

# Comunicação Móvel Inter-Grupo Baseada em TCP sobre Wi-Fi Direct\*

António Teófilo<sup>1,2</sup>, Diogo Remédios<sup>1,2</sup>, Hervé Paulino<sup>2</sup>, and João Lourenço<sup>2</sup>

<sup>1</sup> ISEL – Instituto Superior de Engenharia de Lisboa, IPL, Portugal

{ateofilo,dremedios}@deetc.isel.ipl.pt

<sup>2</sup> NOVA LINCS, Departamento de Informática, Faculdade de Ciências e Tecnologia

Universidade NOVA de Lisboa, Portugal

{herve.paulino,joao.lourenco}@fct.unl.pt

**Resumo** Neste artigo explora-se a utilização de dispositivos móveis Android, *non-rooted*, em cenários de comunicação dispositivo-a-dispositivo utilizando *WiFi-Direct*. As atuais topologias de comunicação inter-grupo com *WiFi-Direct* têm a limitação de requererem: a utilização de *broadcasts*, *multicasts*; ou que os nós tenham de desfazer as ligações e as refazer no sentido inverso; ou um n.º de nós que pode não estar disponível. Neste artigo propõe-se duas topologias de comunicação exclusivamente baseadas em ligações TCP: que tiram partido das características deste tipo de ligações; que apresentam melhor velocidade de comunicação; e que requerem um menor n.º de nós.

**Keywords:** Redes Móveis, Redes móveis Autónomas, Comunicação Dispositivo-a-Dispositivo, Wi-Fi Direct, Wifi, Android

## 1 Introdução

Como os atuais dispositivos móveis, podem comunicar por WiFi (WF), *WiFi-Direct* (WFD), e GSM/UMTS/LTE constata-se que eles podem ser ligados a uma rede externa ou estarem somente interligados entre si. De facto a tecnologia WFD permite formar redes de grupos autónomos de dispositivos, que permite a comunicação em caso de: colapso da infraestrutura de comunicação, resultante de uma excessiva sobrecarga de utilização, como acontece no caso de concentrações maciças de pessoas (eventos desportivos, culturais ou manifestações); ou pela inexistência temporária ou permanente de infraestrutura, como acontece em locais remotos ou em situações de catástrofe.

O WFD [1] permite então o suporte à formação de grupos *ad-hoc*, e a descoberta, autenticação e encaminhamento de mensagens entre nós do mesmo grupo. Contudo, apresenta as seguintes limitações: a comunicação entre dispositivos está limitada ao âmbito do seu grupo; e um grupo pode conter apenas 8 dispositivos.

Existem correntemente propostas para comunicação entre grupos em WFD (ver secção 2), em que recorrem à utilização simultânea das interfaces de WFD

---

\* Trabalho parcialmente financiado pela FCT-MEC, no contexto do projeto Hyrax (CMUP-ERI/FIA/0048/2013) e NOVA LINCS (UID/CEC/04516/2013)

e de WF, mas utilizam: *broadcasts* ou *multicasts* numa das suas interfaces; ou comutam as ligações entre os nós, desfazendo e refazendo as ligações; ou requerem a utilização de um número de nós que pode não estar disponível.

Este artigo foca-se então nas redes de comunicação autónomas com dispositivos Android, *non-rooted* e baseadas em WFD. Não fazer *root* permite que essas topologias possam ser utilizadas diretamente nos dispositivos sem violar a sua garantia, nem obrigar a alterações complicadas para o utilizador comum. Propõe-se então duas novas topologias de comunicação inter-grupo com a utilização exclusiva de ligações TCP e que: requerem um menor número de nós; e comunicam a uma velocidade superior às propostas anteriores.

De seguida, na secção 2, apresentam-se as dificuldades da comunicação entre grupos em WFD e as soluções existentes. Na secção 3 são apresentadas as novas topologias baseadas em TCP. Na secção 4 é analisado o n<sup>o</sup> de nós necessário e a potencial velocidade de comunicação para as várias topologias. Na secção 5 apresentam-se os dados experimentais, terminando-se com uma comparação entre as várias topologias. Na secção 6 conclui-se este artigo.

## 2 Comunicação entre grupos em Wi-Fi Direct

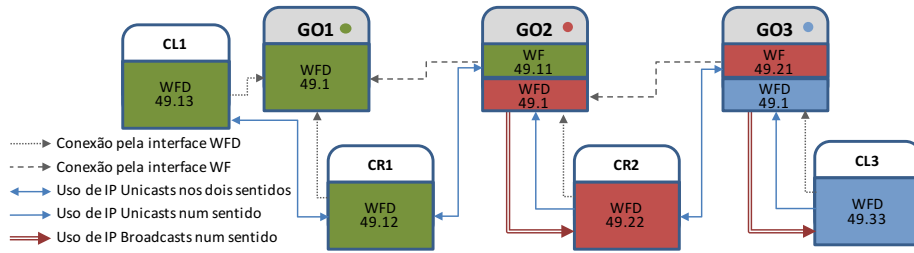
O WFD é um protocolo que define a comunicação entre um grupo de dispositivos. Cada grupo tem um nó coordenador designado de *Group Owner* (GO) que atua como um ponto de acesso WiFi (*Wi-Fi Direct AP*), providenciando um serviço de DHCP e encaminhamento de mensagens, ao nível MAC, entre os nós clientes do grupo. O WFD proporciona ainda a descoberta de nós, a negociação para seleção do GO e autenticação na formação do grupo ou inclusão de novos membros [3].

Um grupo pode-se formar quando um nó faz um convite WFD para o outro (ambos com a interface WFD ativada), ou quando um nó cria um grupo de forma autónoma. Um grupo tem um GO e os nós podem ligar-se a ele: utilizando a interface WFD (CL - *Client*); ou utilizando a interface WF, desde que conheçam o SSID da rede e a *password* (LC - *Legacy Client*).

O estudo da tecnologia WFD tem-se focado na análise do seu desempenho [3,5], mas apenas incidindo num único grupo e com relativamente poucos dispositivos; até 6 em [5]. Em [2] interliga-se a rede móvel *LTE* com a rede WFD, contudo as operações em WFD são dentro um único grupo.

Em cenários com vários grupos, tal como identificado em [4], existe a limitação de não se poder participar, por WFD, simultaneamente em 2 grupos WFD. De facto, 1 nó pode fazer a ponte entre 2 grupos, mas terá que utilizar as suas duas interfaces: WFD e WF.

Verifica-se que cada GO fica com o endereço 192.168.49.1/24 e aos seus clientes, WFD ou WF, é atribuído um endereço 192.168.49.x/24, onde  $x \in [2..244]$ . Para abreviar, doravante, os endereços serão mencionados apenas como 49.x. Como o uso do endereço 49.1 não é vinculativo pela norma, seria bom a API Android possibilitar a indicação do endereço, aquando da criação do GO. Os clientes receberiam essa informação na fase final de conexão ao grupo. Não se podendo fazer isso, surgem dois problemas: todos os GOs têm o mesmo endereço;



**Figura 1.** Um cenário GOCR com três grupos

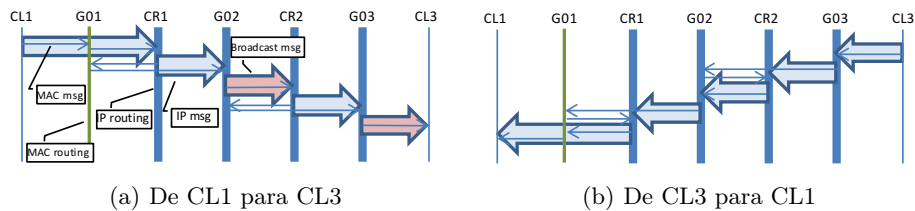
e todas as redes têm o mesmo endereço de rede. Assim: os GOs não comunicam diretamente entre si, pois os pacotes ou ficam retidas na interface *loopback* local ou são rejeitadas no destino, dado que o endereço de origem é o mesmo; e nos dispositivos com as duas interfaces ligadas, estes só comunicam diretamente pela interface que for prioritária. Salienta-se que há dispositivos com comportamentos distintos no uso das interfaces WFD e WF, tal como: não suporte a WFD; só permitir uma interface ativa de cada vez (ex<sup>o</sup>: Motorola G2 2<sup>a</sup> geração); ter como interface prioritária WF (ex<sup>o</sup>: Nexus 7) ou WFD (ex<sup>o</sup>: Nexus 6, Nexus 9).

Seguem-se as topologias de comunicação inter-grupo existentes.

## 2.1 Topologia GOCR

De forma a superar as limitações enunciadas, em [6] e [4] propõe-se uma configuração de nós para estabelecer comunicação bidirecional entre grupos, que utiliza um cliente auxiliar (*Client Relay* - CR) a cada GO, em que os GOs utilizam *broadcasts* para comunicar pela interface não prioritária (interface prioritária: WF). Designa-se esta topologia de GOCR, pois por grupo utiliza 1 GO e 1 CR.

Na Figura 1 apresenta-se um cenário nesta topologia com 3 grupos WFD, com os GOs (GO1, GO2 e GO3), e com dois 2 CR (CR1 e CR2) e 2 CLs (CL1 e CL3). Nele GO1 tem como clientes: CL1, CR1 e GO2. Como GO1 só utiliza a sua interface WFD, não há quaisquer limitações às comunicações CL1-GO1 e CL1-CR1, permitindo estabelecer canais IP em qualquer sentido entre os três. CL1 e CR1 podem comunicar diretamente ao nível IP com GO2, no entanto ao nível físico (MAC) essas mensagens são intermediadas por GO1. O GO2 está ligado pela interface WF (endereço 49.11) ao GO1, pelo que pode receber comunicações como cliente de GO1, mas não diretamente de GO1, pois como ambos têm o mesmo endereço local (49.1) as mensagens são impedidas de ser enviadas ou são ignoradas na chegada por serem consideradas como mensagens locais. Contudo entre CR1 (ou CL1) e GO2 já se consegue comunicar em IP, nos dois sentidos, dado que neste estudo a interface prioritária é WF. Contudo essas comunicações têm GO1 como intermediário ao nível MAC. De GO2 para GO3: GO2 envia as mensagens como *broadcasts* para CR2, pois foi a forma encontrada para conseguir comunicar pela interface não prioritária, estes *broadcasts* são ignorados pelos outros nós; e CR2 faz o encaminhamento para GO3. De GO3 para GO2: GO3 envia as mensagens para CR2, que saem pela sua interface WF;



**Figura 2.** GOCR: fluxo de mensagens

CR2 depois envia as mensagens para GO2 em *unicasts* UDP. Entre GO3 e CL3, como GO3 utiliza as duas interfaces, a comunicação é realizada via *broadcasts*, tal como de GO2 a CR2, e a comunicação de CL3 para GO3 é realizada via *unicasts* UDP, tal como de CR2 para GO2.

Na Figura 2(a) encontra-se um diagrama com as mensagens existentes na comunicação entre CL1 e CL3. De CL1 para CL3: CL1 envia para CR1 um datagrama (UDP) que contém a indicação interna de CL3 como destino, este é enviado ao nível MAC para GO1 e encaminhado por este para CR1; CR1 reenvia-o para GO2, como outro datagrama, passando ao nível MAC por GO1; GO2, por sua vez, reenvia-o como um *broadcast* IP que chega a CR2; CR2 reenvia-o para GO3 como um datagrama, passando ao nível MAC por GO2; e finalmente GO3 envia-o para CL3, novamente como um *broadcast* IP. De CL3 para CL1, tal como se pode observar na Figura 2(b): CL3 envia um datagrama para GO3; GO3 reenvia-o como outro datagrama para o CR2, que ao nível MAC passa por GO2; CR2 reenvia-o como outro datagrama para GO2; GO2 reenvia-o para CR1, passando ao nível MAC por GO1; e CR1 encaminha-o para CL1, passando ao nível MAC por GO1.

A topologia GOCR caracteriza-se então por criar um caminho para as mensagens que passa, ao nível IP, pelos GOs, utilizando um nó auxiliar para permitir a comunicação entre os grupos e por utilizar *broadcasts*. Esta topologia tem o problema da utilização dos *broadcasts* que limitam a velocidade de comunicação ao ritmo de 6Mbps, em vez de toda a largura de banda disponível em WiFi (54Mbps ou mais) que se pode conseguir com os *unicasts* do 802.11g, tal como se pode observar nos resultados apresentados em [4].

## 2.2 Topologia GO2CR

Em trabalho anterior [8] propusemos uma topologia designada de GO2CR (1 GO e 2 CRs), que, entre dois GOs, utiliza dois nós cliente, cada um ligado aos dois GOs, mas com as suas interfaces ligadas de forma simétrica. Desta forma a comunicação pode utilizar apenas datagramas UDP (não necessitando de *broadcasts*). Este trabalho, à semelhança de [4], considera WF como a interface prioritária.

Na Figura 3 apresenta-se um cenário com esta topologia contendo três grupos, e onde apenas estão assinaladas as ligações rádio WFD e WF e as comunicações utilizadas pela topologia. No entanto, neste cenário existem as seguintes ligações com capacidade de conexão IP bidirecional: a) directas: CL1↔GO1,



Considerando um cenário, nesta topologia, designado de  $GO1_{WF}CR_{WFD}-GO2$ , onde o CR está ligado pela interface WF a GO1 e ligado por WFD a GO2, a transferência de informação de GO1 para GO2 tem os seguintes passos: GO1 envia um *multicast* para CR para avisar; GO1 transfere a informação para CR; CR desliga a interface WF; CR transfere por TCP para GO2 toda a informação; CR liga-se por WF a GO1. Na transferência de GO2 para GO1 tem-se: GO2 notifica CR da transferência por *multicast*; CR desliga as duas interfaces; CR liga as duas interfaces mas no sentido contrário ficando  $GO1_{WFD}CR_{WF}-GO2$  e procede de forma análoga como no caso anterior.

Nesta solução é identificado como problemático o tempo dispendido no desfazer e refazer das ligações, que se situa na média dos 6 segundos.

### 3 Topologias de comunicação baseadas em TCP

Nas topologias já apresentadas identifica-se que as suas limitações são: o uso de *broadcasts* ou *multicasts*; a utilização de um n.º de nós que pode não estar disponível (2 CRs); e não utilização de ligações TCP. Nesta secção propõe-se então duas topologias que utilizam exclusivamente TCP e um n.º reduzido de nós.

#### 3.1 Topologia GOCR\_TCP

Esta topologia explora a utilização de ligações TCP, tirando partido de que estas são iniciadas num sentido, mas uma vez estabelecidas proporcionam um canal de comunicação bidirecional. Assim sendo em qualquer uma das topologias anteriores onde existe uma possibilidade do envio de unicasts IP (UDP), pode-se estabelecer um canal TCP iniciado no sentido possível.

Em GOCR, Figura 1, pode-se estabelecer um canal TCP entre: CR1 e GO2; CR2 e GO3, podendo ser iniciado por qualquer um deles, e entre: CR1 e GO1; CR2 e GO2, CL3 e GO3, tendo de ser iniciado por CR1, CR2 e CL3. Desta forma não é necessário o uso de *broadcasts*. Semelhantemente, em GO2CR pode-se estabelecer um canal TCP de: CL1 a, por exemplo, CR12, tendo de ser iniciado por CL1; CR12 a CR32, sendo iniciado por CR12; e CR32 a CL3, podendo ser qualquer um deles o iniciador.

Como as ligações TCP são bidirecionais estas suportam comunicações em ambos os sentidos e portanto permite definir uma nova topologia composta por apenas 1 GO e 1 CR por cada grupo. A esta topologia designamos de GOCR\_TCP e na Figura 5 apresenta-se um cenário com 3 grupos. Neste cenário e doravante será considerada a interface WFD como a interface prioritária. Esta topologia (GOCR\_TCP) utiliza um cliente relay (CR) entre cada dois GOs, mas o CR pode ser utilizado para estender a distância entre GOs, mantendo uma média de 1 nó por alcance de cobertura WiFi. A topologia é então caracterizada por: permitir ligações TCP; utilizar um CR entre cada dois GOs; o encaminhamento ao nível IP pode ser apenas realizado pelos CRs; os GOs podem só fazer o encaminhamento ao nível MAC.

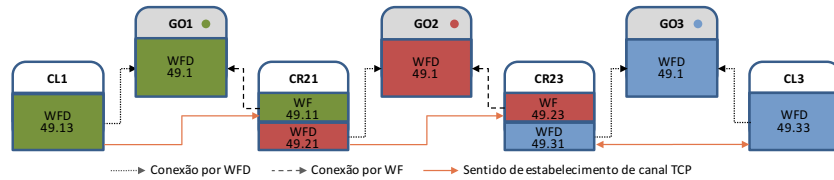


Figura 5. Um cenário GOCR\_TCP com três grupos

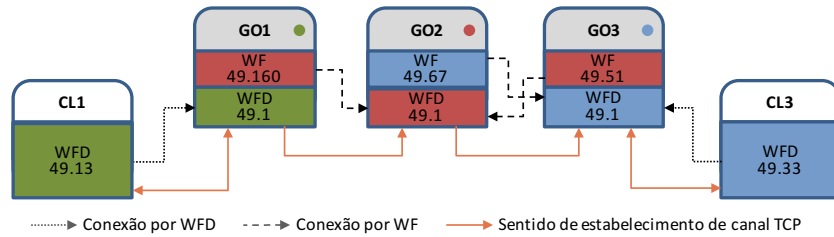


Figura 6. Um cenário GOGO\_TCP com três grupos

### 3.2 Topologia GOGO\_TCP

O Android na suas APIs 21 e 22 (Android 5.0 e 5.1) introduziu o conceito de *Network*, o qual pode ser utilizado para estabelecer sockets TCP (e UDP) também através da interface não prioritária. A esta ação designamos de criar *sockets* com *bind* a uma interface. Assim na topologia anterior podemos agora estabelecer também as seguintes ligações TCP: CR21→CL1; e CR23→CR21. De facto agora todas as ligações TCP existentes podem ser iniciadas por qualquer um dos nós. Um outro facto de salientar é que um GO1 que utilize a sua interface WF para se ligar a um outro GO, GO1 fica com o endereço de cliente na interface WF disponível para receber *unicasts* provindos da sua rede WFD, pois ele com servidor de DHCP resolve o endereço para si próprio. Com base nestas duas funcionalidades, propõe-se uma nova topologia designada de GOGO\_TCP que utiliza um GO ligado diretamente a outro GO por uma ligação TCP. Na Figura 6 pode-se observar um cenário com 3 grupos nesta topologia. Nele GO1 pode abrir uma conexão TCP para GO2, utilizando um *socket* TCP com *bind* a WF e para 49.67. GO2 também consegue ligar-se a GO3, mas para isso GO3 tem de se ligar a GO2 por WF, para ter um endereço nessa interface.

Esta topologia apresenta as melhores características em termos de: n<sup>o</sup> de nós utilizados (1 GO por grupo); cobertura, pois todos os nós são *Group Owners* (maior cobertura que GOCR\_TCP); e permitir a utilização de ligações TCP.

## 4 Análise das topologias

### 4.1 Análise da cobertura geográfica

Para esta análise utiliza-se a medida de “alcance Wi-Fi” (WFR: WF Range) que conceptualmente é a distância que o WF (e WFD) pode alcançar. Assim temos

por topologia, por cada WFR quantos nós necessita e média final de nós por WFR: a) GO2CR: 1 GO e 1 CR (por WFR), média 2/WFR; b) GO2CR: 1 GO e 2 CRs (por 2 WFRs), média 1.5/WFR; c) GO2CR\_TCP: 1 GO e 1 CR (por 2 WFRs), média 1/WFR; e d) GOGO\_TCP: 1 GO (por WFR), média 1/WFR.

## 4.2 Análise da velocidade de comunicação

Para esta análise utiliza-se a medida de “velocidade de comunicação WF em *unicast*” (WFS: WF *Speed*) que conceptualmente é a velocidade máxima ( $V_{max}$ ) potencial que o WF (e WFD) pode alcançar nas transmissões. Ao nível dos canais WiFi considera-se que cada GOs operam num canal distinto e sem interferências entre si. Como na análise a realizar se quer ter em conta o nº de mensagens *unicast* que partilham um mesmo canal, é necessário relacionar os *broadcasts* com os *unicasts*. Considera-se então o termo *Broadcast Factor* (BCF) que corresponde ao nº de *unicasts* equivalente a 1 *broadcast*, em termos de ocupação do canal. Tem-se então  $BCF = WFS / BCS$ , em que BCS é a *broadcast speed*. Considerando-se, para efeitos de referência,  $WFS = 54\text{Mbps}$  e  $BCS = 6\text{Mbps}$ , então  $BCF = 9$ . Os valores concretos apresentados são dependentes destes valores de referência e são assinalados com \*. Assume-se que os nós podem comunicar em simultâneo pelas duas interfaces.

Na configuração GO2CR, da direita para a esquerda, por um GO passa: 1 datagrama IP do GO da sua direita para o seu CR, que requer 2 mensagens MAC no seu canal; e 1 datagrama IP do CR para si mesmo, requerendo portanto 3 mensagens MAC *unicast* no seu canal, resultando  $V_{max} = WFS / 3 = 18\text{Mbps}^*$ . No sentido da esquerda para a direita, passam por um GO: 1 *broadcast* IP de si mesmo para o CR; e 1 datagrama IP do CR para o GO à sua direita, que requer 2 mensagens MAC. No canal do GO passam então um equivalente a  $2 + BCF$  mensagens *unicast*, resultando  $V_{max} = WFS / (2 + BCF) = 4.9\text{Mbps}^*$ . Numa utilização bidirecional simultânea, cada canal terá que suportar 3 mensagens num sentido e  $(2 + BCF)$  no outro, resultando  $V_{max} = WFS / (5 + BCF) = 3.86\text{Mbps}^*$ .

Na configuração GO2CR a comunicação é simétrica e por um GO num sentido passa apenas 1 datagrama IP do CR de um lado para o CR do outro lado. Esse datagrama é reencaminhado ao nível MAC pelo GO, resultando em 2 MAC mensagens *unicast*. Assim  $V_{max} = WFS / 2 = 27\text{Mbps}^*$ . Numa utilização bidirecional simultânea, esta topologia, transporta um total de 4 mensagens, resultando  $V_{max} = WFS / 4 = 13.5\text{Mbps}^*$ .

Na configuração GO2CR\_TCP, a comunicação é simétrica e num sentido só passa 1 mensagem TCP entre os CRs, a qual é reencaminhada pelo GO que intermedeia os CRs, resultando em 2 mensagens MAC *unicast* no canal do GO. Então tem-se  $V_{max} = WFS / 2 = 27\text{Mbps}^*$ . Na comunicação bidireccional simultânea ter-se-á  $V_{max} = WFS / 4 = 13.5\text{Mbps}^*$ .

Na configuração GOGO\_TCP, cada GO estabelece uma ligação TCP, pela interface WF, com o GO seguinte, a qual suporta mensagens nos dois sentidos. Num sentido é necessário apenas o envio de 1 mensagem TCP para comunicar entre GOs. Então  $V_{max} = WFS = 54\text{Mbps}^*$ . Na comunicação bidireccional simultânea ter-se-á  $V_{max} = WFS / 2 = 27\text{Mbps}^*$ .



**Tabela 1.** N° de nós por WFR e velocidade de comunicação

	N° de nós / WFR	Vel. comunicação
GO2CR	2	WFS / 14 (3.86 Mbps*)
GO2CR	1.5	WFS / 4 (13.5 Mbps*)
GO2CR_TCP	1	WFS / 4 (13.5 Mbps*)
GOGO_TCP	1	WFS / 2 (27 Mbps*)

Na Tabela 1 apresenta-se, para as várias topologias, o sumário do n° de nós necessário por WFR e a velocidade de transmissão bidirecional.

## 5 Dados experimentais

Para os testes foram utilizados 3 dispositivos Nexus 6 (N61, N62 e N63), com: quad core; 2.7GHz; 3GB (ram); 32GB (flash); e interface prioritária WFD. Nos cenários apresentados tem-se em consideração que: o *bind* é sempre efetuado para WF; o comportamento de *bind* em UDP é idêntico em TCP; todos os *multicasts* utilizam o endereço 224.1.0.0:10000; os dispositivos foram colocados todos juntos; e os ensaios decorreram com interferências externas com menos de -90dBm. Para a análise das possibilidades de comunicação considera-se os seguintes cenários com 3 dispositivos: 1 GO e 2 CLs; 2 GOs e 1 CL; e 3 GOs.

### Cenário A: $CL1_{WFD} - GO - WFDCL2 / WFCL2$

Considerando-se CL2 ligado a GO por WFD, consegue-se estabelecer ligações TCP, UDP, UDP *broadcast* e UDP *multicast* entre qualquer um dos nós, em que as ligações por *multicast* requerem a utilização da interface WFD. No caso de se considerar ligar CL2 por WF ao GO, todos os comportamentos permanecem idênticos, excepto que CL2, na comunicação por *multicast*, necessita de indicar a interface WF em vez de WFD.

### Cenário B: $GO1 - WFDCL_{WF} - GO2$

Neste cenário o cliente faz de ponte entre os dois grupos e as possibilidades de comunicação constam na Tabela 2. Como o cliente utiliza as duas interfaces ele só consegue enviar *unicasts* com *bind* por WF. Entre GOs nada consegue passar *diretamente* de um lado para outro, pois ambos os GOs estão em redes diferentes e o cliente não faz encaminhamento automático dos pacotes.

### Cenário C: $GO1 - WFGO2 - WFDCL$

Este é um cenário em sequência, em que CL liga-se a GO2 e este a GO1, logo GO2 utiliza as duas interfaces. Na Tabela 3 constam as ligações possíveis, onde se pode verificar que entre GO1 e GO2 não se consegue comunicar de forma direta (exceptuando por *multicast*), pois ambos têm o mesmo endereço (49.1). Entre GO1 e CL nada consegue passar *diretamente* pois estão em redes diferentes.

Numa segunda versão deste cenário coloca-se GO1 também como cliente WF de GO2 (valores na tabela com \*). Deste modo GO1 utiliza as duas interfaces possibilitando o uso de *sockets* com *bind* a WF, entre GO1 e GO2. Aqui os envios

**Tabela 2.** Cenário B:  $GO1 - WFDCL_{WF} - GO2$ 

$GO1 - WFDCL_{WF} - GO2$				
$N61_{49.1} - 49.47N62_{49.167} - 49.1N63$				
	$N61 \rightarrow N62$	$N61 \leftarrow N62$	$N62 \rightarrow N63$	$N62 \leftarrow N63$
TCP	✓	✓	✗	✓
UDP	✓	✓	✗	✓
TCP/bind	-	-	✓	-
UDP BC	✓	✓	✗	✓
UDP MC	WFD✓WFD	WFD✓WFD	WFD✓WFD	WF✓WFD

**Tabela 3.** Cenário C:  $GO1 - WFGO2 - WFDCL$ 

$GO1 - WFGO2 - WFDCL$						
$N61_{49.1} - 49.167N62_{49.1} - 49.160N63$						
	$N61 \rightarrow N62$	$N61 \leftarrow N62$	$N62 \rightarrow N63$	$N62 \leftarrow N63$	$N61 \rightarrow N63$	$N61 \leftarrow N63$
TCP	✗ ✗*	✗ ✗*	✓	✓	✗ ✗*	✗ ✓*
UDP	✗ ✓*	✗ ✓*	✓	✓	✗ ✗*	✗ ✓*
TCP/bind	✗ ✓*	✗ ✓*	-	-	✗ ✓*	✗ ✓*
UDP BC	✓ ✓*	✗ ✓*	✓	✓	✗ ✗*	✗ ✓*
UDP MC	WFD✓WF	WFD✓WF	WFD✓WFD	WFD✓WFD	✗ ✓*	✗ ✓*

são para os endereços de WF, que são 49.51 para GO1 e 49.167 para GO2. Assim já é possível a utilização de *sockets* TCP com *bind* e também o envio de dados entre GO1 e CL, pois GO1 também é cliente de GO2. Os *multicasts* entre GO1 e CL, requerem a utilização de WF por GO1 e WFD por CL.

**Cenário D:**  $GO1_{WF} - GO2 - WFDCL / WFGO3$

Este é um cenário em que GO1 e CL são clientes de GO2, mas em que GO1 utiliza as suas duas interfaces. Na Tabela 4 constam as ligações possíveis, no qual se pode constatar que: GO1 consegue comunicar com CL por TCP com *bind* por WF; GO2 não consegue estabelecer uma ligação com GO1 pois como não utiliza as duas interfaces não pode fazer *bind*. Numa primeira versão o terceiro elemento é um cliente WFD, sendo portanto uma situação normal e já analisada. Numa segunda versão o terceiro elemento é um GO (GO3) ligado a GO2 pela interface WF e como utiliza as duas interfaces consegue estabelecer ligações TCP com *bind* por WF para GO2 e também para GO1, pois GO3 e GO1 são clientes de GO2. O envios de GO1 para GO3 são com *bind* e para o endereço 49.241, e de GO3 para GO1 com *bind* e para 49.51.

**Cenário E:**  $GO1_{WF} - GO2_{WF} - WFGO3$

Este cenário apresenta uma sequência de GOs ( $GO1 \rightarrow GO2 \rightarrow GO3$ ) ligados por WF e onde GO3 liga-se a GO2 por WF de forma a permitir ligações TCP com *bind* entre eles. Assim pode-se estabelecer conexões TCP, com *bind*, entre todos eles. Por limitações de espaço não se apresenta a tabela, mas remete-se a observação do comportamento para os pares: GO1-GO2 do cenário C, que equivalem ao par GO3-GO2 neste cenário; e GO1-GO2 no cenário D, que equivale

**Tabela 4.** Cenário D:  $GO1_{WF} - GO2 - WFDCL / WFGO3$

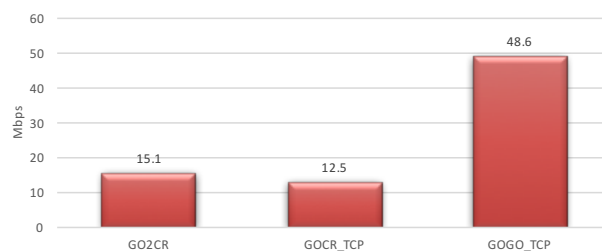
$GO1_{WF} - GO2 - WFDCL / WFGO3$					
$N61_{49.1/49.51} - 49.1N62_{49.1} - 49.160N63 / 49.241N63$					
	$N61 \rightarrow CL$	$N61 \leftarrow CL$	$N62 \leftarrow GO3$	$N61 \rightarrow GO3$	$N61 \leftarrow GO3$
TCP	✗	✓	✗	✗	✗
UDP	✗	✓	✗	✗	✗
TCP/bind	✓	-	✓	✓	✓
UDP BC	✗	✓	✗	✗	✗
UDP MC	WF ✓ WFD	WF ✓ WFD	WFD ✓ WF	WF ✓ WF	WF ✓ WF

a GO2-GO1 neste cenário. Entre GO1 e GO3, além de *unicasts*, consegue-se ligar por TCP com *bind* nos dois sentidos e para os endereços das interfaces WF do nó destino.

### 5.1 Teste às várias topologias

Pretende-se aqui avaliar o comportamento real das topologias analisadas: GO2CR, GO2CR\_TCP e GOGO\_TCP. Para tal iremos utilizar um cenário mínimo com 2 grupos. A topologia GO2CR não permite uma implementação direta caso WFD seja a interface prioritária. Para se simular, essa topologia, de uma forma aproximada, ter-se-ia que utilizar *sockets* UDP com *bind* por WF, entre GOs e entre GO e o CR do grupo seguinte. De forma a evitar apresentar valores que não correspondem à realidade opta-se por não se apresentar valores para esta topologia. A topologia GO2CRSwitch como implica a desconexão e reconexão dos nós e a consequente salvaguarda em memória (provavelmente flash) dos dados em transito, não foi considerada para implementação pelas demoras e recursos que requer. De recordar que o tempo médio de conexão é de cerca de 6 segundos. Seguidamente apresenta-se para cada topologia um cenário mínimo com 2 GOs: a) para GO2CR, cenário GO-2CR-GO; b) para GO2CR\_TCP, cenário GO-CR-GO; e c) para GOGO\_TCP, cenário GO-GO . O cenário GO2CR foi implementado com 4 ligações UDP. O cenário GO2CR\_TCP requereu 2 ligações TCP, que foram utilizadas de forma bidirecional. GOGO\_TCP requereu apenas 1 ligação TCP. É apresentado na Figura 7 o resultado da média de 10 ensaios, da velocidade média de envio de 100MB entre GOs, com eco dos mesmos dados, num envio com mensagens de 1KB.

Como numa ligação GO-CL em UDP consegue-se uma velocidade de transmissão de cerca de 100 Mbps, a topologia GOGO\_TCP, que apenas requer duas transmissões, devido ao eco dos dados, encontra-se no pleno da utilização do canal, com uma taxa média de 48.6 Mbps. Já as outras duas topologias, requerem que o encaminhamento IP seja feito por nós que utilizam as duas interfaces e apresentam valores de quase 1/4 do seu potencial, pois em termos ideais ambas requerem a partilha dos canais por 2 transmissões em simultâneo. A deterioração da velocidade indicia que os dispositivos não devem utilizar as interfaces de forma paralela e independente e que os conflitos no acesso ao canal penalizam



**Figura 7.** Velocidade média para GO2CR, GOCR\_TCP e GOGO\_TCP com 2 GOs

também a comunicação. O cenário GO2CR apresenta 15.1 Mbps de média de envio, mas utiliza UDP e apenas cerca 17.4% dos dados retornam ao GO inicial.

## 6 Conclusões

Neste artigo, propõe-se duas novas topologias (GOCR\_TCP e GOGO\_TCP) de comunicação entre dispositivos Android, *non-rooted*, com as seguintes vantagens em relação às topologias existentes: requerem um menor n<sup>o</sup> de nós por área geográfica; só utilizam ligações TCP; e apresentam melhores valores potenciais de comunicação. A topologia GOGO\_TCP apresenta um valor real de comunicação significativamente superior a todas as configurações existentes.

## Referências

1. Wi-fi direct alliance @ONLINE, June 2014.
2. A. Asadi and V. Mancuso. Wifi direct and lte d2d in action. In *Wireless Days (WD), 2013 IFIP*, pages 1–8, Nov 2013.
3. D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano. Device-to-device communications with wi-fi direct: overview and experimentation. *Wireless Communications, IEEE*, 20(3):96–104, June 2013.
4. C. E. Casetti, C.-F. Chiasserini, L. C. Pelle, C. D. Valle, Y. Duan, and P. Giaccone. Content-centric routing in wi-fi direct multi-group networks. In *IEEE 16th International Symposium on "A World of Wireless, Mobile and Multimedia Networks", WoWMoM 2015, Boston, USA, June 14-17, 2015*, 2015.
5. M. Conti, F. Delmastro, G. Minutiello, and R. Paris. Experimenting opportunistic networks with wifi direct. In *Wireless Days (WD), 2013 IFIP*, pages 1–6, Nov 2013.
6. Y. Duan et al. Wi-fi direct multi-group data dissemination for public safety. In *WTC 2014; World Telecommunications Congress 2014*, pages 1–6, 2014.
7. C. Funai, C. Tapparello, and W. B. Heinzelman. Supporting multi-hop device-to-device networks through wifi direct multi-group networking. *CoRR*, abs/1601.00028, 2016.
8. A. Teófilo, D. Remédios, H. Paulino, and J. a. Lourenço. Group-to-group bidirectional wi-fi direct communication with two relay nodes. In *12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MOBIQUITOUS 2015*, pages 275–276, 2015.