Module 2

Communication de groupe

Résumé

- Modèle de communnication.
- Mécanismes sous-jacents.
- Niveau de garantie offert pour la communication de groupe.

Avec ou sans connexion

- Sans connexion (UDP):
  - *Send* ([destinataire], message);
  - *Receive* ([émetteur], file de stockage);

- Avec connexion (TCP):
  - *Connect* (destinataire, émetteur);
  - Read, Write...
  - *Disconnec*t (identificateur de connexion);

- **Synchrone ou asynchrone:**
  - Le *send* et *receive* sont bloquants; les processus se synchronisent à chaque message. Send attend l'accusé de réception!?!
  - Le *send* est non-bloquant, si la file du système d'exploitation n'est pas pleine, tandis que *receive* est généralement bloquant. Si les paquets arrivent trop vite et la file du système d'exploitation est pleine, un message de ralentir est envoyé et les paquets en surplus peuvent être ignorés.
  - Le *receive* aussi est non-bloquant, le processus vérifie de temps en temps, peut demander un signal, ou utilise un appel système de type select ou epoll.

- **Un ou plusieurs destinataires:**
  - Avec un bus, on peut envoyer un message à n destinataires sur le bus en 1 unité de temps au lieu de n unités de temps pour des messages séparés.
  - Les protocoles Ethernet et IP supportent la diffusion générale (broadcast) ainsi que la multidiffusion (multicast).
  - Gestion des membres d'un groupe de multi-diffusion plus complexe s'ils ne sont pas tous sur le réseau local.
  - Messages sans connexion non fiables (UDP), ou séquences de messages fiables (Pragmatic General Multicast, PGM) avec accusés de réception négatifs.

- **Messages de groupe par multidiffusion**
  - Tolérance aux fautes basées sur des services répliqués sur plusieurs serveurs.
  - Amélioration des performances par des services et données répliqués sur plusieurs serveurs.
  - Découvertes de services dans un environnement de réseautage spontané.
  - Economie de bande passante par multidiffusion des notifications d'évènements, vidéo ou fichiers identiques demandés par de nombreux clients.

- **Messages de groupe**
  - Envoi d'un message aux membres d'un groupe.
  - Sur réception, le message est remis au processus destinataire.
  - Communication non fiable, un seul message UDP est envoyé, par exemple pour annoncer une adresse Ethernet versus IP.
  - Communication fiable: intègre (message livré au plus une fois), valide (message livré éventuellement à tous les processus fonctionnels, i.e. retransmission jusqu'à obtention d'une réponse), atomique (livré à tous ou à aucun).
  - Communication ordonnancée: totalement, causalement, par origine.

- **Messages de groupe atomiques**
  - Tous le reçoivent ou personne ne le reçoit.
  - Envoi à tous du message *non-confirmé* et demande d'accusé de réception. Retransmission au besoin si l'accusé de réception ne vient pas.
  - Si tous ont reçu, message de confirmation pour rendre le message maintenant confirmé disponible aux applications.
  - Si des accusés de réception manquent après un certain temps, annuler le message.
  - Si un client ne reçoit pas de confirmation, il demande l'état à l'envoyeur puis au besoin vérifie avec un autre ordinateur qui peut prendre la relève.

- **Messages de groupe totalement ordonnancés**
  - Chaque message a un numéro de séquence unique, par exemple fourni par un processus séquenceur.
  - Chaque récipiendaire attend d'avoir reçu les messages précédents avant de livrer un message à ses applications.
  - Un récipiendaire peut devoir laisser tomber des messages plus récents s'il en a trop en attente.
  - Un envoyeur peut bloquer s'il a trop de messages en attente d'accusés de réception.

- **Messages de groupe causalement ordonnancés**
  - Les messages en provenance d'un même ordinateur arrivent dans leur ordre d'envoi à chaque récipiendaire: numéro de séquence propre à chaque membre du groupe qui envoie des messages.
  - Les messages reçus en provenance de plus d'un ordinateur arrivent en ordre causal: chaque processus maintient un vecteur de numéros de séquence de messages de groupe vus venant de chaque processus. Un message avec un certain vecteur n'est pas délivré avant que tous les messages de chaque processus n'aient été reçus, jusqu'au numéro dans le vecteur pour ce processus.