Laboratório #05

WiFi Direct

Objectivos

- Introduzir o WiFi Direct.
- Introduzir o emulador de rede Termite WiFi Direct.

Exercício I: Encontrar dispositivos próximos usando WiFi Direct

O objectivo deste exercício é demonstrar como construir aplicações Android que possam detectar a presença de dispositivos próximos usando WiFi Direct. Além disso, mostrar como testar tais aplicações em redes virtuais WiFi Direct emuladas usando o emulador de rede Termite. Para este exercício, é preciso baixar os seguintes pacotes:

- `Termite-Cli.tgz`: Uma distribuição binária da ferramenta cliente Termite.
- `Termite-WifiP2P-API.tgz`: O binário da API Termite WiFi Direct.
- `Termite-WifiP2P-PeerScanner.tgz`: Uma aplicação simples que verifica dispositivos dentro do alcance WiFi do dispositivo.

1. Explorar a aplicação PeerScanner

Baixe a aplicação `Termite-WifiP2P-PeerScanner` e abra-o no Android Studio. Esta aplicação ilustra as etapas básicas para utilizar a API WiFi Direct. Observe, entretanto, que, para operar com o Termite, o PeerScanner utiliza uma versão modificada da API Android WiFi Direct, que pode ser encontrada na pasta "libs".

Compile a aplicação e execute-a no emulador. Em seguida, inspecione o código-fonte da aplicação e interprete-a consultando o guia da API Termite WiFi Direct (`Termite-API.pdf`). Em particular, identifique a finalidade dos seguintes fragmentos de código-fonte:

- receptor de radiodifusão,
- declaração de serviço no manifesto da aplicação,
- código de inicialização no método de criação de actividade,
- ouvintes associados a botões.

Para testar esta aplicação, precisamos emular um WiFi Direct virtual usando Termite. A seguir, explicamos como fazer isso.

joao.costa@isptec.co.ao

2. Instalar e executar o cliente Termite

1. Instalar o Termite: Baixe o cliente Termite e descompacte-o em um directório local. Em seguida, é necessário definir alguns atributos de configuração da instalação do Termite. No directório Termite-Cli, abra o arquivo `etc/platform/X/backends.conf`, onde `X` é sua plataforma de destino: `windows`, `linux` ou `mac`. Por padrão, este ficheiro tem a seguinte aparência:

```
{
    "backends" : [
        {
            "id" : "avd",
            "connector" : "avd",
            "config" : {
                "sdk" : "/Users/joaojdacosta/Library/Android/sdk",
                "vmi" : "Nexus_5_API_21_x86"
            }
        }
    ]
}
```

É necessário actualizar os atributos SDK e VMI. O atributo `sdk` deve ser actualizado com o caminho correcto do Android SDK. Uma maneira simples de determinar isso é abrir o Android Studio e, a seguir, abrir as configurações do projecto. (Observação: no Windows, deve escrever o caminho do SDK usando barras invertidas duplas, por exemplo, "`c:\\my\\path\\to\\sdk`".) Por enquanto, ignore o atributo `vmi`.

2. Executar a ferramenta termite: Abra uma janela de terminal e mude o directório para a pasta `Termite-Cli`. Defina a variável de ambiente `TERMITE_CLI_PATH` para apontar para o local do módulo Termite CLI, ou seja, este caminho. No `Linux` ou `Mac OS`, uma maneira fácil de fazer isso é actualizar o ficheiro `env/platform/{linux,mac,windows}/env_setup.sh` e então, na janela do terminal, executar o comando: `source env/platform/{linux ,mac,windows}/env_setup.sh`. Naturalmente, também pode criar seu próprio ficheiro "`env`". Em seguida, execute o script `./termite.sh`. No Windows, o procedimento é diferente. Para configurar a variável de ambiente, utilize o comando "`set TERMITE_CLI_PATH=...`". A variável de ambiente `TERMITE_PLATFORM` também deve ser configurada de acordo com: `windows`, `linux` ou `mac`. Para executar o cliente Termite, execute o ficheiro em lote `termite.bat`. Se tudo correr bem, espera-se o seguinte resultado:

```
Termite Testbed
  Working Directory = /Users/joaojdacosta/Desktop/Termite-Cli
  Type "help" or "h" for the full command list

avd:simplechat>
```

Termite está pronto para aceitar seus comandos.

3. Emular o movimento do nó em uma rede virtual WiFi Direct

A próxima etapa é testar o `PeerScanner` em uma rede virtual WiFi Direct emulada no Termite. A rede virtual é muito simples. É composto apenas por dois nós: A e B. A é um dispositivo Android e B é um beacon WiFi. Os nós estão, em primeiro lugar, localizados separados uns dos outros e, num segundo estado, tornam-se acessíveis dentro dos seus respectivos alcances de sinal WiFi. Queremos simular esta rede virtual conforme representado na figura abaixo, de modo que ao executar o `PeerScanner` no nó A, B não seja detectado no estado 1, mas seja detectado no estado 2:

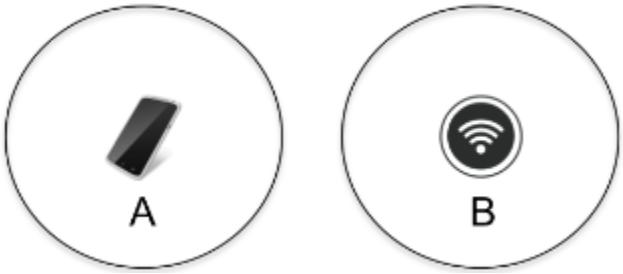

Figura 1: Nós virtuais A e B inacessíveis (estado 1).

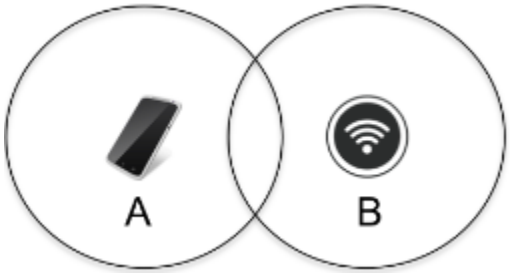

Figura 2: Nós virtuais A e B acessíveis (estado 2).

1. Criar os nós virtuais: No Termite, crie dois dispositivos virtuais, um para A e outro para B, executando o comando `newdevice`. Em seguida, liste os dispositivos da rede virtual actual com o comando `list`.

```
avd:simplechat> newdevice A
avd:simplechat> newdevice B
avd:simplechat> list devices
A 0.0.0.0:1 0.0.0.0:1 0.0.0.0:1
B 0.0.0.0:2 0.0.0.0:2 0.0.0.0:2
```

Por enquanto, ignore os endereços IP exibidos em "`list devices`". Esses endereços serão usados para comunicação entre dispositivos Android emulados. Pode excluir um dispositivo usando um comando específico (`deletedevice [nome]`).

2. Associar um emulador Android ao nó A: A seguir, queremos associar um emulador Android ao nó A. Este emulador será o host da aplicação `PeerScanner`. Para realizar esta associação, precisamos realizar diversas etapas. Primeiro, certifique-se de que um emulador

Android esteja em execução. O Termite deve então detectá-lo executando o comando `list emus`:

```
avd:simplechat> list emus
e1 => online
    name: emulator-5554
```

Em seguida, é necessário configurar os endereços de rede do emulador. Para que a aplicação `PeerScanner` consiga se comunicar com outros dispositivos virtuais e para que a ferramenta Termite se comunique com os emuladores, é necessário realizar algumas operações de redirecionamento de porta. A ferramenta Termite pode ajudar nessa tarefa. Primeiro, precisa atribuir endereços de rede a cada emulador individualmente usando o comando `assignaddresses` da seguinte forma:

```
avd:simplechat> assignaddr e1
avd:simplechat> list emus
e1 => netok
    name: emulator-5554
    addr: [ avaddr = 192.168.0.1:10001, araddr = 10.0.2.2:10011, cvaddr =
127.0.0.1:9001, craddr = 127.0.0.1:9011]
```

A partir desta listagem, o que é importante entender por enquanto é o atributo "`avaddr`" que significa: endereço virtual da aplicação. Essencialmente, esses são os endereços virtuais que o Termite emula para a aplicação em execução neste emulador específico. Por exemplo, para o emulador-5554, o endereço IP virtual visto pela aplicação é 192.168.0.1 e o número da porta onde a aplicação estará a escutar é 10001. A aplicação `PeerScanner` escutará conexões nesta porta. Esses endereços são especificados no ficheiro de configuração do Termite `etc/netprofiles.conf`.

A seguir, devemos vincular o emulador ao dispositivo virtual A. Isso é feito usando o comando `binddevice` conforme ilustrado a seguir:

```
avd:simplechat> binddevice A e1
avd:simplechat> list devices
A 192.168.0.1:10001 10.0.2.2:10011  127.0.0.1:9011
B 0.0.0.0:2 0.0.0.0:2 0.0.0.0:2
```

O comando `list devices` fornece detalhes da sua rede virtual. Na emulação de rede, estes são os nomes a serem usados quando se refere aos nós virtuais, e não aos nomes dos emuladores. Pode ver que após a operação de ligação, os endereços virtuais do dispositivo A foram alterados. Agora estamos prontos para instalar a aplicação no emulador e emular os estados da rede representados na figura acima.

3. Emule a rede virtual: No Android Studio, instale a aplicação `PeerScanner` no emulador. Em seguida, pressione o botão "`WiFi On`" para iniciar o serviço WiFi Direct na aplicação. No console do Termite, execute `ping` e verifique se o nó está online:

```
avd:simplechat> ping
A 127.0.0.1 9011  ONLINE
B 0.0.0.0 2 OFFLINE
```

Isso significa que o serviço Termite está a aguardar na porta 9011 do host local. Essas portas são redirecionadas internamente para a porta 9001 dentro do emulador. Naturalmente, como nenhum emulador Android está vinculado ao nó B, B parece estar offline.

O estado inicial da rede é definido de forma que cada nó da rede fique isolado um do outro. Se pressionar "`In Range`", a lista ficará vazia. Isto corresponde ao estado 1 (ver figura acima).

A partir do cliente Termite é possível modificar a topologia da rede disparando determinados eventos. O primeiro evento relevante que precisamos saber é o "movimento", que diz a um determinado nó para se mover para a vizinhança de outro nó, permitindo que se torne acessível. Assim, para emular o estado 2, podemos instruir o Termite a "mover" A para perto do nó B conforme mostrado a seguir (ao executar os comandos a seguir, certifique-se de que o emulador esteja visível para que você possa ver um `toast` sendo apresentado):

```
avd:simplechat> move A (B)
avd:simplechat> list neighbors
A => B
B => A
avd:simplechat> commit
A 127.0.0.1 9011  SUCCESS
B 0.0.0.0 2 FAIL
```

Verifique se as mensagens do `toast` apareceram dizendo "`Peer list changed`". Clique em "`In range`" para ver quais pares estão disponíveis. Se clicar em "`In network`" a lista ficará vazia porque a rede ainda não está formada. Para formar uma rede é necessário criar um grupo, que veremos na próxima secção.

Na sequência de comandos realizados anteriormente, observe que as alterações na topologia da rede virtual são realizadas localmente no console do Termite. Para propagar as informações de topologia para os nós, deve executar o `commit`. Para separar os nós, pode simplesmente emitir o comando "`move A ()`" seguido de `commit`.

4. Scripts: o Termite oferece suporte a scripts. Em outras palavras, os comandos executados na linha do console podem ser carregados a partir de um ficheiro de script. Os scripts estão localizados na pasta "scripts" da distribuição do cliente Termite. A nomenclatura do ficheiro de script segue a convenção: "`s.termite`", onde "s" é o nome do script. Para executar o script no console, execute `load s`. Verifique, por exemplo, o script "example.termite" e execute-o no console. O comando `wait` espera até que o utilizador pressione uma tecla ou espera até que um determinado tempo limite expire se um valor de tempo limite for fornecido como argumento.

4. Exercícios de aplicação

Para aprender melhor todos esses novos conceitos, emule os seguintes cenários:

- **Cenário 1.** Existem quatro nós: um dispositivo Android (nó N1) e três beacons (B1, B2 e B3). Simule o cenário onde os três beacons são colocados sequencialmente em uma rua, separados uns dos outros, e um utilizador que carrega o dispositivo N1 passa por cada beacon, um por um. Escreva um script que automatize o movimento realizado pelo utilizador. Use o comando wait para introduzir atrasos conforme o utilizador passa por cada beacon.
- **Cenário 2.** Existem dois dispositivos Android: nó N1 e N2. Simule o cenário onde eles se aproximam e se afastam depois de algum tempo. Neste caso, é necessário lançar um segundo emulador e associá-lo ao nó N2.

Exercício II – Formação de grupos WiFi e troca de mensagens

1. Explore a aplicação MsgSender

Baixe a aplicação Android `Termite-WifiP2P-MsgSender` e abra-o no Android Studio. Esta aplicação ilustra como configurar grupos WiFi e trocar mensagens. Permite que os dispositivos formem um grupo WiFi e permite que seus respectivos utilizadores troquem mensagens. Para enviar uma mensagem para outro dispositivo, o utilizador deve escrever o endereço IP virtual do nó de destino, pressionar "`Connect`", a seguir escrever uma mensagem e pressionar "`Send`".

Compile a aplicação e execute-a no emulador. Em seguida, inspecione o código-fonte da aplicação e interprete-o consultando o Guia da API Termite WiFi Direct (`Termite-API.pdf`). Em particular, identifique a finalidade dos seguintes fragmentos de código-fonte:

- `listeners` (ouvintes) associados com o grupo,
- `asynctasks` responsáveis por enviar e receber mensagens.

Para testar esta aplicação, precisamos emular um grupo WiFi Direct virtual usando o Termite, conforme explicamos a seguir.

2. Emular grupos P2P em uma rede WiFi Direct virtual

Para testar a aplicação `MsgSender`, nossa rede alvo é mostrada abaixo. Neste caso, temos dois dispositivos Android virtuais (A e B), ambos a executar a aplicação `MsgSender`. No Estado 3, um grupo é formado entre os nós, sendo A o proprietário do grupo (GO). Neste ponto, eles recebem endereços IP virtuais e podem abrir conexões TCP/IP regulares com base em seus respectivos IPs virtuais.

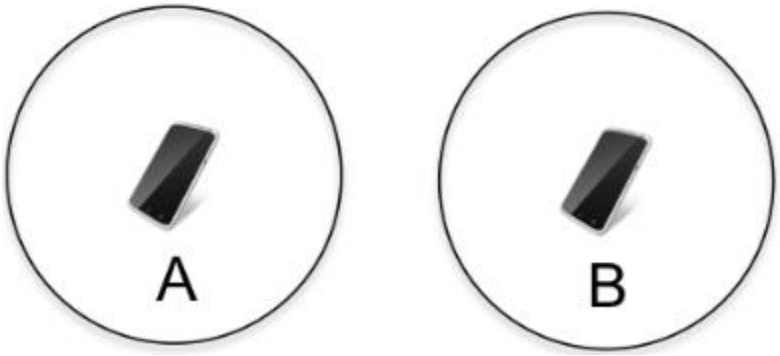

Figura 3: Nós virtuais A e B inacessíveis (estado 1).

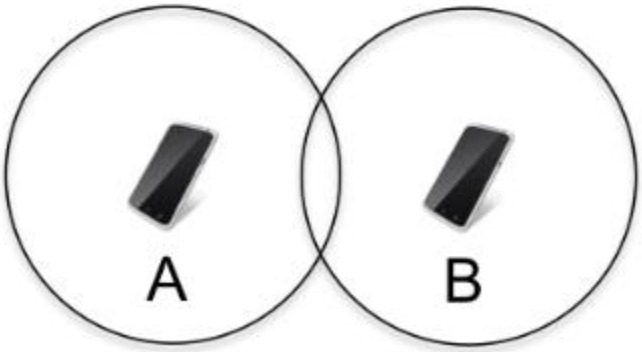

Figura 4: Nós virtuais A e B acessíveis (estado 2).

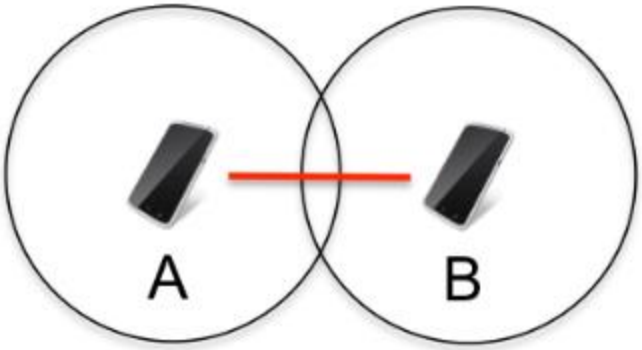

Figura 5: Nós virtuais A e B formam grupo (estado 3).

1. Emular a formação de grupo e a comunicação ponto a ponto: Siga os passos indicados no Exercício I para emular uma rede virtual até o estado 2. Em seguida, forme um grupo Wifi contendo ambos os nós A e B. O nó A será o proprietário do grupo (GO) da rede. Execute estes comandos:

```
avd:simplechat> creategroup A (B)
avd:simplechat> list groups
A => B
avd:simplechat> commit
```

Observe os novos `toast` lendo "`Network membership changed`" e "`Group owner changed`" para o nó A. Em A, aprenda o endereço de B (`192.168.0.2`), digite-o na caixa de texto de edição de A e clique no botão "`Connect`". Isto abrirá soquetes entre ambos os nós,

permitindo-lhes trocar mensagens de forma conversacional. Teste esse recurso. Em A, envie uma mensagem para B.

2. Emular destruição de grupo e quebras de comunicação: Agora, exclua o grupo e veja o que acontece. Para excluir o grupo, na linha de comando execute:

```
avd:simplechat> deletegroup A
avd:simplechat> commit
B 127.0.0.1 9021  SUCCESS
A 127.0.0.1 9011  SUCCESS
```

Também é possível emular nós se afastando e saindo do grupo sem destruir explicitamente o grupo. Para testar isso, recrie o grupo:

```
avd:simplechat> creategroup A (B)
avd:simplechat> list groups
A => B
avd:simplechat> commit
```

Então, em B, abra uma conexão com A. Agora, afaste os nós uns dos outros. Isso fará com que o grupo seja automaticamente destruído. No Termite, execute:

```
avd:simplechat> move A ()
avd:simplechat> list n
A =>
B =>
avd:simplechat> list groups
A =>
avd:simplechat> commit
B 127.0.0.1 9021  SUCCESS
A 127.0.0.1 9011  SUCCESS
```

3. Adicionar membros a grupos pré-existentes: Na listagem anterior, veja que o grupo de propriedade de A ainda existe. Para repetir o cenário em que ambos os nós podem se comunicar, pode novamente adicionar B ao grupo de A:

```
avd:simplechat> move B (A)
avd:simplechat> joingroup B (A)
avd:simplechat> list groups
A => B
avd:simplechat> commit
A 127.0.0.1 9011  SUCCESS
B 127.0.0.1 9021  SUCCESS
```

Verifique se pode abrir novamente um canal entre A e B e fazer com que eles se comuniquem.

Dicas: Para reinstalar a aplicação, basta: instalá-la nos emuladores do Android Studio e no cliente Termite executar o commit, para propagar o estado actual da rede para os dispositivos. Não é necessário repetir todas as etapas de formação da rede do zero se a topologia de rede que

precisa emular já estiver na memória da ferramenta Termite. Para saber qual é o estado actual da rede, execute: `list network`.

3. Exercício de aplicação

Modifique a aplicação `MsgSender` para suportar um modo de operação conversacional, no qual os utilizadores podem trocar múltiplas mensagens na mesma conexão. Na versão actual do `MsgSender`, apenas uma mensagem pode ser enviada a um ponto remoto por conexão.

Apêndice: Dicas úteis

1. Importar a API Termite para o projecto

Para utilizar a API Termite em seu projecto, descompacte o ficheiro de biblioteca `Termite-WifiP2P-API.tgz` no subdiretório `libs` do seu projecto (por exemplo, `MyProject/app/libs`). Em seguida, adicione o seguinte código em seu ficheiro `MyProject/app/build.gradle`:

```
dependencies {
  compile(name:'Termite-WifiP2P-API', ext:'aar')
}
repositories{
  flatDir{
    dirs 'libs'
  }
}
```

2. Utilizar o conector Genymotion

Genymotion é um gestor de máquina virtual baseado em **VirtualBox**. Fornece funcionalidade semelhante ao **AVD**, mas tem desempenho muito melhor e consome menos recursos que o AVD. Para gerir emuladores Genymode no Termite, siga estas etapas:

1. Instalar o Genymotion: Primeiro, precisa instalar o `VirtualBox`. Em seguida, baixe o `Genymotion` e instale-o em sua máquina local.

2. Criar dispositivos virtuais: Inicie o Genymotion e crie um novo dispositivo virtual a partir da imagem "`Custom Phone - 5.0.0 - API 21 - 768x1280`". Atribua um nome, por ex. "`TVD - 5.0.0 - API 21 - 768x1280`". Altere algumas configurações do dispositivo virtual: defina a memória para 512 MB e marque "`Use virtual keyboard for input`". Em seguida, crie dois clones desta imagem de dispositivo virtual usando a linha de comando, conforme mostrado abaixo. Esses clones serão as imagens dos emuladores que serão lançados pelo Termite.

```
$ VBoxManage clonevm "TVD - 5.0.0 - API 21 - 768x1280" --name "TVD - 5.0.0 -
API 21 - 768x1280-Clone0" --register
  $ VBoxManage clonevm "TVD - 5.0.0 - API 21 - 768x1280" --name "TVD - 5.0.0
- API 21 - 768x1280-Clone1" --register
  $ VBoxManage list vms
```

Obs: é muito importante que o nome de cada clone siga a convenção *vdname*-Clone#, caso contrário o Termite não conseguirá gerar emuladores. A numeração do clone # começa em 0.

3. Configurar o Termite: Actualize o backend "genymotion" no ficheiro etc/platform/[mac|linux|windows]/backends.conf. Em particular, actualize os seguintes atributos:

- "sdk": o caminho completo para Android SDK
- "path": o caminho completo para Genymotion
- "vmiprefix": nome original do dispositivo virtual, neste caso: "TVD - 5.0.0 - API 21 - 768x1280"
- "numclones": o número de clones disponíveis, neste caso: 2

4. Actualizar o script init do Termite: Edite o ficheiro de configuração Termite-Cli/etc/termite.conf, e defina o atributo "init" como "simplechat-gm.termite", que corresponde ao nome de um script Termite localizado em Termite-Clique/etc/init.

5. Emular a rede: Reinicie o Termite e proceda conforme indicado anteriormente neste guia para emular uma rede virtual.

Links úteis adicionais

- https://nuno-santos.github.io/termite/wiki-docs/Home.html
- https://developer.android.com/develop/connectivity/wifi/wifip2p
- https://developer.android.com/reference/android/net/wifi/p2p/package-summary