ROCE

PROTOCOLE DE CONNEXION

DÉVELOPPEURS:
Lasseye Benjamin
Dubois Clément
Gasnier Corentin
Mari Lucas
Maitrot Maxime

I - Contexte

Le cœur de notre projet consiste à envoyer à une machine Windows des commandes depuis un mobile Android. Il est important de préciser que le protocole mis en place doit aussi être capable d'envoyer des informations au client Android depuis Windows en cas de changement de paramètres.
Pour réaliser cela, il nous faut établir un protocole de communication entre les deux appareils. Ce document vous présente une liste relativement exhaustive des différents protocoles capables de répondre à notre problématique.

II - RDP

Quand on cherche à communiquer avec Windows depuis une autre machine, on pense rapidement à RDP. En effet le Remote Desktop Protocol permet à un utilisateur disposant des paramètres de connexion RDP d'une session Windows de se connecter à celle-ci grâce à ses identifiants et ainsi accéder aux fichiers, à l'image et au son de cette session.
Bien qu'apprécié pour ses fonctionnalités de contrôle à distance relativement bien sécurisées, RDP reste un protocole lourd dont l'usage est limité par la qualité de la connexion internet de l'utilisateur distant. De plus, seul le presse-papiers de session permet l'échange d'informations entre la machine hôte et la machine à distance.

III - Les sockets

A - Socket unique

Un socket permet à deux appareils connectés au même réseau de communiquer via un tunnel établie sur un port compris entre 1025 et 65535. Ce système repose sur la logique client-serveur. Dans notre cas, on définit la machine Windows comme étant notre serveur hôte et le mobile Android comme client unique.
Le protocole fonctionne comme suit :

1. Lancement d'un socket serveur en écoute sur le port XXXX d'un réseau local (IPv4).
2. Lancement d'un socket client qui pointe sur le port XXXX du réseau local.
3. Poignée de main (Handshake) entre le client et le serveur pour établir la connexion.
4. Création de deux tampons (Buffers) d'écoute à chaque extrémité du socket.

Se faisant, chaque appareil est capable de lire ce que lui envoie l'autre sur son tampon.
Ce protocole est peu coûteux en ressource et permettrait de répondre à la problématique. Mais attention, il est important de préciser que cette méthode ne permet pas d'envoyer deux messages simultanés dans les deux sens. Cette particularité pourrait être à l'origine de bugs majeurs.

B - Sockets doubles

Pour assurer la bilatéralité de la communication on peut penser à une amélioration du protocole précédent en ajoutant un deuxième socket. Ce socket supplémentaire sert à implémenter un socket unique par sens de communication (d'Android à Windows et de Windows à Android) et le protocole s'établit comme suit :

1. Lancement du socket serveur 1 en écoute sur le port 1 d'un réseau local (Windows)
2. Lancement du socket serveur 2 en écoute sur le port 2 d'un réseau local (Android)
3. Lancement du socket client 1 qui pointe sur le port 1 du réseau local (Android)
4. Poignée de main (Handshake) entre les extrémités du socket 1
5. Création d'un tampon (Buffer) sur Windows en écoute d'Android
6. Lancement du socket client 2 qui pointe sur le port 2 du réseau local (Windows)
7. Poignée de main entre les extrémités du socket 2
8. Création d'un tampon (Buffer) sur Android en écoute de Windows

Ainsi on rend autonome les deux sens de la communication en levant les conflits en cas d'un envoi de messages bilatéraux et simultanés .

C - Pour aller plus loin : SSL

Même si la solution des sockets doubles semble fiable, elle n'en garde pas moins des lacunes de sécurité. Pour pallier ce problème, il peut être judicieux de remplacer les simples socket par des tunnels SSL (Secure Socket Layer).
Pour faire simple, il s'agit de reprendre le protocole précédent et de complexifier la poignée de main (Handshake) entre les deux appareils. Grâce à la génération d'un certificat SSL, on va pouvoir assurer que la connexion ne provient pas de n'importe quel appareil mais bien de l'utilisateur de ROCE.

IV - USB

Pour établir une connexion filaire entre Windows et Android, il est nécessaire de passer par l'USB. Android donne accès à un port ADB (Android Debug Bridge) permettant la communication entre Android et une autre machine. Ce protocole nécessite que le client, ici Windows, dispose des packages correspondant et que le serveur, ici le mobile Android dispose de l'option active "Débogage USB".
Dans ce cas particulier c'est la machine Windows qui lance le protocole définit comme suit :

1. Listing des appareils compatibles ADB branchés sur les ports USB.
2. Lancement du port ADB sur l'appareil choisi.

Ce protocole gère automatiquement les conflits d'envoi de message simultanés

V - Conclusion

Dans le cadre de notre projet, nous avons décidé de nous concentrer sur la solution du **double socket**. Même si la solution SSL permet d'augmenter la sécurité du produit, la contrainte de temps nous oblige à assurer un produit fonctionnel avec un compris sur une technologie légèrement moins sécurisée mais plus facile à appréhender.
En plus de la connexion via double socket et de sorte à rendre possible son utilisation sans

réseau local, ROCE devra être implémenter de sorte à pouvoir intégrer l'utilisation du port ADB à l'avenir.