

EDWiN : leveraging device-to-device communications for Efficient data Dissemination over Wi-Fi Networks

Lyes Hamidouche, Sébastien Monnet, Frédéric Bardolle, Pierre Sens, Dimitri Refauvelet

▶ **To cite this version:**

Lyes Hamidouche, Sébastien Monnet, Frédéric Bardolle, Pierre Sens, Dimitri Refauvelet. EDWiN : leveraging device-to-device communications for Efficient data Dissemination over Wi-Fi Networks. The 31st IEEE International Conference on. Advanced Information Networking and Applications (AINA-2017), Mar 2017, Taipei, Taiwan. hal-01515372

HAL Id: hal-01515372
https://inria.hal.science/hal-01515372

Submitted on 27 Apr 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.

EDWiN : leveraging device-to-device communications for **E**fficient data **D**issemination over **Wi**-Fi **N**etworks

Lyes Hamidouche*‡, Sébastien Monnet†, Frédéric Bardolle‡, Pierre Sens* and Dimitri Refauvelet‡
* Sorbonne Universités, UPMC Univ Paris 06, CNRS, Inria, LIP6
4 Place Jussieu, 75005 Paris, France
Email: Firstname.Lastname@lip6.fr
†LISTIC/Université Savoie Mont Blanc
5 chemin de bellevue, 74944 Annecy le Vieux
Email: Sebastien.Monnet@univ-smb.fr
‡ Magency
60, rue de Wattignies, Paris 75012, France
Email: firstname@magency.fr

***Abstract*—An emerging usage is to rely on mobile devices (Smartphones or tablets) for large-scale events. They can be used for many applications like live voting or chatting, but also to access all the data related to an event. However, in such case, handling mobile devices trying to access data simultaneously is difficult. A Wi-Fi access point can only handle a limited amount of devices. Current solutions, relying on pre-loading data on the devices or over-sizing the network equipments are not satisfying and are not even always possible. We propose an approach that leverages the capability of mobile devices to interact directly through device-to-device (D2D) communications. Our solution can be tuned to choose the right level of parallelization to cope with radio interferences, it also provides the ability to adjust the trade-off between efficiency and energy consumption. We evaluate our approach using a discrete event simulator. The results show that our approach using D2D communications brings a 30% gains with our simulation.**

***Index Terms*—Distributed systems, data dissemination, Wi-Fi Direct, wireless networks**

I. Introduction

Nowadays, mobile technologies have become very popular. Most people own a mobile device, such as a smartphone or a tablet. These devices now have good computing and networking capabilities, they offer many possibilities. In this paper, we focus on a particular use: large scale events. The participants of such events can use their[1] smartphones or tablets to communicate and participate to the different activities. Magency [1] is a company organizing such large scale events, it relies on the use of mobile devices. The participants use their mobile devices to interact (e.g. to vote, play quiz, post status, etc.) and to access data related to the event such as presentations, technical informations, or seating plans. All the devices will access the same set of data, which size can approach the Gigabyte (average sizes are usually between 100 and 300 Megabytes)[2]

During events, the mobile devices communicate through a Wi-Fi network. The infrastructure consists of a server and the required hardware to set-up a Wi-Fi network such as routers, switches and access points. When multiple devices access simultaneously to the same voluminous data, the network is congested. This phenomenon causes an increase in download completion time as Figure 1 illustrates. We analyse the impact of the number of parallel transfers on the download time while downloading a 100 Megabytes piece of data, using a Wi-Fi network. We observe that the download completion time grows proportionally with the number of parallel downloads (the bandwidth is shared among the devices).

During large-scale events, it is important that all the devices can access all the data quickly. However, as we just discuss, the dissemination of a large amount of data using a classical Wi-Fi network is not efficient. In order to face this problem, Magency teams prepare the devices before every event and pre-load the data that is going to be used during the event. Even if this method permits to avoid network congestion, it has an important time and resource cost. Mobile devices can also communicate with direct connections; our approach uses these connection as a mean to increase network overall throughput and reduce the infrastructure load.

The peer-to-peer P2P paradigm has already been used to tackle scalability problems of centralized client/server architectures. Solutions like P2P distributed hash tables [2], [3], P2P file sharing systems [4], [5], P2P distributed file systems [6], [7], [8], P2P distributed computing [9], [10], or P2P distributed caching [11] have been proposed. However, most of the research concerning the P2P paradigm focused on the

[1]Or devices lent by the event organizers.

[2]These informations have been collected from an analyse of events covered by Magency

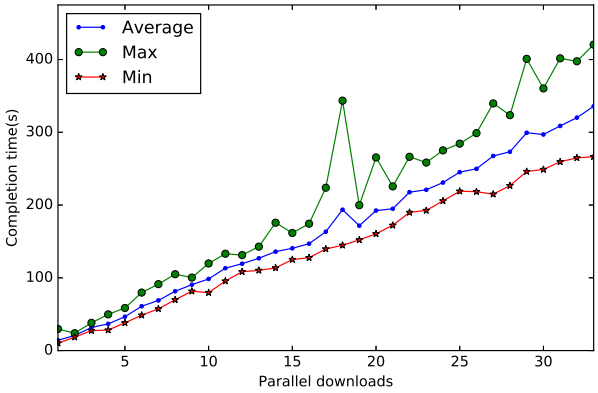


Figure 1. Measures on completion time for the dissemination of a 100 MBytes file overa set of 33 devices with one server and one access point

use of wired networks where peers are constantly supplied with electricity power and rely on a stable and high capacity networking communications.

Mobile technologies do not share the same properties with the traditional wired networks. In fact, a mobile device has a limited battery capacity and communicates through radio-based supports such as cellular connections, Wi-Fi or Bluetooth. Unfortunately, these technologies are subjects to interferences. Although the permanent increase of their computation performances and networking capabilities, mobile devices are still constrained by wireless networks interferences and limited battery life.

Device to device (D2D) technologies allow mobile devices to communicate through a direct connection without using any infrastructure. This is an opportunity to use nearby connections for collaborative networking. In research, collaborative D2D communication have been used for questions such as networks offloading [12], collaborative streaming [13], [14], [15] or file sharing [16].

The authors of [13], [14] present MoVi and its extension MoVi+, a distributed video streaming approach that uses a mix of the devices downlinks and opportunistic direct communications among devices to increase network overall throughput. This solution relies on a central tracking of the network. With the same goal, the authors of [15] propose MicroCast, a solution that exploits Wi-Fi overhearing and network coding to exceed the low data rates of conventional broadcast and improve overall network throughput for collaborative streaming on mobile devices. These solutions are designed for on-demand video streaming. The constraints are quite different from ours: the performance requirement is weaker (no need to download the video much faster than the player plays it) but the delivery order of the video segments is constrained. Finally, Bluetorrent [16], a P2P file sharing application using the Bluetooth technology is also close to our work. Their experiments have shown a performance improvement using their approach in comparison with traditional access point (AP) based content sharing. Yet, our approach targets bigger scales and data volumes and can not rely on the Bluetooth technology which data rates and ranges are not adequate.

Our approach architecture is inspired by MoVi [13], [14]: it is decentralized (we use direct connections among the devices for data propagation) but relies on a central control. A centralized tracking mechanism helps us to maintain an accurate information about the network state at a very low cost (saving synchronization messages among the devices).

In this paper, we present EDWiN : a hybrid approach using D2D communications with a central tracking to data dissemination performance over Wi-Fi networks, our contributions are:

- A distributed, bittorent-like [17], data dissemination mechanism using D2D connections, relying on a centralized decision;
- The ability to consider the effects of interferences by varying the number of devices networking simultaneously;
- A clustering method providing the ability to homogenize the networking activity time for a better control on energy consumption.

The following of this paper is is organized as follows: Section II describes the related works, Section III presents our approach, followed by its evaluation in Section IV and finally, Section V concludes and gives our perspectives.

II. Related Work

The existing P2P file sharing solutions [17], [4], [5] take benefit of multiple data sources (peers) to increase the speed of data retrieval. Many distributed file systems like Google File System (GFS) [6] Hadoop distributed file system (HDFS) [7] or OceanStore [8] rely on the P2P paradigm to provide efficient and scalable file system services, treating aspects such as availability scalability and resilience.Distributed hash tables like PAST [2] and Chord [3] are among the most used data structures to coordinate P2P systems. Maintaining such structures, implies a cost in terms of exchanged messages in order to keep the network state information up to date for every devices. Mobile Adhoc Networks (MANETs) are a kind of Mobile P2P systems. The network overhead of routing messages incurs additional bandwidth and energy consumption. In [18], the authors present different routing protocols for P2P file sharing over MANETs. In such networks, where scalability needs a careful maintenance, there is always a trade-off to make between scalability, routing, implementability, maintainability and energy efficiency. Many works have proposed P2P file sharing protocols for MANETs. Li Liu and al. [19] have analysed and classified research works around P2P file sharing over MANETs such as ORION[20], 7DS [21] and LocP2P [22]. These research works focus on dynamic network topologies, where nodes have low bandwidths, a limited storage space and short battery autonomy.

To the best of our knowledge, MoVi [13], [14], MicroCast [15] and bittorrent [17] are the closest approaches from ours. [13], [14], [15] share design aspects with EDWiN. Like MOVi [13], [14] and Microcast [15], we use D2D communications

for collaborative downloads from centralized data source. Furthermore, we use a "*rarest first*" chunk replication policy, like bittorrent [17].

MicroCast [15] uses a mix between cellular connections and nearby devices to increase the overall network throughput. The authors implemented a pseudo-broadcast solution that uses both unicast and Wi-Fi overhearing to distribute small data segments to the mobile devices. On the other side, MoVi [13], [14] leverages opportunistic adhoc connections among devices to increase data throughput for video streaming. Thanks to a central monitoring of the network interference fluctuations, MoVi exploits a mix use of downlink and ad hoc Wi-Fi communications among mobile devices to significantly improve the overall system throughput. This study shows that leveraging on best channel conditions and highest data rates not only improves data throughput, but also maintains a power consumption on mobile devices similar with a conventional VoD system (from AP to device). To stream media, it is mandatory to deliver the pieces of data in a given order. However, it is not mandatory to deliver data faster than the video player reads it. In our case, the goal is to deliver all the data to all the devices as fast as possible, regarding-less which pieces of data come first. In [16], the authors present Bluetorrent, a P2P file sharing solution for Bluetooth devices. This technology is constrained by a limited communication range and a small bandwidth. Whereas, in Wi-Fi networks we can handle networks with larger transmission ranges and higher data throughput than Bluetooth, we can then handle bigger data volumes.

Wi-Fi [23] technology is constantly improving: (*i*) higher data rates (Multi-Gbps with "*ad*" standard [24]), (*ii*) better D2D communication capabilities with Wi-Fi Direct [25] and (*iii*) neighbour awareness with Wi-Fi Aware [26]. In [27], the authors presented an extended overview of Wi-Fi Direct. Their study highlighted interesting properties of this networking technology: fast group formations and high data rates. Wi-Fi Direct is mostly used for Wireless Display and one-to-one file transfers, mainly in the Android ecosystem. Researches around this technology tackled different problems such as content sharing [28], [29] and cellular traffic offloading [12]. To the best of our knowledge, there is no approach that treated the problem of voluminous data dissemination over dense Wi-Fi Networks.

III. Leveraging D2D communications for Efficient data Dissemination over Wi-Fi Networks

In this section we detail our approach. Firstly, we describe our network model and detail our network representation. Then, we describe our dissemination algorithm. We conclude by presenting a variant of our algorithm for energy efficiency.

A. Model

We consider a system composed of a server, an access point and mobile devices as illustrated by Figure 2. The devices try to access to the same data, hosted by the server, at the same time. They can communicate with the access point and among them, through D2D connections. Our approach aims at reducing the dissemination time of a voluminous piece of data, partitioned in *chunks*, over a Wireless LAN (WLAN). For our large-scale events case study, this will provide the ability to pre-load all the event-related data on all the devices at the beginning of an event.

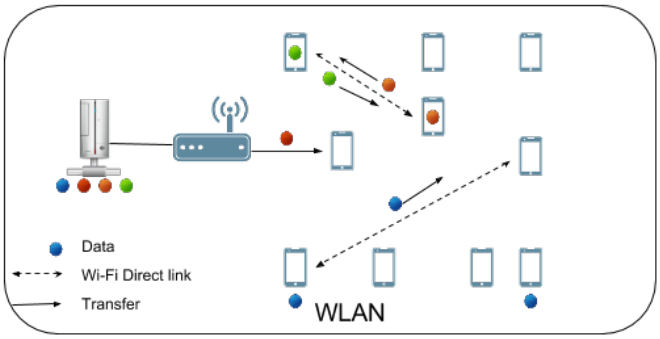


Figure 2. Wireless LAN with device-to-device Wi-Fi Direct connections between mobile devices

Parallel D2D exchanges can increase the network's overall throughput. Our algorithm main behaviour is to quickly generate multiple sources for each data chunk. With such a networking infrastructure, it is easy to maintain a global network state information at a negligible cost thanks to small control messages. The server acts as a scheduler for the exchanges that occur between mobile devices. The scheduling is based on a global knowledge of the network state.

As preliminary step of our study, we have run experiments to measure the average transfer and connection times using Magency Wi-Fi Direct equipped devicesWe observed an average connection establishment time under 4 seconds and data transfer rates around 8Mbps for D2D exchanges. During these experiments we have observed that while two devices A and B communicate, a transfer from A to B is not halved in the presence of a transfer from B to A. We thus favor bidirectional D2D exchanges rather than unidirectional ones when possible.

1) Network representation and assumptions: As we consider networks where all the devices are located in the same place, we assume that they are able to establish Wi-Fi Direct connections with all their neighbors [3]. In order to avoid the congestion of the network infrastructure and to quickly obtain new sources, only one transfer at a time from the server/access point is allowed. A second transfer would take half the bandwidth anyway, the global throughput would thus not be increased. All the other exchanges occur between mobile devices pairs through D2D connections. Thus, the server bandwidth is saved for other uses (e.g., messaging, notifications push, ...). A node can be in the following states: $Idle$, $Uploading$, $Downloading$ or $Exchanging$. The scheduler maintains a table that represents the current state of

[3]During our large scale events (conferences and seminars) this is true at the beginning of the event, during a "welcome" time and during plenary sessions.

the network shown in table I. This table (NST, for Network State Table) contains for each node: (*i*) its identifier in the network, (*ii*) its state and (*iii*) the chunks it hosts. Based on this information the scheduler orchestrates the exchanges within the network (the dissemination algorithm is detailed below).

As discussed in Section II, it is complex to run reproducible experiments on mobile networks. Furthermore, as precise a mathematical model for interferences is, it often remains far from the reality: unfortunately, interferences depend on many environmental factors and they are unavoidable. However, there is a consensus on the fact that more parallel communications will induce a higher probability of radio interferences. Our approach takes into consideration the presence of interferences by offering the possibility to tune the degree of parallelism in the network. In our algorithm, the maximal number of parallel exchanges among the mobile devices allowed at a given time is a parameter.

The centralized monitoring of the network allows the scheduler to control the parallelism degree. It also provides the ability to have a control over the Wi-Fi Direct group formations, facilitating the creation of D2D communication and enabling short delay connections.

Id.	State	Chunks
0	Uploading	0,1,2,3,4
1	Exchanging	1,2,3
2	Exchanging	0,4
3	Downloading	0,1
...		

Table I
NETWORK STATE TABLE (NST)

Besides the NST, the server maintains some parameters for the dissemination:

- $nbTransfers$: the number of current transfers;
- Max_{Par}: the number of authorized devices to exchange;
- Max_{AP}: maximum chunk count for an AP download;
- Max_{D2D}: the limit of chunks between in a D2D transfer;

B. Dissemination algorithm

For a network of N mobile devices, the NST contains $N+1$ entries. There are two types of nodes: the scheduler (1) and the mobile devices (N). The algorithm 1 describes the tasks of the scheduler, these tasks are: $INIT$ and $RECEIVE$. The $INIT$ task requires as an input the network state table NST and the chunk set CS which represents the chunk identifiers of the disseminated data. The $RECEIVE$ task deals with transfer completion notifications. The mobile devices that download the data have a unique task $RECEIVE$ that deals with the received messages; this task is described in Algorithm 2.

The $INIT$ task Initializes all the entries of the NST with empty chunk sets, except the first one, for the server (which host the data at initialization). It then starts the dissemination by sending the Max_{AP} chunks to a first device. The target devices are randomly chosen and the transferred chunks are selected with the "*rarest first*" policy. The chosen mobile device state is set to $Downloading$ and the AP state is set to $Uploading$.

When a device i finishes an operation (down/upload) it notifies the scheduler with a message $< Ack, Cs, j >$. The mobile device i here acknowledges the scheduler that it received the chunk set Cs from the device j.

From line 2 to 4, the scheduler first updates the corresponding NST entries by adding the downloaded chunks identifiers to the entry of i, and updating the devices availability states. As downloading through the AP is preferred to D2D exchanges, the scheduler first checks if the AP is $Idle$. If it is the case, it initiates a download by a randomly selected device using the "*rarest first*" chunk selection policy. From line 22 to 35, the scheduler computes the D2D exchanges between devices pairs while there are available pairs for exchanging and while the limit of parallel exchanges Max_{par} has not

Algorithm 1 Scheduler's algorithms

1: **procedure** INIT(NST,Cs)
2: $NST[0].chunks \leftarrow Cs$
3: $NST[i].chunks \leftarrow \emptyset, \forall\ i \in [1, N]$
4: $NST[i].state \leftarrow Idle, \forall\ i \in [1, N]$
5: $node \leftarrow random(N)$
6: $C \leftarrow rarestChunks(node, Max_{AP})$
7: $send(< Down, C, 0 >, node)$
8: $NST[0].state \leftarrow Uploading$
9: $NST[node].state \leftarrow Downloading$
10: $nbTransfers++$
11: **end procedure**

12: Upon receive $< Ack, Cs, j >$ from i
13: **procedure** RECEIVE($< Ack, Cs, i >$,j)
14: $NST[dest].chunks \leftarrow NST[dest].chunks \bigcup Cs$
15: **if** $NST[dest].state$ = $Exchanging$ **then**
16: $NST[dest].state \leftarrow Downloading$
17: $NST[j].state \leftarrow Uploading$
18: **else**
19: $NST[dest].state \leftarrow Idle$
20: $NST[j].state \leftarrow Idle$
21: **end if**
22: $p \leftarrow selectExchangePair(NST, Max_{D2D})$
23: **while** $p \neq Nil$ **and** $nbTransfers < Max_{Par}$ **do**
24: **if** $p.type = Down$ **then**
25: $send(< Down, p.Cs1, p.n2 >, p.n1)$
26: $NST[p.n1].state \leftarrow Downloading$
27: $NST[p.n2].state \leftarrow Uploading$
28: **else**
29: $send(< Exch, p.Cs1, p.Cs2, p.n2 >, p.n1)$
30: $NST[p.n1].state \leftarrow Exchanging$
31: $NST[p.n2].state \leftarrow Exchanging$
32: **end if**
33: $nbTransfers+ = 2$
34: $p \leftarrow selectExchange(NST)$
35: **end while**
36: **end procedure**

been reached. There are two types of downloads, the line 22 provides the ability to chose a pair p of devices, selected from the available nodes of the network. The selection is based on the hamming distance between the chunk sets of each device. The structure of p contains the informations that characterizes a D2D exchange: (*i*) the devices identifiers $n1$ and $n2$, (*ii*) the exchanged chunk sets $Cs1$,$Cs2$ and (*iii*) the nature of the exchange $type$. The $type$ specifies if one of the chunk set is empty ($Download$) or not ($Exchange$). $Cs1$ (resp. $Cs2$) is the chunk set that $n1$ (resp. $n2$) downloads from $n2$ (resp. $n1$).

Algorithm 2 Mobile devices task

```
1:  procedure RECEIVE(< type, cs1, cs2, src >)
2:      if src = 0 then
3:          download(Cs1, 0)
4:      else
5:          ConnectWith(src)
6:          if type = Down then
7:              download(Cs1, src)
8:          else msgType = Exch
9:              send(< Down, Cs2, myId >, src)
10:             download(Cs1, src)
11:         end if
12:     end if
13:     send(< Ack, cs1, src >, Sched)
14: end procedure
```

Algorithm 2 describes the behavior of the client. It is executed when the client receives an instruction from the scheduler. We identify three cases: (*i*) a download from the server, (*ii*) a D2D download and (*iii*) a D2D exchange. When the id of the source src is equal to zero, the device directly downloads data from the server. Otherwise, it first establishes a connection with src by calling the method $connectWith(src)$ before downloading the chunk set $Cs1$. After finishing the download, a device sends an acknowledgment message to notify the scheduler of the end of the transfer.

C. Clustering and energy efficiency

In our preliminary measurements we have observed a data rate of 8 Mbps for D2D exchanges using Apple iPad Mini tablets. The transfer time using D2D communications is more that 10 times longer when comparing with a direct download from the server. If a transfer lasts longer, a device consumes more energy.

We propose a variant in which we partition the network into clusters/groups. The server disseminates data to clusters, one by one (from cluster 0 to cluster $n-1$). While the server disseminates data to a cluster i, the previous cluster $(i-1)$ starts the dissemination to the next one $(i+1)$. After finishing the dissemination to a cluster, the server serves the following one which has already been pre-filled with the D2D transfers. This strategy provides the ability to limit the number of devices operating at the same time, and thus, to reduce the energy usage while still taking advantage of parallel transfers with D2D connections.

AP bw	D2D Transf. bw.	Exchange bw.	Connection delay
10 Mbytes/sec	820 Kbytes/sec	700 Kbytes/sec	4 sec

Table II
DELAYS AND BANDWIDTH CALIBRATION FOR THE SIMULATIONS

IV. EXPERIMENTATIONS AND EVALUATION

A. Evaluation conditions

We evaluated our approach using Peersim[30], a discrete event simulator. As a preliminary step, we have measured transfer delays between devices provided by Magency: a production server, a UniFi UAP-AC-PRO [31] and apple iPad mini tablets. In order to avoid latencies due to slow disk accesses, we placed data in a RAMDisk. The experiments have permitted to observe the behaviour of an infrastructure under an intensive workload (see Figure 1) and to measure realistic latencies and connection establishment delays for devices using Wi-Fi Direct connection.

Our prototype is composed of two layers: (*i*) an applicative layer implementing the exchanges between the devices and (*ii*) a transport layer representing the network, calibrated with the measures from our preliminary experiments (which are summarized in table II). Our default configuration consist of a network of 100 mobile devices (which corresponds to the number of devices that can be managed by a professional access point nowadays).

We have chosen to not simulate the network interferences because they depend on the environment and are hard to model indeed, and even real experiments are hardly reproducible. Thus, it is important to notice that our simulator gives results in an ideal, interference-free environment. However, in our simulations, we evaluate the impact of the number of parallel transfers, impacting the interference level. In real life, the parallelization degree should depend on the presence or not of network interferences.

To evaluate the benefit of our approach, we compare it against a centralized one (based on one access point delivering the data to all the nodes). We also run experiments to evaluate the role of the parameters of our algorithms on the completion time. Finally the impact of our clustering based method for energy saving is measured.

The evaluated metrics are:

- **Completion time :** the time it takes disseminate the data to all the devices;
- **Activity time :** the time spent networking by the mobile devices.

In our simulations, we considered a default network size of 100 mobile devices and a default data size of 100 Mbytes, network latencies have been calibrated with the values in table II, the default chunk size of 5Mbytes.

B. Scalability

The goal of our approach, is to disseminate the data as fast as possible to all the devices of the network. We have observed

through experimental measures that, using a single access point, the dissemination delay is proportional to the number of devices accessing the data simultaneously (see Figure 1). In our approach, we use multiple sources in order to reduce this time. Figure 3 represents the evolution of the number of completed devices in time by varying the number of maximum devices exchanging simultaneously in the network. We observe that the gain, in completion time, brought by our approach increases with the number of devices exchanging simultaneously. Obviously, increasing the parallelization degree accelerate the transfers and reduce the completion time. For example, when 33 nodes are authorized to download simultaneously, the gains are around 40% compared to the centralized version. The "17 nodes" plot represents a dissemination with 17 devices authorized to download simultaneously, which represents 8 devices exchanging with each other and a node downloading from the server (AP). It is important to notice that these results are obtain in our interference-free simulated environment. Depending on the environmental conditions and the technology in use, it may be impossible to reach 33 concurrent transfers.

With our architecture, thanks to the central tracking of the network it is possible to monitor the channels in use in order to define the communication channels by taking care to favor non-overlapping ones. Modern APs and devices are able to communicate in both 2.4GHz and 5Ghz bands. Which provides the ability to use the 3 non overlapping channels of 2.4GHz frequencies and 8 non overlapping frequencies in 5GHz in Wi-Fi networks. Thus, the "17 nodes" plot seems reasonable.

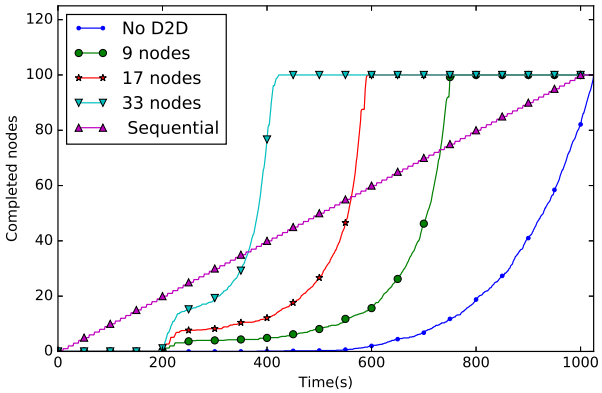


Figure 3. The impact of the number of parallel connections on data dissemination

In order to measure the scalability of our approach, we have run simulations with fixed parameters, except for the network size, which varies from 50 to 200. This range represents the typical average number of devices that can be observed during Magency events. The results of these simulations are presented by Figure 4. We observe that as the network size grows, the benefits of using EDWiN, our approach, grow.

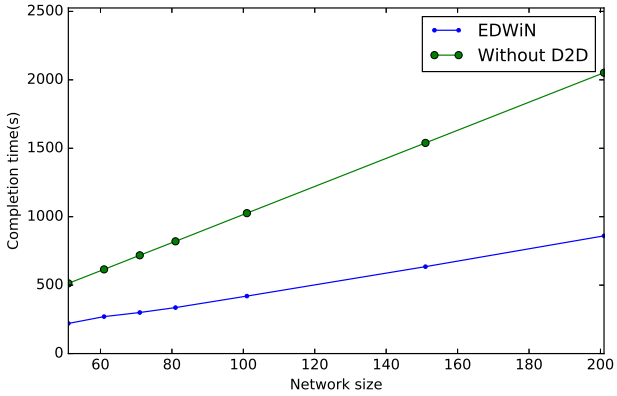


Figure 4. The impact of network size on the completion time

C. Dissemination parameters impact

Figure 5 shows the impact of chunk partitioning on the completion time. The figure shows that the better performance is obtain with $10Mb$ chunks. Chunks should be small enough to quickly create new sources, if they are too big it takes a long time to download them, and thus to create a new available source. However, if they become too small, a performance degradation can be observed. This performance degradation is due to the time the mobile devices spend establishing connections.

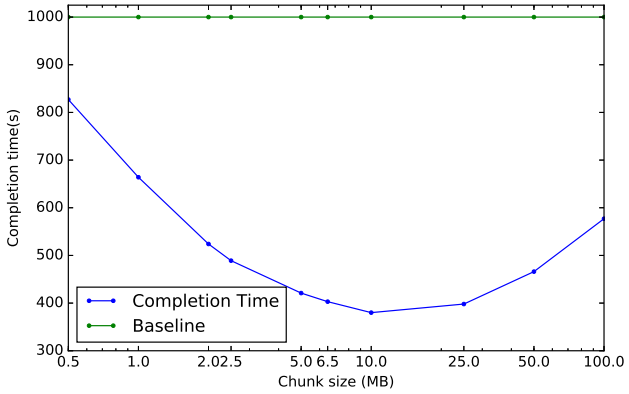


Figure 5. Impact of chunk size

Figure 6 shows the impact of the connection establishment delay on the performance. The connection delay strongly depends on the type of device used. We observe that its impact is important: with long connection establishment delays, the completion time increases and approaches the one of a centralized dissemination solution.

We also evaluate the impact of the Max_{AP} parameter, the maximum number of chunks that can be downloaded from the server in one download. When Max_{AP} is high, more chunks are downloaded from the server and the devices spend less time exchanging. On Figure 7 we can observe that while

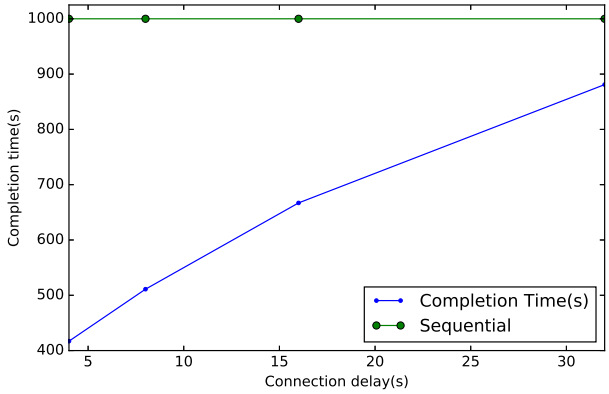


Figure 6. Impact of connection time on the completion time

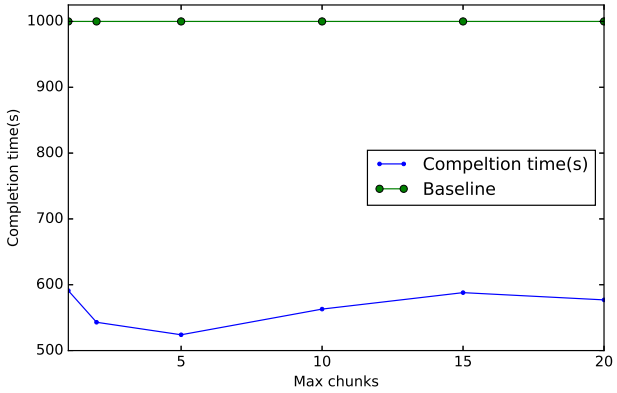


Figure 8. Impact of the maximum number of exchanged chunks between mobile devices

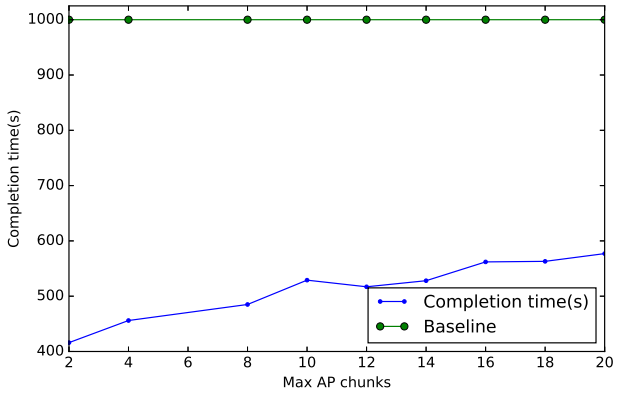


Figure 7. Impact of the maximum number of downloaded chunks from the AP

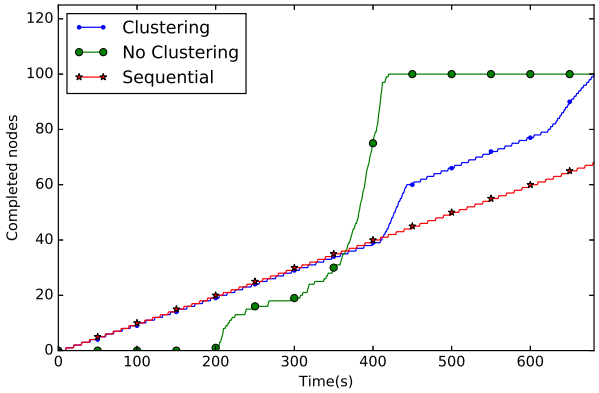


Figure 9. With vs without clustering

devices get more chunks from the server, they exchange less with each other, which causes an increase of the completion time.

Figure 8 represents the impact of the Max_{D2D} parameter on the completion time, Max_{D2D} being the maximum number of chunks two mobile devices can exchange at one time using a D2D connection. We observe an optimum time when Max_{D2D} is set to 5 chunks. If this parameter is too small, time will be lost in establishing connections, if it is too high it may reduce the parallelization degree. Of course, this parameter is closely coupled with the chunk size and depends on the connection establishment delay.

D. Clustering and Energy efficiency

We now evaluate the variant of our approach consisting in partitioning the device set into clusters to reduce the networking activity (see III-C for details). Our clustering method permits to reduce the energy consumption, as shown by Figure 10 but it has a performance cost as Figure 9 illustrates it (with groups/clusters of 20 devices). Our approach provides the ability to tune the trade-off energy consumption/performance by setting the group size.

E. Discussion

The use of D2D communications helps to improve the overall network throughput and permits to obtain interesting gains: more than 40% while letting 9 devices exchanging simultaneously.

It is important to consider environmental conditions of wireless networks. Tuning the parallelization degree of the network permits to consider the presence of wireless interferences and their impact on the performance of wireless communication. Furthermore, all the results presented in this section a obtain by simulating an interference-free environment. It is also important to correctly tune the data chunk size and the number of chunks that can be downloaded simultaneously. It should not be too small in order to amortize the connection establishment cost, and not too big in order to improve the parallelization degree. Finally, as the number of simultaneous networking devices is necessarily limited to reduce the effect of network interferences, the manner this limit is imposed is important. By clustering the set of devices into groups, it is possible to save energy by allowing mobile devices in inactive groups to switch into power save mode.

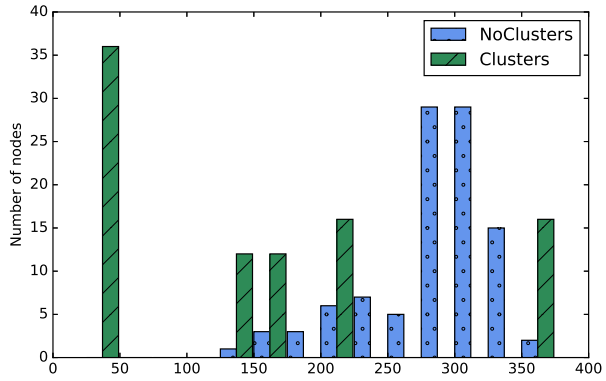


Figure 10. Comparison between the activity time with and without clustering

V. Conclusion

In this paper, we have presented EDWiN, our approach that leverages D2D connections to improve data dissemination within a large set of mobile nodes. Our solution is pragmatic, it is based on a centralized coordination and can be implemented at application layer (no need to change the networking protocols). EDWiN is tunable: it is possible to reduce the parallelization degree if network interferences are detected, it is also possible to choose between performance and energy consumption.

We plan to experiment an implementation of our solution using Magency's mobile devices during different large-scale events and evaluate EDWiN's adaptability.

References

[1] "Magency," http://www.magency.fr/.

[2] P. Druschel and A. Rowstron, "Past: A large-scale, persistent peer-to-peer storage utility," in *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*. IEEE, 2001, pp. 75–80.

[3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01)*, 2001, pp. 149–160. [Online]. Available: http://portal.acm.org/citation.cfm?doid=383059.383071

[4] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," *Proceedings First International Conference on Peer-to-Peer Computing*, pp. 1–11, 2001.

[5] J. Liang, R. Kumar, and K. W. Ross, "Understanding kazaa," 2004.

[6] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, p. 29, 2003.

[7] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010*, 2010.

[8] D. Bindel, Y. Chen, P. Eaton, D. Geels, R. P. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zhao, and J. Kubiatowicz, "Oceanstore: An architecture for global-scale persistent storage," *Asplos*, vol. 35, no. 11, pp. 190–201, 2000.

[9] "SETI@HOME," http://setiathome.ssl.berkeley.edu/.

[10] "Folding@Home," http://folding.stanford.edu/.

[11] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proceedings - IEEE INFOCOM*, 2010.

[12] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "3GPP LTE traffic offloading onto WiFi Direct," in *2013 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2013*, 2013, pp. 135–140.

[13] H. Yoon, J. Kim, F. Tan, and R. Hsieh, "On-demand video streaming in mobile opportunistic networks," in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*. IEEE, 2008, pp. 80–89.

[14] H. Lee, J.-Y. Yoo, and J. Kim, "Movi+: Opportunity extension for mobile peer-to-peer video on demand," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*. IEEE, 2013, pp. 247–252.

[15] L. Keller, C. Fragouli, and U. C. Irvine, "MicroCast : Cooperative Video Streaming on Smartphones Categories and Subject Descriptors," *MobiSys 2012*, pp. 57–69, 2012.

[16] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla, "Bluetorrent: Cooperative content sharing for bluetooth users," *Pervasive and Mobile Computing*, vol. 3, no. 6, pp. 609–634, 2007.

[17] J. A. Johnsen, "Peer-to-peer networking with BitTorrent," 2005.

[18] G. Ding and B. Bhargava, "Peer-to-peer file-sharing over mobile ad hoc networks," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, 2004, pp. 104–108.

[19] L. Liu, Y. Jing, Y. Zhang, and B. Xia, "A Survey on P2P File Sharing Algorithms over MANETs," *Do Not Cite - Only Use*, vol. 2, pp. 109–115, 2013.

[20] A. Klemm, C. Lindemann, and O. P. Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 4. IEEE, 2003, pp. 2758–2763.

[21] M. Papadopouli and H. Schulzrinne, "A Performance Analysis of 7DS: A Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users," in *Proc. of IEEE Advances in Wired and Wireless Communications*, 2001, pp. 1–4.

[22] Y. C. Tung and K. C. J. Lin, "Location-assisted energy-efficient content search for mobile peer-to-peer networks," in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2011*, 2011, pp. 477–482.

[23] Wi-Fi Alliance, "Wi-Fi CERTIFIED Wi-Fi Direct," *October*, no. October, p. 6, 2010.

[24] E. Perahia, C. Cordeiro, M. Park, and L. L. Yang, "IEEE 802.11ad: Defining the next generation multi-Gbps Wi-Fi," *2010 7th IEEE Consumer Communications and Networking Conference, CCNC 2010*, no. November 2008, 2010.

[25] W.-F. Alliance, "Wi-Fi Peer-to-Peer (P2P) - Technical specification," 2014. [Online]. Available: http://www.wi-fi.org/download.php?file=/sites/default/files/private/Wi-Fi_P2P_Technical_Specification_v1.5.pdf

[26] ——, "Neighbor Awareness Networking Technical Specification - Version 1.0," 2015.

[27] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *Wireless Communications, IEEE*, vol. 20, no. 3, pp. 96–104, 2013.

[28] C. 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 *Proceedings of the WoWMoM 2015: A World of Wireless Mobile and Multimedia Networks*, 2015.

[29] T. N. Duong, N. T. Dinh, and Y. Kim, "Content sharing using P2PSIP protocol in Wi-Fi direct networks," in *2012 4th International Conference on Communications and Electronics, ICCE 2012*, 2012, pp. 114–118.

[30] "Peersim," http://peersim.sourceforge.net/.

[31] "Unifi AC - 802.11ac Dual-Radio Access Points," https://dl.ubnt.com/datasheets/unifi/UniFi_AC_APs_DS.pdf.