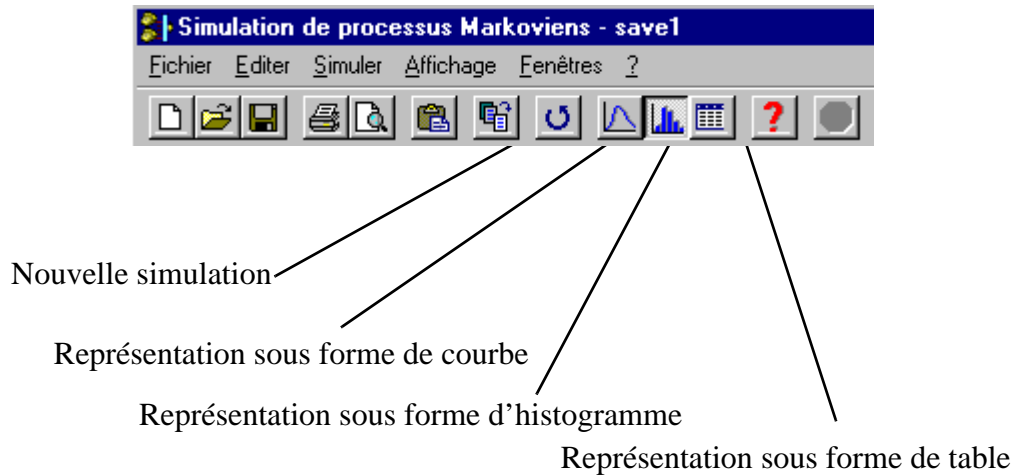
Il n’est pas nécessaire de l’installer sur les postes de travail; il est possible de l’utiliser directement à partir du serveur. Du fait d’un problème non encore élucidé, une utilisation répétée du programme peut conduire à une erreur (“plantage”). Il est donc préférable de relancer le programme à chaque nouvelle simulation.

Il est par contre nécessaire de connecter PC_Public comme un disque réseau.

4.2.2 La barre d’outils

Le programme est équipé de menus et de barres d’outils pour faciliter la manipulation. La barre d’outils est représentée à la Figure 2, page 8. Elle permet de démarrer une simulation, de la configurer, et de choisir son mode d’affichage. Les autres symboles sont assez bien connus des utilisateurs, même occasionnels, de programmes Windows.

FIGURE 2. Barre d'outils du programme Markov



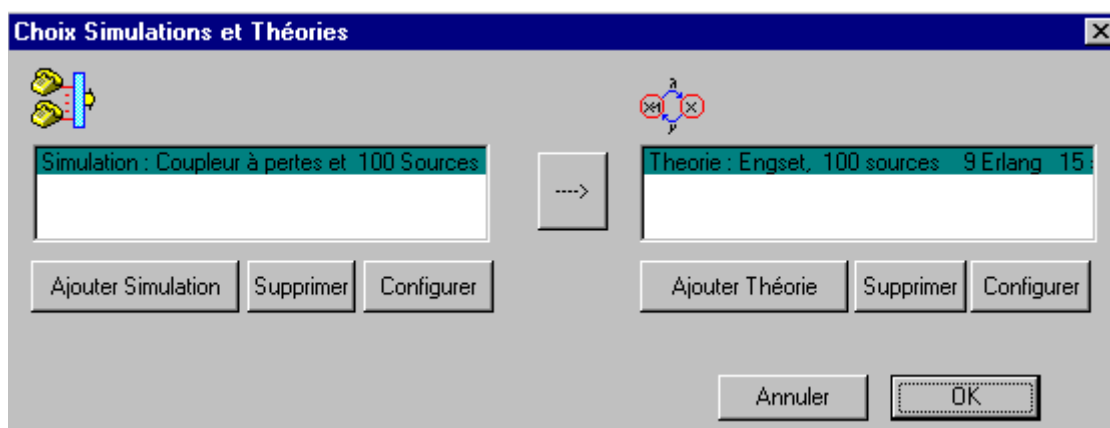
4.2.3 Démarrage de la simulation

Le dialogue d'initialisation de la simulation réclame l'entrée d'une configuration à simuler, et d'une théorie avec laquelle on comparera les résultats de la simulation. Au besoin, le programme extrapole automatiquement une théorie à partir d'une configuration de simulation (bouton central, marqué d'une flèche vers la droite).

La partie gauche de la fenêtre permet d'ajouter des configurations à simuler (il est possible de faire plusieurs simulations simultanément), alors que la partie droite permet l'introduction des paramètres d'une courbe théorique de référence.

Le dialogue est représenté à la Figure 3, page 9.

FIGURE 3. Démarrage de la simulation



4.2.4 Configuration de la simulation

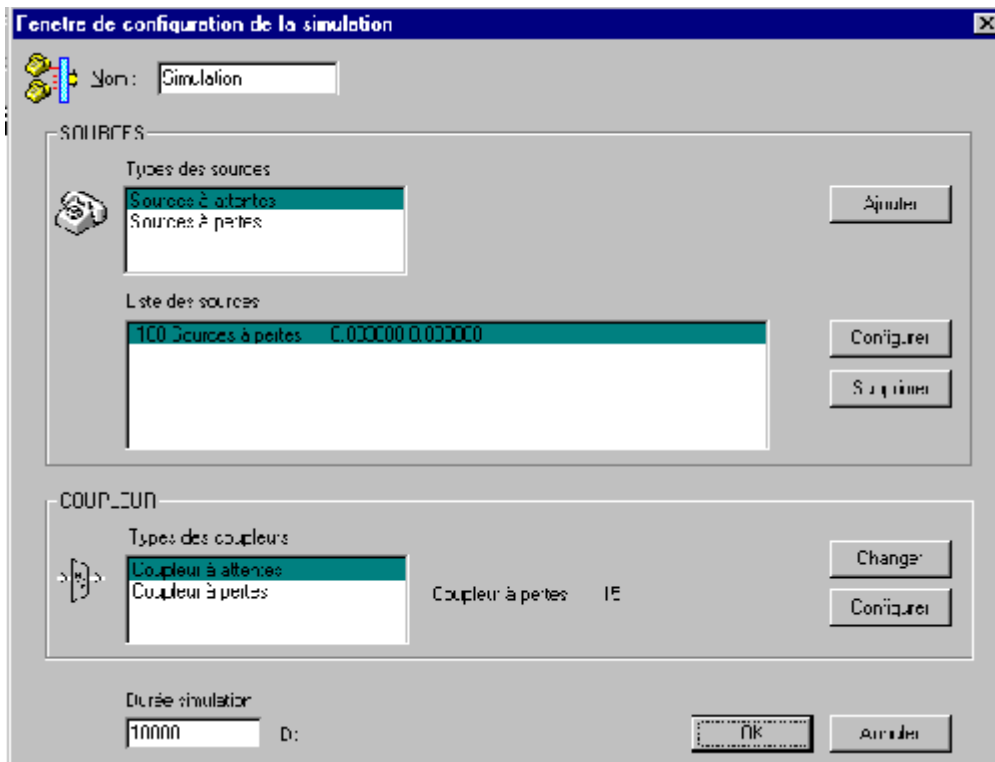
Le dialogue de configuration de la simulation est représenté à la Figure 5, page 11. Il est conseillé de donner un nom à la simulation à des fins de documentation de l'impression. Le type de sources à ajouter peut être à pertes ou à attentes : dans notre cas, ce sera généralement des sources à pertes.

La fenêtre suivante permet l'introduction de différents groupes de sources, chacun pouvant comporter des sources de caractéristiques différentes. Dans la plupart des cas, nous utiliserons des sources homogènes.

La dernière partie concerne les coupleurs, qui seront des coupleurs à pertes dans notre cas. Tant pour les sources que pour les coupleurs, le bouton configurer permet de paramétrer correctement un coupleur.

La durée de la simulation est à introduire en unités de temps de simulation.

FIGURE 4. Dialogue de configuration de la simulation



4.2.5 Configuration des sources

Les sources identiques sont configurées au moyen du dialogue de la Figure 5, page 11.

FIGURE 5. Dialogue de configuration des sources

Configuration d'une source à pertes

Nombre de sources identiques : 100

Caractéristiques statistiques

Probabilité de sollicitation : 0.003

Probabilité de libération : 0.03

Télétrafic

Trafic généré par la source : 0.1 Erlang

Durée moyenne d'un appel : 33.3333 Dt

La durée d'un appel est mesurée en temps de cycle du programme. Cette durée influence directement les probabilités de sollicitation et de libération.

Annuler OK

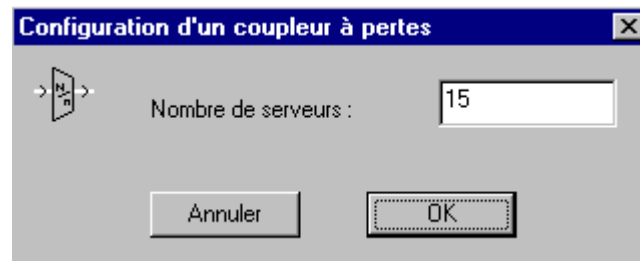
On entrera de préférence le trafic par source en Erlang, et la durée de l'appel en unités de temps de simulation (Dt). Généralement, on préférera utiliser des secondes pour le cas de processus téléphoniques. Les quantités mathématiques seront en principe automatiquement mises à jour par le programme.

4.2.6 Configuration des serveurs

Le nombre de lignes (ou nombre de serveurs) doit être défini par le dialogue de la Figure 6, page 12. Ce dialogue ne nécessite pas d'explications particulières, semble-t-il.

FIGURE 6.

Nombre de serveurs



4.2.7 Configuration de la théorie

Le modèle théorique à utiliser (ou Théorie) doit être défini par le dialogue de la Figure 7, page 13. Il est vivement conseillé d'introduire une appellation claire au modèle théorique, de manière à être en mesure d'identifier aisément une courbe sur le diagramme final.

Les modèles sont prédéfinis (Poisson, Erlang, Engset). En fonction du modèle choisi, il peut être ou ne pas être nécessaire d'introduire également le nombre de sources, ou un maximum d'occupations (cas de Poisson).

FIGURE 7. Dialogue de configuration de la théorie

Configuration du calcul théorique

Nom :

Choix du modèle :

- Engset
- Erlang
- Poisson

Trafic offert :

Nombre de serveurs :

Nombre de sources :

Nombre d'occupations maximum :

OK Annuler

4.3 Les fichiers de données de trafic

Les données de trafic ont été acquises par des centraux téléphoniques. Le format original, propriétaire au fournisseur du central, a été converti en fichiers texte contenant N lignes. La première contient l'heure à chaque quart d'heure. Les lignes suivantes contiennent des valeurs pour chacun des quart d'heure d'une journée. Ces valeurs sont séparées par des virgules (,), les textes sont marqués par des guillemets (""). La première colonne de chaque ligne contient le nom de la série de données associée à cette ligne. Les valeurs numériques sont des valeurs réelles, et c'est un point (.) qui est utilisé pour séparer la partie entière de la partie fractionnaire.

Ce format devrait pouvoir en principe être importé dans tout tableur digne de ce nom. Il faut signaler que certaines versions de Lotus 1-2-3 n'acceptent pas d'importer des lignes contenant 97 champs.

Ces fichiers sont localisés sur le serveur du laboratoire, mais peuvent également être fournis sur disquette pour ceux qu'un travail à la maison motive plus.

Ces fichiers contiennent les données numériques sous forme de valeurs entières. Une valeur non valable (n'ayant pas été mesurée) est remplacée par -1.

4.4 L'outil d'évaluation

L'évaluation peut se faire de diverses manières, mais l'utilisation d'un tableur (à défaut d'outil plus spécifique) permet une évaluation confortable. La plupart des tableurs utilisent, par défaut, un tabulateur pour séparer les champs. Il est probablement nécessaire, selon le tableur utilisé, de configurer le séparateur dans le dialogue d'importation adéquat.

On a décrit ci-après la manière d'effectuer ces calculs répétitifs dans un tableur comme Excel.

5.0 Manipulation

5.1 Faisceau bidirectionnel de faible importance

Il s'agit dans ce cas de l'évaluation d'un faisceau reliant une zone semi rurale de l'Emmental Bernois (la vallée, pas le fromage du même nom) à un central de la ville de Berne. Les données sont contenues dans le fichier **lag_buz.dat**, et comprennent :

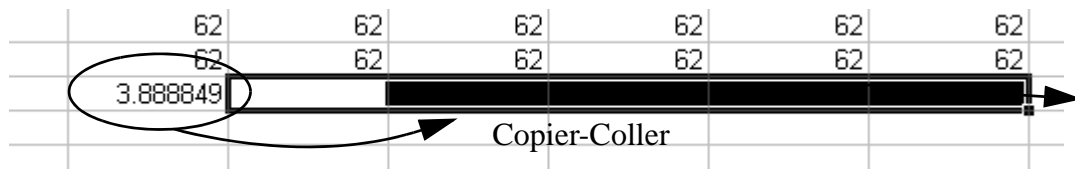
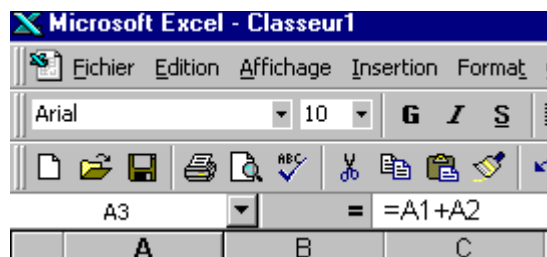
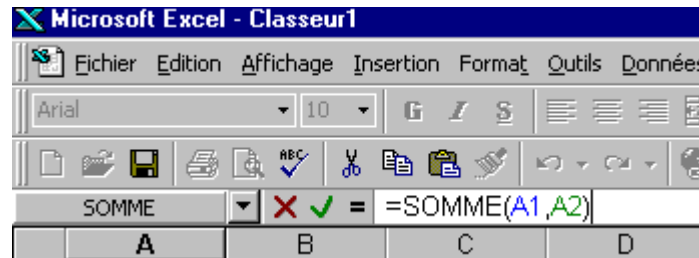
- Le trafic entrant
- Le trafic sortant
- Le nombre de lignes (serveurs)

Les données sont mesurées sur une semaine. On déduira les quantités suivantes pour l'évaluation :

5.1.1 Trafic global

Pour chaque jour mesuré, on convertit le trafic entrant et sortant en un trafic bidirectionnel, qui représente la charge globale du faisceau. Ce trafic ainsi obtenu va servir à l'évaluation du faisceau. La somme peut se faire très simplement avec Excel : Calculer la somme sur une cellule, puis copier cette somme sur toute la ligne. Copier ensuite cette ligne sur toutes les lignes consécutives (4 lignes), et on obtient ainsi les lignes désirées.

FIGURE 8. Constitution de la somme de deux lignes de données avec Excel 97



	b2	b2	b2	b2	
Copier	3.888849	0.284442	0.244442	0.408885	0.608885
	2.617751	0.73777	1.62665	1.608873	0.63555
Coller					

5.1.2 Valeur hebdomadaire

Pour cette semaine, représenter sur un graphique un “jour moyen” représentant la moyenne des jours de la semaine (trafic global), et en tirer la valeur hebdomadaire. On prendra soin bien sûr de calculer au préalable les valeurs horaires, quart d’heure par quart d’heure, en uti-

lisant les possibilités du tableur de manière appropriée, en utilisant la fonction MOYENNE de Excel sur les 4 cases consécutives (le quart d'heure courant y compris, soit une zone du genre **Bn:En**), et en copiant/collant ensuite le résultat sur toute la ligne, puis la ligne sur quatre autres lignes consécutives. On en profitera pour déduire les valeurs de l'heure chargée pour chacun des jours évalués.

La même fonction MOYENNE sera utilisée pour calculer le jour "moyen", alors que la fonction MAX peut être utilisée sur une zone **Bn:CSn** pour le calcul de la valeur hebdomadaire.

5.1.3 Evaluation

Evaluer les pertes de ce faisceau sur la base des données mesurées. Que se passe-t-il en cas de surcharge de 20% ? Ce faisceau vous semble-t-il correctement dimensionné ?

5.1.4 Simulation

Certaines simulations peuvent être relativement longues, et s'étendre sur plusieurs minutes. Ne pas se désespérer trop tôt !

Simuler ce faisceau dans les conditions d'utilisation de la mesure. On répartira, à cet effet, le trafic offert (à estimer en fonction des mesures effectuées) sur un millier de sources, la durée des occupations étant d'environ 180 secondes. Ces hypothèses sont réalistes et correspondent assez exactement aux conditions réelles qui prévalaient dans cette région au moment où la mesure a été effectuée. Les résultats obtenus sont-ils cohérents avec la mesure ?

Suite à la venue d'une grande multinationale dans la zone de raccordement, on pense que le nombre de raccordements va doubler dans les quatre années à venir. Simuler ce cas de figure, comparer à une prévision selon Erlang, et proposer des mesures à prendre.

5.2 Mesure sur un jour

Les mêmes données, mais ne concernant qu'un jour, pour un autre faisceau de la région bernoise sont stockées sur le fichier **bei01.csv**. Critiquer le dimensionnement de ce faisceau, suggérer un redimensionnement. On s'appuiera essentiellement sur la théorie mathématique selon Erlang pour aboutir à des conclusions étayées, et on pourra vérifier ensuite le cas de figure par une simulation.

5.3 Faisceau bidirectionnel de grandeur moyenne (facultatif)

On étudie ici un faisceau convoyant un trafic bidirectionnel. dans la région bernoise. Il s'agit d'un faisceau entre un central de raccordement de la ville de Berne et le central de transit de Berne. Les mesures comprennent (fichier **bol_bei.dat**) :

- Le trafic entrant
- Le trafic sortant
- Le nombre de lignes (serveurs)

Le faisceau a été mesuré durant deux semaines complètes. On se propose d'en tirer les données d'évaluation suivantes, en plus du **trafic global**, de l'**heure chargée**, des **valeurs journalières** et des **valeurs hebdomadaires** :

5.3.1 Valeur mensuelle

Tirer des calculs précédents la valeur mensuelle (sur deux semaines) de ce faisceau.

5.3.2 Interprétations

Sur la base de ces mesures, peut-on dire si le faisceau est correctement dimensionné, en justifiant la réponse donnée ?

5.4 Trafic généré par des ordinateurs

Le fichier **ta2.tra** contient du trafic (entrant et sortant) généré par des ordinateurs. Ces ordinateurs n'ont travaillé que pendant une partie de la journée (jusque vers 10h00), mais leurs utilisateurs ont constaté un nombre anormal d'échecs de tentatives de connexion. Ce nombre est stocké dans le fichier **ta2.los**.

Quelles différences peut-on constater relativement à un trafic généré par des humains ? Pourquoi les échecs sont-ils aussi nombreux ? Peut-on, sur la base de ces mesures, modéliser les sources (taux de sollicitation, durée moyenne d'une communication) ? Si oui, peut-on comparer les données mesurées à une simulation réaliste d'un système à pertes ?

5.5 Trafic de route (destinations et vias) (facultatif)

Le fichier **lsp_dest.all** contient l'ensemble des destinations mesurées par un central de transit de la région lausannoise. Cette mesure a eu lieu en février 1993, à une époque où avait encore cours une ancienne

forme de numérotation. L'indicatif de destination se retrouve dans le nom des données (comme dans **027.1** (TRAFFIC VALUE)). Les mesures concernent une semaine complète, du lundi au vendredi.

On s'intéresse dans ce cas au trafic de transit concernant une certaine zone. On s'intéresse au volume de trafic traité par ce central pour les zones suivantes :

- l'agglomération zurichoise et ses environs
- Genève et environs
- la Suisse romande

Pour chacun de ces cas de figure, on calculera

- L'heure chargée pour chacun des jours de la semaine.
- La valeur hebdomadaire
- Le diagramme journalier d'un jour de semaine "moyen".

5.6 Centraux privés (PBX)

Mettez-vous dans la peau d'un employé d'un grand opérateur de télécommunications de notre pays. Celui-ci reçoit un coup de téléphone de la part d'une PME ayant un central domestique équipé de plusieurs lignes de raccordement au réseau public. La personne qui fait le téléphone est le (la) centraliste, et son numéro est le 031.207.11.11 (région bernoise).

La PME estime que ses factures sont exagérément élevées, et demande un diagnostic. Pouvez-vous effectuer ce diagnostic sur la base des données comprises dans le fichier contenant les centraux privés reliés au central de raccordement auquel est également relié le PBX en question ? (Fichier : **PBX.dat**).

6.0 Simulation de coupleurs à attente

Les manipulations effectuées jusqu'ici concernent des systèmes que l'on traite généralement comme systèmes à pertes. Les systèmes à attente sont plus fréquemment utilisés dans les systèmes téléinformatiques, pour traiter des informations ne nécessitant pas une transmission conforme du point de vue temporel.

On a conçu quelques petits programmes en Java permettant de simuler les cas de systèmes à attente les plus fréquents, et on se propose dans cette partie de la manipulation de comparer ces divers cas de figure. Les étudiants intéressés, et disposant de connaissances du langage Java sont vivement encouragés à s'intéresser aux programmes. Toute suggestion d'amélioration est bienvenue. Un des inconvénients de Java est sa lenteur; cette lenteur devrait par corollaire vous permettre d'approfondir le contexte théorique de la simulation de trafic.

6.1 Les outils utilisés

Les programmes de simulation sont composés d'objets (presque forcément, en Java) qui interagissent pour effectuer une simulation aussi valable que possible. On a volontairement négligé, dans un premier temps, l'aspect présentation de ces outils de simulation (mais si vous vous sentez motivés pour définir et implémenter un tel interface, considérez-vous comme très bienvenu !). De fait, ils se présentent sous la forme d'une simple fenêtre d'introduction de commandes.

Pour chaque cas de figure, une classe bien définie effectue la simulation. On se servira de la machine virtuelle **java** fournie par Sun pour effectuer les simulations. La machine virtuelle Java de Sun Microsystems se trouve sur les postes de travail : “\JDK1.2\bin\java”.

Les classes responsables de la simulation sont situées sur le serveur du laboratoire, dans : “\eint06\Profs\IICT\Laboratoires\Telecom\Reseaux\Trafic\Java Simulator”. **On recopiera les fichiers .class dans le répertoire de travail** : les programmes java devront **impérativement** être exécutés dans le **répertoire de travail**, et **pas dans le répertoire original**.

Les résultats de la simulation sont inscrits sur un fichier (tmp.dat) dans le répertoire de travail. Ce fichier est un fichier pouvant être lu par un tableur (genre Excel), et dont le format correspond au format utilisé pour les données de trafic, à l'exception du séparateur de données, qui se trouve être un tabulateur au lieu de la virgule précédemment utilisée (ce qui incidemment vous épargnera une opération lors de l'importation du fichier dans Excel).

Ces résultats contiennent la statistique des occupations de la file, la statistique des attentes des sollicitations dans la file, ainsi que quelques données auxiliaires utiles pour formater les résultats de manière correcte.

6.1.1 Sources statiques et sources dynamiques

On parle de source statique dans le cas de sources qui sont occupées pendant le temps de l'attente et le temps de traitement. Ceci est le cas de trafic téléphonique, ou de l'attente au guichet de banque ou au guichet de poste. Ce cas de figure a été traité en théorie au cours, encore que de manière assez superficielle. Une seule source ne peut jamais générer un trafic apte à surcharger le serveur.

On parle de source dynamique dans le cas de sources qui se contentent d'effectuer une notification, et qui continuent de vaquer à leurs occupations une fois la notification faite. On parle aussi dans ce cas de sources événementielles. Ce cas correspond à beaucoup de systèmes de mesure d'événements, soit des barrières lumineuses, des compteurs de particules, etc... Dans ce cas, une seule source peut surcharger un ou plusieurs serveurs. Au fait, comment une seule source peut-elle surcharger un serveur ?

6.1.2 Le logiciel de simulation

Après avoir copié les classes nécessaires dans un répertoire de travail (D:\Travail, par exemple), on définira ensuite la variable d'environnement pour l'interpréteur Java, de manière à lui permettre de localiser les classes nécessaires :

```
C:\Travail>SET CLASSPATH=.;<Répertoire de Java>\jre
```

Le "." indique le répertoire courant (là où vous avez copié les classes pour la simulation), et l'autre répertoire définit les classes standard Java. Le logiciel de simulation **TestQ** (Attention à respecter la casse, en Java !) se lance par le biais de la machine virtuelle Java, par exemple, en utilisant **java** :

```
C:\Travail><Répertoire de Java>\bin\java TestQ arg1 arg2 arg3 arg4  
arg5 ....
```

argx a pour format arg=Valeur, où valeur est une chaîne de caractères représentant une valeur en virgule flottante (*double*) ou entière (*int*), selon le cas. argx ne contient pas d'espaces, les espaces étant utilisés pour séparer les divers arguments de la ligne de commande Java. Les arguments possibles sont :

- c : représente le taux de sollicitation d'une source (double)
- h : représente le temps moyen (ou constant, selon le cas) de traitement par un serveur (double)
- q : représente la longueur de la file d'attente (int)
- n : représente le nombre de serveurs (int)
- N : représente le nombre de sources (int)
- t : représente la durée de la simulation en cycles de programme (int)
- m : représente le modèle de file d'attente à utiliser (int)

m peut prendre les valeurs suivantes :

- 0 : Serveurs multiples, temps de traitement constant, sources dynamiques.
- 1 : Serveur unique, temps de traitement constant, sources dynamiques. Equivaut à $m=0$ avec $n=1$.
- 2 : Serveur multiple, temps de traitement aléatoire, sources dynamiques.
- 3 : Serveur multiple, temps de traitement aléatoire, sources dynamiques, **une file d'attente par serveur.**

Par défaut, on a :

- $c=2.2$ (2.2 sollicitations par seconde et par source)
- $h=0.12$ (120 ms de temps de traitement, moyen ou constant)
- $q=100$ (100 cases dans la file d'attente)
- $N=4$ (4 sources, soit un trafic global à écouler de 1.056 E)
- $n=2$ (2 serveurs pour écouler ce trafic)
- $t=100000$ (100000 itérations de temps calcul, insuffisant en principe pour obtenir une statistique valable)
- $m=0$ (temps de traitement constant)

L'unité de temps de simulation (c'est-à-dire, le temps effectif que représente un cycle de programme) correspond à $h/10$. Ceci détermine la finesse de la simulation, comme le faisait Dt dans le programme de Mr Bussard. Ceci définit aussi le temps qui s'écoulerait dans la réalité (temps simulé).

6.1.3 Temps de calcul

On a dit plus haut que ces programmes étaient écrits en Java, et que ce fait pénalisait le temps d'exécution. Le temps d'exécution varie linéairement en fonction du nombre de sources et du nombre de serveurs, mais aussi (non linéairement !) en fonction de l'algorithme uti-

lisé. Avant de vous lancer dans une simulation de 50000 sources avec 400 serveurs, veuillez s'il vous plaît accorder un minimum d'attention aux quelques lignes qui vont suivre.

On a effectué un test comparatif basé sur les données suivantes :

- 4 sources ($N=4$)
- 2 serveurs ($n = 2$)

on a obtenu, en utilisant un Pentium II à 266 MHz exécutant Windows NT4 Server, les chiffres suivants :

- Mode 1 ($m=1$) : de 12 à 14 secondes pour 10^7 pas de simulation.
- Mode 2 ($m=2$) : de 16 à 19 secondes pour 10^7 pas de simulation.
- Mode 3 ($m=3$) : de 170 à 185 secondes pour 10^7 pas de simulation.

Notons que 10^7 est un chiffre que l'on qualifiera de bon, sans plus. Il correspond à une durée réelle de 3,6 heures, en utilisant les valeurs par défaut du programme pour les autres paramètres ($h = 130$ ms, donc cycle simulé de 13 ms).

On pourrait vraisemblablement obtenir un gain de performances d'un facteur 1.5 en optimisant le programme Java, et un gain de performances de l'ordre de 10 à 15 au moins en réécrivant le programme en C++. Que ceux qui se sentent frustrés par la lenteur de la simulation considèrent cette remarque comme un défi; répétons-le, les sources sont disponibles !

6.2 File d'attente classique infinie

Il est impossible de simuler une file d'attente effectivement infinie, si bien que le terme "infini" est à traduire par "très grand" dans notre cas de figure. On utilisera une valeur "grande" par rapport aux résultats que l'on peut s'attendre à obtenir. A vous de choisir cette valeur.

6.2.1 Sources dynamiques, Temps de traitement variable

Dans ce cas de figure ($m = 2$), on s'intéresse à la statistique résultant d'un nombre de sources donné servies par un certain nombre de serveurs au travers d'une file d'attente unique (banalisée). On simulera trois cas de figure :

- Système peu chargé
- Système moyennement chargé
- Système surchargé

Pour les trois cas de figure, on utilisera le même nombre de sources et de serveurs, en faisant varier soit le taux de sollicitation des sources, soit le temps moyen de traitement. Dans les trois cas, on calculera la durée moyenne d'attente (y compris les sollicitations ne devant pas attendre) et on comparera la valeur obtenue au calcul théorique, lorsque l'on disposera des outils nécessaires pour effectuer ce calcul (le fait que les sources soient statiques ou dynamiques n'intervient pratiquement pas dans ce calcul). On essaiera, dans la mesure du possible, de représenter cette durée moyenne sur le graphique des probabilités d'attente, et on essaiera de mettre en évidence la distribution théorique des attentes en fonction de la distribution obtenue par simulation. On représentera aussi la statistique des occupations de la file; ces deux statistiques sont à représenter dans tous les cas de simulation, de manière semblable aux figures ci-après.

En guise de point de comparaison, on a utilisé, lors de la préparation de ce labo, les valeurs suivantes :

4 sources, 2 serveurs. 0.18 Erlang par serveur pour un système peu chargé, 0.8E en charge moyenne, et 0.96E pour un système chargé.

FIGURE 9.

Exemple de statistique des attentes

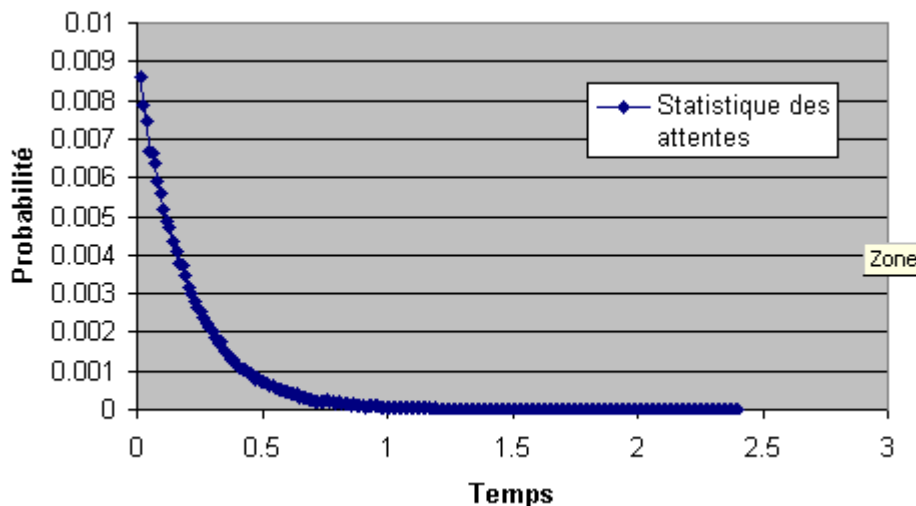
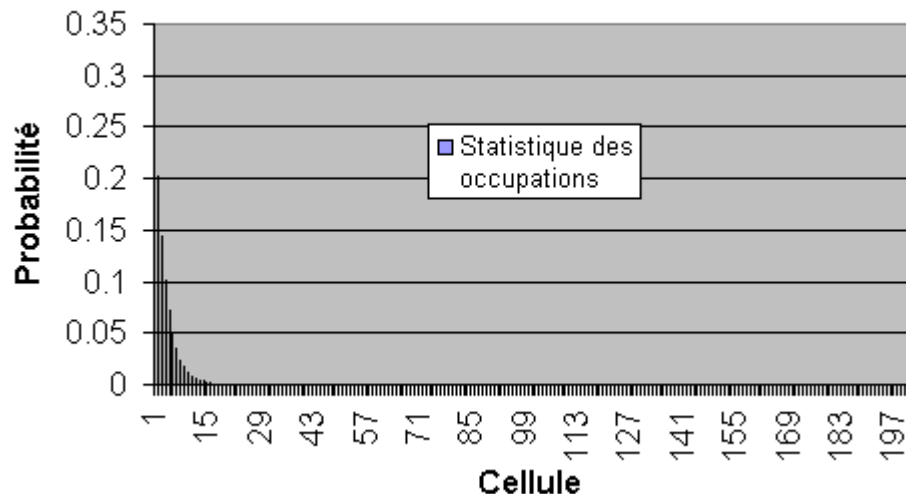


FIGURE 10.

Exemple de statistique des occupations



6.2.2 Sources dynamiques, temps de traitement constant

Répéter l'exercice précédent avec un temps de traitement constant, comme c'est le cas fréquemment en informatique. Ceci correspond au modèle 0 ($m=0$). Essayer de discuter et d'interpréter les différences éventuelles.

6.2.3 Sources dynamiques, temps de traitement variable, file d'attente par serveur

Répéter l'exercice Section 6.2.1, page 22 avec une file d'attente par serveur, sur le modèle des guichets de poste bien connus. Ceci correspond au modèle 3 ($m=3$). Discuter et interpréter les différences éventuelles.

7.0 Critiques et suggestions

Après avoir subi ces diverses manipulations avec des sentiments allant de l'enthousiasme (on peut rêver...) à la frustration la plus totale, il serait sympathique pour vos successeurs que vous critiquiez cette manipulation, et suggériez des améliorations, des compléments, voire de nouvelles idées que l'on pourrait implémenter dans ce domaine.

Toute critique, même incendiaire, sera la bienvenue. Cette partie du laboratoire ne peut en aucun cas faire l'objet d'une sanction, que ce

soit au point de vue de la note ou à tout autre point de vue; la proximité du diplôme ne doit donc pas vous empêcher d'exprimer vos griefs, même acerbes, voire vous permettre un défolement bénéfique pour votre état nerveux général. En revanche, une critique constructive (même franchement défavorable, mais permettant une amélioration du produit) ou des suggestions intéressantes pour de futures versions de ce laboratoire (si tant est qu'il y ait de futures versions) ne manquera pas de vous attirer la reconnaissance de l'auteur de ces lignes.