Duan *et al. EURASIP Journal on Wireless Communications and Networking* (2015) 2015:181
DOI 10.1186/s13638-015-0405-3

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Software-defined networking-based resource management: data offloading with load balancing in 5G HetNet

Xiaoyu Duan, Auon Muhammad Akhtar and Xianbin Wang*

Abstract

The explosive growth of mobile data traffic and the shortage of the available spectral resources have created new challenges for future cellular networks. In particular, resource management in heterogeneous network environment has become a critical issue. In this paper, we propose software-defined networking (SDN)-based resource management algorithms for future cellular network. Specifically, in this work, we have a threefold objective: i) alleviate spectrum shortage concerns by efficiently offloading traffic over the Wi-Fi network, ii) address network congestion by optimally balancing loads across multiple cells and iii) achieve the aforementioned objectives while taking network conditions and the end user quality-of-service (QoS) requirements into consideration. To this end, we present SDN-based partial data offloading and load balancing algorithms. The proposed algorithms exploit an SDN controller's global view of the network and take optimized resource allocation decisions. We analyze the performance of the proposed algorithms under realistic network model. Moreover, we also present an analytical framework to quantify the delay incurred due to the SDN-based data processing and forwarding. Our analysis and system-level simulations show that the proposed load balancing algorithm significantly improves the equilibrium extent and network stability as compared to the baseline algorithms. On the other hand, the proposed partial data offloading algorithm is shown to satisfy end user's quality-of-service while saving a significant amount of cellular resources.

Keywords: HetNet; Load balancing; Quality of service; Resource management; Software-defined networking; Wi-Fi data offloading

1 Introduction

Over the last decade, cellular networks have witnessed an unprecedented growth in mobile data traffic, mainly due to the explosive development of smart and media-rich mobile devices. These smart devices enable ubiquitous mobile internet access, traffic-intensive social applications, and cloud-based services [1]. According to Cisco's networking visual index report [2], the data traffic is expected to grow at a compound annual rate of 57 % by 2019, a tenfold increase over 2014. Current cellular networks are not capable of sustaining such high traffic volumes. Consequently, fifth generation (5G) is being touted as the next-generation cellular standard. The 5G networks are envisioned to have a densified heterogeneous network (HetNet) architecture, combining multiple radio access technologies (multi-RATs) into a single holistic network [3]. In order to optimize resource utilization and to support the predicted traffic volumes, 5G is expected to utilize a multitude of technologies including device-to-device (D2D) communications, data offloading, load balancing, spectrum sharing, etc.

Data offloading is seen as an enabling solution to address the spectrum shortage concerns. Offloading refers to the use of complementary networks for delivering the data originally targeted for the cellular networks [4]. As shown in [5], 55 % of data usage occurs at home and 26 % occurs in office or hotspots. Thus, the already deployed Wi-Fi access points (APs) become a natural solution for the operators to execute data offloading. In addition to spectrum shortage, network congestion is another

*Correspondence: xianbin.wang@uwo.ca
Department of Electrical and Computer Engineering, University of Western Ontario, London, ON, Canada

critical challenge for the cellular networks. Congestion results from the extremely large number of bandwidth-hungry smart devices. To alleviate network congestion, load balancing, which balances data distribution across multiple resources, e.g., multiple macrocells, presents an attractive solution [6]. Thus, offloading, along with load balancing, are used to address spectrum shortage and network congestion issues and they optimize the resource utilization in cellular networks.

With the built-in Wi-Fi card in the smartphones, and the ability to shift data traffic from the expensive licensed bands to the free unlicensed bands (2.4 and 5 GHz), Wi-Fi presents an attractive offloading technology for the operators. Therefore, many works have been done on Wi-Fi-based offloading. For example, the algorithm presented in [7] predicts the future Wi-Fi connectivity and delays suitable data transfers until a Wi-Fi network becomes available. Sou et al. in [8] propose a more flexible Wi-Fi offloading model, by introducing mobile Internet protocol (IP) flow into the core network. The authors in [9, 10] present a distributed offloading solution based on non-cooperative game theory, where macro cellular base stations (BSs) and third party APs try to achieve the highest utility for offloading traffic. In load-balancing-related research, traditional schemes usually use the received signal strength or resource utilization ratio to make load balancing decisions, like the basic distributed mobility load balancing algorithm introduced in [6]. Haydar et al. also study access selection between heterogeneous networks by considering quality-of-service (QoS) and present an algorithm which improves load balancing performance in [11].

All of the aforementioned works show significant performance improvements in terms of resource management, however, technical challenges still exist, especially when the envisioned 5G HetNet architecture is taken into consideration. Firstly, the uncoordinated Wi-Fi cells will be deployed overlay to the heterogeneous cellular cells [12, 13], resultantly, resource management will be challenging in this two-tier architecture. Secondly, offloaded data will be routed directly to Internet through the Wi-Fi backbone, which is not under control of the wireless operators since the Wi-Fi networks are usually owned by third parties. Consequently, the operators are unwilling to lose control over the valuable subscriber information [1], not to mention the security risks introduced due to the loosely controlled nature of the Wi-Fi networks. Moreover, existing network protocols have been designed for current levels of network density, and thus, they will become the performance bottlenecks in case of the highly densified small cell deployments in 5G. For example, existing load balancing algorithms are mostly distributed, which causes ping-pong handovers due to the lack of global information [14]. There will also be unnecessary frequent mobility management messages exchanged in dense cells. Therefore, it comes as no surprise that operators are seeking to introduce intelligence while exerting greater controls in 5G HetNet resource management [15].

In this paper, we consider software-defined networking (SDN) [16], a programable network structure, as an enabling solution to apply intelligence and control in 5G HetNets [15]. The idea of programmable networks has been around for many years. Nevertheless, the advent of the Openflow interface has given a new life to SDN [17]. OpenFlow was first introduced in [18], where the authors provide a uniform interface for researchers to program flow entries and to run experiments on Ethernet switches, without having any knowledge about the internal workings of the switch. In the context of wireless communications, the authors in [19] introduce OpenRoads—an open SDN platform which improves mobility management in HetNets. Related work in [20] further provides an SDN approach for handover management in heterogeneous networks and the real-time test bed shows significant performance improvement in the QoS of the real-time videos.

In this paper, we propose SDN-based partial data offloading and load balancing algorithms to alleviate spectrum shortage concerns and to address the network congestion issues. The proposed algorithms exploit an SDN controller's global view of the network to achieve the aforementioned objectives, while taking network conditions and the end user QoS requirements into consideration. With an overall view of the network, the SDN controller has visibility over the offloaded data and can deliver subscribed content from vendors even after offloading, which is important for the operators [4]. Moreover, SDN provides an open and easily reconfigurable interface, which will enable the operators to efficiently modify and update/upgrade network policies. Our contributions also include quantifying the delay incurred due to the SDN-based data processing and forwarding and analyzing the performance improvements of the proposed algorithms under realistic network model. Results show that SDN-based data offloading algorithm decreases threshold miss probability and saves significant amount of cellular resources simultaeously. It is also shown that SDN-based load balancing achieves better resource allocation with higher network throughput and smaller number of handovers, as compared to the baseline algorithms.

The remainder of this paper is organized as follows: Section 2 outlines the network model considered in this work. SDN-based data offloading and load balancing schemes are presented in Section 3. The performance of SDN and the proposed algorithms is analyzed in Section 4. Simulation results are shown in Section 5 while the conclusions are drawn in Section 6.

2 Network model

We consider a HetNet environment consisting of cellular BSs and Wi-Fi APs, as shown in Fig. 1. The BSs communicate over the licensed band while Wi-Fi APs communicate over the unlicensed band. The Wi-Fi APs are located randomly within each cell. OpenFlow protocol is implemented onto BSs, APs, and switches in order to enable the SDN controller to easily control these network components via OpenFlow secure channel. It is assumed that all of the BSs and the Wi-Fi APs are connected to the OpenFlow switches, as depicted in Fig. 1. For the cellular network, the switches are co-located with the BSs, i.e., each BS has its own OpenFlow-enabled switch. While the switches of the macrocells are connected to the core network, the femtocell switches are connected directly with the Internet. On the other hand, for the Wi-Fi APs, the switches are located within the ISP infrastructure and each switch is connected to multiple Wi-Fi APs. All of the switches are controlled by a centralized SDN controller. Given the fact that the controller is only a program running on a server, it can be placed anywhere within the network, even in a remote data center [21].

Since the SDN controller is a fundamental component of the proposed architecture, it is imperative to discuss the SDN framework in detail. The controller application consists of three core modules, namely, authentication and charging (AC) module, offload manager (OM), and the load balancing (LB) module. The AC includes the identity management and charging record generation functionalities, which are used to enforce admission control and subscription-based charging, respectively. Once authenticated by the AC, a mobile user can access all of the available network resources. These resources are either owned by the network operator or they are leased from the available Wi-Fi networks. Next, for the OM module, the service-level rules describe the traffic features and characteristics which are required by the offload manager, e.g., related IP addresses, port numbers, bit rates, and delay sensitivity. These features and characteristics are based on the traffic flow template (TFT) filter [22] and the related QoS descriptions. Finally, the LB module includes the load measurement and mobility management functionalities, which collect cell load ratio and execute the load balancing algorithms. The complete SDN framework is presented in Fig. 2, where it can be seen that the SDN applications (AC, OM, and LB) utilize the global view of the APs to improve network management.

Note that the following proposed algorithms are based on a pre-condition that the cellular operator owns the Wi-Fi APs or there are mutual agreements between them. In this way, Wi-Fi resources are available to release cellular network burden, and all the cell loads are collected and monitored easily in load balancing procedure.

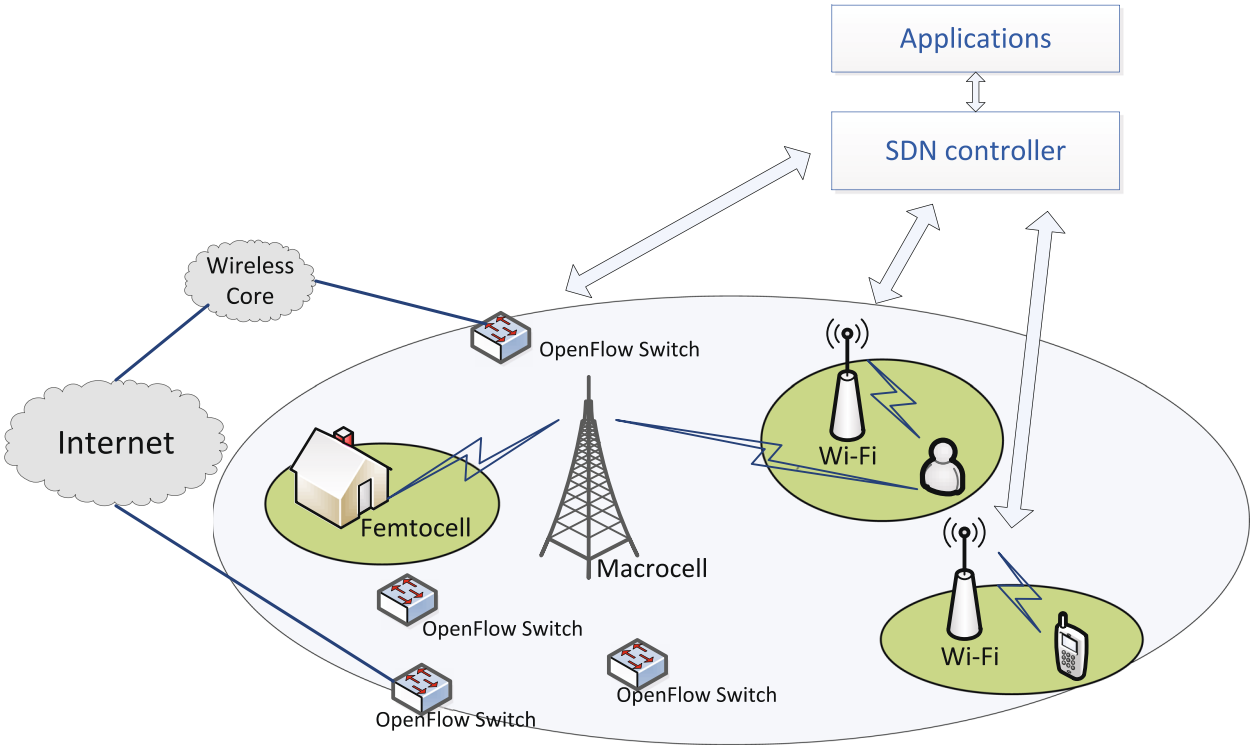

Fig. 1 SDN-based wireless heterogeneous network structure with control plane design

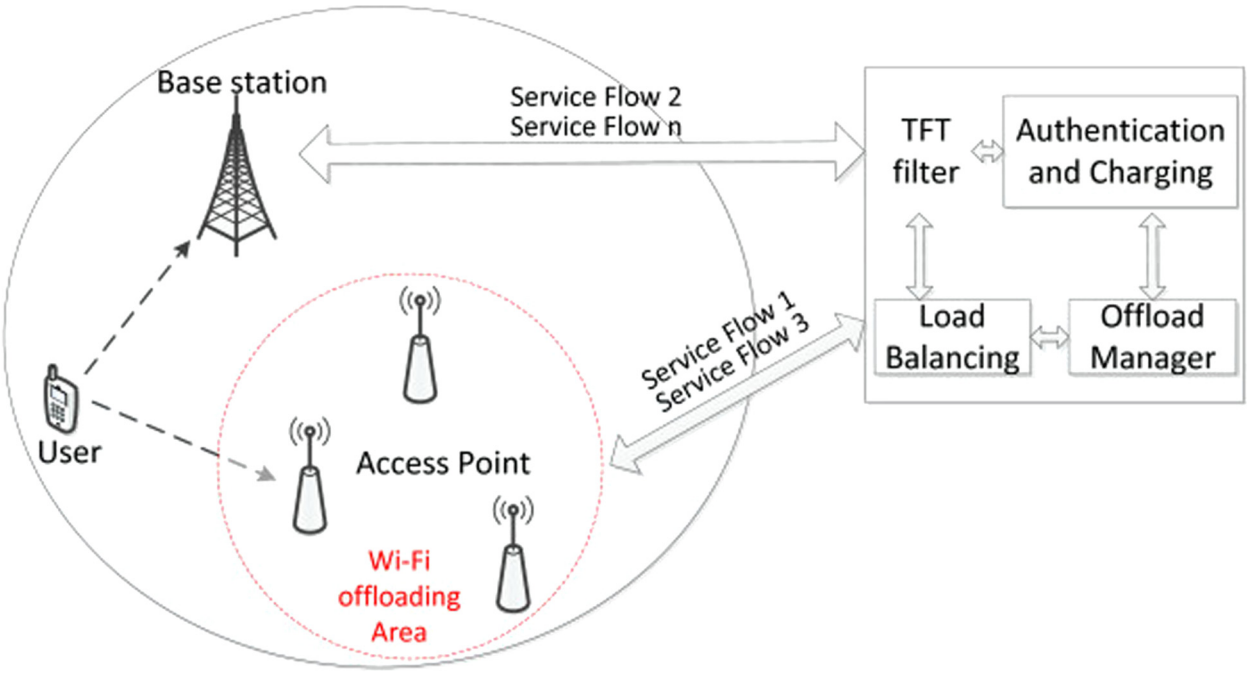

Fig. 2 SDN-based offloading scenario and module framework

3 SDN-based resource management

In this section, we present SDN-based resource management algorithms for the heterogeneous network shown in Fig. 1. To this end, we attempt to alleviate spectrum shortage and network congestion in the cellular network by using SDN-enabled mobile data offloading and load balancing. More specifically, to address the shortage of spectrum, data from the cellular network is offloaded onto a Wi-Fi network whenever the cellular users move within the Wi-Fi range. On the other hand, in order to deal with network congestion, load balancing is utilized to evenly distribute traffic across multiple cells. For both these cases, all of the decision-making is accomplished by the centralized SDN controller. As mentioned previously, the introduction of SDN facilitates efficient coordination between the cellular and Wi-Fi networks. Moreover, the performance of load balancing can also be improved significantly by exploiting the controller's view of the whole network. In the next subsection, we present an SDN-based data offloading algorithm.

3.1 SDN-based partial data offloading algorithm

In this subsection, we present an SDN-based partial data offloading algorithm. Here, partial offloading refers to the fact that only part of the user data is offloaded onto the Wi-Fi network, while the remaining traffic is transferred across the cellular network. To elaborate further, if the Wi-Fi network is unable to meet the requirements of the delay and loss sensitive flows, OM only offloads a limited amount of data. As we show later through analysis and simulations, the proposed algorithm improves application performance by taking various service requirements into consideration.

The proposed algorithm proceeds as follows: when a mobile user moves within the Wi-Fi coverage area and starts data transmission, the data will be queued in OpenFlow switch and awaits further processing. The OM then executes a delay-threshold-based selection algorithm, which selects the appropriate traffic flows for partial offloading. Due to the fact that Wi-Fi provides high data rates without QoS guarantees in the unlicensed band, the choice of offloading traffic should be discreet. Our goal here is to reduce cellular usage by leveraging Wi-Fi connectivity when available, but to do so without affecting application performance [7]. The algorithm used to select the appropriate flows for offloading works are explained below:

Step 1) The OM collects the traffic flow requirements of all the users within the Wi-Fi area and calculates the resources required by each existing user (notice that with SDN, the operator can easily incorporate various functionalities in the OM to collect required information from users). Assuming each user has a traffic demand u_i and a link rate r_i^w to the Wi-Fi AP, every Wi-Fi user's resource demand is computed as $\theta_i^w = u_i/r_i^w$ [23]. The available Wi-Fi resources for offloading can then be estimated accordingly.

Step 2) The OM calculates the amount of data that can be transferred within the delay tolerance threshold over the Wi-Fi network. If Wi-Fi cannot transfer all of the data before the deadline, partial offloading is executed. Otherwise, all of the data is offloaded onto the Wi-Fi network. Next, we calculate the application specific delay tolerance threshold, T_s, and the amount of data, V_s, that can be transferred within T_s.

3.1.1 Calculation of T_s

OM estimates user mobility direction and predict future paths [21]. Assume that a mobile user enters and leaves a Wi-Fi offloading area at times t_{in} and t_{out}, respectively, as shown in Fig. 3. Consequently, the Wi-Fi connection time is $t_c = t_{\text{out}} - t_{\text{in}}$. Figure 3 also shows the residual time of t_c, i.e., t_r, which is the duration of data transfer from t. In this scenario, the delay tolerance threshold T_s is equal to the difference between the application specific deadline T_d, which is obtained from the TFT filter in the network model, and the waiting time during SDN processing, denoted by D. The calculation of D will be explained later in the next section.

3.1.2 Calculation of V_s

Note that V_s is the amount of data in bytes that will be transmitted. Let b_1 and b_2, respectively, denote the bandwidths allocated by the cellular network (primary resource) and the Wi-Fi network (secondary resource). As shown in [8], the smaller value between t_r and T_s should be used to calculate the actual transmitted data volume. Therefore, V_s can be written as follows:

$$V_s = b_1 T_s + b_2 \min(t_r, T_s). \tag{1}$$

Based on the value of V_s, OM decides between partial and total offloading, as shown in Algorithm 1.

Algorithm 1 Partial data offloading

```
procedure PDO(T_s)
    T_s: delay threshold
    d: Total file size
    V_s2 = b_2 min(t_r, T_s): size in bytes to be transferred in Wi-Fi within T_s
    if d < V_s2 then
        offload all data to Wi-Fi and update d
    else
        if d_1 ≤ V_s2 and d = d_1 + d_2 then
            send d_1 on Wi-Fi, d_2 on cellular concurrently and update d
        end if
    end if
end procedure
```

Here, the percentage of b_1 and b_2 are specified in next section. The principle behind Algorithm 1 is elaborated below: if $d > V_{s2}$, it will be split into two parts. The first part, d_1, is the volume that can be transmitted through the Wi-Fi network before the deadline, while the remaining part, i.e., $d - d_1$, will be transmitted through the cellular network at the same time. Updating d enables the network to keep track of the volume that has been transmitted in

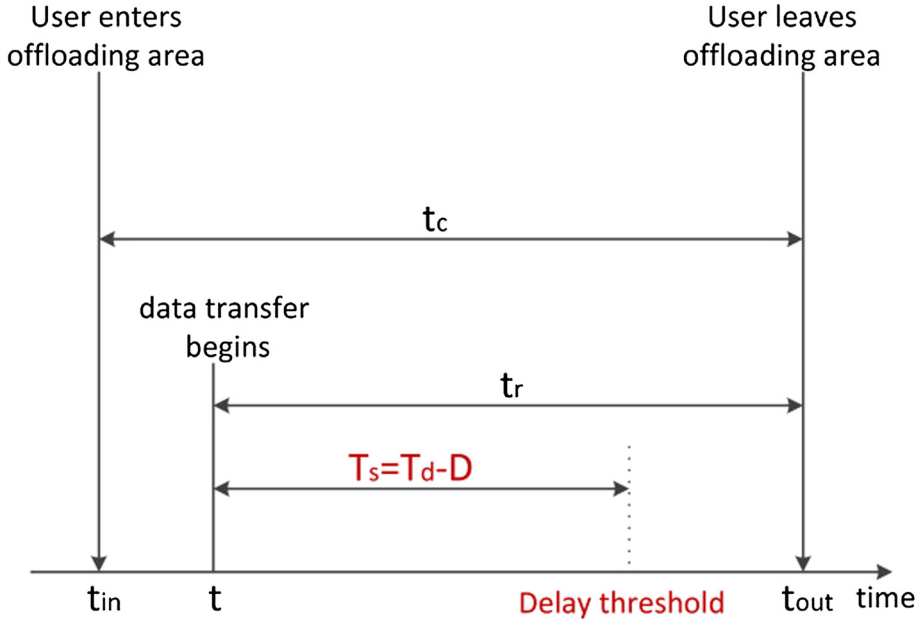

Fig. 3 SDN-based offloading time diagram model

the two paths. The procedure ends once all of d has been transmitted.

Note that the delay caused by packet re-assembling process in application layer will be negligible. The reason is that in practical implementation, the data is not distributed on per packet basis, instead, it is distributed in bursts between the cellular and Wi-Fi networks. With concurrent transmission, these bursts will arrive at the receiver in approximately the same time, thus minimizing the waiting period required for all the packets to arrive at the receiver device. Therefore, partial offloading method reduces the overall delay in the packet arrival process.

Step 3) OM updates the record of available Wi-Fi resources after offloading and the QoS that Wi-Fi can provide during partial data offloading (PDO) algorithm. This record is maintained for a specific amount of time, and it enables rapid decision making when a new traffic flow arrives.

3.2 SDN-based load balancing mechanism

Load balancing [6] has the ability to reduce network congestion over an area by distributing user traffic across neighboring APs or BSs. With load balancing, a proportionate share of wireless traffic can be guaranteed for better resource utilization. Since it is difficult to guarantee QoS with Wi-Fi, load balancing and vertical handovers of the edge users are seen as the enabling solutions to tackle network congestion. Both these strategies improve QoS by equally distributing the traffic load across the network [4].

Although load balancing has its advantages, yet, there are scenarios where it can prove to be counter-productive, e.g., in low-latency applications such as voice or live (unbuffered) video streaming. More specifically, for mobile users, a high number of handovers impacts the voice quality and it makes the streaming video jittery. With SDN-based load balancing, however, one can take advantage of the controller's view of the whole network. This makes it easier to find the optimal neighboring cell for load balancing with minimum handovers. SDN-based load balancing proceeds as follows:

Step 1) When a source cell i becomes overloaded, it sends a request to the controller, enquiring about the target neighboring cell for load balancing. A cell i is identified as overloaded when its load ratio LR_i exceeds a certain threshold. The load ratio LR_i is formulated as [6]:

$$\mathrm{LR}_i = w_1 * U_i + w_2 * R_i, \tag{2}$$

where U_i is the proportion of the number of user equipments (UEs) to cell i's maximum UE capacity, R_i is the ratio of the used resource blocks to the total resource blocks in a cell i, while w_1 and w_2 represent the weight parameters which provide the operators with the option to give higher preference to either U_i or R_i.

Step 2) The LB module calculates all of the neighboring cells' environment state ES_j, which is the average load state of each neighbor cell j's adjacent cells, excluding cell i (denoted as layer 2 in Fig. 4).

The environment state of the neighboring cell j can be written as

$$\mathrm{ES}_j = (\mathrm{LR}_{1j} + \mathrm{LR}_{2j} + ... + \mathrm{LR}_{nj})/n, \tag{3}$$

where n is the number of the neighbor cell j's layer 2 cells.

Step 3) The LB module computes the overall state OS_j of each neighboring cell of the source cell i and selects the target cell with smallest overall state value for load balancing.

The overall state of each neighboring cell j is a combination of its own load and its environment [24], which is computed as:

$$\mathrm{OS}_j = \mu \mathrm{LR}_j + (1 - \mu)\mathrm{ES}_j, \tag{4}$$

where μ is the influence degree of the neighbor's load and its environment. μ is decided by network operator according to network performance, i.e., whether neighbor cell's load or its environment is more important for network performance improvement in a specific type of network. It is important to mention that the proposed algorithm only decides the appropriate target cell. In order to maintain radio link quality after handover, the choice of edge users remains the same as that with the existing algorithms [6]. That is to say, only the source cell edge users, which are close to this selected target cell with good enough link quality will be handed over to the target cell.

In the traditional distributed load balancing scenarios, there is a possibility that multiple overloaded cells choose the same target cell for load balancing, causing a new overload situation or ping-pong handovers [6, 14]. Distributed solution thus takes more rounds to achieve an optimized state. However, SDN-based load balancing mechanism uses an overall network view when selecting the target cell, which decreases handover times and improves system performance.

Moreover, by maintaining a list of BSs, SDN-based load balancing encourages new clients to associate with the least loaded BS upon arrival. Then, after a certain amount of time, if there exists an overloaded cell, the controller uses load balancing mechanism to handover appropriate cell edge users. Along with partial data offloading algorithm, it is expected that SDN-based module framework will achieve an optimized overall system state.

4 Performance analysis

In this section, we analyze the performance of the resource management schemes introduced in the previous section. To this end, we begin by formulating the

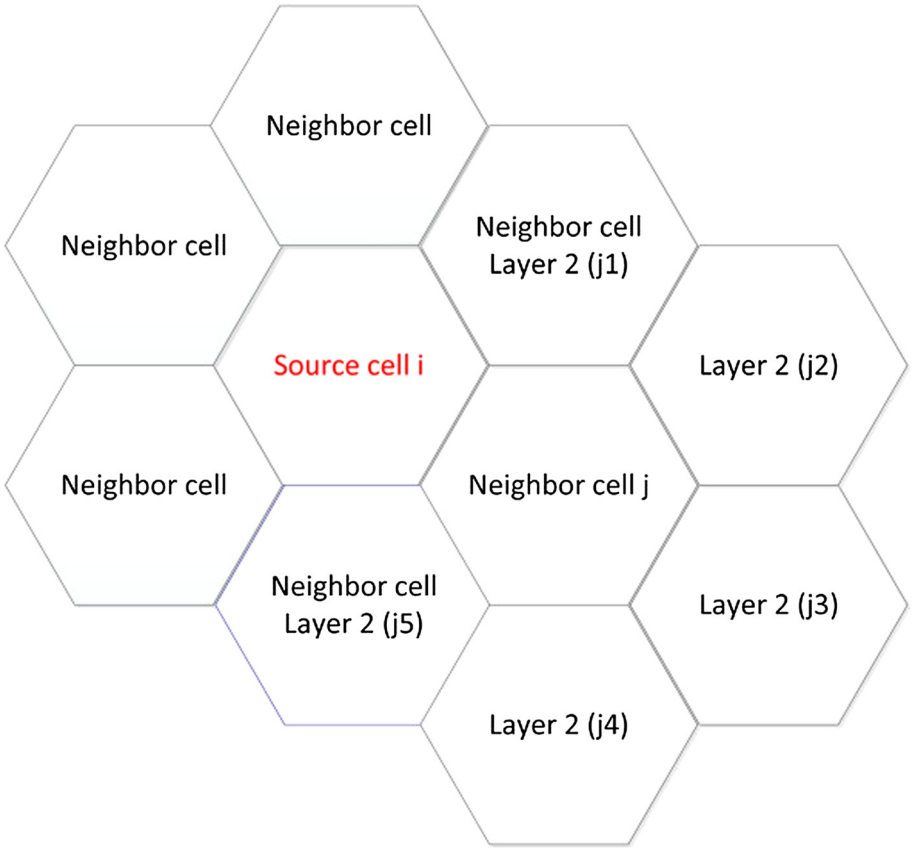

Fig. 4 SDN-based load balancing with network view

delay incurred due to the processing at the SDN-based controller and switches.

4.1 SDN network delay *D*

Generally, the average delay introduced by SDN depends on the state of the flow table within the switch, i.e., whether or not the switch's flow table contains a rule for the incoming traffic flow. Figure 5 shows the queuing model for the controller and the switch.

In Fig. 5, λ is the data arrival rate while μ_s and μ_c represent the processing rates at the switch and the controller, respectively. As shown in the figure, if the packet arriving at the switch is the first packet of a new data flow (i.e., no flow entry in switch matches its IP addresses obtained from TFT filter), the switch forwards this packet to the controller over the OpenFlow channel. The controller decides the optimal forwarding rule for this packet and returns it to the switch. Moreover, the controller also

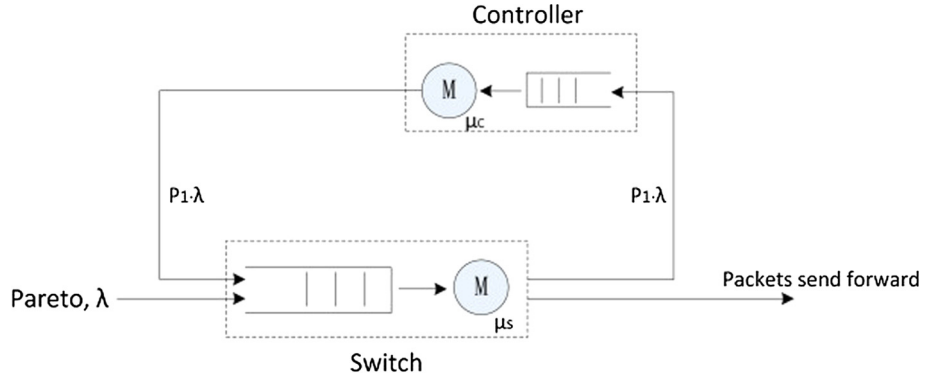

Fig. 5 A model for SDN switch and controller

pushes the corresponding forwarding rule (flow entry) into the flow table at the corresponding switches. Subsequent packets from the same flow are forwarded by switches directly based on the newly installed forwarding rule. It is worth mentioning here that the following analysis is based on the assumption that the incoming traffic is the transmission control protocol (TCP), i.e., a given source node initially transmits a single packet to initiate the TCP handshaking procedure and the actual data is transmitted once the TCP session is established. As a final remark, it should be noted that we did not go into other details about TCP, since SDN model is the focus of this research. Nonetheless, the TCP congestion window applies to the performance with and without SDN, so its impact will remain the same.

Assuming P_1 represents the probability that there is no flow entry in the OpenFlow switch for the incoming packet, the average packet delay, D, can be written as

$$D = D_1 \times P_1 + D_2 \times (1 - P_1), \tag{5}$$

where D_1 is the delay incurred if the switch has to forward the packet to the controller while D_2 represents the delay which occurs when the switch forwards the data directly, i.e., a forwarding rule for the packet already exists. It has been shown in [25] that in a normal productive network carrying end-user traffic, a switch observes new flows with a probability of 0.04, hence, we use this value for the probability P_1.

The delay D_1 can be written as

$$D_1 = T_C + 2T_{\text{PROP}} + 2T_{\text{sw}}, \tag{6}$$

where T_{sw} and T_C, respectively, represent the delays at the switch and the controller, including the queuing and the processing delays. Moreover, T_{PROP} denotes the propagation delay between the controller and the switch. Note that T_{sw} in (6) is multiplied by 2 since the packet has to make two passes through the switch. In the first pass, it is forwarded to the controller for further processing, and in the second pass, it is forwarded along the data path once a forwarding rule has been established. On the other hand, $2T_{\text{PROP}}$ accounts for the propagation delays when the packet is sent to and from the controller. The delay D_2 is equal to T_{sw} only, i.e.,

$$D_2 = T_{\text{sw}}. \tag{7}$$

Next, we derive the delays T_{sw} and T_C. Here, it is imperative to mention that in order to simplify the analysis, it is assumed that the packets returning from the controller have no impact on the queuing delay at the switch. This assumption is reasonable for two reasons: firstly, the probability P_1 is relatively small. Secondly, the size of the data returning from the controller is also very small as compared to the size of the data buffered in the queue, i.e., controller returns only a single packet at a given time instance while the traffic at switch arrives in the form of multiple flows containing a large number of data packets. The validity of this assumption is verified later through simulations in Section 5. At this stage, it is worth mentioning that although these returned packets have little influence on the queuing delay for other arriving flows, nevertheless, the performance of the flows that do have to go through the controller will still be impacted. That is why the controller processing delay T_c cannot be ignored.

4.1.1 T_{sw}

In order to derive T_{sw}, the inter arrival process at the switch is modeled with Pareto distribution [26–28], with shape parameter α and scale parameter k [29]. Furthermore, the switch processing times are modeled as Poisson distributed with rate parameter μ_c. With the aforementioned assumption and the Pareto/M/1 queuing model, the switch waiting time T_{sw} is given as [28]

$$T_{\text{sw}} = \frac{1}{\mu_s(1 - z)}, \tag{8}$$

where z is the root of the Laplace transform of the inter-arrival time distribution function. The root z is given by (9) shown on top of the next page[26]. In (9), $\Gamma^*(-\alpha + 3, \mu(1-z)) = \Gamma(-\alpha+3)P(-\alpha+3, \mu(1-z))$, where $P(-\alpha+3, \mu(1 - z))$ is the incomplete gamma function.

$$\begin{aligned} z = \alpha\,[\mu(1-z)]^{\alpha} \Bigg\{ & \frac{1}{\alpha e^{\mu(1-z)}[\mu(1-z)]^{\alpha}} \\ & - \frac{1}{\alpha(\alpha-1)e^{\mu(1-z)}\,[\mu(1-z)]^{\alpha-1}} \\ & + \frac{1}{\alpha(\alpha-1)(\alpha-2)e^{\mu(1-z)}\,[\mu(1-z)]^{\alpha-2}} \\ & - \frac{1}{\alpha(\alpha-1)(\alpha-2)} \left[\Gamma(-\alpha+3) - \Gamma^*(-\alpha+3, \mu(1-z))\right] \Bigg\} \end{aligned} \tag{9}$$

4.1.2 T_C

The arrival rate at the controller is Poisson distributed since the departure rate at the switch is Poisson [30]. Therefore, T_C can be calculated using the waiting time equation for $M/M/1$ queuing as

$$T_C = \frac{1}{\mu_c(1 - \rho_c)}. \tag{10}$$

In the above equation, ρ_c represents the controller utilization and it is equal to $P_1.\lambda/\mu_c$ (see Fig. 5).

Using (6)–(10) in (5), one can find the average packet delay with SDN-based data forwarding.

4.2 Performance analysis of SDN-based PDO algorithm

In [31], Li et al. prove that at the application level, the size of most of the web traffic, including multimedia files and Internet documents, is a combination of long-tailed distribution process and forms Pareto distribution. Therefore, this paper considers the data file d to be transmitted in the offloading session with Pareto distribution. Resultantly, the cumulative distribution function $F_d(X)$, which is the probability that d is smaller than some number X, is formulated as:

$$F_d(X) = 1 - (\frac{k}{X})^{\alpha} \; for X \geq k \tag{11}$$

In order to analyze the performance of the partial data offloading scheme, we use two key performance indicators: the first indicator is the delay threshold miss probability, P_{miss}, which is the probability that the Wi-Fi network is unable to meet the service latency requirements; the second indicator is amount of data d_1 that is offloaded from the cellular network. The probability P_{miss} is given as

$$P_{\text{miss}} = Pr[\, d > V_s] \,. \tag{12}$$

Using the cdf of d from (11) and the value of V_s from (1), the above equation can be written as [8]

$$\begin{aligned} P_{\text{miss}} = & \int_{t=0}^{T_s} [1 - F_d(b_1 T_s + b_2 t)]\, r_c(t) dt \\ & + [1 - F_d((b_1 + b_2) T_s)] \int_{t=T_s}^{\infty} r_c(t) dt, \end{aligned} \tag{13}$$

where $r_c(.)$ is the probability density function of the residual time t_r. From (13), it can be seen that the probability P_{miss} consists of two parts: the first part reflects the scenario where the time t that a user spends in the Wi-Fi connection area is smaller than the application deadline time T_s. In this situation, the transmission volume in the Wi-Fi network is $b_2 t$. On the other hand, the second part of the equation addresses the scenario where the duration of a user's stay in the Wi-Fi area is larger than the application deadline time. Accordingly, the transmission volume of Wi-Fi network in this situation is equal to $b_2 T_s$.

By setting b_1 in (13) equal to 0, one gets P_{miss} for the scenario when all of the traffic is offloaded onto the Wi-Fi network. Recall that t_r is the residual life of t_c. According to the distribution of the asymptotic residual life from the renewal theory [32], the probability density function $r_c(.)$ can be formulated as:

$$r_c(t_r) = \frac{1 - P(T < t_r)}{E[\, t_c]} = \frac{1 - F_c(t_r)}{E[\, t_c]}, \tag{14}$$

where $E[\, t_c]$ is the expected value of the connection time t_c while $F_c(.)$ is the distribution function of t_c.

Assuming that t_c follows Erlang distribution with parameters (n, λ_e), which is a widely used traffic model, the distribution function $F_c(.)$ of connection time t_c can be formulated as [32]

$$F_c(t_c) = \frac{\gamma(n, \lambda_e t_c)}{(n-1)!} = 1 - \sum_{m=0}^{n-1} \frac{1}{m!} e^{-\lambda_e t_c} (\lambda_e t_c)^m, \tag{15}$$

where $\gamma(.)$ is the lower incomplete gamma function and $E[\, t_c]$ is equal to n/λ_e. By using (14) and (15) in (13), one obtains P_{miss} as

$$\begin{aligned} P_{\text{miss}} = & \left(\frac{\lambda_e}{n}\right) \sum_{m=0}^{n-1} \int_{t=0}^{T_s} \left(\frac{k}{b_1 T_s + b_2 t}\right)^{\alpha} \frac{e^{-\lambda_e t} (\lambda_e t)^m}{m!} dt \\ & + \left(\frac{k}{(b_1 + b_2) T_s}\right)^{\alpha} \left\{ 1 - \left(\frac{1}{n}\right) \sum_{m=0}^{n-1} \left[1 - \sum_{j=0}^{m} \frac{e^{-\lambda_e T_s} (\lambda_e T_s)^j}{j!} \right] \right\}, \end{aligned} \tag{16}$$

where $T_s = T_d - D$. P_{miss} indicates the performance of partial offloading scheme as a function of the delay tolerance threshold T_s. One can find the improvement of proposed algorithm by comparing P_{miss} under various primary bandwidths b_1.

The second indicator, the amount of offloaded data, i.e., d_1, which is served by the Wi-Fi network can be expressed as:

$$d_1 = \begin{cases} \dfrac{b_2}{b_1 + b_2} d, \text{if } d > b_2 t_r \\ b_2 t_r, \text{otherwise}, \end{cases} \tag{17}$$

which means that if data volume is larger than the Wi-Fi capability, the data is transmitted concurrently in cellular and Wi-Fi networks. Otherwise, all the data is offloaded onto the Wi-Fi network. Eq. 17 can be re-formulated as

$$d_1 = \frac{b_2 d}{b_1 + b_2} Pr[\, d > b_2 t_r] + b_2 t_r Pr[\, d \leq b_2 t_r] \tag{18}$$

Using a similar procedure as that presented in [8], the above equation can be written as

$$d_1 = \frac{b_2 d}{b_1 + b_2} [\, 1 - R_c \left(\frac{d}{b_2}\right)] + b_2 \int_0^{d/b_2} t_r r_c(t_r) dt_r, \tag{19}$$

where $R_c(.)$ is the distribution function of t_r and $1 - R_c(\frac{d}{b_2})$ shows the probability that the value $\frac{d}{b_2}$ is larger than time value t_r. Using (14), one obtains $R_c(.)$ as

$$R_c(t_r) = \int_{t=0}^{t_r} r_c(t)dt = \left(\frac{1}{n}\right)\sum_{m=0}^{n-1}\left[1 - \sum_{j=0}^{m}\frac{e^{-\lambda_e t_r}(\lambda_e t_r)^j}{j!}\right] \quad (20)$$

Finally, by replacing t_r in the above equation with $\frac{d}{b_2}$ and using the resulting equation in (19), one gets the offloaded data volume d_1 as

$$d_1 = \frac{b_2 d}{b_1 + b_2}\left\{1 - \left(\frac{1}{n}\right)\sum_{m=0}^{n-1}\left[1 - \sum_{j=0}^{m}\frac{e^{-\lambda_e x}(\lambda_e x)^j}{j!}\right]\right\} + \frac{b_2}{\lambda_e n}\sum_{m=0}^{n-1}(m+1)\left[1 - \sum_{j=0}^{m+1}\frac{e^{-\lambda_e x}(\lambda_e x)^j}{j!}\right], \quad (21)$$

where $x = d/b_2$. In particular, $\frac{d_1}{E[d]}$ describes the proportion of the data that has been offloaded onto Wi-Fi network.

4.3 Performance analysis of SDN-based load balancing algorithm

The performance of SDN-based load balancing mechanism is measured by the number of handover times, the equilibrium extent of the network and the throughput of the network.

Equilibrium extent is defined as the degree of load balanced across the entire network [6] and it can be written as

$$\nabla(t) = \frac{(\sum_c \rho_c)^2}{|N|\sum_c(\rho_c)^2}, \quad (22)$$

where N is the number of cells and ρ_c is the load density of cell c. Obviously, the network resource is better utilized if the load is more evenly balanced across the network. In this paper, we use the overall throughput of the network, which is calculated from the signal-to-noise ratio (SNR) and related bit error rate (BER), to measure the performance of the network resource utilization.

5 Performance evaluation

In this section, we evaluate the performance of SDN and the proposed algorithms. To this end, we begin by evaluating the performance of SDN-enabled switches and controller.

5.1 Performance evaluation of SDN

In this section, the performance of the SDN framework, shown in Fig. 5, is evaluated in terms of network utilization and the incurred delay. For the simulation setup, the service times of the controller and switch was set to be 0.33 ms and 9.8 μs , respectively [30, 33]. To account for different types of data traffic, we used different values for the shape parameter α of the Pareto arrivals at the switch. More specifically, values of $\alpha = 1.5$ and $\alpha = 2.5$ were used. Moreover, we also evaluate the performance when the arrival process at the switch is Poisson distributed. All the simulation results were averaged to 10,000 number of iterations.

Figure 6 shows the SDN delay as a function of the network utilization ρ. Recall that in the queuing model introduced in Section 4.1, it was assumed that the packets returning from the controller to the switch cause negligible delay in the switch queue. To justify this assumption, in Fig. 6, we also plot the SDN delay which occurs when the effect of the packets returning from the controller is not ignored (model 2 in Fig. 6). From the figure, it can be seen that the simulation results with and without the aforementioned assumption are approximately the same and therefore, our assumption is justified. Moreover, it can also be seen that the theoretical and the simulation results also match very closely with each other, thus verifying the validity of the analysis in Section 4.1.

As mentioned previously, different α values represent different types of application traffic [1]. It can be seen that with decreasing α, the network delay increases under the same utilization rate, which shows that the smaller traffic flows cause greater burden on the SDN network due to the higher arrival rate. Traditional telecommunication voice traffic, shown by Poisson arrival, has the lowest delay since it does not participate in the SDN offloading procedure. Based on the above discussion, it can be concluded that SDN-based solution is more suitable for applications which are less sensitive to latency.

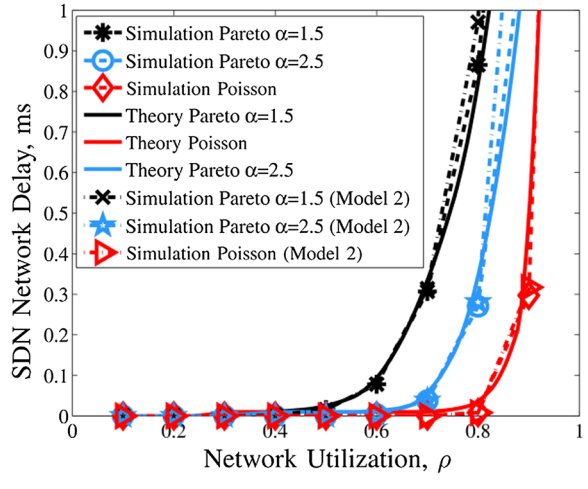

Fig. 6 SDN network delay versus network utilization using different SDN queuing model

5.2 Performance evaluation of SDN-based partial data offloading

In this section, we evaluate the performance of the proposed partial data offloading scheme. To this end, all the simulations were conducted in MATLAB and for each simulation round, a user randomly moved across the offloading area with residence time t_c, which followed an Erlang distribution [8]. A download traffic of size d was initiated at a given time instance t whenever a user resided within the offloading area. For demonstration purposes, it was assumed that the session data traffic d had a shape parameter of $\alpha = 1.5$ and scale parameter of $k = E[d](\alpha - 1)/\alpha$ [29]. The default Wi-Fi bandwidth, b_2, was set to 5 Mb/s.

Figure 7 shows the threshold miss probability P_{miss} as a function of the delay threshold, T_s, in seconds. For this simulation, it is assumed that the residence time t_c is 30 min while the average data size $E[d]$ is 10 MB. Partial offloading algorithm is used to allocate different amounts of primary bandwidth, b_1. When $b_1 = 0$, the data is offloaded completely onto the Wi-Fi network, and this reflects the existing full offloading algorithms [4]. Figure 7 shows that when the delay threshold is small, especially from 5 $\sim$ 20 s, partially offloading the data can effectively improve the threshold assurance. That is to say, if an application is sensitive to delay, the use of partial offloading improves the application performance by 20 $\sim$ 50 %. Finally, it can also be seen from the figure that if the application is not sensitive to delay, e.g., delay threshold is higher than 20 s, the offloading performance in terms of threshold miss probability becomes constant for a given primary bandwidth. Moreover, it can be seen that generally, partial offloading outperforms full offloading in terms of the threshold miss probability. This happens because the proposed partial offloading takes application delay requirement into consideration.

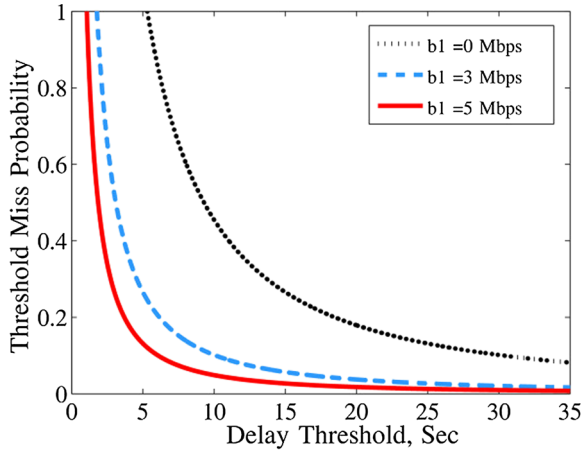

Fig. 7 Performance of partial offloading algorithm in threshold miss probability versus delay threshold

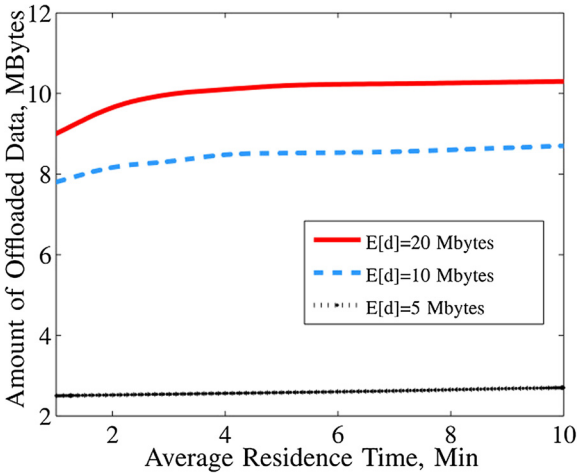

Fig. 8 Offloaded data volume by Wi-Fi data offloading in terms of different arriving traffic volumes

Figure 8 plots the amount of offloaded data as a function of the average residence time, t_r. It is obvious from the figure that when the average residence time is larger than 6 min, the amount offloaded data remains unchanged. The reason is that although the session offloads more data if the user stays longer in the offloading area, the maximum amount of offloaded data is limited by $E[d]$. Figure 8 shows that the amount of offloaded data increases significantly with increasing $E[d]$. This is reasonable because when a cellular network has higher loads, offloading will play a more significant role.

5.3 Performance evaluation of SDN-based load balancing

For the simulation, we consider a densified 37-cell (4-layered) hexagonal layout. All of the simulations are conducted in MATLAB. For the smaller cells, the inter site distance is set equal to 300 m [3] and the wrap around technique is used to avoid boundary effect [6]. We assume that every user requests a constant bit rate of 1 Mbps. Assuming a bandwidth of 10 MHz, the cell capacity in this case will be 15 UEs per cell. The main simulation parameters are given in Table 1 [6].

Table 1 Simulation parameters

Parameters	Values
System bandwidth	10 MHz
Cell layout	Hexagonal grid, 37 cell sites, with wrap-around technique
Cell radius	150 m
Pathloss	$-38.4 - 35.0 \log 10R$ (distance between UE and AP)
Shadowfading	Log-normal with standard deviation 8 dB
Antenna gain	−7 dB
eNB Tx power	46 dBm
Traffic model	CBR 1 Mbps full buffer traffic

Moreover, we artificially create heavy load concentration in the 37 densified cells to show the performance of load balancing algorithms. At the beginning, a total number of 365 UEs are uniformly dropped according to the setting: 7 cells with 20 UEs (overload), 15 cells with 10 UEs, and 15 cells with 5 UEs (low load). Each UE engages a random walk in the areas and changes direction every 2.5 s [24]. We assume half of the UEs would not move out of their original cell to ensure that the heavy load concentration will not be broken by UEs' random walks.

For the simulation of proposed SDN-based load balancing mechanism, we set up two reference scenarios: traditional load balancing and distributed mobility load balancing (DLB) [6]. In traditional load balancing, the decisions are made based on the received signal strengths. On the other hand, in DLB, the handover parameter is adjusted dynamically according to cell load measurements $\mathrm{LOAD}_c = \min(\frac{\sum_{u\in c} N_u}{N_{\mathrm{tot}}}, 1)$ [6], which is the ratio of required resources, N_u, of all the users u to the total number of resources, N_{tot}, in the cell c. That is to say, when cell load of the source cell exceeds a certain threshold, load balancing is executed and edge users are handed over to the target cell with higher SNR or lower load ratio, respectively.

Figure 9 depicts the number of handover instances as a function of the simulation runtime. A lower number of handovers is desirable since frequent handovers affect service quality and user experience. It can be seen from the figure that SDN-based LB has fewer number of handovers as compared to the other two scenarios. This happens because with the better knowledge of all the cell load states and trends, SDN-based LB is able to select the most suitable target cells more efficiently while DLB takes more rounds, consequently requires more time to achieve the optimized state.

Figure 10 shows the extent of equilibrium between cells as a function of the simulation runtime. It can be seen

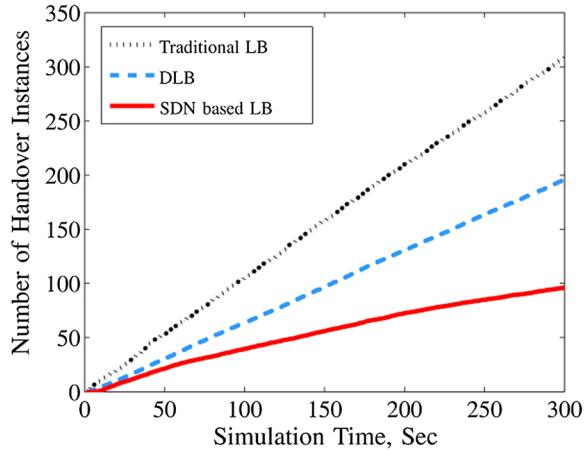

Fig. 9 Cumulative handover times as a function of the simulation time

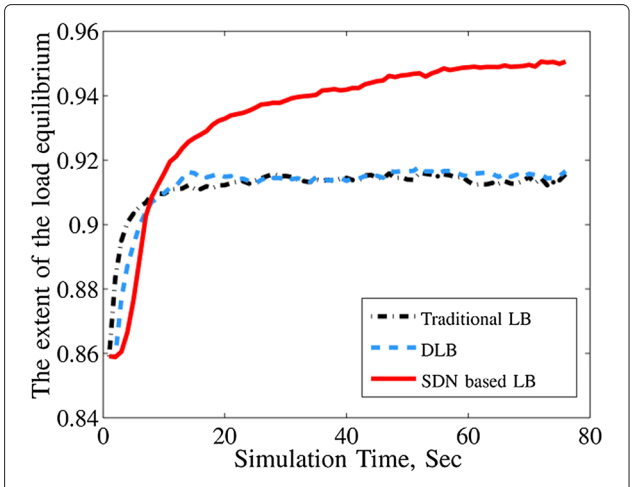

Fig. 10 The extent of equilibrium as a function of the simulation time

that with increasing simulation time, traffic load becomes more balanced with all the LB mechanisms. However, SDN-based LB outperforms the baseline methods. This is because the baseline methods only have a limited view of the network. Notice that at the start of the simulation, SDN-based LB performs worse than the other two methods. This happens because the baseline methods choose the neighboring cell with the lowest load as target cell for LB, which provides faster balancing. However, the target cell eventually becomes overloaded, if its surrounding environment is mostly highly loaded. On the other hand, with an overall view of the network, SDN-based LB uses the overall state to select the target cell and thus guarantees a long-term advantage in equilibrium extent.

The load balancing performance can be further verified by comparison of the network throughput, as shown in Fig. 11. Obviously, the trend of throughput is similar to the load balancing extent in Fig. 10. This is easy to understand:

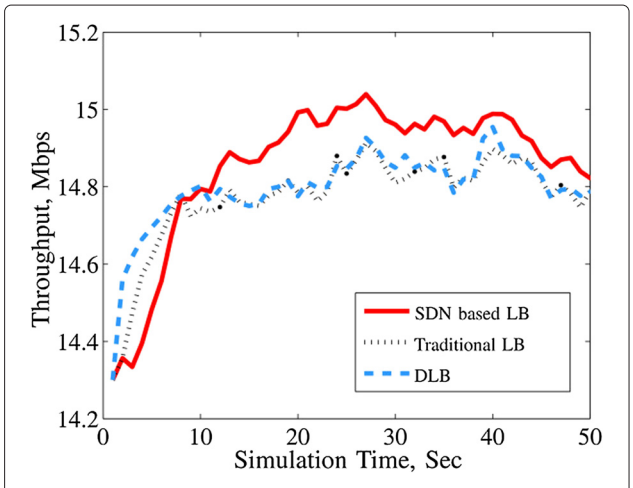

Fig. 11 Network throughput comparison as a function of the simulation time

when the load is more balanced within the network, the resources are utilized more efficiently, rendering a higher throughput for the whole network.

6 Conclusions

Due to the increased data traffic and the co-existence of different radio access technologies, efficient resource management is a key challenge for future 5G networks. In this paper, we presented SDN-based Wi-Fi data offloading and load balancing algorithms. The new algorithms utilized the controller's global view of the network to take more informed decisions for efficient resource management. We also analyzed the performance of the proposed algorithms under realistic load conditions. To this end, we first introduced a queuing model with Pareto arrivals to analyze the processing and forwarding delays incurred due to the SDN architecture. Then, we analyzed the performance of the proposed SDN-based partial data offloading scheme in terms of the threshold miss probability and the amount of data offloaded successfully onto Wi-Fi. Through simulations, it was shown that partial data offloading saves primary resources and decreases threshold miss probability by 20 % $\sim$ 50 %, which ultimately improves the application performance at the user end. Furthermore, the simulation results also confirmed that SDN-based LB outperforms the baseline methods by minimizing the number of required handovers by 50 % and by balancing the loads more evenly across multiple cells. Our results and discussions showed that the delay incurred by SDN is well within the acceptable limits for most applications. Particularly, it was shown that SDN-based solutions perform better for large data traffic with high delay tolerance. All in all, SDN is shown to be a suitable enabling technology for introducing intelligence within the wireless networks and for providing fine-grained control to the network operators.

Competing interests

The authors declare that they have no competing interests.

Received: 20 December 2014 Accepted: 1 June 2015
Published online: 23 June 2015

References

1. K Lee, A Member, J Lee, S Member, Y Yi, Mobile data offloading : how much can Wi-Fi deliver? IEEE/ACM Trans. Netw. **21**(2), 536–551 (2013)
2. Cisco Visual Networking Index: global mobile data traffic forecast update, 2014-2019. Accessed 3 February 2015
3. JG Andrews, S Buzzi, W Choi, SV Hanly, ALACK Soong, JC Zhang, What will 5G be? IEEE J. Sel. Areas Commun. **32**(6), 1065–1082 (2014)
4. A Aijaz, H Aghvami, M Amani, A survey on mobile data offloading: technical and business perspectives. IEEE Wireless Commun. **20**(2), 104–112 (2013)
5. Informa (Firm), Mobile broadband access at home. (Informa Telecoms & Media, London, UK
6. R Kwan, R Arnott, R Paterson, R Trivisonno, M Kubota, On mobility load balancing for LTE systems. IEEE Veh. Technol. Conf. Fall. **1**(5), 6–9 (2010)
7. A Balasubramanian, R Mahajan, A Venkataramani, *Augmenting mobile 3G using WiFi*. ACM MobiSys, USA, 2010)
8. S-i Sou, Mobile data offloading with policy and charging control in 3GPP core network. IEEE Trans. Veh. Technol. **62**(7), 3481–3486 (2013)
9. G Lin, I George, H Jianwei, T Leandros, *Economics of mobile data offloading*. (IEEE INFOCOM Workshop, Italy, 2013)
10. J Lee, Y Yi, S Chong, Y Jin, *Economics of WiFi offloading: trading delay for cellular capacity*. (IEEE INFOCOM, Italy, 2013)
11. J Haydar, A Ibrahim, G Pujolle, A new access selection strategy in heterogeneous wireless networks based on traffic distribution. Wireless Days 1st IFIP. **1**(5), 24–27 (2008)
12. 3GPP, Study of heterogeneous networks management. (TS 32.835 Release 12, 2013)
13. H Zhang, X Chu, W Guo, S Wang, Coexistence of Wi-Fi and heterogeneous small cell networks sharing unlicensed spectrum. IEEE Commun. Mag. **53**(3), 158–164 (2015)
14. D Lopez-Perez, I Guvenc, X Chu, in *IEEE International Conference on Communications (ICC)*. Theoretical analysis of handover failure and ping-pong rates for heterogeneous networks (Ottawa, 2012)
15. P Demestichas, A Georgakopoulos, D Karvounas, K Tsagkaris, V Stavroulaki, J Lu, 5G on the horizon: key challenges for the radio-access network. IEEE Veh. Technol. Mag. **8**(3), 47–53 (2013)
16. Open Networking Foundation, *OpenFlow Switch Specification*. [Online]. http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf, Accessed 28 Feb 2011
17. BAA Nunes, M Mendonca, X-n Nguyen, K Obraczka, T Turletti, A survey of software-defined networking : past, present, and future of programmable networks. IEEE Commun. Surv. Tutor. **16**(99), 1–18 (2014)
18. N McKeown, H Balakrishman, T Anderson, *Openflow: Enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev, ACM*. (Stanford University, California, 2008)
19. K-k Yap, M Kobayashi, R Sherwood, N Handigol, T-y Huang, M Chan, N Mckeown, OpenRoads: empowering research in mobile networks. ACM SIGCOMM Comput. Commun. Rev. **40**(1), 125–126 (2010)
20. P Dely, A Kassler, N Bayer, H Einsiedler, C Peylo, L Chow, B Collins, N Bambos, M Daniel, S Miguel, A software defined networking approach for handover management with real-time video in WLANs. J. Modern Transportation. **21**(1), 58–65 (2013)
21. K-K Yap, R Sherwood, M Kobayashi, T-Y Huang, M Chan, N Handigol, N McKeown, G Parulkar, in *Proceedings of ACM SIGCOMM workshop*. Blueprint for introducing innovation into wireless mobile networks (India, 2010)
22. 3GPP, Technical specification group services and system aspects; policy and charging control architecture. (TS 23.203 Ver. 9.5.0, June 2010)
23. C-H Ko, H-Y Wei, On-demand resource-sharing mechanism design in two-tier OFDMA femtocell networks. IEEE Trans. Veh. Technol. **60**(3), 1059–1071 (2011)
24. L Zhang, Y Liu, M Zhang, S Jia, X Duan, in *IEEE International Conference on Communication Technology (ICCT)*. A two-layer mobility load balancing in LTE self-organization networks (China, 2011)
25. M Jarschel, D Schlosser, S Oechsner, in *Proceedings of the 2011 23rd International Teletraffic Congress*. Modeling and performance evaluation of an OpenFlow architecture (USA, 2011)
26. GR Dattatreya, *Performance Analysis of Queuing and Computer Networks. Chapman Hall/CRC Press*. (Taylor Francis Group, University of Texas at Dallas. U.S.A, 2008)
27. AC Gilbert, W Willinger, A Feldmann, Data networks as cascades: investigating the multifractal nature of the Internet. ACM SIGCOMM Comput. Commun. Rev. **28**, 42–55 (1998)
28. KS Munasinghe, A Jamalipour, in *IEEE Wireless Communications and Networking Conference (WCNC)*. Evaluation of session handoffs in a heterogeneous mobile network for Pareto based packet arrivals (Hungary, 2009)
29. ME Crovella, A Bestavros, Self-similarity in world wide web traffic: evidence and possible causes. IEEE/ACM Trans. Netw. **5**(6), 835–846 (1997)
30. M Jarschel, *An Assessment of Applications and Performance Analysis of Software Defined Networking*. (Dissertation, University Wurzburg, 2014)
31. X Li, H Lu, H Lu, QoS analysis of self-similar multimedia traffic with variable packet size in wireless networks. IEEE Veh. Technol. Conf. Fall. **1**(5), 2–5 (2013)
32. RG Gallager, *Stochastic Processes: Theory for Applications*. (Cambridge University Press, New York, USA, 2014)
33. SG Amin Tootoonchian, Y Ganjali, in *Proc. HotICE*. On controller performance in software-defined networks (USA, 2012)