

Nouveaux protocoles de transport

Etudes d'approfondissement

Laurent Torride - Romain Roux - RICM5

Professeur : Florence Perronnin

13 *novembre* 2009



Sommaire

- 1 Motivations
- 2 Les protocoles SCTP/DCCP
- 3 Propriétés
- 4 Performances
- 5 Bilan
- 6 Démonstrations

Plan

- 1 Motivations
 - La couche de transport
 - Rappels TCP/UDP
 - Les nouveaux besoins
 - Les lacunes de TCP/UDP
- 2 Les protocoles SCTP/DCCP
- 3 Propriétés
- 4 Performances
- 5 Bilan
- 6 Démonstrations

La couche de transport

- Couche 4 du modèle OSI
- Permet le transfert fiable, efficace et économique d'informations
- Gère les communications bout en bout entre les processus
- Doit répondre à des qualités de service variables

UDP

Caractéristiques

- Transmission de données de manière simple : adresse IP et numéro de port
- Mode non connecté
- Aucun contrôle de flux ni de congestion

Utilisation

- Applications demandant à transmettre des paquets rapidement
- Peu d'importance si une partie des données est perdue
- Streaming, jeu en réseau
- DNS/TFTP

TCP

Caractéristiques

- Mode connecté : session TCP à 3 phases (établissement de la connexion, transferts de données, fin de la connexion)
- Contrôle de flux (contrôle des tampons, algorithme de Nagle)
- Contrôle de congestion (slow start...)

Utilisation

- 95 % du trafic de Internet
- Applications les plus courantes :
HTTP(s)/FTP/POP3/SMTP

Les nouveaux besoins

- Avoir un compromis entre fiabilité (TCP) et rapidité (UDP)
- Améliorer la sécurité des protocoles
- Avoir un débit constant. Par exemple pour les vidéos en streaming
- Profiter d'un maximum de bande passante sans bloquer les autres

Les lacunes de TCP/UDP

TCP

Attaque possible par deni de service

Chemin coupé : Connexion perdu

Nécessité d'une couche d'abstraction pour structurer les données

UDP

Aucun contrôle de congestion

Pas de contrôle d'erreur

N'est pas TCP-Friendly

Plan

- 1 Motivations
- 2 Les protocoles SCTP/DCCP
 - Présentation de SCTP
 - Présentation de DCCP
- 3 Propriétés
- 4 Performances
- 5 Bilan
- 6 Démonstrations

Présentation de SCTP (1)

Initiation de la connexion

Protocole défini en 2000 par l'IETF
Spécifié dans la RFC 2960
Développé pour la téléphonie sur IP
Différentes API

Présentation de SCTP (2)

Envoi de données et contrôle de session

On parle de blocs d'informations (chunks)
Il existe deux types de blocs, de données (DATA) ou de contrôle

blocs de contrôles

- Blocs SACK (fonctionnent de la même manière que TCP)
- Blocs HEARTBEAT/HEARTBEAT_ACK (conserver la session active)
- Blocs ERROR : Information de problèmes ou d'erreurs de connexion
- Blocs d'ouverture et de fermeture de connexion
INIT/INIT_ACK/COOKIE_ECHO/COOKIE_ACK /
SHUTDOWN/ABORD/SHUTDOWN_ACK

Format des paquets SCTP

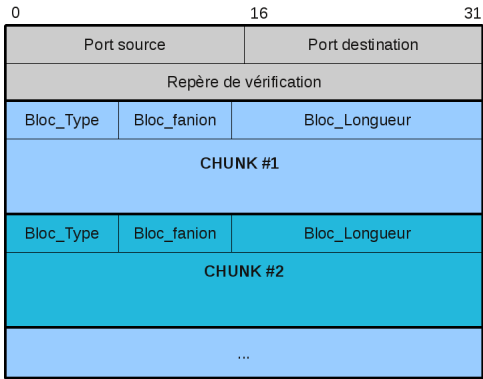


FIG.: Un paquet SCTP contient un nombre variable de blocs (chunks)

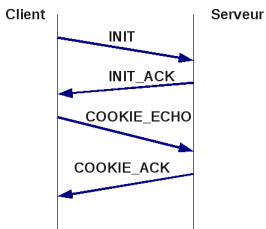
Présentation de SCTP (3)

Initiation de la connexion

Poignée de main effectuée par 4 échanges.

Introduction d'un "cookie" afin d'éviter l'attaque par deni de service

Allocation des ressources de la connexion au moment de la réception de COOKIE_ECHO



Présentation de SCTP (4)

Fermeture de la connexion

Deux méthodes sont à disposition :

- Fermeture courtoise : graceful shutdown (SHUTDOWN)
- Fermeture non polie : abortive shutdown (ABORT)

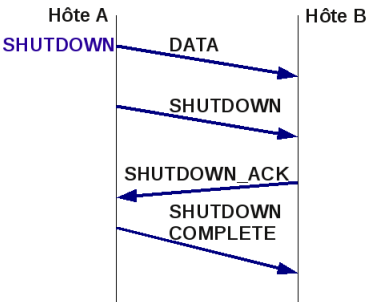


FIG.: Fermeture courtoise

Présentation de DCCP (1)

DCCP

Protocole orienté message avec contrôle de congestion.
Plusieurs choix de contrôle possibles.
Pas de retransmission en cas de perte ou erreur.
Mais connaissance du problème.

Plusieurs phases

- Initiation de la connexion
- Transfert de données
- Fin de la connexion

Présentation de DCCP (1)

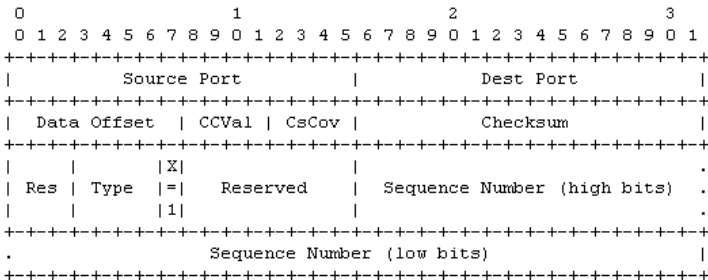


FIG.: L'entête DCCP - Source : RFC 4340

Présentation de DCCP (2)

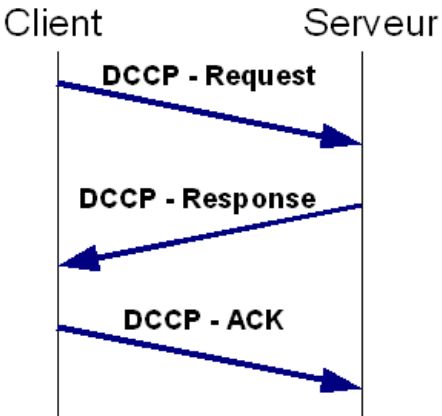


FIG.: Initiation d'une connexion DCCP

Présentation de DCCP (3)

Les requêtes DCCP

- **DCCP-Request :**
Spécifie les ports utilisés, l'application auquel on se connecte et le contrôle de congestion utilisé.
- **DCCP-Response :**
Réponse du serveur indiquant les options acceptées.
- **DCCP-ACK :**
Contient un numéro d'acquittement.

Présentation de DCCP (4)

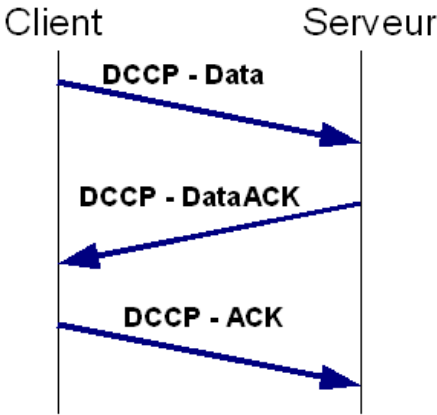


FIG.: Transfert de données DCCP

Présentation de DCCP (5)

Les requêtes DCCP

- **DCCP-Data :**
Paquet de base contenant des données.
- **DCCP-DataACK :**
Contient un numéro d'acquittement et des données de niveau applicatif.

Présentation de DCCP (6)

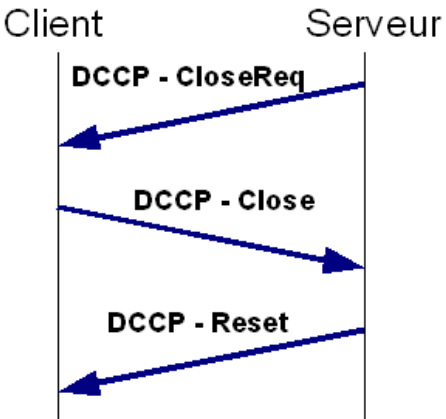


FIG.: Fin d'une connexion DCCP

Présentation de DCCP (7)

Les requêtes DCCP

- **DCCP-CloseReq :**
Demande au client une fermeture de la connexion.
- **DCCP-Close :**
Demande / confirmation de la fermeture de la connexion.
- **DCCP-Reset :**
Fermeture de la connexion. Est aussi utilisé en cas de problème (mauvais ports, mauvaises options, ...).
- **DCCP-Sync et DCCP-SyncACK**
Paquets utilisés après une suite de paquets perdus afin de rétablir la connexion.

Plan

- 1 Motivations
- 2 Les protocoles SCTP/DCCP
- 3 Propriétés**
 - Détection des pertes
 - Contrôle de congestion
 - Multidiffusion
 - Flux multiples
 - Récapitulatif des critères
- 4 Performances
- 5 Bilan

Détection des pertes avec DCCP

DCCP - Détection des pertes

- Détection effectuée grâce aux numéros de séquence.
- Numéros définis lors de l'initialisation de la connexion (DCCP-Request et DCCP-Response).
- Envoi de n'importe quel paquet DCCP → numéro de séquence + 1.
- DCCP-ACK contient le numéro de séquence du dernier paquet reçu mais possède son propre numéro de séquence.

Une perte est détectée lorsque l'acquittement d'un paquet n'est pas reçu.

Détection des pertes de messages avec SCTP

Identification des blocs

- Chaque bloc de données (DATA chunk) est identifié par son TSN (Transmission Sequence Number)
- Les TSN sont consécutifs

Acquittement des messages par SACK

- Envoi du dernier TNS reçu en séquence. (TSN Ack cumulé)
- Acquittement sélectif grâce aux Gap Ack
- Envoi des TNS dupliqués

Conséquences

- Seuls les blocs perdus sont réémis
- Gains de performances face à TCP

Détection des pertes de messages avec SCTP

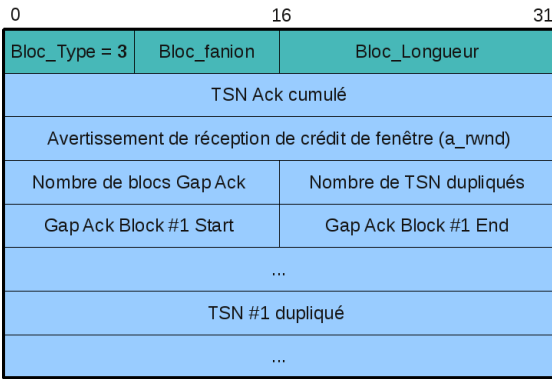


FIG.: Bloc d'acquiesement sélectif (SACK)

Congestion du réseau avec DCCP

2 types de contrôle de congestion

- TCP-like Congestion Control
 - Adapté en cas de besoin de bande passante
- TCP-Friendly Rate Control
 - Utilisé pour garder un taux de transfert assez constant

Optimisés pour des paquets de taille fixe envoyés dans un intervalle de temps variable.

TCP-like Congestion Control

Définitions

- *pipe* : nombre de paquets sur le réseau.
- *cwnd* (congestion window) : fenêtre de contrôle de congestion. Correspond au nombre maximum de paquets autorisé sur le réseau.
- *ssthresh* (slow-start threshold) : seuil de passage du mode slow-start au mode normal.

Condition à respecter

$pipe < cwnd$. Dans le cas contraire on ne peut plus envoyer de paquet.

TCP-like Congestion Control

Envoi d'un paquet : $pipe + 1$.

Reception d'un ACK : $pipe - 1$.

A la réception de 3 ACK étant supérieurs à un paquet : paquet perdu, $pipe - 1$.

Timeout : $pipe$ remis à zéro.

Une fois un paquet considéré perdu, l'éventuelle réception de son ACK ne décrémente pas $pipe$.

TCP-like Congestion Control

Mode slow-start

- Condition : $cwnd < ssthresh$
- Tous les 2 ACKs reçus $\rightarrow cwnd + 1$
- Augmentation du débit très rapide

Mode "normal"

- Condition : $cwnd \geq ssthresh$
- Chaque fenêtre de paquets bien reçue par le destinataire $\rightarrow cwnd + 1$

TCP-Friendly Rate Control

Mode slow-start

- Le débit est environ doublé à chaque RTT (Round-Trip Time)
- S'arrête lors de la perte d'un paquet
- A l'initialisation : 2 paquets envoyés par RTT

Mode "normal"

- Le débit est calculé en fonction du RTT des paquets
- A la réception d'un paquet ACK, le client calcule une valeur *nofeedback*
- Si aucun paquet ACK n'est reçu pendant ce temps alors le débit est divisé par deux

Contrôle de congestion avec SCTP

Au récepteur

Le récepteur indique le nombre d'octets qu'il est prêt à recevoir
Champ Advertised Receiver Window Credit (a_rwnd)

Dans le réseau

Comme pour TCP, utilisation des algorithmes Slow Start, Fast Retransmit et le Fast Recovery.



Multidiffusion

Possibilité d'utiliser plusieurs adressages pour le même destinataire
Envoi des blocs DATA sur 1 seul lien (adresse primaire)
Tolérance au pannes réseaux

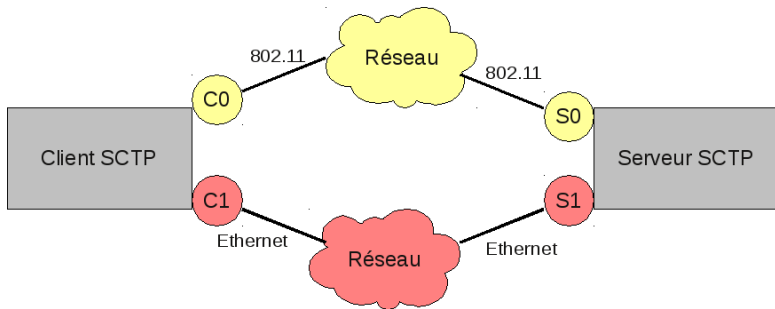
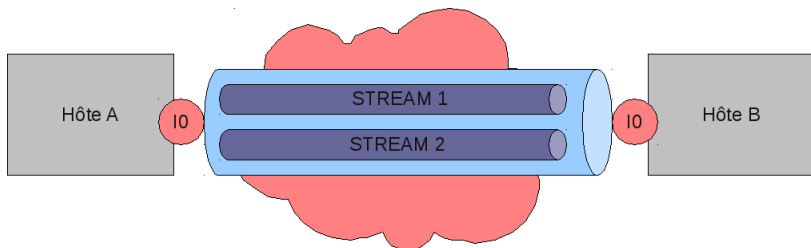


FIG.: La détection de la perte de connexion se fait grâce à des blocs de type HEARTBEAT

Flux multiples

- Différents flux dans un même flux (ex : images et code html téléchargé en même temps)
- Identification d'un flux dans le bloc : Stream sequence number



Récapitulatif des critères

Caractéristique	UDP	TCP	SCTP	DCCP
Orienté connexion	non	oui	oui	oui
Transport fiable	non	oui	oui	non
Transport non fiable	oui	non	oui	oui
Messages bornées	oui	non	oui	oui
Arrivée ordonnée des messages	non	oui	oui	non
Arrivée non ordonnée des messages	oui	non	oui	oui
Vérification des données	Oui (16 bits)	Oui (16 bits)	Oui (32 bits)	Oui (16 bits)
Contrôle de congestion	non	oui	oui	oui
Flux multiples	non	non	oui	non
Propriété de multidiffusion	non	non	oui	non
Algorithme de Nagle	non	oui	oui	non

FIG.: source : wikipedia

Plan

- ① Motivations
- ② Les protocoles SCTP/DCCP
- ③ Propriétés
- ④ Performances**
- ⑤ Bilan
- ⑥ Démonstrations

Performances avec SCTP

- Performances brutes en retrait face à TCP
- La navigation Internet pourrait être optimisée

Plan

- 1 Motivations
- 2 Les protocoles SCTP/DCCP
- 3 Propriétés
- 4 Performances
- 5 Bilan**
 - DCCP
 - SCTP
- 6 Démonstrations

Sources

DCCP

- http://en.wikipedia.org/wiki/Datagram_Congestion_Control_Protocol
- <http://www.bortzmeyer.org/4340.html>
- <http://www.linuxfoundation.org/en/Net:DCCP>
- http://www.linuxfoundation.org/en/Net:DCCP_Testing
- <http://wand.net.nz/iam4/dccp>

RFC

- <http://tools.ietf.org/html/rfc3448>
- <http://tools.ietf.org/html/rfc4340>
- <http://tools.ietf.org/html/rfc4341>
- <http://tools.ietf.org/html/rfc4342>

Sources

SCTP - Sites internet

- <http://www.ibm.com/developerworks/linux/library/l-sctp>
- <http://sawwww.epfl.ch/SIC/SA/SPIP/Publications/spip.php?article242>
- <http://tdrwww.exp-math.uni-essen.de>
- <http://www.linux-france.org>
- <http://datatag.web.cern.ch>
- Wikipedia

SCTP - Rapports

- Henrik Osterdahl : A comparison of TCP and SCTP performance using the HTTP protocol
- Jayesh Rane, Nitin Kumbhar, Kedar Sovani : Exploiting Multi-homing in SCTP for High Performance

Plan

- 1 Motivations
- 2 Les protocoles SCTP/DCCP
- 3 Propriétés
- 4 Performances
- 5 Bilan
- 6 **Démonstrations**
 - SCTP : Analyse du multiflux
 - SCTP : Multihoming

Démonstrations avec SCTP

- Utilisation de machines sous Linux Noyau 2.6 disposant de l'implémentation LKSCTP.
- lksctp_tools et programme utilisant la librairie libsctp



SCTP : Analyse du multflux

Le serveur envoi sur 2 flux :

- Flux 1 : date système
- Flux 2 : texte 'Hello %n' où n incrémente

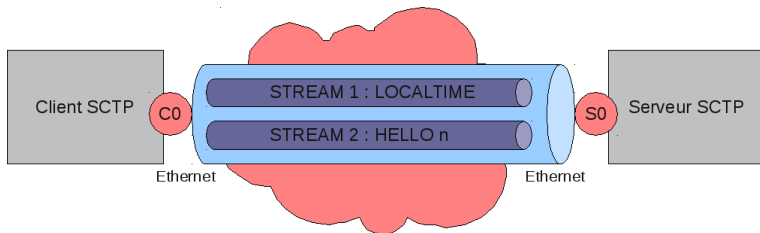


FIG.: Schéma de la démonstration

SCTP : Multihoming

- Utilisation de `sctp_darn` avec plusieurs liens réseaux
- Perte d'un lien

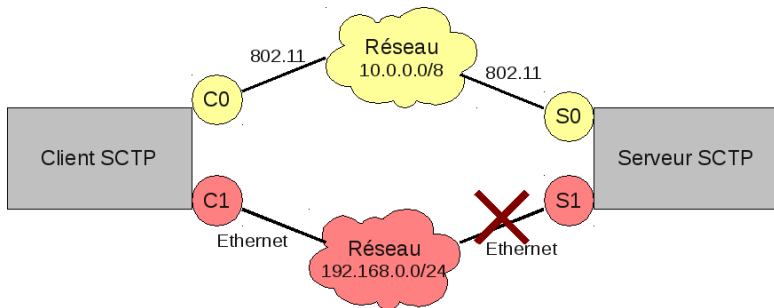


FIG.: Schéma de la démonstration