



**HAL**  
open science

# SDN-based Wi-Fi Direct Clustering for Cloud Access in Campus Networks

Thi-Mai-Trang Nguyen, Lyes Hamidouche, Fabien Mathieu, Sébastien Monnet, Syphax Iskounen

► **To cite this version:**

Thi-Mai-Trang Nguyen, Lyes Hamidouche, Fabien Mathieu, Sébastien Monnet, Syphax Iskounen. SDN-based Wi-Fi Direct Clustering for Cloud Access in Campus Networks. *Annals of Telecommunications - annales des télécommunications*, 2018, 73 (3), pp.239–249. 10.1007/s12243-017-0598-z . hal-01567735

**HAL Id: hal-01567735**

**<https://hal.sorbonne-universite.fr/hal-01567735>**

Submitted on 29 Jun 2018

**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.

## SDN-based Wi-Fi Direct Clustering for Cloud Access in Campus Networks

Thi Mai Trang Nguyen · Lyes  
Hamidouche · Fabien Mathieu ·  
Sébastien Monnet · Syphax Iskounen

Received: date / Accepted: date

**Abstract** Mobile cloud is changing the way to enroll teaching activities in university campus. Lectures and lab sessions can be carried out directly from tablets in a class room by accessing a server in the cloud. In this paper, we address the problem of high density cloud access with wireless devices in campus networks. We propose to use Wi-Fi Direct clustering to solve the problem of Quality of Service (QoS) degradation when a high number of wireless devices want to access a content in the cloud at the same time. A centralized software-defined network controller is used in our proposed architecture to capture the network state and organize the Wi-Fi Direct groups. The optimized number of clusters can be calculated in function of the number of devices in the room. By simulations, we show that we can provide a better QoS in terms of download time and application's throughput by reducing the interference in this dense wireless network environment.

**Keywords** Wi-Fi Direct · software-defined networking · cloud network access

---

Thi Mai Trang Nguyen  
LIP6, Sorbonne Université, UPMC, CNRS, 4 Place Jussieu, 75005 Paris, France  
Tel.: +33-1-4427-7125  
E-mail: Thi-Mai-Trang.Nguyen@lip6.fr

Lyes Hamidouche  
Magency, 60 Rue de Wattignies, 75012 Paris, France  
LIP6, Sorbonne Université, UPMC, CNRS, 4 Place Jussieu, 75005 Paris, France

Fabien Mathieu  
NOKIA Bell Labs, 91620 Nozay, France

Sébastien Monnet  
LISTIC, Université Savoie Mont Blanc, 5 Chemin de Bellevue, 74944 Annecy le Vieux,  
France

Syphax Iskounen  
LASS, CNRS, 7 Avenue du Colonel Roche, 31400 Toulouse, France

## 1 Introduction

Cloud communications and networks since their invention have shown an important impact on our society. Starting by online file storage services such as Google Drive and Dropbox, today's cloud networks provide users with numerous advanced services with mobile access. For the education sector, mobile cloud services are changing the way that we teach at universities. Tablets have become a common tool for users and can be given to students at the beginning of the lecture with slides and lecture notes. Demonstrations can be easily done by video streaming through a Wi-Fi access available in the auditorium. Teaching materials can be updated with less cost and effort compared to the printed version. For numerous teachings in computer science, lab sessions can be done over tablets via an Internet connection to a distant server where the experiments are really running. Small tablets also consume less energy than fixed PC, making the campus greener.

However, providing Quality-of-Service (QoS) in dense wireless networks is a challenge. Many wireless devices in a small geographic area can drastically degrade the overall network performances and decrease the user's Quality-of-Experience (QoE). Two main reasons are the interference generated by simultaneous transmissions and the heavy traffic load requested by all the devices through the access point. With the high density of devices set at a high transmission power to reach the access point in the auditorium, the number of transmission collisions at the Medium Access Control (MAC) layer is very high. Collided frames are erroneous and must be retransmitted by the transmitter with an additional waiting Back-off time. As a consequence, data transmission delay and packet loss ratio are very high in the network.

Wi-Fi Direct [1], a new technology widely available in tablets and smartphones, allows users to communicate directly to each other. Two nearby Wi-Fi users can activate Wi-Fi Direct communications to exchange pictures. A group of users can form a Wi-Fi Direct group to communicate without the necessity of having an access point. Recognizing the high automation and flexibility offered by Wi-Fi Direct to form user groups, we propose in this paper to use the Wi-Fi Direct clustering to reduce the interference in dense Wi-Fi campus networks. By dividing a large number of Wi-Fi users into smaller groups, combined with frequency assignment and power control techniques, we can reduce the overall interference level in the network and increase network performances. For the Wi-Fi Direct group formation to be efficient, a centralized controller following the Software Defined Networking architecture is used for network control.

While the main research motivation of this paper is focused on campus networks, other use cases such as events or conference venues can also take advantage of Wi-Fi Direct-based D2D communications. This technology can reduce data dissemination time in large scale events. There are companies (e.g. Magency [2]) that are specialized in providing mobile technology-based solutions to animate events. During an event, the attendees receive devices that they use to participate in different activities. Depending on the needs of each

event, the number of devices simultaneously connected to the network can go up to thousands. The size of the data used by devices can go up to more than 1 GB. Using traditional Wi-Fi networks poses serious problems of scalability. Even if some practical solutions can be used to meet user's needs such as oversizing the network infrastructure or pre-loading data onto the devices, they only partially solve the problem. In fact, these solutions cost additional resources in terms of hardware and time required. Furthermore, when a high number of access points are physically close to each other, network throughput can be drastically reduced due to wireless interferences.

The remainder of this paper is organized as follows. In section 2, we give an overview on the Wi-Fi Direct technology and SDN architecture. Section 3 presents our SDN-based architecture for Wi-Fi Direct clustering. Section 4 presents our implementation of Wi-Fi Direct in the OMNeT++ simulator and evaluates the performances of the proposed architecture. Finally, section 5 concludes the paper.

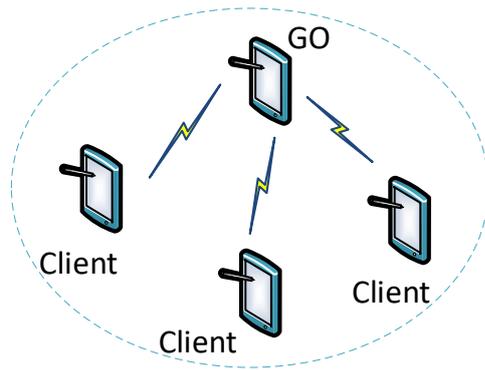
## 2 Background

### 2.1 Dense Wi-Fi networks

Since the introduction of Wi-Fi networks at the beginning of the 21st century, this technology has been widely deployed in university campus, event centers and hotspots. The density of Wi-Fi hosts and access points have also been significantly increasing during the last five years. Especially, with the coming of smartphones and tablets with Wi-Fi integrated, users are moving to use Wi-Fi as the default connectivity to access cloud-based applications. Wi-Fi is also a principle wireless technology to offload mobile traffic. In most of campus networks, wireless connectivities are totally based on Wi-Fi. Following the Cisco VNI report [3], 60 percent of mobile traffic has been offloaded over Wi-Fi and small-cell networks in 2016. Wi-Fi traffic from both mobile devices and Wi-Fi only device contributes 42 percent of the total IP traffic in 2015 and will account for 49 percent of total IP traffic by 2020. Total public Wi-Fi hotspots will grow six-fold, from 94 million in 2016 to 541.6 million by 2021. By 2021, there will be 8.3 billion handheld or personal mobile devices all over the world.

In high density Wi-Fi networks, the Quality-of-Service is degraded. Maity and al. [4] have shown that TCP download performance degrades significantly in a dense network, e.g. a network with 20-30 clients per access point. Especially, in a classroom scenario, a student may need up to 229 seconds to download a quiz file of 200KB and up to 478 seconds to complete the download of a reference file of 5MB. In a residential scenario, the number of clients per access point is small but there can be up to 500 access points per square kilometer area [5] resulting in high interference and poor network performances.

The solutions to dense Wi-Fi network performances have been proposed mainly in three directions - Device-to-Device (D2D) communications, power control and channel selection. D2D communications [6] allow direct communi-



**Fig. 1** Wi-Fi Direct mode

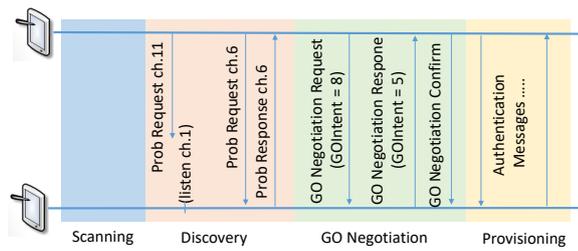
cations between devices to reduce the traffic going through the access points. Power control and channel selection techniques [5] enable an efficient interference management.

## 2.2 Wi-Fi Direct mode

In the Wi-Fi networks, two communication modes have been defined in the IEEE 802.11 standard: the infrastructure mode and the ad-hoc mode. The infrastructure mode, the most widely deployed mode, is based on a network infrastructure composed of access points deployed by a network administrator. When a user wants to send data to another, the data is transmitted to the access point which will retransmit the data to the destination, even if the two users are close to each other. In the ad-hoc mode, there is no access point. Devices in the transmission range of each other can communicate directly as a single-hop communication.

Recently, the Wi-Fi Alliance has defined the Wi-Fi Direct mode to facilitate the D2D communications among nearby Wi-Fi users. In a Wi-Fi Direct group (Figure 1), one of the nearby devices is elected as Group Owner (GO) which plays the role of access point. The other devices are clients and communicate to each other via the GO. Wi-Fi Direct can be seen as a combination of the infrastructure mode and the ad-hoc mode because we still have an AP (the GO) like the infrastructure mode but this AP is also a client as in the ad-hoc mode.

A standard Wi-Fi Direct group formation (Figure 2) is composed of four steps: scanning, discovery, GO negotiation and provisioning [7]. First, a device scans to verify if there is an existing group. If there is no beacon message sent by an existing GO, the device moves on to the second step to discover a peer. The device switches between scanning and listening states over randomly chosen channels until there are two devices sending the ProbRequest message and receiving the ProbResponse message over the same channel. The third



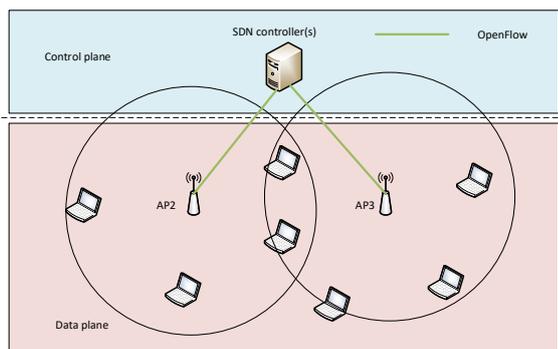
**Fig. 2** Standard Wi-Fi Direct group formation

step, GO negotiation, is a three-way handshake. Each peer communicates a GO Intent value which is randomly chosen between 0 and 14. The device with the greater GO Intent value becomes GO. In the last step, the peers exchange authentication messages to secure the communications.

Wi-Fi Direct is a strong candidate to offload cellular traffic [8]. The ability of quickly creating groups and disseminating information of Wi-Fi Direct can be used in mission-critical communications for public safety [9]. Proactive routing protocols can be used with Wi-Fi Direct to build Mobile Ad-hoc Networks (MANET) [10]. An open-source software has been also developed with Android devices for opportunistic data transmission with Wi-Fi Direct[11].

### 2.3 Software-Defined Networking in Wi-Fi Networks

SDN is a new concept which separates the network control from data forwarding (Figure 3). The data plane includes network devices which forward user data. The control plane is composed of one or more SDN controllers which have a consistent and global view to control the network. The SDN controllers communicate with the APs via the an SDN protocol such as OpenFlow [12].



**Fig. 3** SDN in Wi-Fi networks

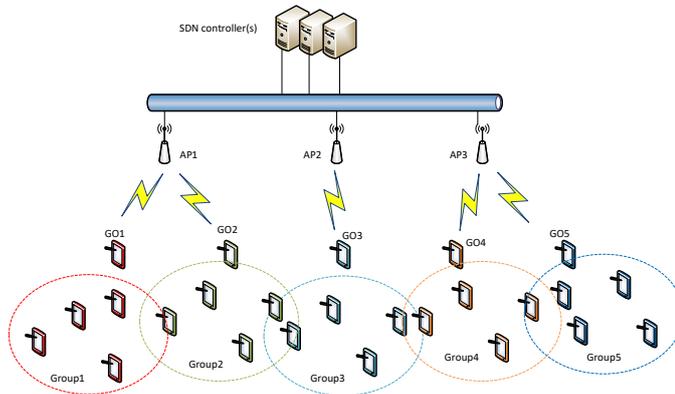
In Wi-Fi networks, the centralized view of the SDN controller can efficiently optimize the complex wireless environment. In [13], SDN has been used for interference management. The SDN abstractions provide a global view of channel quality and interference in the network and allow quick actions on reconfiguration such as channel assignment. The authors in [14] proposed an architecture using SDN for the management of virtual Wi-Fi access points. Flow tables in virtual access points are reprogrammed when users move in the network to provide seamless handovers. In [15], SDN has been used for channel assignment in dense Wi-Fi networks. Channel assignment problem is modeled as an optimization problem and implemented in the SDN controller to react when the interference level in the network gets higher than a predefined threshold. The centralized wireless management enabled by SDN can improve up to 12 Mbps of channel capacity measured at physical layer compared with the uncoordinated one.

When applied to wireless networks, the SDN concept is not limited to flow configuration function initially proposed by the OpenFlow protocol. SDN in wireless networks takes advantage of the centralized view of the controller which is very useful for efficient channel assignment, power control, interference management and resource allocation. Some proposed solutions are compatible with OpenFlow while others are not OpenFlow compliant [16]. Many research contributions are related to the control of virtualized and programmable radio networks.

### 3 SDN-based Wi-Fi Direct Clustering

#### 3.1 Network architecture

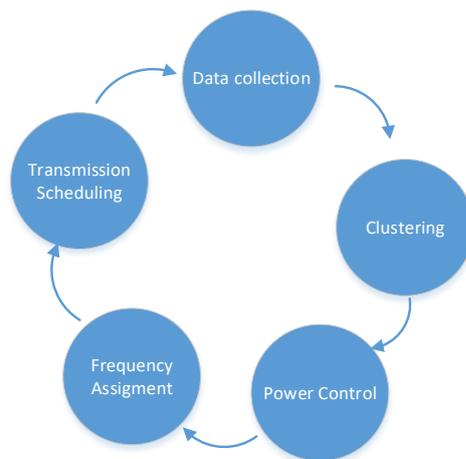
In this paper, we propose an SDN-based architecture for Wi-Fi Direct clustering as illustrated in Figure 4.



**Fig. 4** Network architecture

In the control plane, one or more SDN controllers are connected to the network of APs. When a new user wants to join the Wi-Fi network, SDN controllers and the overlay authentication service ask the user for information not only for user authentication such as user name and password but also the information related to the position of the user such as the seat number in an auditorium. Based on the number of users and their positions, the SDN controller can decide to form Wi-Fi Direct groups to optimize the network performances. For this purpose, the SDN controller can activate certain users to be GOs, configure their transmission powers and assign them the frequencies to be used. The SDN controllers also tell the other devices the Wi-Fi Direct group that they should join.

In the data plane, a device acting as a GO plays the role of an AP for the clients within its group and the role of client vis-a-vis the APs in the campus network infrastructure. In other words, it should be able to forward data between a client in the group and the infrastructure AP. From a network architecture point of view, the GO in our architecture is equivalent to an ad-hoc node which plays both roles of host and router with a routing policy controlled by the SDN controller. An OpenFlow compliant protocol can be used for the communication between the GO and the SDN controller for this purpose. For other devices which are not GOs, the data transmission is simpler. They select the specified GO in the same way that they select an AP among the available APs in the surrounding and use the GO as their single point of attachment to the network. All data from a client to the network will be sent to its GO.



**Fig. 5** SDN control loop

With this architecture, network administrators can develop advanced control algorithms for the SDN controller such as clustering, power control, chan-

nel assignment or data transmission scheduling. The centralized view on network topology at the SDN controller can enable efficient clustering and interference management algorithms especially for dense wireless networks. The control in the above-presented network architecture can be summarized by a five-step control loop as shown in Figure 5. The control loop repeats periodically at a frequency configurable by network administrator or every time there is a new user coming in the room. When a student changes the seat, the seat number can be updated through the application developed by the event planner. Indoor positioning techniques can also be used to detect the position of the user. However, in the case of campus networks, requesting users to update the seat number through an application is feasible. If the moving device is a client and the user moves out of the range of its GO, it will automatically detect the presence of the new GO close to the new seat thanks to beacon messages and perform the group joining procedure. If the moving device is a GO, the clients of this group will detect its absence due to the absence of beacon messages. As a consequence, the clients will try to connect to the access point of the auditorium using the infrastructure mode, similar to the case of new arrivals. The SDN controller can intelligently detect this case and assign a new GO node for the group. The details of these control steps are presented in the following subsections.

### 3.2 Clustering Algorithm

It is possible for the SDN controller to use different clustering algorithms. In this paper, we use the K-means algorithm [17] which divides a set of  $N$  nodes in the network into  $K$  groups called clusters by minimizing the sum of the distances from the nodes belonging to a given group to the center of gravity of that group, called the *centroid*. Applied to our network architecture presented in Figure 4, the node that is closest to a centroid is designed as GO for the corresponding cluster. Each cluster corresponds to a Wi-Fi Direct group.

$K$ -means algorithm is an unsupervised learning algorithm which is used to solve clustering problems. It is used to compute  $K$  clusters out of a given set of  $N$  elements based on their positions. The aim of this algorithm is to find  $K$  centroids, one for each cluster.

Our network  $\mathcal{N}$  is composed of a set of  $N = |\mathcal{N}|$  devices. For ease of notation, we map the devices in  $\mathcal{N}$  to the integers  $1, \dots, N$ . Device positions are stored in a set  $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$ , where  $p_j = (x_j, y_j)$  is the position of device  $j$ .  $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$  is the partition of  $\mathcal{N}$  into  $K$  clusters computed with the  $K$ -means algorithm.  $\mathcal{O} = \{o_1, o_2, \dots, o_K\}$  is the associated set of centroids ( $o_k = (x_{o_k}, y_{o_k})$  denotes the coordinates of the centroid of cluster  $C_k$ ).

Informally, the  $K$ -means works as follows: initially,  $K$  centroids are randomly chosen among the coordinates of the  $N$  devices. After some iterations, the  $K$ -means algorithm returns a set  $\mathcal{C}$  of  $K$  clusters, with the associated cen-

troids, which forms a Voronoi diagram: each node belongs to the cluster whose centroid is closest to it. This property is summarized by Equation (1):

$$\forall C_k, C_l \in \mathcal{C}, C_k \neq C_l, \forall j \in C_k, \text{dist}(j, o_k) < \text{dist}(j, o_l), \quad (1)$$

where  $\text{dist}$  is the Euclidean distance:

$$\text{dist}(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

---

**Algorithm 1** K-means algorithm
 

---

```

1: function K-MEANS( $\mathcal{N}, K$ )
2:    $finished \leftarrow false$ 
3:    $\mathcal{O} \leftarrow \text{randomCentroids}(\mathcal{N}, K)$  ▷ Select random centroids
4:   while  $!finished$  do
5:      $\mathcal{O}' \leftarrow \mathcal{O}$  ▷ Save the old centroids
6:      $\mathcal{C} \leftarrow \text{assignClusters}(\mathcal{O}, \mathcal{N})$  ▷ Compute new clusters
7:      $\mathcal{O} \leftarrow \text{calculateCentroids}(\mathcal{C})$  ▷ Compute the new centroids
8:     if  $\mathcal{O} == \mathcal{O}'$  then
9:       return  $\langle \mathcal{C}, \mathcal{O} \rangle$  ▷ The centroids did not change
10:    end if
11:  end while
12: end function

```

---

Algorithm 1 details the procedure which computes the cluster partition  $\mathcal{C}$  and the centroids  $\mathcal{O}$ . The inputs of these algorithms are the description of the set  $\mathcal{N}$ , (including the set of positions  $\mathcal{P}$ ), and the number of the clusters to calculate,  $K$ . As explained above, the first centroids are chosen with random values among the locations of the devices. Then the function iterates until we converge to a set of centroids that doesn't change. This means that the centroids of the  $i^{th}$  and the  $(i + 1)^{th}$  iterations are the same. At every loop, we first save the values of the centroids of  $\mathcal{O}$  in order to compare them with the new centroids. Then, the  $\text{assignClusters}(\mathcal{O}, \mathcal{N})$  function builds a cluster  $\mathcal{C}$  by computing, for each point  $j \in \mathcal{N}$ , the minimal distance between  $j$  and every cluster centroid  $o_k$ , and assigning  $j$  to its closest centroid.

Algorithm 2 details the instructions of this function.

---

**Algorithm 2** Assign Clusters
 

---

```

1: function ASSIGNCLUSTERS( $\mathcal{O}, \mathcal{N}$ )
2:    $\mathcal{C} \leftarrow \{C_1 = \emptyset, C_2 = \emptyset, \dots, C_{|\mathcal{O}|} = \emptyset\}$ 
3:   for  $i \in \mathcal{N}$  do
4:      $k = \arg \min_{l \in [1, K]} \text{dist}(i, o_l)$  ▷ Find the closest centroid
5:      $C_k = C_k \cup \{i\}$  ▷ Add device to cluster of closest centroid
6:   end for
7:   return  $\mathcal{C}$ 
8: end function

```

---

Back to Algorithm 1, the cluster set computed by the function  $\text{assignClusters}(\mathcal{O}, \mathcal{N})$  will then be used next to compute new centroids in

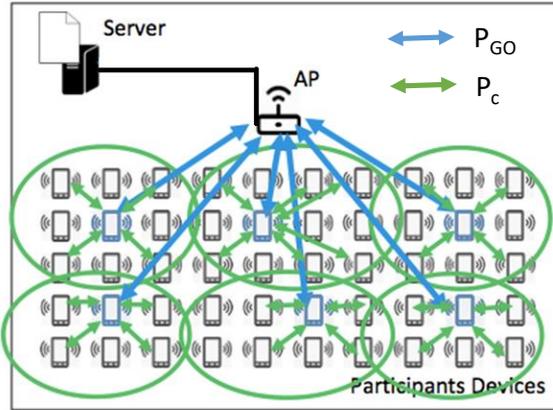
line 7. For each cluster  $C_k$ , the distance between its centroid and the points is minimal in comparison with the other centroids of  $\mathcal{O}$ . After this step, function  $calculateCentroids(\mathcal{C})$  computes the new centroids of the new clusters. The new centroid set  $\mathcal{O}$  contains  $K$  centroids. The coordinates of the centroid  $o_k$  of a cluster  $C_k \in \mathcal{C}$  are obtained by averaging the coordinates of its points (see (2)).

$$x_{o_k} = \frac{\sum_{i \in C_k} x_i}{|C_k|}, y_{o_k} = \frac{\sum_{i \in C_k} y_i}{|C_k|} \quad (2)$$

If the centroids did not change after an iteration ( $\mathcal{O} = \mathcal{O}'$ ), the algorithm stops iterating and returns the optimal  $k$ -partition for the network. For each cluster, the device that is located in the closest position to the corresponding centroid will be elected as GO for data dissemination.

### 3.3 Interference management and transmission scheduling

As indicated in Figure 5, after the Wi-Fi Direct clustering step, the SDN controller calculates the transmission powers of GO and client in order to minimize the interference in the network. Based on the positions and the role of the devices, two types of transmission power are calculated as illustrated in Figure 6.



**Fig. 6** Power control combined with Wi-Fi Direct clustering

Each GO is assigned with a high transmission power  $P_{GO}$  so that it can communicate with the access point in the auditorium. Non-GO devices are only assigned with a small transmission power  $P_c$  which is sufficient to reach the GO of the group.

After the power control procedure (Figure 5), the SDN controller verifies if there may be interference between groups. Two clients at the border of two clusters can interfere each other if they use the same frequency. To avoid interference as much as possible, the SDN controller builds an interference graph. Each node in the interference graph represents a cluster. An edge connecting two nodes in the graph represents the possible interference between members of the two corresponding groups. With this interference graph, the SDN controller can apply a coloring algorithm [18] for frequency assignment.

Finally, the data transmission is scheduled into two steps : data transmission between an access point and GO, and data transmission between a GO and their clients. For this purpose, the Notification of Absence (NoA) [1] protocol defined in Wi-Fi Direct is used by the GO to switch between the Wi-Fi Direct mode and the infrastructure mode. This protocol allows the GO to notify the clients in the group that it will be absent for a moment then switch into the infrastructure mode to get the data via the access point.

To some extent, the interference management problem in Wi-Fi Direct is similar to Femtocell networks. In Femtocell networks [19], a set of femto base stations are deployed in the same coverage area of macro base stations. A user can be associated to a femtocell or a macrocell. The femto base stations in cellular networks are equivalent to the GO nodes in our architecture. The macro base stations in cellular networks are equivalent to the Wi-Fi access points in campus networks. Algorithms for frequency assignment and power control designed for Femtocell networks may be applied to Wi-Fi Direct. However, the main differences between Femtocells and Wi-Fi Direct reside in the medium access control mechanism and the orthogonality of the wireless channels. The access to radio medium in Wi-Fi networks is based on random access while cellular networks rely on radio resource allocation. While the frequencies in Femtocell networks are orthogonal, the channels in Wi-Fi networks are almost partially overlapping. These differences make the interference management in Wi-Fi Direct more difficult and the algorithms designed for Femtocells less applicable to the Wi-Fi Direct environment.

By dividing the whole network into smaller groups and reducing the transmission power within the groups, our approach can reduce the overall interference in the network and increase the network performances. However, the transmission power of the GO devices is higher than the rest of the devices. Residual energy state can be included as a parameter in the GO selection algorithm run by the SDN controller to select the GO nodes in addition to the position-based clustering method. It is possible to change the GO role to another device at a reasonable frequency to balance the battery lifetime among devices in the network.

### 3.4 Optimized number of clusters

The K-means algorithm can only be applied when the number of clusters has been determined. It is necessary for the SDN controller to determine the number of clusters  $K$  in the network before applying the K-means algorithm.

While a complete, exact determination of an optimal value of  $K$  is beyond the scope of this paper, we propose as a rule of thumb to choose  $K$  to be of the same order as  $\sqrt{N}$ . This choice is motivated by considering and solving the following simplified performance model:

- We suppose to begin with that all devices share the same frequency (the more realistic case of multiple frequencies will be deduced afterwards).
- We assume that the communications can be described by a disk graph: each device  $p$  has a given communication range  $\rho_p$ .  $p$  can send information to any device in the disk of center  $p$  and radius  $\rho_p$ , but will also create interferences with all other devices inside that disk.  $p$  has no impact outside its communication range.
- We make the assumption that the overall performance of the system is non-increasing with the number of interferences. In other words, to maximize the performance, we need to minimize the total number of interferences.

Each of the  $K$  GO needs to communicate with the access point. As the distance between a GO and the access point is of the same order as the dimensions of the room, we expect that a typical device will receive interferences from a non-negligible fraction of the GOs, i.e.  $\alpha K$  for some constant  $K$ .

Then, each device needs to communicate with its GO, so we can expect that the transmission range of a typical non-GO device will have the same order of magnitude as the radius of a cluster. Assuming that the devices are uniformly distributed in the room, this means that the number of devices within the interference range of a typical non-GO device should be of the same order as the size of a typical cluster, i.e.  $\beta \frac{N}{K}$  for some constant  $\beta > 0$ .

In the end, the typical number of interferences perceived by a device should be like  $\alpha K + \beta \frac{N}{K}$ , which is minimized for  $K = \sqrt{\frac{\beta}{\alpha} N}$ .

Intuitively,  $\alpha$  and  $\beta$  should be similar, as they both measure the same type of geometrical ratio, only at different scales (room and cluster respectively). This leads to  $K \approx \sqrt{N}$ .

Note that the same type of reasoning should apply if one uses multiple frequencies: both  $\alpha$  and  $\beta$  would be roughly divided by the number of orthogonal frequencies available<sup>1</sup>, leaving the result unchanged. Similarly, the transmission modeling could be refined, for example by considering some AWGN channel instead of a simple disk range, but this should not change the main reason of the  $\sqrt{N}$  rule. That means that the interference comes:

- from one part from the communications by the  $K$  GOs to the AP,

<sup>1</sup> IEEE 802.11 has 11 frequency bands in the US, 13 in the UE; however, the number of *orthogonal* channels is about 3 in both cases.

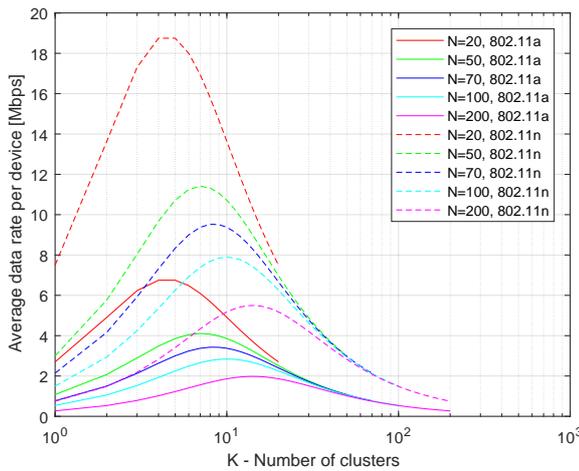
- from another part from the communications by devices to their GO, which impact a number of devices proportional to the typical size of a cluster,  $\frac{N}{K}$ .

Note that in a dynamic context, the value of  $N$  may fluctuate with time, changing the optimal number of clusters. As modifying the cluster architecture may be costly, it would be best to delay the modification with some hysteresis mechanism. For example, one could change the number of clusters only when the difference between current and actual  $K$  is greater than a given threshold.

## 4 Performance evaluation

### 4.1 Clustering performance evaluation

In this section, we evaluate the data rate that a client can receive in function of the number of clusters that we have in the network. Let's consider a network of  $N$  devices divided into  $K$  clusters. If we suppose that medium access control mechanism is fair to all the device sharing the same medium, each device will have the same probability to access the channel and share the channel capacity in an equitable manner. Figure 7 shows the average data rate that a non-GO device should receive following the assumptions presented in Section 3.4. All devices share the same frequency. The capacity of the channel is shared between the  $\frac{N-K}{K}$  devices within a cluster and the  $K$  GOs.



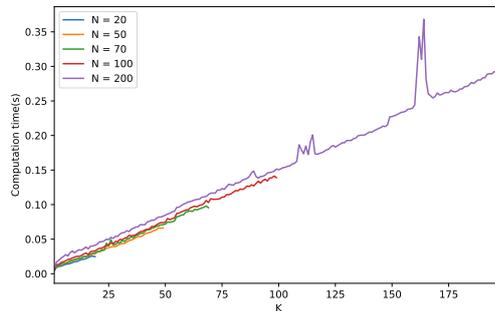
**Fig. 7** Data rate evaluation in 802.11g at 54 Mbps and in 802.11n at 150 Mbps

As we can see in Figure 7, for an 802.11g network working at 54 Mbps, the optimized average data rate can be obtained when the number of clusters  $K \approx \sqrt{N}$ . For a class room of 20 students, the optimized number of clusters is 4 or 5 offering a data rate of 6.75 Mbps per device. For an auditorium of 200

students, the optimized number of clusters is 10 or 11 with the offered data rate per device of 1.86 Mbps or 1.91 Mbps respectively. It is worth noting that these results on data rate are theoretical and do not take into account the overhead introduced by protocol headers at different layers and the reduction of data rate due to the Back-off mechanism in 802.11 medium access control protocol.

For recent 802.11 standards offering higher data rates such as 802.11n or 802.11ac, the optimized data rate that a user can obtain is also higher. As illustrated in Figure 7, in the case of 802.11n network working at 150 Mbps, each user in a classroom of 20 students can receive a data rate of 18.75 Mbps and each user in an auditorium of 200 students can receive a data rate of 5.32 Mbps. Standards supported by a device can also be used as a parameter in the GO selection algorithm run by the SDN controller. A device supporting among others a more recent 802.11 standard with higher data rate should be a better candidate for a GO node.

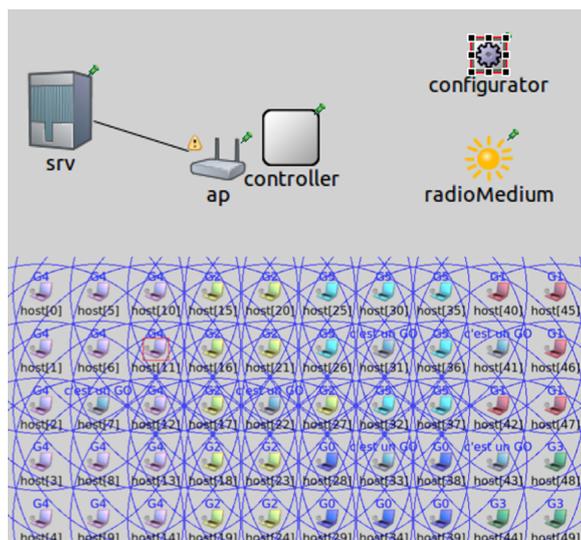
The clustering computational time depends on the number of the devices in the network  $N$  and the number of clusters  $K$ . Figure 8 presents the measurements of computational time of the  $K$ -means algorithm in function of number of clusters to be formed. We can see that the computation time of the  $K$ -means algorithm linearly increases with the number of cluster  $K$  and the size of the network  $N$ . However, for a configuration of 200 devices and 200 clusters, the computation time does not exceed 0.3 seconds which is acceptable for a campus network.



**Fig. 8** K-Means algorithm computation time measurement

#### 4.2 Simulation with OMNeT++

In order to evaluate the data transmission performances in a more realistic conditions, we have implemented Wi-Fi Direct in OMNeT++ [20]. The simulated network topology is presented in Figure 9.



**Fig. 9** Simulation topology

In this network, 50 tablets are arranged in a grid topology composed of five rows and ten columns. The horizontal and vertical distances between two adjacent devices are one meter which is approximately equivalent to the distance between two adjacent seats in an auditorium. The data rate of the Ethernet connection between the server and the access point is 100 Mbps. An SDN controller is used to control the Wi-Fi Direct clustering and data transmission in the network. The radio parameters configured in the simulation are presented in Table 1.

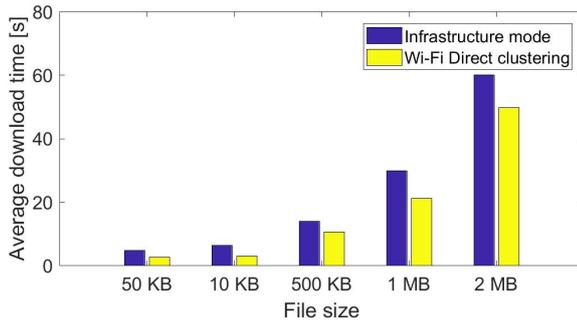
**Table 1** Simulation parameters

Parameter	Value
Operation mode	802.11g
Carrier frequency	2.4 GHz
Channel bandwidth	20 MHz
Radio transmission rate	48 Mbps
Receiver sensitivity	-85 dBm
SINR threshold	4 dB
Background noise power	-90 dBm
GO transmission power	1.47 dBm
Client transmission power	-23 dBm

To evaluate the performances, we have defined two test scenarios. In the first scenario, all the 50 nodes download a file directly from the access point using the infrastructure mode. In the second scenario, the 50 nodes are clus-

tered into  $K$  Wi-Fi Direct groups. Transmission power of non-GO devices is limited to only reach their GO. Data are transmitted between a device within a cluster and the access point via the GO of the cluster.

Figure 10 shows the average download time in function of the file size when the 50 nodes are clustered into 4 Wi-Fi Direct groups. The file size has been varied between 50 KB, which can correspond to an excel file containing the time table of the class, and 2 MB which can be the size of a heavy PDF file of 50 slides for a lecture. These simulation results show that Wi-Fi Direct clustering can reduce up to 34% the average download time compared to the Wi-Fi infrastructure mode.



**Fig. 10** Average download time at  $K = 4$

To understand the the origin of performance gain in terms of download time, we consider the relationship between the packet error rate and the download time. Figure 11 shows the download time of a 50 KB file and the packet error rates in three cases of clustering,  $K = 1$  (i.e. the infrastructure mode),  $K = 4$  and  $K = 8$ . The packet error rates explain the gain in download time that we have obtained. When all user devices use a high transmission power to communicate with the access point to download the file simultaneously, the interference level is high leading to a high packet error rate of 21% and a high average download time of 4.68 seconds. When using Wi-Fi Direct clustering and reducing the transmission power of the non-GO devices, the overall interference in the network is also reduced, resulting in much lower packet error rates, 14.1% in the case of 4 clusters and 13.08 % in the case of 8 clusters.

To analyze the impact of the number of clusters on the overall performance, we measure the download time, the group formation time and the packet error rate in the cases of 4, 6, 8 and 10 clusters. Figure 12 shows that the configuration of 8 clusters can offer better performances in terms of download time and packet error rate than the configuration of 4 clusters, 6 clusters or 10 clusters. This observation also confirms our calculation of the optimized number of clusters should be  $K \approx \sqrt{N}$ .

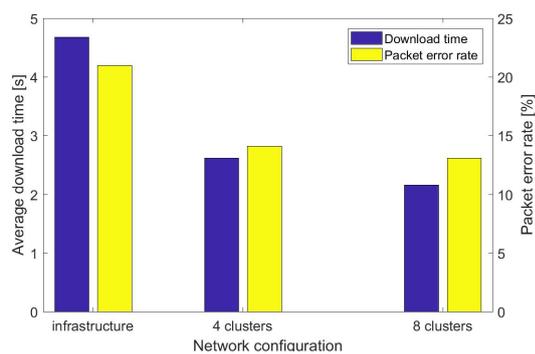


Fig. 11 Impact of Clustering on packet error rate and download time

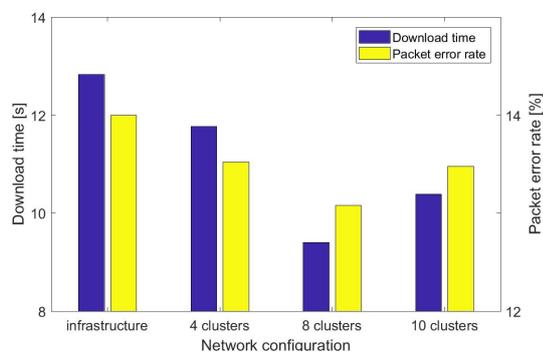


Fig. 12 Optimized number of clusters

## 5 Conclusion

In this paper, we have proposed the use of SDN to control the Wi-Fi Direct clustering in campus networks. This approach is particularly useful in dense wireless networks such as auditorium and class rooms. When dividing a dense wireless networks into smaller Wi-Fi Direct groups, we can reduce the transmission power of the clients in each group hence reduce the overall interference in the network. We also propose a simple method to calculate the optimized number of clusters. The simulation results have shown that our proposed architecture gets better performances in terms of download time and packet error rate compared to the Wi-Fi infrastructure mode. In the future work, we will consider data caching, multicast transmission and develop a prototype with Android tablets to be used in university campus.

## References

1. D. Camps-Mur, A. Garcia-Saavedra and P. Serrano, *Device-to-Device Communications with Wi-Fi Direct: Overview and Experimentation*, IEEE Wireless Communications Magazine, Vol. 20, No. 3, pp. 96-104, June 2013.
2. Magency, <http://www.magency.me>
3. Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021*, White paper, March 2017.
4. M. Maity, B. Raman and M. Vutukuru, *TCP Download Performance in Dense Wi-Fi Scenarios: Analysis and Solution*, IEEE Transactions on Mobile Computing, Vol. 16, No. 1, pp. 213-227, January 2017.
5. A. Baid and D. Raychaudhuri, *Understanding Channel Selection Dynamics in Dense Wi-Fi Networks*, IEEE Communications Magazine, Vol. 53, No. 1, pp. 110-117, January 2015.
6. Green Communications, *Green PI: Device2Device wearable networks*, White paper, URL: <http://www.green-communications.fr>, 2016
7. Wi-Fi Alliance, *Wi-Fi Simple Configuration Technical Specification, Version 2.0.5*, August 2014.
8. A. Pyattaev, K. Johnsson, S. Andreev and Y. Koucheryavy, *3GPP LTE traffic offloading onto Wi-Fi Direct*, IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Shanghai, China, 2013.
9. R. Rajadurai, K. S. Gopalan, M. Patil, and S. Chitturi, *Enhanced Interworking of LTE and Wi-Fi Direct for Public Safety*, IEEE Communications Magazine, Vol. 54, No. 4, pp. 4046, April 2016.
10. K.Liu, W. Shen, B. Yin, X. Cao, L. X. Cai and Y. Cheng, *Development of Mobile Ad-hoc Networks over Wi-Fi Direct with Off-the-Shelf Android Phones*, Proceedings of IEEE ICC, Kuala Lumpur, Malaysia, May 2016.
11. L. Amaral, R. Sofia, P. Mendes and W. Moreira *Oi! - Opportunistic Data Transmission based on Wi-Fi Direct*, Proceedings of IEEE INFOCOM, San Francisco, April 2016.
12. N. McKeown, T. Anderson, H. Balakrishman, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, *OpenFlow: Enabling Innovation in Campus Networks*, ACM SIGCOMM Computer Communication Review, Vol. 38, N.2, pp. 69-74, April 2008.
13. R. Riggio, M. K. Marina and T. Rasheed, *Interference Management in Software-Defined Mobile Networks*, Proceedings of IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, Canada, May 2015.
14. O. Stiti, O. Braham and G. Pujolle, *Virtual OpenFlow-Based SDN Wi-Fi Access Point*, Proceedings of IEEE Global Information Infrastructure and Networking Symposium (GIIS), Guadalajara, Mexico, October 2015.
15. M. Seyedbrahimi, F. Bouhafs, A. Raschella, M. Mackay and Q. Shi, *SDN-Based Channel Assignment Algorithm for Interference Management in Dense Wi-Fi Networks*, Proceedings of European Conference on Networks and Communications (EuCNC), Athens, Greece, June 2016.
16. I. T. Haque and N. Abu-Ghazaleh, *Wireless Software Defined Networking: A Survey and Taxonomy*, IEEE Communications Surveys and Tutorials, Vol.18, No. 4, Fourth Quarter 2016.
17. I. Cabria and I. Gondra, *Potential-K-Means for Load Balancing and Cost Minimization in Mobile Recycling Network*, IEEE Systems Journal, Vol. 11, No. 1, pp. 242-249, March 2017.
18. D. Brelaz, *New Methods to Color the Vertices of a Graph*, Communications of the ACM, Vol. 22, No. 4, pp. 251-256, April 1979.
19. N. Saquib, E. Hossain, L.B. Le, D.I. Kim, *Interference Management in OFDMA Femtocell Networks: Issues and Approaches*, IEEE Wireless Communications Magazine, Vol. 19, No. 3, pp. 86-95, June 2012.
20. S. Iskounen, T.M.T Nguyen and S. Monnet, *Wi-Fi Direct Simulation for INET in OMNeT++*, OMNeT++ Community Summit, Brno, Czech Republic, September 2016.