Licence Magistère STIC **13 juin 2003**

Examen du cours Réseaux

Durée : 2h. **Tous les documents sont autorisés.**
NB : Toutes les réponses doivent être clairement justifiées. Le barème est indicatif.

Détection d'erreurs (7 points)

Soit C un code cyclique de longueur n et dimension k sur le corps $\mathbb{K} = \mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$. On rappelle qu'un mot de code de C peut être représenté comme un polynôme sur $\mathbb{K}[x]/(x^n - 1)$ ou un n-uple, et que tous les mots de C sont des multiples d'un polynôme (appelé générateur) $g(x)$, avec $deg\, g(x) = n - k$. Les calculs des mots de C est fait à l'aide d'un circuit composé de registres de décalage et de portes logiques. Le poids d'un mot de $\mathbb{F}_2$ est le nombre de composantes non-nulles (le mot 01000100 est de poids 2).

Question 1 Donner le circuit qui calcule les mots de code correspodant au polynôme générateur $g(x) = \text{CRC-12} = x^{12} + x^{11} + x^3 + x^2 + x + 1$.

Pour un mot $z(x) \in \mathbb{K}[x]/(x^n - 1)$, on dira que $z(x)$ est un mot reçu dont l'erreur de transmission est $\eta(x)$ s'il existe $c(x) \in C$ tel que $z(x) = c(x) + \eta(x)$ (c'est-à-dire, $z(x)$ provient d'un mot de C entaché des erreurs données par $\eta(x)$).

On appelle syndrome polynômial d'un mot $z(x) \in \mathbb{K}[x]/(x^n - 1)$, le reste de la division de $z(x)$ par $g(x)$ dans $\mathbb{K}[x]$. Il resulte qu'un syndrome non-nul signale une erreur dans le mot reçu.

Question 2 Le polynôme générateur CRC-12 détecte-t-il l'erreur de transmission $\eta_1(x) = x^{10}$? La même question pour l'erreur $\eta_2(x) = x^{16}$ (on suppose que $n > 16$) ?

Question 3 Montrez qu'un code cyclique (n, k) non-trivial sur $\mathbb{F}_2$ détecte l'ensemble des erreurs de poids 1.

Question 4 On appelle paquet d'erreurs d'étendue L un mot de $\mathbb{F}_2^n$ dont les seules composantes non nulles sont parmi L composantes successives, dont la première et la dernière sont non nulles. Formellement, c'est un mot $(x_1, ..., x_n) \in \mathbb{F}_2^n$ tel que il existe j, $1 \leq j \leq n - L + 1$ et $x_i = 0$ si $i < j$ ou $i > j + L - 1$ et $x_i \neq 0$ si $i = j$ ou $i = j + L - 1$.

Montrez qu'un code cyclique (n, k) non-trivial sur $\mathbb{F}_2$ détecte l'ensemble des paquets d'erreurs d'étendue inférieure ou égale à $n - k$.

Routage Multicast (8 points)

Les protocoles des niveaux transport et réseau étudiés en cours sont des protocoles **unicast** car ils assurent le transport des données entre **un** hôte émetteur et **un** hôte récepteur.

Dans ce problème, on s'intéresse au transfert des données entre **un** hôte émetteur et **un groupe** de hôtes récepteurs. L'émetteur et les récepteurs forment un groupe de **multicast**, identifiés par une adresse spécifique (en plus de leur adresse IP). Pour transférer des données, l'émetteur forme **un seul** paquet (comme dans le transfert unicast) en précisant comme adresse destination l'adresse de multicast du groupe. Ce paquet est dupliqué par les routeurs du réseau chaque fois qu'il faut le faire suivre sur plusieurs liens sortants pour atteindre les récepteurs.

Le problème du routage multicast est d'établir pour chaque routeur du réseau quels sont les liens sortants sur lesquels il doit envoyer un paquet ayant une adresse de multicast donnée. Il s'agit donc d'établir un arbre de couverture entre l'émetteur et les récepteurs.

Par exemple, dans la figure 1, le groupe de multicast d'adresse δ est formé par les machines A.2, B.1, E.2 et F.2 et la source est la machine A.2. Quand le routeur A reçoit un paquet à transmettre au groupe de multicast δ, et dans sa table de routage à l'entrée δ il a les routeurs C et B, il duplique ce paquet vers C et B. Si B a dans sa table de routage pour l'adresse δ, en plus de son réseau, le routeur D, il duplique le paquet et l'envoie à B.1 sur son réseau local et à D.

Dans la suite, on considère que les liens entre les routeurs ont un coût constant et le même dans les deux directions. Le coût de la connexion d'un hôte à son routeur est nul.

Question 1 : Sans tenir compte du routage multicast, indiquer comment les routeurs peuvent construire un arbre de couverture tel que le coût de la transmission entre l'émetteur et chaque récepteur est minimal. On suppose que chaque routeur connaît le graphe complet du réseau.

Dans le cadre du routage multicast, on considère maintenant que chaque routeur connaît les routeurs auxquels sont connectés les récepteurs du groupe multicast.

Question 2 : Indiquer comment la solution proposée à la question 1 peut s'adapter pour introduire dans les tables de routage les informations concernant les adresses de multicast.

En pratique, les routeurs du réseau n'ont pas d'informations sur les groupes de multicast. Ils doivent construire l'arbre de couverture utilisé

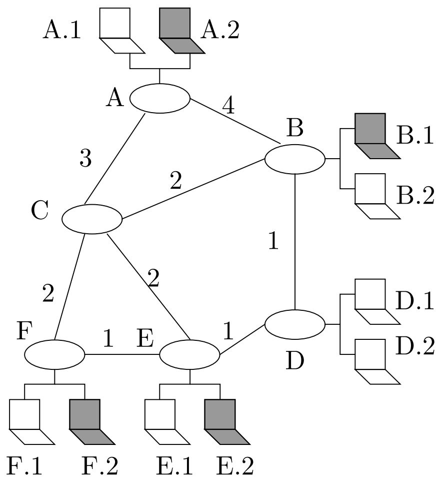

FIG. 1 – Exemple de groupe multicast.

par le multicast en utilisant un protocole spécial. Dans la suite, on suppose que les groupes de multicast ne peuvent pas évoluer avec le temps. De plus, chaque routeur a l'information sur le chemin minimal vers tout autre routeur.

On considère le protocole suivant. Dans une première étape, l'émetteur envoie une datagramme de construction de l'arbre `T-CONS` avec l'adresse du groupe comme destination (et son adresse comme source). Les routeurs considèrent les datagrammes de type `T-CONS` uniquement si elles arrivent sur le lien qui correspond au chemin minimum vers l'émetteur. Si oui, ils envoient ce datagramme sur tous les autres liens. Sinon, ils ignorent ce datagramme.

Question 3 : Montrer sur l'exemple donné dans la figure 1 comment circulent les datagrammes de type `T-CONS` dans le réseau.

La datagramme `T-CONS` arrive aux hôtes finaux. Ces hôtes répondent avec une datagramme `G-REJ` s'ils ne font pas partie du groupe. Si un routeur reçoit sur tous les liens sur lesquels il a envoyé `T-CONS` une réponse `G-REJ`, il envoie également un `G-REJ` sur le lien vers la source et élimine l'adresse de multicast de sa table de routage.

Question 4 : Donner l'algorithme local (avec communications asynchrones) à exécuter par chaque routeur pour implémenter le protocole ci-dessus. Préciser comment sont calculées les informations de routage pour les groupes multicast en utilisant ce protocole. Appliquer cet algorithme sur l'exemple donné.

Protocole de communication multicast (8 points)

Comme dans le problème précédent, on considère un protocole de transmission **multicast**, mais on s'interesse au transport des données et on considère le routage comme résolu. Le protocole doit permettre la transport des données rapide mais pas nécessairement fiable entre un émetteur et plusieurs récepteurs à travers un réseau non fiable (avec pertes de messages) qui supporte de multicast (un message émis est dupliqué pour atteindre tous les récepteurs). Plus précisément, les canaux sont non fiables, mais qu'ils ne le sont pas totalement, c'est-à-dire, qu'un message est transmis à chaque récepteur de temps en temps (infiniment souvent).

Le service du protocole est d'assurer que pour chaque message émis, la couche supérieure (de chaque récepteur) reçoit soit le message, soit une indication de perte. De plus, les messages, ou les indications de pertes correspondantes, sont délivrés dans l'ordre, sans duplications. Par exemple, si la séquence de messages $abcd$ est émise, un récepteur peut délivrer la séquence $abcd$ elle même, ou bien, $a\perp cd$ si b a été *définitivement perdu* (malgré les efforts pour le récupérer), ou $a\perp c\perp$ si b et d ont été définitivement perdus, ou alors $\perp\perp\perp\perp$ si tous les messages ont été perdus. En fait, dans le protocole, les récepteurs chercheront à récupérer les messages perdus (qu'ils n'ont pas reçus) afin de les délivrer, mais on ne peut exclure (ou on peut tolérer) la perte définitive de quelques messages, pourvu que chacune de ces pertes soit signalée à la couche supérieure (par le message spécial $\perp$). Noter que les séquences délivrées par les différents récepteurs ne sont pas nécessairement les mêmes.

On considère dans un premier temps la version suivante du protocole : Chaque message est accompagné par un numéro de séquence (un entier positif) qui indique sa position dans le flot de données à transmettre. Pour réduire le nombre des messages en circulation, le protocole est basé sur le principe des accusés de réception *négatifs* (NAK) : les récepteurs envoient des NAK lorsqu'ils se rendent compte (grâce aux numéros de séquence) qu'ils n'ont pas reçu un des messages envoyés.

L'émetteur possède une fenêtre d'une taille maximale T qui définit l'intervalle des messages qu'il a émit et qui sont disponibles dans sa mémoire. Du fait qu'il n'y a pas d'accusés de réception, l'émetteur fait progresser périodiquement sa fenêtre, c'est-à-dire, la borne inférieure de sa fenêtre est incrémentée périodiquement. En incrémentant cette borne, l'émetteur perd le message dont elle est le numéro, et libère de la place dans sa mémoire.

Si le récepteur a reçu (et délivré à la couche supérieure) tous les messages de numero $\leq i$, et s'il reçoit un message de numero $j > i+1$, alors il garde le message j dans sa mémoire, et il emet des NAK pour tous les messages dans l'intervalle $[i+1, j-1]$. A la réception de ces NAK, l'émetteur envoie en multicast (à tous les récepteurs) les messages correspondants dont il dispose.

Question 1 : Quels sont les défauts de cette version ?

On considère maintenant que les messages émis sont accompagnés de la valeur de la borne inférieure de la fenêtre de l'émetteur (au moment de l'émission).

Question 2 : Indiquer comment cette version améliore la précédente et comment elle réalise le service requis.

Question 3 : Donner les algorithmes locaux de l'émetteur et des récepteurs.

On considère que l'émetteur reçoit de la couche supérieure un message à transmettre tous les δ unités de temps. On considère aussi que la fenêtre de l'émetteur est incrémentée tous les p unités de temps, et que le temps de transmission d'un message entre l'émetteur et un récepteur (dans un sens comme dans l'autre) est τ unités de temps. De plus, on admet les hypothèses simplificatrices suivantes : seuls les messages d'origine peuvent être perdus (les NAK et les retransmissions ne le sont jamais), et il ne peut jamais y avoir plus de n messages perdus consécutifs (par exemple, $n = 1$ correspond au cas où les canaux ne peuvent jamais perdre deux messages consécutifs).

Question 4 : Donner une condition (une contrainte sur les paramètres mentionnés ci-dessus) pour quíl satisfasse la propriété de sûreté suivante : "le protocole est totalement fiable" (tous les messages sont délivrés par les récepteurs sans qu'aucun ne soit définitivement perdu).