



HAL
open science

A Collaborative Video Download Application Based on Wi-Fi Direct

Haotian Sha, Argyrios G. Tasiopoulos, Ioannis Psaras, George Pavlou

► **To cite this version:**

Haotian Sha, Argyrios G. Tasiopoulos, Ioannis Psaras, George Pavlou. A Collaborative Video Download Application Based on Wi-Fi Direct. 14th International Conference on Wired/Wireless Internet Communication (WWIC), May 2016, Thessaloniki, Greece. pp.147-158, 10.1007/978-3-319-33936-8_12. hal-01434849

HAL Id: hal-01434849

<https://inria.hal.science/hal-01434849>

Submitted on 13 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Collaborative Video Download Application Based on Wi-Fi Direct

Haotian Sha, Argyrios G. Tasiopoulos, Ioannis Psaras, and George Pavlou

Dept. of Electronic and Electrical Engineering
University College London

{haotian.sha.14, argyrios.tasiopoulos, i.psaras, g.pavlou}@ucl.ac.uk

Abstract. We developed a collaborative video download application, which allows several users to form a local group via peer-to-peer (P2P) links and collaboratively help one user in the group to download a high quality video. Cellular interface and P2P interface are simultaneously utilised in the application to guarantee quality of experience (QoE) when the cellular connection is disrupted. The application is based on Wi-Fi Direct technology and it exploits the service discovery function of the Wi-Fi Direct framework.

Apart from the implementation itself, we also model collaborative video streaming over the cellular network under periodic cellular disconnections (*e.g.*, the commuters experience poor Internet services when trains travel through tunnels). Our purpose is to exploit the gains of the collaborative download between users in the context of periodic disconnections and limited cellular downlink rate. We simulate several scenarios to find the requirements of the application that guarantee seamless and undisturbed playback.

Keywords: Collaborative Download, Wi-Fi Direct, Service Discovery, Seamless Playback

1 Introduction

People today have growing demand for mobile Internet services and especially video streaming. The applications related to video streaming are increasingly popular over mobile devices, as people want to watch HD video through their mobile devices when they are in cafe or commuting by train. There are, however, several challenges which lower the QoE of video streaming applications.

In underground systems, video streaming is extremely challenging, since the cellular connection is physically disrupted when the trains travel between stations. Often, even when trains are in stations and signal is available, streaming cannot necessarily be supported with acceptable QoE.

In order to deal with the above issues, collaborative video download systems [8][6][7][9][10] based on mobile P2P networks have attracted attention recently. However, these studies focus on downloading through Wi-Fi interface rather than cellular interface. There are also studies, *e.g.*, in [3][2], utilising P2P and cellular

links to disseminate content, taking the social ties and geographical proximity into account. They focus, however, on offloading current cellular network rather than video streaming.

In contrast to the above studies, in this paper, we utilise multiple interfaces of mobile devices to improve the QoS of streaming applications in constrained cellular connection environments. We build models to find the requirements of seamless playback in various situations, depending on several parameters, such as the video data rate, the cellular downlink rate and the device-to-device (D2D) data transmission rate.

We implement a collaborative video download application which allows several users who are in the vicinity to form a local group via Wi-Fi Direct and collaboratively help one user in the group download an HD video through the cellular network. The problems above could be effectively solved by “*helper nodes*”, who download content on behalf of the user who wants to watch some video. They then transmit the downloaded content using Wi-Fi Direct. The collaborative download system we design is shown in Figure 1(a). For the purposes of this study, we do not consider incentives of “*helper nodes*”, but assume that micropayments can complement and incentivise participation.

We choose Wi-Fi Direct for our application since it can transfer data faster and over longer distances compared to Bluetooth. We exploit the service discovery function in Wi-Fi Direct framework, which allows the users (*i.e.*, group members) to identify the user who requires help for video streaming (*i.e.*, group owner) and connect to the group owner to form a many-to-one wireless local network. We assume that an HD video is divided into chunks at the server and each chunk is assigned a unique *identifier*. Once the group of the owner and the helpers is formed, each member is assigned with a number of chunk identifiers to download on behalf of the group owner. We then use the AsyncTask class in java to allow the group owner to simultaneously receive video chunks from multiple group members, which is the key component for achieving seamless playback.

We model the process of video chunk downloading and delivering between users in several different cases, taking into account all the required parameters.

The rest of the paper is organised as follows. In section 2, we describe the implementation of the collaborative video download application on the Android platform. In section 3, we build models for different scenarios to analyse the requirements of seamless playback and the gains of collaborative download. At last, we conclude our findings in section 4, where we also discuss future developments of collaborative download systems.

2 Implementation of Collaborative Video Download Application

The implementation of our application is comprised of four main components: *i*) establishing local device-to-device (D2D) network, *ii*) downloading and/or streaming video, *iii*) video chunk delivery and *iv*) playing received video chunks.

Establishing Local D2D Network. In Wi-Fi Direct, the service discovery technique allows a device to discover the services published by devices in the vicinity without being connected to a network. The service here is a Bonjour service information object, which has three parameters ¹:

- instanceName (e.g. “MyPrinter”).
- serviceType (e.g. “ipp.tcp”).
- txtMap, which is a TXT record with key/value pair in a map containing the information of the service.

The `DnsSdServiceResponseListener` in Wi-Fi Direct framework is utilised to receive the actual service description and connection information. We ask the user who requires help for downloading an HD video (group owner) to publish a service with an instance name of “GroupOwner”. The *helper nodes or group members* start the process of discovering a service with the instance name of “GroupOwner” and receive the service description and connection information through `DnsSdServiceResponseListener`. Once the connection information of group owner is obtained, the helpers connect to the group owner and establish a many-to-one local D2D network.

Streaming/Downloading Video Chunks. An HD video is divided into chunks at the server and each chunk is assigned a unique *identifier*. This can be realised through the *Information-Centric Networking* (ICN) concept (e.g., [1][5][4]). In our application, the group owner streams directly from the source. In the meantime, the group members download the later parts of the video. After downloading, the group members transfer the downloaded content to the group owner.

We utilise the `VideoView` class in java to implement streaming. `AsyncTask` class in java is utilised to run the downloading process in the background and publish the results in the UI thread.

Video Chunk Delivery. In Wi-Fi Direct, there are “broadcast intents” which inform our application when certain events occur. We instantiate an intent filter and create a new `WifiDirectBroadcastReceiver` class which extends the `BroadcastReceiver` class to listen for the intents we are interested in. The application starts different `AsyncTasks` or threads (transferring chunks or receiving chunks) according to the various intents. The logical flowchart is shown in Figure 1(b).

We use different TCP ports to receive different video chunks. This is to allow the group owner device to receive video chunks from different group members simultaneously. We name the received chunks according to the port numbers which they come from to distinguish different chunks. We use TCP ports with smaller numbers to receive chunks in the former part of the video and TCP ports with bigger numbers to receive chunks in the latter part of the video. This way, we can sort the video chunks according to their names from low to high and play them in order.

¹ [Android/sdk/docs/reference/android/net/wifi/p2p/nsd/WifiP2pDnsSdServiceInfo.html](https://developer.android.com/sdk/docs/reference/android/net/wifi/p2p/nsd/WifiP2pDnsSdServiceInfo.html)

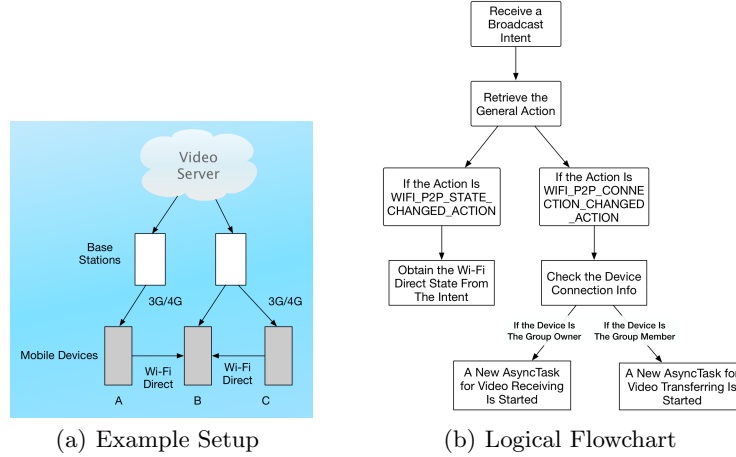


Fig. 1. Overall Setup and Logical Flowchart

Playing Received Video Chunks. When the group owner device has received the video chunks from group members, it needs to play the received chunks seamlessly and in order. `VideoView` class is a wrapper around the `MediaPlayer`, which is usually used to play video files on Android platform. In order to achieve seamless playback, we need to sort the received video chunks before the group owner finishes streaming the first few video chunks.

3 Modelling and Evaluation

As is clear from the implementation aspects presented above, the main parameters that affect seamless playback are the *video data rate*, x , the *cellular downlink rate*, y , and the *WiFi-Direct transferring rate*, z . In this Section, we model three separate cases as regards these parameters.

- Scenario 3.1: $z > y > x$, plus periodic disconnections. We first consider the case where the cellular downlink rate is faster than the video data rate. In this case, clearly, the video can be streamed without playback disruptions. For this reason, we add periodic disconnections to emulate public transportation systems, such as underground train networks.
- Scenario 3.2: $z > x > y$. As a second case, we consider the case where the video playback rate is faster than the cellular downlink. This means that playback disruptions will inevitably reduce the perceived QoE as perceived by the end user.
- Scenario 3.3: $z > x > y$, plus periodic disconnections. Finally, we add periodic disconnections to the above case, in order to emulate a more challenging and more realistic case of an underground train network.

We note that in all cases, the WiFi-Direct rate (*i.e.*, the D2D transfer rate) is faster than both the video data playback rate and the cellular rate.

In each of the above cases, we study the relations between the parameters in Table 1 under the requirement of achieving undisrupted playback. The modelling parameters and the related notation is listed in Table 1.

Parameters We Consider	
Video Data Rate	x (KBps)
Cellular Downlink Rate	y (KBps)
Wi-Fi Direct Transferring Rate	z (KBps)
Connection Period	C (s)
Disconnection Period	D (s)
Number of Group Members	N
Buffered content in terms of Playback time	T_b (s)
Playback duration of a Video Chunk	T_p (s)
Rounds for the Group Owner Receiving Video Chunks from the Group Members	n
Total Number of Chunks of a Video	V
Time for Playing Video in C	T_{pc} (s)
The Extra Buffered Video Content	A (KB)

Table 1. Parameters that may affect the video streaming experience of our application.

3.1 Scenario 3.1: $z > y > x$, Plus Periodic Disconnections.

In this case, the group owner can stream more content in connection period C than he can watch, since $y > x$. Therefore, the group owner can play the extra buffered content in the upcoming disconnection period. The group members can utilise the time that the group owner plays the extra buffered content to transfer video chunks to the group owner. The first requirement of seamless playback is that group members transfer the downloaded content to the group owner before the owner's buffered content is played-back. This requirement is shown in inequality (1).

$$\frac{(y-x)C}{x} \geq \frac{Cy}{z} \quad (1)$$

The second requirement of seamless playback is that the video chunks the group owner receives from group members can cover the disconnection period. This case is shown in inequality (2).

$$\frac{CyN}{x} \geq D \quad (2)$$

A special case here is when the cellular downlink rate y is so fast that the extra buffered content could cover the following disconnection period, hence, there is no need for helper users. This subcase is shown in inequality (3).

$$\frac{(y-x)C}{x} \geq D \quad (3)$$

We present 4 evaluation scenarios for this case in Table 2. The purpose is to study how many group members are required in order to realise seamless playback, as well as the relation between transferring rate and video quality.

Scenario 3.1.1	$x = 350KBps$, $z = 10MBps$, $C = 240s$, $D = 900s$, and y and N are variables
Scenario 3.1.2	$y = 600KBps$, $z = 10MBps$, $C = 240s$, $D = 900s$, and N and x are variables
Scenario 3.1.3	$y = 600KBps$, $C = 240s$, $D = 900s$, $N = 5$, and z and x are variables
Scenario 3.1.4	$C = 240s$, $D = 900s$, $x = 550KBps$, $y = 600KBps$, and z and N are variables

Table 2. Scenarios for case when $z > y > x$, plus periodic disconnections

Scenario 3.1.1. According to inequality (1), we find that in order to meet the first requirement of seamless playback the cellular downlink rate should be $y \geq 362.39KBps$. In turn, according to inequality (3), we find that if $y \geq 1662.5KBps$ there is no need to have helper nodes, as the cellular downlink is so fast that can take the user through the disconnection period and into the next connection period. The relation between y and N is shown in Figure 2(a).

We can see from Figure 2(a) that as the cellular rate y increases, less group members are required.

Scenario 3.1.2. The relation between the video data rate x and the minimum number of required helper nodes N is shown in Figure 2(b).

As expected, we can see from Figure 2(b) that the relation between x and N is linear. That is, the group owner can improve the quality of the video he receives by adding more group members.

Scenario 3.1.3. Figure 2(c) shows the relation between the WiFi Direct transferring rate and the video data rate under the requirements of seamless playback.

We can see from Figure 2(c) that as video quality increases, the user requires faster transferring rate between user devices. The gradient of the figure is increasing dramatically when $x \geq 500KBps$. This means that fast device-to-device transferring rate is necessary if the user wants to watch HD video.

Scenario 3.1.4. Figure 2(d) shows the relation between transferring rate and the required number of group members.

We can see from Figure 2(d) that increasing transferring rate can reduce the required number of helpers. In this case, the group owner requires only two helpers when the transferring rate is $12,000KBps$, which is within the range of the WiFi Direct specification.

Scenario 3.2: $z > x > y$. In this case, the group owner needs to buffer video content in advance of starting the playback, since the cellular rate is smaller than the video playback rate, $y < x$. We therefore, need to guarantee that the buffered video content, T_b , can make up the content that the user cannot stream in T_p secs, that is, in duration equal to the playback time of one chunk. This requirement is shown in inequality (4).

$$yT_b \geq (x - y)T_p \quad (4)$$

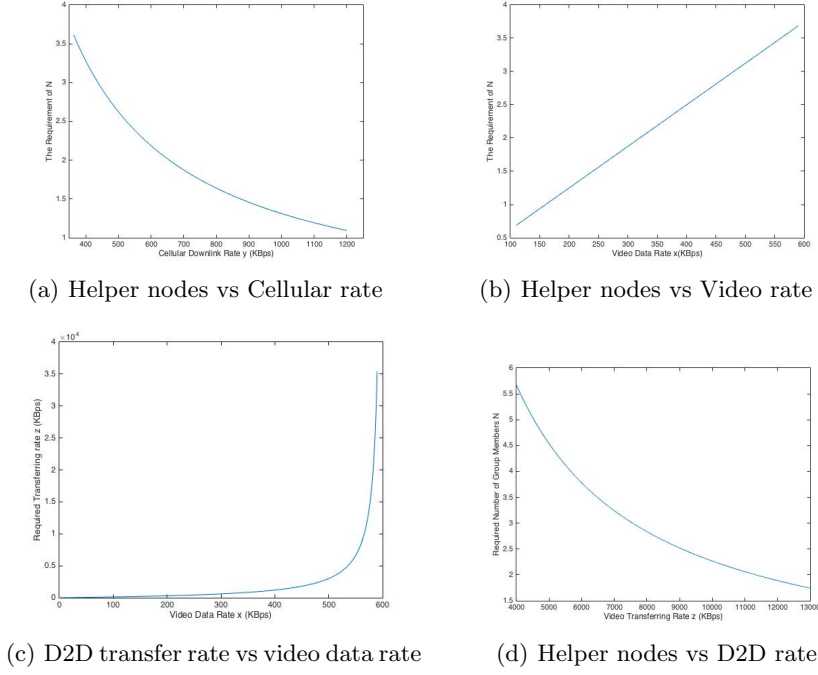


Fig. 2. Scenario 3.1: $z > y > x$, plus periodic disconnections.

We assume that at the point when the group owner starts streaming the first video chunk, each group member has already downloaded one video chunk and is in the process of transferring it to the group owner. After the first round of receiving content from the N helper nodes, the group owner would have N chunks ready for playback. Therefore, the second round of downloading content and transferring it to the group owner is longer than the first. This is because the first round needs to guarantee that video chunks have been downloaded and sent to the group owner before the first chunk finishes, whereas in the second round, the group owner has N chunks to playback before he runs out of buffered content. That said, we assume that each group member can download and transfer r_2 chunks during the second round, and the requirement of r_2 is shown in (5).

$$NT_p \geq \frac{r_2 x T_p}{y} + \frac{r_2 x T_p}{z} \quad (5)$$

The left part of inequality (5) is the playback time of N chunks, and the right part of inequality (5) is the time each group member needs in order to download and transfer r_2 chunks. Inequality (5) can be simplified as follows.

$$r_2 \leq \frac{Nyz}{zx + yx} \quad (6)$$

Similarly, the number of chunks that each group member can download and transfer during the third round, r_3 , is shown in inequality (7).

$$r_3 \leq \left(\frac{Nyz}{xz + xy}\right)^2 \quad (7)$$

Generalising inequalities (6) and (7), we conclude that the number of chunks that each group member can download during the n^{th} round, r_n , is:

$$r_n \leq \left(\frac{Nyz}{xz + xy}\right)^{n-1} \quad (8)$$

Based on Eq. (8), we find that in order to stream V chunks without disruptions, the group owner needs to complete n rounds of receiving from the helper nodes, according to:

$$\begin{aligned} V \leq & 1 + N\left(\frac{Nyz}{xz + xy}\right)^0 + N\left(\frac{Nyz}{xz + xy}\right)^1 \\ & + N\left(\frac{Nyz}{xz + xy}\right)^2 + \dots + N\left(\frac{Nyz}{xz + xy}\right)^{n-1} \end{aligned} \quad (9)$$

We present three evaluation scenarios for the case where the video playback rate is faster than the cellular downlink rate. We focus on the number of rounds needed in order for the group owner to receive the video chunks required to achieve uninterrupted playback. Note that in this scenario, we assume no physical disruption due to disconnection.

Scenario 3.2.1	$x = 450KBps, z = 10MBps, y = 200KBps, V = 50$, and n and N are variables
Scenario 3.2.2	$x = 450KBps, z = 10MBps, y = 200KBps, N = 4$, and V and n are variables
Scenario 3.2.3	$x = 450KBps, y = 200KBps, N = 4, V = 50$, and z and n are variables
Scenario 3.2.4	$x = 450KBps, y = 200KBps, z = 10MBps, n = 4$, and N and V are variables

Table 3. Scenarios for situation $z > x > y$

Scenario 3.2.1. The number of helper nodes, N , should be equal or greater than 3 in order to achieve uninterrupted playback. The relation between N and n is shown in Figure 3(a).

The group owner can effectively reduce the receiving rounds by adding more group members, as shown in Figure 3(a). However, we observe that the group owner does not need to have too many helper nodes. For example, the group owner needs to complete almost the same number of rounds when there are 6 group members and 12 group members.

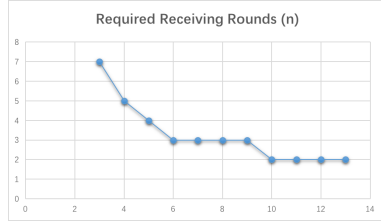
Scenario 3.2.2. The relation between the total number of video chunks, V , and the number of rounds n is shown in Figure 3(b).

We can see from Figure 3(b) that our application is very effective for long videos. For example, a video with 25 chunks and a video with 105 chunks require the same number of rounds. This is because of the fact that the rounds get longer as the video progresses and therefore, more chunks can be downloaded in one round (see Eq. (5) and related discussion).

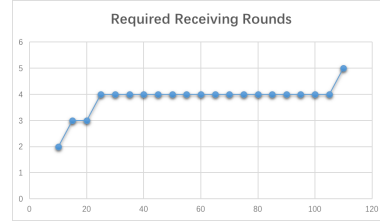
Scenario 3.2.3. The relation between the WiFi transfer rate z and the required number of rounds n is shown in Figure 3(c).

We observe from Figure 3(c) that compared to the number of group members and the size of the video presented before, the WiFi Direct transfer rate has less effect on the number of required rounds. We see that for any transfer speed above 1,500KBps, the required number of rounds stays the same and equal to 5. As noted before, this is because, after the first round, the number of chunks downloaded by the helper nodes increases exponentially.

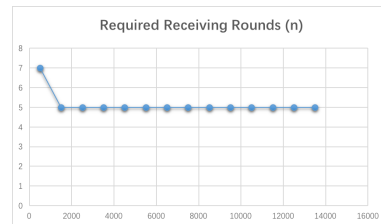
Scenario 3.2.4. The relation between the total number of chunks, V , of the video and the number of group members, N , is shown in Figure 3(d). As expected, we see from Figure 3(d) that with more group members the group owner can download longer videos.



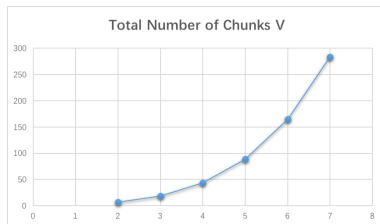
(a) The x-axis is N , and the y-axis is n .



(b) The x-axis is V , and the y-axis is n .



(c) The x-axis is $z(KBps)$, and the y-axis is n .



(d) The x-axis is N , and the y-axis is V .

Fig. 3. Results for situation $z > x > y$.

Scenario 3.3: $z > x > y$, Plus Periodic Disconnections. In this last case, we add periodic disconnections into the previous evaluation scenario. Therefore, the user should not only buffer content in advance, T_b , to avoid disruption due to slow downlink rate, but also buffer some extra content A to cover the upcoming disconnection period D . These requirements are shown in (10) and (11).

$$yT_b \geq (x - y)T_{pc} + A \quad (10)$$

$$\frac{A}{x} \geq \frac{yC}{z} \quad (11)$$

We also need to guarantee that the video content received from group members can cover the upcoming disconnection period. This requirement is shown in inequality (12).

$$\frac{NyC}{x} \geq D \quad (12)$$

We present four scenarios to study how much content should the group owner buffer and how long should the connection period be in order to overcome the disconnections and the limited cellular downlink rate. We also study the relation between the WiFi transfer rate and the length of the disconnection period that can be overcome without playback disruptions.

Scenario 3.3.1	$x = 350KBps$, $C = 300s$, $D = 500s$, $N = 4$, $z = 10MBps$, and T_b and y are variables
Scenario 3.3.2	$z = 10MBps$, $x = 350KBps$, $T_{pc} = 200s$, $D = 900s$, $N = 5$, and C and y are variables
Scenario 3.3.3	$x = 350KBps$, $y = 300KBps$, $T_b = 30s$, $C = 150s$, $N = 5$, and z and D are variables
Scenario 3.3.4	$y = 300KBps$, $T_{pc} = 200s$, $N = 5$, $D = 900s$, $z = 2000KBps$, and x and T_b are variables

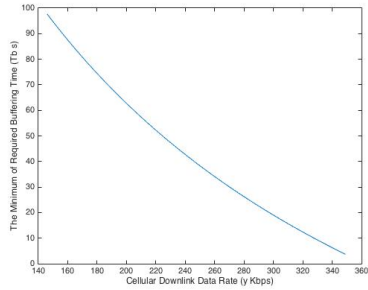
Table 4. Scenarios for situation $z > x > y$, plus periodic disconnections

Scenario 3.3.1. Figure 4(a) shows the relation between the required buffering time T_b and cellular downlink rate y . We see from Figure 4(a) that as the cellular downlink rate, y , increases, the required buffering time is becoming shorter. The group owner only requires about about 10s for buffering when the cellular downlink rate is 330KBps.

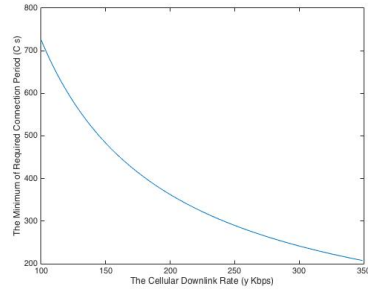
Scenario 3.3.2. Figure 4(b) shows the relation between cellular downlink rate y and the duration of the connection period C that guarantees undisrupted playback. In Figure 4(b), we see that as y grows, the required connection period, C , is becoming shorter. Still, however, in order to stream HD video, the connection period needs to be in the order of a couple of minutes.

Scenario 3.3.3. Figure 4(c) shows the relation between the WiFi Direct transfer rate and the length of the disconnection period that the user can overcome without disruptions. We see from Figure 4(c) that the faster the WiFi Direct rate the longer the disconnection period that the user can tolerate. For example, for WiFi Direct rate equal to 9000KBps (which is still within the range of the WiFi Direct specification), the user can tolerate up to 1400s of disconnection.

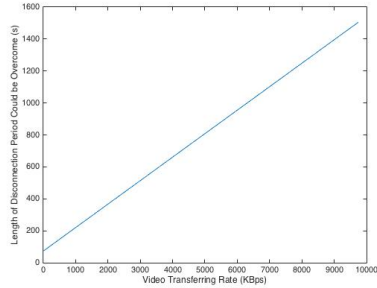
Scenario 3.3.4. Finally, in Figure 4(d) we present the relation between video data rate and the amount of content that the user needs to have buffered in terms of playback time. From Figure 4(d) we see that streaming high quality video requires long buffering time. The user requires about 300s when watching a video with 600KBps data rate.



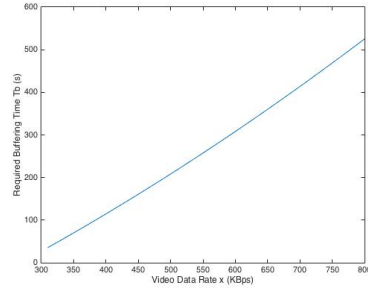
(a) Buffering Time vs Cellular Rate



(b) Connection Period vs Cellular Rate



(c) Disconnection Period vs D2D Rate



(d) Buffering Time vs Video Rate

Fig. 4. Scenario 3.3: $z > x > y$, plus periodic disconnections.

4 Conclusions

We have developed a collaborative video download application based on Wi-Fi Direct to improve the QoE of video streaming when the cellular connection is

constrained or physically disrupted. We have built simple models in order to study the requirements of our application to achieve seamless and undisrupted playback. Under the requirement of undisrupted playback, we study the relation between several parameters that affect playback in order to exploit the potential performance gains of our application.

Our work can be considered as a crowd-computing application. As next steps, we plan to develop a third party platform which allows users to exchange resources. For example, helper nodes spend 3G/4G data to help one user (*i.e.*, the group owner) download an HD video. In turn, the user who gets help rewards the helpers with electronic vouchers in return for their services.

Acknowledgements

This work was supported by EU FP7/NICT (GreenICN project) grant no. (EU) 608518/(NICT)167 and H2020 UMOBILE project, grant no. 645124.

References

1. W. Chai et al. Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services. *ComMag, IEEE*, pages 112–120, 2011.
2. B. Han et al. Cellular traffic offloading through opportunistic communications: a case study. In *CHANTS, ACM*, pages 31–38, 2010.
3. S. Ioannidis et al. Optimal and scalable distribution of content updates over a mobile social network. In *INFOCOM, IEEE*, pages 1422–1430, 2009.
4. D. Kutscher et al. Icn research challenges. *IRTF, IRTF Internet Draft—work in progress*, 2, 2014.
5. G. Pavlou, N. Wang, W. K. Chai, and I. Psaras. Internet-scale content mediation in information-centric networks. *Annals of Telecommunications, Springer*, 68(3-4):167–177, 2013.
6. M. Ramadan et al. Implementation and evaluation of cooperative video streaming for mobile devices. In *PIMRC, IEEE*, pages 1–5, 2008.
7. H. Seferoglu et al. Cooperative video streaming on smartphones. In *Allerton Conf. on Communication, Control, and Computing, IEEE*, pages 220–227, 2011.
8. M. Stiemerling and S. Kiesel. A system for peer-to-peer video streaming in resource constrained mobile environments. In *U-NET Workshop, ACM*, pages 25–30, 2009.
9. A. G. Tasiopoulos et al. Mind the gap: modelling video delivery under expected periods of disconnection. In *CHANTS, ACM*, pages 13–18, 2014.
10. A. G. Tasiopoulos et al. Tube streaming: Modelling collaborative media streaming in urban railway networks. In *IFIP Networking*, 2016.