

LINGI2315 Design of embedded and real-time systems

Projet: relai dhcp sur PIC

Arnaud Delmotte INFO22MS
Arnaud Schils SINF21MS
Sébastien De Fauw INFO21MS

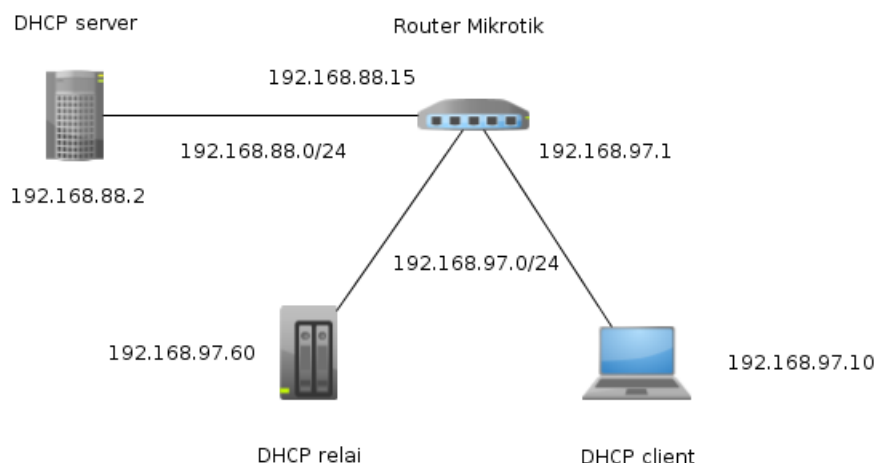
18 mai 2013

1 Documentation pour l'utilisateur et l'installateur

1.1 Exemple d'utilisation du relai

Nous avons branché le relai au routeur mikrotik, sur son interface « local ». Le serveur DHCP (dhcpd) a été branché sur l'interface « gateway » de ce routeur. Le réseau local auquel est attaché le serveur DHCP a un subnet différent de celui auquel est attaché le relai.

Le serveur DHCP est configuré pour attribuer des adresses aux deux subnets : dans notre cas 192.168.88.0/24 (subnet local du DHCP) et 192.168.97.0/24 (subnet local du relai).



Le relai devrait être utilisé dans un environnement similaire à celui décrit ci-dessus, mais ce n'est qu'un exemple. Nous avons implémenté un « vrai » relai, suivant rigoureusement les instructions de l'énoncé. Il devrait fonctionner dans n'importe quel environnement correspondant à l'utilisation que l'on fait d'un tel dispositif.

Une fois le relai allumé, il n'y a rien à configurer. La spécification de l'adresse IP du serveur DHCP et de la gateway se fait dans le fichier source avant compilation et transfert sur PIC. L'adresse MAC de la gateway est trouvée automatiquement par une requête ARP.

1.2 Compiler, installer sur le PIC et tester le programme

Dans le fichier MainDemo.c l'adresse IP du serveur DHCP doit être encodée sous forme de nombres décimaux dans les define DHCP_IP8, DHCP_IP16, DHCP_IP24 et DHCP_IP32. DHCP_IP8 doit contenir les 8 premiers bits de l'adresse IP (ex : 192) DHCP_IP16 les 16 bits suivants (ex : 168) et ainsi de suite.

Dans le fichier **include/TCPIPConfig.h** il faut modifier si nécessaire l'IP du PIC, le mask et l'IP de la gateway dans les define correspondant.

Bien que le code du relai se trouve dans le fichier MainDemo.c, il est nécessaire de recompiler tous les fichiers du dossier soumis car certains fichiers de la librairie fournie ont été modifiés. Il suffit de d'exécuter le makefile et de transférer le fichier .hex obtenu sur le PIC.

Le test du programme peut se faire simplement en tentant d'utiliser le relai DHCP suivant la configuration décrite dans la première section. La LED0 est sensée clignoter toutes les secondes.

2 Documentation pour le programmeur

2.1 Choix structurels et d'implémentation

Un des choix que nous avons fait est d'identifier la source d'un message DHCP (serveur ou client) en fonction de l'adresse IP et non en fonction du type de message (Offer, Request, Ack,...) pour des raisons évidentes de sécurité. Il ne faut pas que n'importe quel client puisse jouer le rôle du serveur DHCP.

Nous devons donc récupérer l'adresse IP source du paquet UDP reçu. Or l'API UDP actuelle fournie ne permet pas de le faire, et mixer l'utilisation des API des couches IP (couche network) et UDP (couche transport) se révèle être assez périlleux. Nos expériences ont montré que le PIC se comporte de manière hasardeuse si on utilise des fonctions des API IP et UDP sur un même paquet UDP reçu.

Nous avons donc créé une nouvelle fonction **StackTaskHelp** dans le fichier **TCPIP_Stack/StackTsk.c**. Nous l'utilisons à la place de la fonction **StackTask** d'origine. Elle nous permet de récupérer l'adresse IP source du paquet en train d'être traité.

Plus généralement nous avons suivi d'assez près notre diagramme ASG. Les différents cas de la machine à état sont gérés dans différentes fonctions. Nous avons essayé de minimiser la duplication de code en créant des sous fonctions utilisées dans plusieurs états.

Les contraintes parallèles du diagramme sont vérifiées dans la boucle principale du programme. Nous n'utilisons pas les interruptions associées au Timer1 car celles-ci sont déjà utilisées dans la librairie pour le réseau. Nous utilisons donc les fonctions du fichier **TCPIP_Stack/Tick.c** pour récupérer le temps écoulé.

2.2 Quelle est la fonction du programme (spécification) ?

Le programme implémenté répond aux requirements de l'énoncé. Plus concrètement :

- Le relai implémente la fonctionnalité d'un simple relai : transfert des Discovery et Request reçus du client au serveur DHCP et des Offer et Ack du serveur au client.
- Une fois un ACK reçu depuis le serveur, le client concerné par celui-ci est ajouté au POOL du relai. Notons que le lease time est modifié dans le paquet avant d'être passé au client : il doit être de 5 minutes.
- Lorsque ce client effectue de nouvelles Request alors qu'il est déjà enregistré dans le POOL, le relai lui répond directement, sans en informer le serveur DHCP. A chaque Request reçue, le compteur d'échéances manquées est remis à zero. Si le client manque 5 échéances il est retiré du POOL et un message Release est envoyé au serveur DHCP.
- Si la deadline du lease time du serveur va bientôt expirer (< 5 minutes), une nouvelle request est envoyée du relai au serveur. Une fois le nouveau ACK reçu, les informations du POOL sont mises à jour (deadline, ...).

2.3 Quels sont les détails techniques du PIC ou de la carte qu'il faut avoir en tête pour comprendre le programme ?

L'élément principal à comprendre est l'utilisation de l'API réseau fournie avec le PIC. Par exemple, il faut bien savoir coordonner l'utilisation simultanée de différents sockets. En effet, les différentes fonctions s'appliquent sur le socket actif. Le socket actif change suivant l'appel de certaines fonctions de la librairie.

Il faut également être conscient que le type int est sur 16bits sur cette plateforme. Une compréhension basique de l'utilisation des timers peut aussi être utile.

3 Conclusion

Nous avons trouvé ce projet intéressant et amusant car il comporte à la fois une partie systèmes embarqués et une partie réseau. Il nous permet de comprendre plus concrètement comment sont conçus les différents dispositifs présents sur les réseaux informatiques (routeurs, switches et diverses middleboxes). Notons que avoir à notre disposition un environnement de développement comportant une interface JTag Uart (comme pour la partie hardware du cours) aurait largement facilité la détection de bug lors de la réalisation du projet.