FEMOS

FEMOS: Fog-Enabled Multi-tier Operations Scheduling in Dynamic Wireless Networks

Shuang Zhao, Yang Yang, Ziyu Shao, Xiumei Yang, Hua Qian, and Chengxiang Wang, *Fellow, IEEE*

***Abstract*—Fog computing has recently emerged as a promising technique in content delivery wireless networks to alleviate the heavy bursty traffic burdens on backhaul connections. In order to improve the overall system performance, in terms of network throughput, service delay and fairness, it is very crucial and challenging to jointly optimize node assignments at control tier and resource allocation at access tier under dynamic user requirements and wireless network conditions. To solve this problem, in this paper, a fog-enabled multi-tier network architecture is proposed to model a typical content delivery wireless network with heterogeneous node capabilities in computing, communication and storage. Further, based on Lyapunov optimization techniques, a new online low-complexity algorithm, namely "Fog-Enabled Multi-tier Operations Scheduling" (FEMOS), is developed to decompose the original complicated problem into two operations across different tiers. Rigorous performance analysis derives the tradeoff relationship between average network throughput and service delay, i.e., [O(1/V), O(V)] with a control parameter V, under FEMOS algorithm in dynamic wireless networks. For different network sizes and traffic loads, extensive simulation results show that FEMOS is a fair and efficient algorithm for all user terminals (UTs) and, more importantly, it can offer much better performance, in terms of network throughput, service delay, and queue backlog, than traditional node assignment and resource allocation algorithms.**

***Index Terms*—Fog Computing, 5G, Internet of Things (IoT), Quality of Experience (QoE), Throughput, Delay.**

I. Introduction

In recent years, global mobile data traffic has experienced explosive growth. It is expected to grow to 49 exabytes per month by 2021, a sevenfold increase over 2016 [2]. Current wireless technologies, such as 4G and WiFi, do not have localized data analysis and processing capabilities so that they cannot handle such a bursty traffic increase. As machine-type communications (MTC) have been adopted in future 5G networks [3]–[5], new flexible network architectures and service strategies are desperately needed to support more and more data-centric and delay-sensitive Internet-of-Things (IoT) applications [6], such as smart city, environment surveillance, intelligent manufacturing, and automatic driving. For future 5G and IoT applications in complex environments, raw measurement data of multiple radio channels at different communication scenarios can be found at an open-source website: *www.wise.sh*. If only centralized cloud computing architecture is applied to those various IoT applications, it is envisaged that the underlayer communication networks, especially backhaul connections, will face heavy bursty traffic burdens and experience dramatic performance degradation. On the other hand, the Moore's Law has significantly driven down the prices of computing and storage devices, more and more smart network nodes and user terminals are deployed and connected into modern communication networks. They provide a rich collection of ubiquitous local computing, communication and storage resources. In view of this technological trend, the concept of fog computing is proposed to enable computing anywhere along the cloud-to-thing continuum [7], [8]. In other words, fog-enabled network architecture and services can effectively leverage those local resources to support fast-growing data-centric and delay-sensitive IoT-applications in regional environments, thus reducing backhual traffic transmissions and centralized computing needs, and at the same time, improving the overall network throughput performance and users' quality of experience (QoE) [9]–[11].

Without loss of generality, let us consider a content delivery wireless network consisting of heterogeneous smart nodes with different computing, communication and storage capabilities. As user terminals (UTs) are moving around and can make requests of any contents at anytime anywhere, it is obvious that popular contents should be placed in multiple neighboring nodes of a UT according to their resources and capabilities. In doing so, most content delivery requests are handled in local network segments, service delay and backhaul traffic transmission can be greatly reduced, thus minimizing the needs for expensive centralized computing resources. Besides traffic localization and service delay, the overall network throughput and fairness among different UTs are crucial performance metrics for network operators and service providers, and therefore need to be analyzed and improved simultaneously.

As an interesting and promising way to relieve the backhaul pressures and reduce the service latency, the cache-enabled content delivery networks have been widely studied. Bastug *et al.* in [12] explored the significant gains in terms of the outage probability and the average delivery rate by having cache-enabled small base station. Li *et al.* in [13] designed distributed caching optimization algorithms via belief propa-

Shuang Zhao, Yang Yang and Xiumei Yang are with Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences (CAS), CHINA. Ziyu Shao is with ShanghaiTech University, CHINA. Hua Qian is with Shanghai Advanced Research Institute, CAS, CHINA. Chengxiang Wang is with Heriot-Watt University, UK. Shuang Zhao is also with the University of Chinese Academy of Sciences. Corresponding author: Yang Yang (yang.yang@wico.sh).

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61671436 and the Science and Technology Commission Foundation of Shanghai under Grant No. 16511104204.

Part of this work will be presented at IEEE Global Communications Conference (Globecom), Singapore, Dec. 2017 [1].

gation to minimize the downloading latency. Liu *et al.* in [14] studied the cache placement problem in Fog-Radio Access Network (F-RAN) and developed transmission aware cache placement strategies. Most of those works mainly focus on file placement in the design of content delivery networks, while dynamic association between the UTs and access nodes were rarely investigated, which also directly affects UTs' QoE, and therefore is the focus of this research.

On the other hand, the topic about taking the advantages of available computing and storage resources at network edges has attracted significant attentions recently. Mao *et al.* in [15] provided a comprehensive survey of the state-of-the-art mobile edge computing (MEC) research with a focus on joint radio-and-computational resource management. In [16] a joint radio and computational resource management algorithm for multi-user MEC system is proposed, which efficiently utilized the powerful computation resource at the MEC server. Shanmugam *et al.* in [17] focused on minimizing the transmission delay through taking advantages of the distributed storage capacities at the network edges. Shih *et al.* in [18] introduced the F-RAN architecture which can provide ultra low-latency service. Pang *et al.* in [19] studied the latency-driven cooperative task computing in F-RAN networks, where the F-RAN node can offload its computation tasks to the nearby load-free F-RAN nodes. Different from previous research works, which mainly focused on improving the utilization of computing or storage resources at network edges, this research applies the concept of fog computing to deal with a more complex multi-tier network architecture with heterogeneous node capabilities and dynamic network resources, in terms of computing power, storage capacity, transmission power, and communication bandwidth.

Specifically, in this paper, the content delivery wireless network is modeled with access tier and control tier, where (i) a node in access tier is typically located close to the UTs and is called “fog access node" (FAN). An FAN caches a subset of popular contents depending on its limited computing and storage capabilities. Due to restricted transmission power and dynamic wireless environment, the communication channels between an FAN and its neighboring UTs are unreliable and time-varying. (ii) a node in control tier manages a group of FANs through reliable but expensive backhaul connections, and is called “fog control node" (FCN). An FCN is much more powerful in terms of computing power and storage capacity than FAN. However, the FCN is physically not possible or economically not feasible to communicate directly with any UTs. It is more efficient for an FCN to execute control operations for achieving regional performance objectives with a centralized approach. Under this realistic multi-tier system model, it is very challenging to simultaneously address the following problems in real-time.

(1) When popular contents are randomly cached at different FANs, how to identify the most feasible FAN for every UT's request in order to maximize network throughput and global fairness?
(2) Under dynamic wireless network conditions and fading channel characteristics, how to effectively allocate communication bandwidth for associated FAN-UT pairs in order to minimize service delay?

In particular, our main contributions are summarized as follows:

- Based on the concept of fog computing, a fog-enabled multi-tier network architecture is proposed to model a content delivery wireless network, which takes full advantages of heterogeneous node capabilities in computing, communication and storage.
- Based on this network model and Lyapunov optimization techniques, an online (real-time) low-complexity fog-enabled multi-tier operations scheduling (FEMOS) algorithm is developed to jointly optimize node assignments at control tier and resource allocation at access tier. Specifically, the inherent mixed nonlinear integer programming problem in this multi-tier network architecture is fully investigated and its structural information is exploited to decompose the original complicated problem into two operations, i.e. centralized assignment of access node (CAAN) scheme at the FCN and dynamic online bandwidth allocation (DOBA) scheme at FANs.
- A general analytical framework is then developed to evaluate the performance of FEMOS algorithm and, more importantly, to characterize the tradeoff relationship between average network throughput and service delay. Compared with traditional node assignment and resource allocation algorithms, simulation results show that FEMOS can offer much better performance, in terms of network throughput and service delay.

The rest of this paper is organized as follows. The system model is presented in Section II. Under the fog-enabled network architecture, the problem is formulated and the responding analytical framework is developed in Section III. The online fog-enabled multi-tier operations scheduling algorithm is proposed in Sections IV. The performance analysis for the proposed algorithm is conducted in Section V and simulation results are shown in Section VI. Section VII concludes this paper.

II. System Model

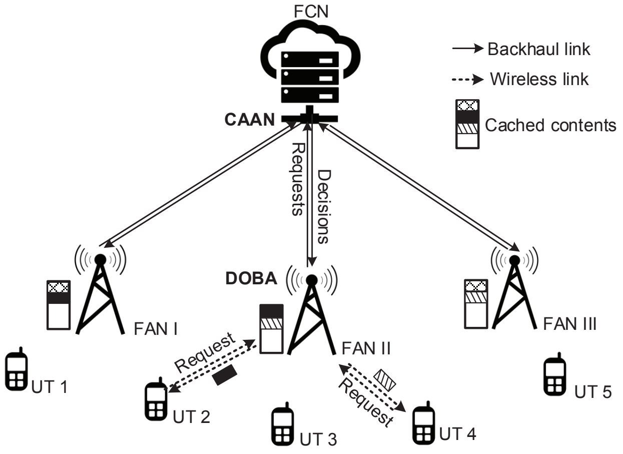
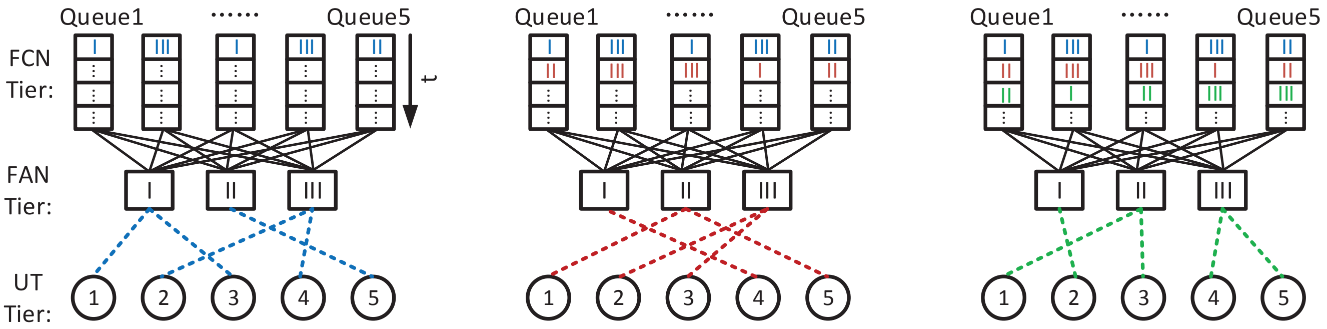


Fig. 1: A sample fog-enabled multi-tier network with three fog access nodes (FANs) and five user terminals (UTs).



(a) $t = 1$.

(b) $t = 2$.

(c) $t = 3$.

Fig. 2: Illustration of dynamic FAN assignment.

We consider a fog-enabled multi-tier network with heterogeneous nodes as shown in Fig. 1, which involves a fog control node (FCN) tier, a fog access control node (FAN) tier and a set of multiple stationary or low-mobility UTs in the region under consideration. Each UT possesses small storage capacity, minor communication ability and little or no computation ability. UTs request files to be downloaded from FAN tier through wireless links. Due to the restricted transmission power and dynamic wireless environment, the communication channels between a FAN and its neighboring UTs are unreliable and time-varying. Each FAN in FAN tier is equipped with limited storage capacity, medium computation ability but strong communication ability. All of them cached a subset of popular files. Through reliable backhaul links, FANs are connected with a FCN, which is next to the cloud and core network, has the global information about the network and is the server of the file library. The FCN has both sufficient storage capacity and powerful computation ability. We emphasize here that in our model, the backhaul links transmit control information from the FCN to FANs and files cached at each FANs are only refreshed at off-peak times. To be clear, if the UT requested file is not cached on the FANs, the FAN will not fetch the file from the operation center through the backhaul link. Under these conditions, we can see more clearly the benefit of dynamic assignment of FAN and resource scheduling.

Denote the set of FANs as $\mathcal{H}$, the set of UTs as $\mathcal{U}$, and the file library as $\mathcal{F}$. The large scale fading and small scale fading coefficients seen by each FAN are assumed to be mutually independent. We assume that the network operates in a slotted system, indexed by $t \in \{0, 1, 2, \ldots\}$ and the time slot length is $\mathcal{T}$. Take Fig. 1 along with Fig. 2 as an example, where $\mathcal{H} = \{\text{I, II, III}\}$, $\mathcal{U} = \{1, 2, 3, 4, 5\}$. During each slot, UT u broadcasts its content request for file with only one type $f \in \mathcal{F}$, to the FAN tier. The arrived but not yet served requests will be queued in the request buffers at the FCN, as shown in Fig. 2. The FCN determines the dynamic FAN assignment for UTs at beginning of each slot, to optimize the network throughput in a memoryless pattern, based on the network states, request queue length and disregarding all such previous decisions. Each UT requested files will be then transmitted by its associated FAN through the wireless link. The queue length of current unserved request buffers will in turn influence the FCN's decision about FAN assignment in the next slot. Each FAN then independently implements its per-slot scheduling policy including the bandwidth and service rate allocation over the UTs associated with it. Consistent with the above setting, the requested file f can only be downloaded from the FAN that has cached it. The service rate would be zero if the requested file is not cached on the FAN that the UT is associated with. For ease of reference, we list the key notation of our system model in TABLE I.

TABLE I: Summary of key notations

Notation	Description
$\mathcal{U}$	Index set of the UTs
$\mathcal{H}$	Index set of the FANs
$\mathcal{F}$	Index set of the files
$\mathcal{E}(\tilde{\mathcal{E}})$	UT-FAN (FAN-File) association set
M	Maximum connectable UTs for each FAN in one time slot
t	Index set of the time slots
$\mathcal{T}$	The length of one time slot
$x_{uh}(t)$	Association indicator between UT u and FAN h in time slot t
y_{hf}	Association indicator between FAN h and file f
$A_u(t)$	Requested files amount by UT u in time slot t
$I_{uf}(t)$	File requested indicator in time slot t
$C_{uh}(t)$	Maximum achievable service rate over link (u, h) in time slot t
$\nu_{uh}(t)$	Allocated bandwidth proportion for UT u in time slot t
$\mu_u(t)$	Achievable service rate for UT u in time slot t
$\mu_{uf}(t)$	Allocated service rate for requested file f in time slot t

A. FAN Assignment and File Placement

Define the FAN assignment (UT-FAN association) as a bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{H}, \underline{\mathcal{E}})$, where $\underline{\mathcal{E}}$ contains edges for the pairs (u, h) such that there exists a potential transmission link between FAN $h \in \mathcal{H}$ and UT $u \in \mathcal{U}$. We assume $\mathcal{G}$ varies in different time slots. Let $\mathbf{X}(t)$ denote a $|\mathcal{U}| \times |\mathcal{H}|$ association matrix of $\mathcal{G}$ between UTs and FANs in time slot t, where $|\mathcal{U}|(|\mathcal{H}|)$ denotes the cardinality of the set $\mathcal{U}(\mathcal{H})$, $\mathbf{X}(t) \triangleq \left[x_{uh}(t)\right]_{u,h}$. Here $x_{uh}(t) = 1$ if $(u, h) \in \underline{\mathcal{E}}$, and 0 otherwise.

Define the file placement (FAN-File association) as a bipartite graph $\widetilde{\mathcal{G}} = (\mathcal{H}, \mathcal{F}, \widetilde{\mathcal{E}})$, where edges $(h, f) \in \widetilde{\mathcal{E}}$ indicates that files with type f are cached in FAN h. The file set cached at each FAN is $\mathcal{N}$, $\mathcal{N} \subseteq \mathcal{F}$, with $|\mathcal{N}|$ different file types.

In this paper, we focus on dynamic FAN assignment and the resource scheduling problem. As said before, the backhaul updates the storages at a time scale much larger than the time scale of UTs placing file requests. Therefore, assuming fixed file placement (FAN-File Association) is justified. Let $\mathbf{Y}$ denote a $|\mathcal{H}|\times|\mathcal{F}|$ file placement matrix of $\widetilde{\mathcal{G}}$ and $\mathbf{Y} \triangleq \left[y_{hf}\right]_{h,f}$. Here, $y_{hf} = 1$ if $(h, f) \in \widetilde{\mathcal{E}}$, and 0 otherwise.

We assume that each UT can be associated with at most one FAN and each FAN can associate with at most M UTs in one time slot. Thus $\mathbf{X}(t)$ should be chosen from the feasible set $\mathscr{A}$,

$$\mathscr{A} = \left\{ \mathbf{X}(t) \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{H}|} \,\middle|\, \begin{array}{l} \sum_{u\in\mathcal{U}} x_{uh}(t) \leq M,\ \forall h \in \mathcal{H}; \\ \sum_{h\in\mathcal{H}} x_{uh}(t) \leq 1,\ \forall u \in \mathcal{U}. \end{array} \right\}. \tag{1}$$

B. UT Traffic Model

All UTs are assumed to generate file request traffic randomly in each time slot, and this traffic generation is independent of the FCN's operation.

Let $\mathbf{A}(t)$ denote the request arrival vector in time slot t and $\mathbf{A}^T(t) \triangleq [A_1(t), \cdots, A_{|\mathcal{U}|}(t)]$, where random variable $A_u(t)$ (with the unit kbits) denotes the requested amount in time slot t and the operation $(\cdot)^T$ denotes vector transposition. Here we assume that $A_u(t)$ is *i.i.d.* with $\mathbb{E}\{A_u(t)\} = \lambda_u$, and there exists a positive constant $A_{\max}$ such that $0 \leq A_u(t) \leq A_{\max}$.

Let $\mathbf{I}(t)$ denote the $|\mathcal{U}| \times |\mathcal{F}|$ requested file type matrix in time slot t and $\mathbf{I}(t) \triangleq \left[I_{uf}(t)\right]_{u,f}$. Here $I_{uf}(t) = 1$ if the requested file type by UT u is f in time slot t, and 0 otherwise. We assume that each UT can request at most only one type of file in one time slot, which means that the row weight of $\mathbf{I}(t)$ is at most 1. The requested probability of each file $f \in \mathcal{F}$ is subject to Zipf distribution [20].

C. The Transmission Model

The wireless channels between UTs and FANs are assumed to be flat fading channels [21], and all FANs transmit at constant power. We assume that the additive white Gaussian noise (AWGN) at the UTs follows Gaussian distribution with $\mathcal{N}(0, \sigma^2)$. Note that the maximum service rate of UT u can be obtained if it has been allocated the total bandwidth by its associated FAN. Then the maximum backlog that can be served in time slot t over link $(u, h) \in \mathcal{E}$ is given by

$$C_{uh}(t) = \mathcal{T} B_h(t) \cdot \mathbb{E}\left[\log_2\left(1 + \frac{P_h g_{hu}(t)|s_{hu}|^2}{\sigma^2 + \sum_{h'\in\mathcal{H}\backslash h} P_{h'} g_{h'u}(t)|s_{h'u}|^2}\right)\right], \tag{2}$$

where $B_h(t)$ is the total bandwidth of FAN h in time slot t, P_h is the transmit power of FAN h, $g_{hu}(t)$ is the large scale fading from FAN h to UT u which contains pathloss and shadow, and s_{hu} is the small scale fading which follows the Rayleigh distribution. For simplicity, currently implemented rate adaption schemes [22] [23] are consistent in assuming, slowly varying pathloss coefficients $g_{hu}(t)$ change across slots in an *i.i.d.* manner, and each FAN h being aware of $g_{hu}(t)$ for all $u \in \mathcal{U}$ at the beginning of each time slot t.

We also assume that each FAN h serves its associated UTs by using orthogonal FDMA or TDMA, which is consistent with most current wireless standards. Let $\nu_{uh}(t)$ be the proportion of bandwidth allocated to UT u by FAN h. Then $\nu_{uh}(t)$ satisfies $0 < \nu_{uh}(t) \leq 1$ when $x_{uh}(t) = 1$, otherwise $\nu_{uh}(t) = 0$. Denote $\boldsymbol{\nu}(t) \triangleq \left[\nu_{uh}(t)\right]_{u,h}$ as the bandwidth allocation matrix, which is chosen from the feasible set $\mathscr{B}$,

$$\mathscr{B} = \left\{ \boldsymbol{\nu}(t) \in \mathbb{R}_+^{|\mathcal{U}|\times|\mathcal{H}|} \,\middle|\, \begin{array}{l} \sum_{u\in\mathcal{U}} \nu_{uh}(t) x_{uh}(t) \leq 1; \\ \nu_{uh} = 0 \text{ if } x_{uh} = 0,\ \forall h \in \mathcal{H}. \end{array} \right\}. \tag{3}$$

Let $\mu_u(t)$ denote the amount of backlog that can be served for UT u in time slot t with maximum value $\mu_{\max}$, which is called service rate hereafter. Define $\boldsymbol{\mu}^{\mathrm{T}}(t) \triangleq [\mu_1(t), \ldots, \mu_{|\mathcal{U}|}(t)]$. Note that each UT can associate with at most one FAN in a time slot, thus $\mu_u(t)$ can be expressed as below:

$$\mu_u(t) = \sum_{h\in\mathcal{H}} C_{uh}(t)\nu_{uh}(t)x_{uh}(t), \forall u \in \mathcal{U}. \tag{4}$$

D. Queueing

In each time slot, the arrived requests of all UTs will be queued in the request buffers at the FCN. We assume that FCN has $|\mathcal{F}|$ request buffers for each UT $u \in \mathcal{U}$. Denote the queue length of the amount of request for file with type f at the beginning of the tth time slot as $Q_{uf}(t)$. Define $Q_u^{\text{sum}}(t) \triangleq \sum_{f\in\mathcal{F}} Q_{uf}(t)$ and denote $\boldsymbol{Q}^T(t) = [Q_u^{\text{sum}}(t), \cdots, Q_{|\mathcal{U}|}^{\text{sum}}(t)]$ as the queue length vector. We assume that all queues are initially empty, *i.e.*, $Q_{uf}(0) = 0, \forall u \in \mathcal{U}, f \in \mathcal{F}$.

Let $\mu_{uf}(t)$ denote the service rate for the requested file f scheduled by the FCN according to a certain queueing discipline [24], such as FIFO, LIFO or Random discipline. We adopt fully-efficient scheduling policy given in [24] for queues, which means:

$$\sum_{f\in\mathcal{F}} \mu_{uf}(t) = \mu_u(t), \tag{5}$$

where $\mu_u(t)$ is defined in (4) and $\mu_{uf}(t) = 0$ if $y_{hf} = 0$.

The queue length $Q_{uf}(t)$ is updated in every time slot t according to the following rules:

$$Q_{uf}(t+1) = \left[Q_{uf}(t) - \mu_{uf}(t)\right]^+ + A_u(t) \cdot I_{uf}(t), \tag{6}$$

where $[x]^+ = \max\{x, 0\}$.

The queueing process $Q_{uf}(t)$ is stable if the following condition holds [25]:

$$\overline{Q}_u^{\text{sum}} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[Q_u^{\text{sum}}(\tau)] < \infty. \tag{7}$$

III. Problem Formulation and Analytical Framework

In this section, we will first introduce the performance metrics in subsection III-A, namely, the time-averaged throughput of the network and the average delay per UT experiences. An average throughput maximization problem with file request queues stability constraints will then be formulated in subsection III-B. In subsection III-C, we will give the analytical framework based on Lyapunov optimization techniques.

A. Performance Metrics

We focus on the total throughput of the network. Therefore, we adopt the time-averaged sum service rate of different UTs in the network as the performance metric, which is defined as follows:

$$\phi_{av} \triangleq \overline{\phi(\boldsymbol{\mu})} = \sum_{u\in\mathcal{U}} \overline{\mu}_u, \tag{8}$$

where $\overline{\mu}_u(t)$ is the averaged expected service rate of UT u, i.e., $\overline{\mu}_u = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}[\mu_u(\tau)]$.

To support for transmission latency sensitive applications, i.e., online video, service delay is a key metric needs to be considered [26]. According to Little's Law [27], the average service delay experienced by each UT is proportional to the averaged amount of its unserved requests waiting at the FCN, which is sum of queue length for different files. Thus, the average delay per UT can be computed as the ratio between the average queue length and the mean traffic arrival rate, which is shown as follows,

$$\Lambda_{av} \triangleq \frac{\sum_{u\in\mathcal{U}} \overline{Q}_u^{\text{sum}}}{\sum_{u\in\mathcal{U}} \lambda_u}. \tag{9}$$

B. Average Network Throughput Maximization Problem

The system objective is to find a feasible FAN assignment $\mathbf{X}(t)$ and bandwidth allocation $\boldsymbol{\nu}(t)$ to maximize the average network throughput while maintaining the stability of all the queues in the network. The average network throughput maximization problem can be formulated in $\mathcal{P}1$:

$$\begin{aligned} \mathcal{P}1: \max_{\mathbf{X}(t),\boldsymbol{\nu}(t)} \quad & \overline{\phi(\boldsymbol{\mu})} \\ \text{s.t.} \quad & \overline{Q}_u^{\text{sum}} < \infty,\ \forall u\in\mathcal{U}, \\ & \mathbf{X}(t)\in\mathcal{A},\ \boldsymbol{\nu}(t)\in\mathcal{B}\ \forall t, \end{aligned} \tag{10}$$

where the requirement of finite $\overline{Q}_u^{\text{sum}}$ corresponds to the strong stability condition for all the queues [25]. Queueing stability implies that the buffered file requests are processed with finite delay. We will show that our proposed algorithms guarantee upper bounds for $\overline{Q}_{uf}$ and thus achieve the bounded service delay.

***Remark** 1:* It is not difficult to identify that $\mathcal{P}1$ is a highly challenging stochastic optimization problem with a large amount of stochastic information to be handled (including channel conditions and request buffer state information) and two optimization variables to be determined, which requires to design an online operation and scheduling scheme for such a network. Besides, to maximize the network throughput, it is essential to jointly optimize the FAN-UT association and the resource allocation, which is always a complicated mixed integer programming problem. Further, the optimal decisions are temporally correlated due to the random arrival traffic demands. Furthermore, the FCN needs to reduce the delay per UT while maintaining the average network throughput, which requires the FCN to maintain a good balance between network throughput and average delay.

C. Lyapunov Optimization Based Analytical Framework

In the following, we focus on solving this challenging problem $\mathcal{P}1$ by using the Lyapunov optimization technique [25], with which we can transfer the challenging stochastic optimization problem $\mathcal{P}1$ to be a deterministic per-slot problem in each time slot.

We first define a quadratic Lyapunov function as follows:

$$L(\boldsymbol{Q}(t)) \triangleq \frac{1}{2}\boldsymbol{Q}^{\mathrm{T}}(t)\boldsymbol{Q}(t) = \frac{1}{2}\sum_{u\in\mathcal{U}} (Q_u^{\text{sum}}(t))^2. \tag{11}$$

We then define a one-slot conditional Lyapunov drift $\Delta(\boldsymbol{Q}(t))$ as follows:

$$\Delta(\boldsymbol{Q}(t)) \triangleq \mathbb{E}\{L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t))|\boldsymbol{Q}(t)\}. \tag{12}$$

Accordingly, the one-slot conditional Lyapunov *drift-plus-penalty* function is shown as follows:

$$\Delta_V(\boldsymbol{Q}(t)) = \Delta(\boldsymbol{Q}(t)) - V\mathbb{E}\{\phi(\boldsymbol{\mu}(t))|\boldsymbol{Q}(t)\}, \tag{13}$$

where $V > 0$ is the policy control parameter. We first establish the upper bound of $\Delta_V(\boldsymbol{Q}(t))$ under any feasible policy $\mathbf{X}(t)$ and $\boldsymbol{\nu}(t)$, as specified in *Lemma 1*.

***Lemma** 1*: For any feasible control decision $\mathbf{X}(t)$ and $\boldsymbol{\nu}(t)$ for $\mathcal{P}1$ such that $\mathbf{X}(t)\in\mathcal{A}$, $\boldsymbol{\nu}(t)\in\mathcal{B}$, $\Delta_V(\boldsymbol{Q}(t))$ is upper bound by

$$\begin{aligned} \Delta_V(\boldsymbol{Q}(t)) \quad \leq \quad & \mathcal{K} - \mathbb{E}\left\{\sum_{u\in\mathcal{U}} \left[V + Q_u^{\text{sum}}(t)\right]\mu_u(t)\middle|\boldsymbol{Q}(t)\right\} \\ & + \mathbb{E}\left\{\sum_{u\in\mathcal{U}} Q_u^{\text{sum}}(t)A_u(t)\middle|\boldsymbol{Q}(t)\right\}, \end{aligned} \tag{14}$$

where $\mathcal{K}$ is a constant.

Proof: Please refer to Appendix A. ■

Lemma 1 provides the upper bound for the conditional *Lyapunov drift-plus-penalty* function $\Delta_V(\boldsymbol{Q}(t))$, which plays a significant role in the FEMOS.

IV. Fog-Enabled Multi-tier Operations Scheduling Algorithm

In this section, we introduce the fog-enabled multi-tier operations scheduling (FEMOS) algorithm, which mainly consists of two stage operations in per time slot: centralized assignment of access node (CAAN) and dynamic online bandwidth allocation (DOBA). We present the DOBA in subsection IV-A and describe the CAAN in subsection IV-B. In subsection IV-C, we propose the FEMOS algorithm and provide the corresponding computation complexity analysis.

A. Dynamic Online Bandwidth Allocation (DOBA)

To solve Problem $\mathcal{P}1$ based on the Lyapunov optimization method [25], it needs to design an algorithm to minimize the upper bound of the *Lyapunov drift-plus-penalty* term in each time slot. Ignoring the constant components in the upper bound of $\Delta_V(\boldsymbol{Q}(t))$ and rearranging them, the upper bound minimization problem is converted to:

$$\min_{\mathbf{X}(t)\in\mathcal{A},\boldsymbol{\nu}(t)\in\mathcal{B}} -\sum_{u\in\mathcal{U}} \left[V + Q_u^{\text{sum}}(t)\right]\mu_u(t). \tag{15}$$

Note that $Q_u^{\text{sum}}(t)$ is observed at the beginning of each time slot, which can be viewed as constant per time-slot. Therefore the upper bound minimization only depends on $\mu_u(t)$, which involves the FAN assignment and bandwidth allocation. For convenience, we define $W_{uh}(t) \triangleq [V + Q_u^{\text{sum}}(t)]C_{uh}(t)$, which is constant per time-slot. From the definition of $\mu_u(t)$ in (4), the upper bound minimization of $\Delta_V(\boldsymbol{Q}(t))$ is transferred to the following equivalent problem:

$$\mathcal{P}_{\text{AR}}: \max_{\mathbf{X}(t)\in\mathcal{A},\boldsymbol{\nu}(t)\in\mathcal{B}} \quad \sum_{h\in\mathcal{H}}\sum_{u\in\mathcal{U}} W_{uh}(t)\nu_{uh}(t)x_{uh}(t) \quad (16)$$

$\mathcal{P}_{\text{AR}}$ is a joint optimization problem of access node assignment and bandwidth allocation. The main idea of the proposed FEMOS algorithm is to solve the deterministic optimization problem $\mathcal{P}_{\text{AR}}$ in each time slot. By doing so, the amount of request waiting in the queues can be maintained at a small level and the network throughput can be maximized at the same time.

Note that $\mathcal{P}_{\text{AR}}$ is a nonlinear integer programming problem, for which the computational complexity of the brute-force search is prohibitive. By exploiting the structure information of $\mathcal{P}_{\text{AR}}$, we transfer problem $\mathcal{P}_{\text{AR}}$ to the following problem $\mathcal{P}'_{\text{AR}}$, which is proven equivalent with $\mathcal{P}_{\text{AR}}$:

$$\begin{aligned} \mathcal{P}'_{\text{AR}}: \max_{\mathbf{X}(t)} \quad & \sum_{h\in\mathcal{H}}\sum_{u\in\mathcal{U}} W_{uh}(t)x_{uh}(t) && (17)\\ \text{s.t.} \quad & \sum_{u\in\mathcal{U}} x_{uh}(t) \le 1, \ \forall h \in \mathcal{H}, \\ & \sum_{h\in\mathcal{H}} x_{uh}(t) \le 1, \forall u \in \mathcal{H}, \quad x_{uh}(t) \in \{0,1\}, \end{aligned}$$

and the corresponding bandwidth allocation for each UT $u \in \mathcal{U}$ can be expressed as follow:

$$\nu_{uh}(t) = \begin{cases} 1 & x_{uh} = 1, \forall h \in \mathcal{H}, \\ 0 & \text{otherwise}. \end{cases} \quad (18)$$

***Lemma** 2* : The FAN assignment solution obtained by the transformed problem $\mathcal{P}'_{\text{AR}}$ along with the bandwidth allocation given by (18) coincides with the solution for $\mathcal{P}_{\text{AR}}$.

Proof: See Appendix B. ■

Through the equivalent problem transformation described above, the solution to joint optimization of FAN assignment and bandwidth allocation are decoupled in two stage operations: centralized assignment of access node (CAAN) and dynamic online bandwidth allocation (DOBA). Specifically, the CAAN will be executed at FCN with powerful computation ability by solving $\mathcal{P}'_{\text{AR}}$. Once the CAAN is established, the FANs will executed DOBA as (18) independently.

B. Centralized Assignment of Access Node (CAAN)

The dynamic assignment of FANs in each time slot can be obtained by solving the problem $\mathcal{P}'_{\text{AR}}$. Note that the constraint of maximum connectable UTs for each FAN in $\mathcal{P}_{\text{AR}}$ is M, while it reduces to 1 in $\mathcal{P}'_{\text{AR}}$. To further understand the dynamic FAN assignment scheme under the constraints of $\mathcal{P}'_{\text{AR}}$, we take Fig. 3 as an example, where the colored dash lines represent the UT-FAN association in time slot $t = 1,\ 2,\ 3$ and the shadowed queues represent the responding UTs didn't access the network in those time slots. In time slot $t = 1$, UT 2, UT 3 and UT 5 are associated with FAN I, III and II, respectively and all of them obtained the total bandwidth. The arrival requests of not associated UTs (UT 1 and UT 3) will be queued in request buffers at the FCN, waiting for being served in the next time slot.

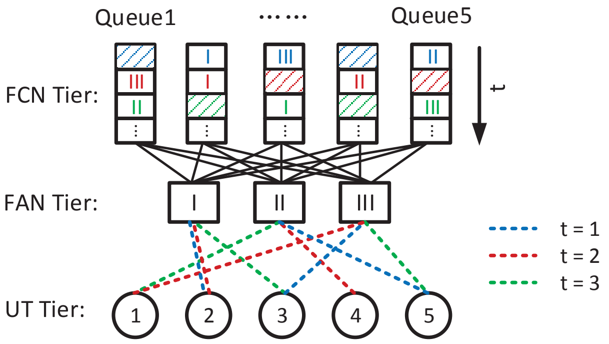


Fig. 3: Illustration of dynamic FAN assignment scheme under the constraints in $\mathcal{P}'_{\text{AR}}$.

To solve $\mathcal{P}'_{\text{AR}}$ efficiently, we demonstrate that the problem $\mathcal{P}'_{\text{AR}}$ can be formulated as the maximization of a normalized modular function subject to intersection of two partition matroid constraints in the following. This structure can be exploited to design computationally efficient algorithms for Problem $\mathcal{P}'_{\text{AR}}$ with provable approximation gaps. The definitions of matroid and submodular function are given in Appendix G.

First, we define $\mathcal{E}$ as ground set $\mathcal{E} = \{(u,h)|u \in \mathcal{U}, h \in \mathcal{H}\}$, that consists of all possible tuples and where each tuple (u,h) denotes an association of UT u to FAN h. Further, we define set $\mathcal{E}_u = \{(u,h)|h \in \mathcal{H}\}$, which consists of all tuples for each UT $u \in \mathcal{U}$, along with set $\mathcal{E}^h = \{(u,h)|u \in \mathcal{U}\}$, which consists of all tuples for each FAN $h \in \mathcal{H}$. Next, we proceed to show that $\mathcal{P}'_{\text{AR}}$ can be solved by a simple greedy algorithm. Towards this end, we define $\mathcal{W}_{\underline{\mathcal{E}}} \triangleq \sum_{(u,h)\in\underline{\mathcal{E}}} W_{uh}(t)$, where $\underline{\mathcal{E}} \subseteq \mathcal{E}$. We also define a family of sets $\mathcal{I}$, as the one that includes each subset $\underline{\mathcal{E}}$ of $\mathcal{E}$ such that the subset meets the UT-FAN limits in $\mathcal{P}'_{\text{AR}}$. Specially,

$$\mathcal{I} = \left\{ \underline{\mathcal{E}} \subseteq \mathcal{E} \middle| \left|\underline{\mathcal{E}} \cap \mathcal{E}_u\right| \le 1,\ \forall u;\ \left|\underline{\mathcal{E}} \cap \mathcal{E}^h\right| \le 1,\ \forall h \right\}. \quad (19)$$

Then, we offer the following result.

***Lemma** 3*: The problem $\mathcal{P}'_{\text{AR}}$ can be formulated as the maximization of a normalized modular function subject to two partition matriod constraints on ground set $\mathcal{E}$.

Proof: See Appendix C. ■

As a consequence of *Lemma 3*, we can leverage the famous result in [28], which shows that a simple effective greedy algorithm yields a constant factor approximation for the problem of maximizing a normalized non-negative modular set function subject to matroid constraints. Specializing the approximation guarantee of [28] to our case with two matroid constraints, the problem $\mathcal{P}'_{\text{AR}}$ can be approximately solved using greedy FAN assignment algorithm with a constant-factor $\frac{1}{2}$ approximation guarantee.

Algorithm 1 The Greedy FAN Assignment Algorithm

```
Initialize $\underline{\hat{\mathcal{E}}} = \emptyset$, $\mathcal{E}' = \mathcal{E}$
**Repeat** Determine $(u^\star, h^\star) \in \mathcal{E} \setminus \underline{\hat{\mathcal{E}}}$ as the solution to $\max_{\underline{\hat{\mathcal{E}}} \cup (u^\star, h^\star) \in \mathcal{I}} W_{u^\star h^\star}(t)$
    Update $\underline{\hat{\mathcal{E}}} = \underline{\hat{\mathcal{E}}} \cup (u^\star, h^\star)$
    Update $\mathcal{E}' = \mathcal{E}' \backslash (u^\star, h^\star)$
**Until** $\mathcal{E}' = \emptyset$
**Output** $\underline{\hat{\mathcal{E}}}$
```

The greedy FAN assignment algorithm is described in Algorithm 1. It starts with an empty set $\underline{\hat{\mathcal{E}}} = \emptyset$ and $\mathcal{E}' = \mathcal{E}$. It iteratively adds the element $(u^\star, h^\star)$ with highest marginal value while satisfying the matroid constraints in each step. The marginal value of an element $(u^\star, h^\star)$ is defined as the gain of adding $(u^\star, h^\star)$ into the UT-FAN association set $\underline{\hat{\mathcal{E}}}$, given by $W_{u^\star h^\star}(t)$, when $\underline{\hat{\mathcal{E}}} \cup (u^\star, h^\star) \in \mathcal{I}$. In our greedy FAN assignment algorithm, once the element $(u^\star, h^\star)$ is added to $\underline{\hat{\mathcal{E}}}$, it should be deleted from set $\mathcal{E}'$. The greedy FAN assignment algorithm stops until $\mathcal{E}'$ becomes $\emptyset$ during the iteration.

C. Fog-Enabled Multi-tier Operations Scheduling (FEMOS) Algorithm

The proposed FEMOS algorithm is summarized in Algorithm 2, which will be implemented at both FCN tier and FAN tier in practice. Specifically, in each time slot, within the global network information and queue length of each request buffer, the FCN will execute CAAN by run the Greedy FAN Assignment Algorithm. The assignment decision is then transmitted to the FAN tier. Next, each FAN performs the DOBA independently and schedule the service rate for the associated UT under full efficient scheduling policy. Finally, the FCN updates the request buffers for each UT, the length of which will influence the operations scheduling in next slot.

Algorithm 2 Fog-Enabled Multi-tier Operations Scheduling (FEMOS) Algorithm

```
Set $t = 0$, $\boldsymbol{Q}(0) = \mathbf{0}$;
**While** $t < t_{\text{end}}$, **do**
    At beginning of the $t$th time slot, observe $A_u(t)$, $g_{hu}(t)$ and $Q_{uf}(t)$;
    CAAN: $\mathbf{X}^\star(t)$ is obtained by run Algorithm 1;
    DOBA:
        $\nu^\star_{uh}(t) = \begin{cases} 1 & x^\star_{uh}(t) = 1 \\ 0 & \text{otherwise}; \end{cases}$
    Schedule the service rates $\mu^\star_{uf}(t)$ to the queues $Q_{uf}(t)$ according to (5) with any pre-specified queueing discipline;
    Update $\{Q_{uf}(t)\}$ according to (6) for each UT based on $\mathbf{X}^\star(t)$, $\boldsymbol{\nu}^\star(t)$ and $\mu^\star_{uf}(t)$;
    $t \leftarrow t + 1$.
**end While**
```

***Remark** 2 :*

- Interestingly, a closer inspection of $\mathcal{P}'_{\text{AR}}$ reveals that in each time slot each FAN can associate with at most only one UT and the number of UTs that can dynamically associate with FANs depends on the number of FANs, which is $|\mathcal{H}|$. What makes our greedy FAN assignment algorithm outstanding from the common assignment method which directly select FAN with the best channel condition for each UT or each FAN associates with the UT with the longest queue length is that, we provide a quantitative analysis and selection method for UT-FAN association.
- In each time slot, the association between UTs and FANs depends on corresponding UT-FAN pair gain $W_{uh}(t) = [V + Q_u^{\text{sum}}(t)]C_{uh}(t)$. The greedy FAN assignment algorithm greedily selects the UT-FAN pair with the largest $W_{uh}(t)$ within the feasible set. For each FAN h, it will associate with one UT with either large queue backlog or good channel condition. If V is small, i.e., $V \ll Q_u^{\text{sum}}(t)$, both the queue backlog and the channel condition will determine the decision on UT-FAN association. The FAN h will associate with UT with largest $Q_u^{\text{sum}}(t)C_{uh}(t)$. Conversely, if V is large enough, i.e., $V \gg Q_u^{\text{sum}}(t)$, the FAN h will more effectively invoke the willingness to associate with a UT with a good channel condition $C_{uh}(t)$. Under large V, the UT with weak channel conditions can not access the network for a long time, leading to large accumulated queue length and influence on the UT-FAN decision in turn.
- Therefore, the parameter V actually controls the FANs' willingness to serve UTs, i.e., performing UT-FAN association. In other words, it controls the trade-off between network throughput and transmission delay.

In each time slot, the computational complexity for FEMOS primarily remains in the greedy-UA algorithm. According to Algorithm 1, the computational complexity of the greedy FAN assignment algorithm is $O(|\mathcal{U}||\mathcal{H}|\Gamma^\star)$, where $\Gamma^\star$ denotes the computational complexity of searching for the element $(u^\star, h^\star)$. The greedy algorithm starts with an empty set and at each step, it adds one element with the highest marginal value to the set while maintaining the feasibility of the solution. Since the objective function is modular, the marginal value of the elements decreases as we add more elements to the set $\underline{\hat{\mathcal{E}}}$. Thus, in one iteration, if the largest marginal value is zero, then the algorithm should stop. In our case, there would be at most $|\mathcal{U}||\mathcal{H}|$ iterations. Each iteration would involve evaluating the marginal value of at most $|\mathcal{U}||\mathcal{H}|$ elements. Then, the computational complexity of searching for the element $(u^\star, h^\star)$ is $O(\Gamma^\star) = O(|\mathcal{U}||\mathcal{H}|)$. Thus the overall computational complexity for the greedy FAN assignment algorithm is $O(|\mathcal{U}|^2|\mathcal{H}|^2)$.

V. Performance Analysis

In this section, we provide the main theoretical results for FEMOS, which characterize the lower bounds for average network throughput as well as the upper bounds for the average sum queue length of the requests of all the UTs. Additionally, the tradeoff between the network average throughput and average delay will also be revealed.

Note that the greedy FAN assignment algorithm is efficient and guarantees a tight $\frac{1}{2}$-approximation for $\mathcal{P}'_{\text{AR}}$, i.e., the worst case is at least 50% of the optimal solution. The FAN assignment $\mathbf{X}(t)$ and bandwidth allocation $\boldsymbol{\nu}(t)$ is then a suboptimal solution to the problem $\mathcal{P}_{\text{AR}}$, which we refer to as *imperfect scheduling* [29], [30]. Therefore, in each time slot, the centralized assignment of access node and dynamic online bandwidth allocation policy for $\mathcal{P}_{\text{AR}}$ yields a transmission rate $\boldsymbol{\mu}(t) \in \mathcal{R}$ that satisfies:

$$\sum_{u\in\mathcal{U}}\Big[V+Q_u^{\text{sum}}(t)\Big]\mu_u(t) \geq \beta \max_{\boldsymbol{\mu}(t)\in\mathcal{R}}\Big\{\sum_{u\in\mathcal{U}}\Big[V+Q_u^{\text{sum}}(t)\Big]\mu_u(t)\Big\}, \tag{20}$$

where β is constant with $\beta = \frac{1}{2}$.

The parameter β in (20) can be viewed as a tuning parameter indicating the degree of precision of imperfect scheduling. Notice that when $\beta = 1$ it reduces to the case with perfect scheduling of $\mathcal{P}_{\text{AR}}$. Define ϕ^{opt} as the optimal average network throughput associated with the problem $\mathcal{P}1$, augmented with the rectangle constraint $\mathcal{R}$, and $\overline{\boldsymbol{\mu}} \in \mathcal{R}$, where $\mathcal{R}$ is chosen large enough to contain the optimal time average service rate vector $\overline{\boldsymbol{\mu}}$. Let $\boldsymbol{\mu}^{*,0}(t)$ denote the transmission rate under the optimal solution to $\mathcal{P}1$. The following β-reduced problem turns out to be a good reference to study imperfect scheduling.

***β*-reduced problem**:

$$\begin{aligned} \max_{\mathbf{X}(t),\boldsymbol{\nu}(t)} \quad & \overline{\phi(\boldsymbol{\mu})} \\ \text{s.t.} \quad & \boldsymbol{\mu}(t) \in \beta\mathcal{R}, \\ & \overline{Q}_u^{\text{sum}} < \infty,\ \forall u \in \mathcal{U}, \\ & \mathbf{X}(t) \in \mathcal{A},\ \boldsymbol{\nu}(t) \in \mathcal{B},\ \forall t. \end{aligned} \tag{21}$$

Let $\boldsymbol{\mu}^{*,\beta}(t)$ denote the transmission rate under the optimal solution to the β-reduced problem above and denote ϕ_β^{opt} as the corresponding optimal throughput. In the following *Lemma 4*, we will demonstrate the relationship between $\mathcal{P}1$ and β-reduced problem in terms of their optimal values.

***Lemma** 4*: Let $\boldsymbol{\mu}^{*,0}(t)$ be the optimal transmission rate for the $\mathcal{P}1$. Then the transmission rate to the β-reduced problem is $\boldsymbol{\mu}^{*,\beta}(t) = \beta\boldsymbol{\mu}^{*,0}(t)$.

Proof: See Appendix D. ∎

Lemma 4 also implies the relationship between the optimal average throughput of $\mathcal{P}1$ and β-reduced problem, which is $\phi_\beta^{\text{opt}} = \beta\phi^{\text{opt}}$.

Next, we characterize the performance of FEMOS under *i.i.d.* system randomness and assume there exists constants ϵ and ϕ_ϵ such that the following *slater-type conditions* holds:

$$\lambda_u - \mathbb{E}\{\mu_u(t)\} \leq -\epsilon,\ \forall u \in \mathcal{U}, \tag{22}$$

$$\mathbb{E}\{\phi(\boldsymbol{\mu}(t)\} = \phi_\epsilon. \tag{23}$$

Note that the assumptions (22) and (23) ensure the strong stability of the queues in the system, which are generally assumed in the network stability problems [25] and guarantee that there exists a stationary and randomized FAN assignment and bandwidth allocation policy.

***Lemma** 5*: For any alternative policy including FAN assignment $\mathbf{X}(t) \in \mathcal{A}$ and bandwidth allocation $\boldsymbol{\nu}(t) \in \mathcal{B}$, we have:

$$\Delta_V(\boldsymbol{Q}(t)) \leq \mathcal{K} - V\phi_\epsilon - \beta\epsilon\sum_{u\in\mathcal{U}} Q_u^{\text{sum}}(t). \tag{24}$$

Proof: See Appendix E. ∎

Based on *Lemma* 4 and *Lemma* 5, the performance bounds of FEMOS are derived in the following theorem. The term ϕ_{av}^{FEMOS} is defined as the long term expected average network throughput of FEMOS and $\Lambda_{av}^{\text{FEMOS}}$ is defined as the long term expected average service delay per UT.

Theorem 1: For the network defined in section II, the centralized assignment of access node and dynamic online bandwidth allocation policy obtained through FEMOS algorithm achieves the following performance:

$$\begin{aligned} \phi_{av}^{\text{FEMOS}} &\triangleq \liminf_{t\to\infty} \phi\Big(\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\phi(\boldsymbol{\mu}(\tau))\}\Big) \\ &\geq \beta\phi^{\text{opt}} - \frac{\mathcal{K}}{V}, \end{aligned} \tag{25}$$

$$\begin{aligned} \Lambda_{av}^{\text{FEMOS}} &\triangleq \frac{1}{\sum_u \lambda_u}\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_u \mathbb{E}\{Q_u^{\text{sum}}(t)\} \\ &\leq \frac{\mathcal{K} + V(\phi^{\text{opt}} - \phi_\epsilon)}{\beta\epsilon\sum_u \lambda_u}. \end{aligned} \tag{26}$$

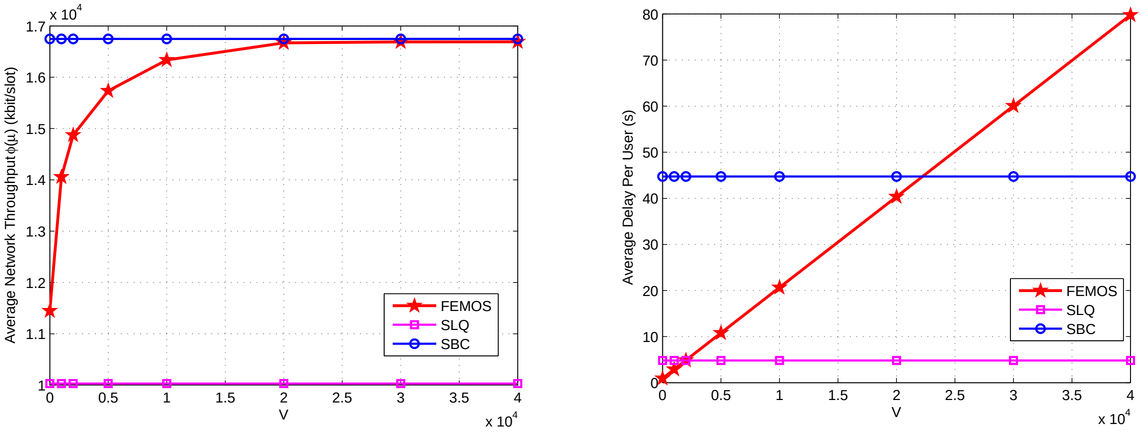
Proof: See Appendix F. ∎

***Remark** 3 :* Theorem 1 shows that under the proposed fog-enabled multi-tier operation scheduling algorithm, the lower bound of average network throughput increases inversely proportional to V, while the upper bound of average service delay per UT experienced increases linearly with V. If a larger V is used to pursue the better network throughput performance, it will introduce severe service delay. Hence, there exists an $[O(1/V), O(V)]$ tradeoff between these two objects. Through adjusting V, we can balance the network throughput and service delay.

VI. Simulation Results

We consider a network with $|\mathcal{H}| = 9$ fixed FANs and $|\mathcal{U}| = U$ randomly deployed UTs. Each FAN can associated with at most $M = 12$ UTs. The size of the file types in the file library is set to $|\mathcal{F}| = 1000$ and the cached file types of each FAN is set to $|\mathcal{N}| = 500$. The system region has a size of $20 \times 20\ m^2$. Each UT requests files with type $f \in \mathcal{F}$ according to the Zipf distribution with parameter η_r, i.e., $p_f = \frac{f^{-\eta_r}}{\sum_{i\in\mathcal{F}} i^{-\eta_r}}$ [31] and $\eta_r = 0.56$ in our simulations. A FIFO queueing discipline is applied in the simulations. The simulation results are averaged over 3000 constant time slots with $\mathcal{T} = 100$ milliseconds intervals.

We assume that each FAN operates on a $B_h = 18$MHz bandwidth (100 resource blocks with 180 kHz for each resource block) and transmits at a fixed power level $P = 20\ W$. Besides, the noise power σ^2 is assumed to be $2 \times 10^{-7}\ W$. Based on the WINNER II channel model in small-cell scenarios [32], the pathloss coefficients between the FAN h and UT u is defined by $g_{hu}(t) = 10^{-\frac{\text{PL}(d_{hu}(t))}{10}}$, where $d_{hu}(t)$ is the distance from FAN h to UT u in time slot t, and $PL(d) = A\log_{10}(d) + B + C\log_{10}(f_0/5) + \mathcal{X}_{dB}$, where f_0 is the carrier frequency, $\mathcal{X}_{dB}$ is a shadowing log-normal variable with variance σ_{dB}^2; $A = 18.7$, $B = 46.8$, $C = 20$, and $\sigma_{dB}^2 = 9$ in line-of-sight (LOS) condition; $A = 36.8$, $B = 43.8$, $C = 20$, and $\sigma_{dB}^2 = 16$ in non-line-of-sight (NLOS) condition. Each



(a) Network Throughput vs. Parameter V.

(b) Service Delay Per UT vs. Parameter V.

Fig. 4: Comparison of FEMOS and other FAN assignment schemes, $A_{\max} = 100$ kbits, $U = 100$.

link is in LOS or NLOS independently and randomly, with probabilities of $p_l(d)$ and $1 - p_l(d)$, respectively, where

$$p_l(d) = \begin{cases} 1 & d \leq 2.5\ m \\ 1 - 0.9(1 - (1.24 - 0.6\log(d))^3)^{1/3} & \text{otherwise.} \end{cases} \tag{27}$$

A. *Performance Comparison with Other Assignment Scheme*

In this subsection, we evaluate the performance of the proposed FEMOS algorithm and compare its Greedy FAN assignment with the other two dynamic schemes: *Select Best Channel* (SBC) and *Select Longest Queue* (SLQ). In SBC scheme, each FAN associates with the UT with the best channel condition between them in each time slot. In SLQ scheme, in each time slot, the UTs with top-$|\mathcal{H}|$ queue length access the network and each of them associates with the FAN with the best transmission condition.

Fig. 4 first validates the theoretical results for the proposed FEMOS algorithm derived in Theorem 1. The average network throughput performance is shown in Fig. 4(a), while the average service delay per UT is shown in Fig. 4(b). It can be observed from Fig. 4(a) that the average network throughput obtained by the Greedy FAN assignment in FEMOS increases as V increases and converges to the maximum value when V is sufficiently large. Meanwhile, as shown in Fig. 4(b), the average service delay experienced by per UT of FEMOS increases linearly with the control parameter V. This is in accordance with the analysis in remark 2 that with V increasing, the importance of request queue backlogs decreases, which makes the dynamic assignment bias towards good channel conditions. Those observations verify the $[O(1/V), O(V)]$ tradeoff between average network throughput and average queue backlog as demonstrated in Theorem 1 and remark 3.

Fig. 4 also compares the Greedy FAN assignment in FEMOS with SBC and SLQ assignment schemes. We observe that the average network throughput performance of SBC stabilizes around 1.7×10^4 kbits/slot, which approaches to the maximum value of FEMOS. However, the average network throughput performance deteriorates severely under SLQ assignment scheme for any V. The average service delay in both SBC and SLQ assignment schemes are much larger than that of Greedy assignment in FEMOS when $V < 2000$. Specifically, when $V = 1$, the average delay in FEMOS approaches to zero, but they are $5s$ and $45s$ for SLQ and SBC schemes respectively. Those comparisons demonstrate the advantages of the Greedy in FEMOS.

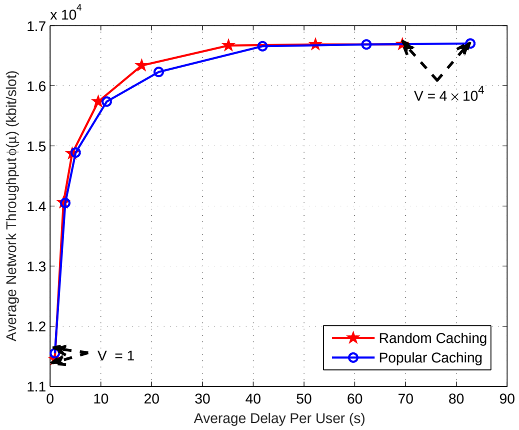
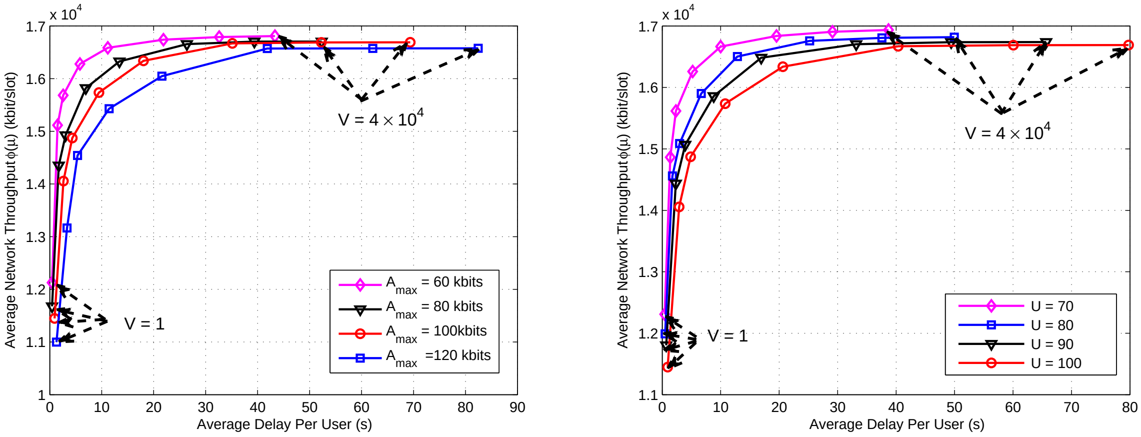


Fig. 5: Network throughput-delay tradeoff for FEMOS under different caching strategies.

B. *FEMOS's Performance under Different Caching Strategies*

In Fig. 5, we investigate the impact of different caching strategies on the network throughput-delay tradeoff performance of the proposed FEMOS algorithm. We compare the following two caching strategies:

- Random Caching: the random caching strategy, where each FAN caches $\mathcal{N} = 500$ files randomly from the file library $\mathcal{F}$.



(a) Throughput-delay tradeoff under different workload, $U = 100$.

(b) Throughput-delay tradeoff under different UT amount, $A_{\max} = 100$ kbits.

Fig. 6: Network throughput-delay tradeoff for FEMOS under different parameters.

- Popular Caching: the popular caching strategy is a heuristic caching strategy, where each FAN caches the most $\mathcal{N} = 500$ popular files [33].

In general, we can see that under both the random and the popular caching strategies, the service delay increases as network throughput increases, which indicates that a proper V should be chosen to balance the two objects. In addition, we can observe that the FEMOS algorithm with random caching strategy outperforms that with the popular caching strategy, in terms of the average delay performance. Specifically, the FEMOS algorithm with random caching can reduce service delay roughly 16% over the popular caching for the V value of 2×10^4.

C. FEMOS's Performance under Different Parameters

In Fig. 6, we further explore the relationship between network throughput and average service delay of FEMOS algorithm regarding to different workloads ($A_{\max}$) and different UT amount (U). Specifically, in Fig. 6(a) we explore the average network throughput versus average service delay in terms of different amount of workloads ($A_{\max} = 60$ kbits, 80 kbits, 100 kbits and 120 kbits, respectively). The random caching strategy is adopted here. It can be observed that for a given control parameter V, the proposed FEMOS algorithm with lower workload obtains better network throughput performance, and experiences shorter service delay. For example, when $V = 2000$, the average delay per UT experienced in FEMOS with $A_{\max} = 120$ kbits is 3.2 s, which is about twice larger than that with $A_{\max} = 60$ kbits. The average network throughput obtained by FEMOS with $A_{\max} = 60$ kbits is 1.512×10^4 kbits/slot when $V = 2000$, while it is 1.317×10^4 kbits/slot obtained by FEMOS with $A_{\max} = 120$ kbits. The reason behind these observations is, according to (25) and (26) in *Theorem 1*, both the lower bound of the average network throughput ϕ_{av}^{FEMOS} and the upper bound of average service delay $\Lambda_{av}^{\text{FEMOS}}$ are determined by parameters V and $\mathcal{K}$, where $\mathcal{K} = \frac{|\mathcal{U}|}{2}(\mu_{\max}^2 + A_{\max}^2)$. Larger workload leads to larger $\mathcal{K}$, and thus, a smaller average network throughput and a longer average service delay are obtained.

In Fig. 6(b), we show the impacts of UT amount U on the network throughput-delay tradeoff performance of FEMOS algorithm. Similar phenomenon exists for the curves with different workload. Network throughput-delay tradeoff also exists under different UT amount. Besides, it can be observed that by increasing U, the average service delay increases and the average network throughput decreases for a given V. This is also consistent with *Theorem 1* that the UT amount U influences the value of parameter $\mathcal{K}$, and thus impacts both throughput and service delay performance. Intuitively, in this case along with different workload scenarios above, the FCN has to fully utilize the network resources to serve UTs' traffic demand, and either high workload or large UT amount certainly will deteriorate both network throughput and delay performance.

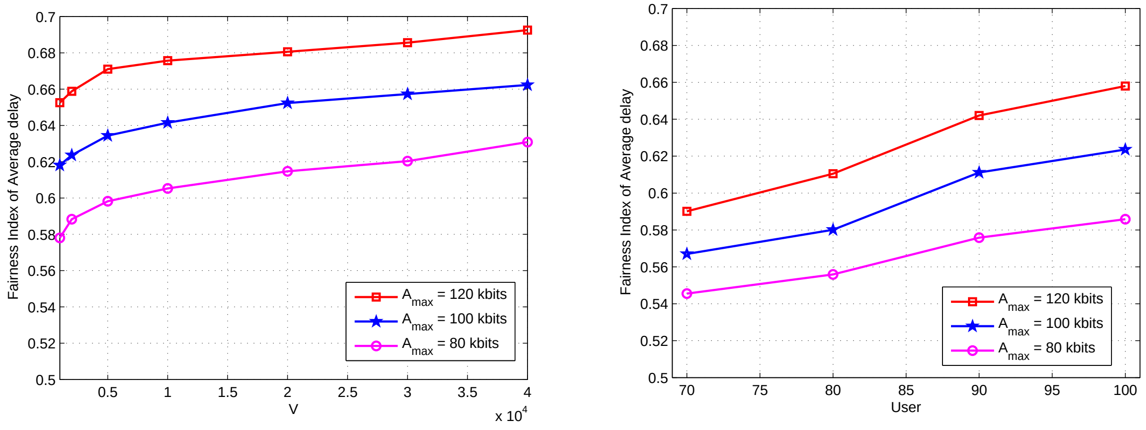
D. Fairness Comparison for FEMOS

Each UT in the network expects to gain the resource as well as the QoE fairly. Thus fairness among the UTs is also a significant indicator to characterize a network performance [34]. In this part, we reveal the fairness among the UTs in FEMOS algorithm. As to UTs, the major fairness concern is the experienced average service delay for each of them. We adopt the fairness index of average delay defined in [35], which is given by

$$F_d = \frac{\left(\sum_{u=1}^{|\mathcal{U}|} \Lambda_u\right)^2}{|\mathcal{U}| \sum_{u=1}^{|\mathcal{U}|} \Lambda_u^2}, \tag{28}$$

where Λ_u is the average service delay experienced by UT u, $\Lambda_u = \overline{Q_u^{\text{sum}}}/\lambda_u$.

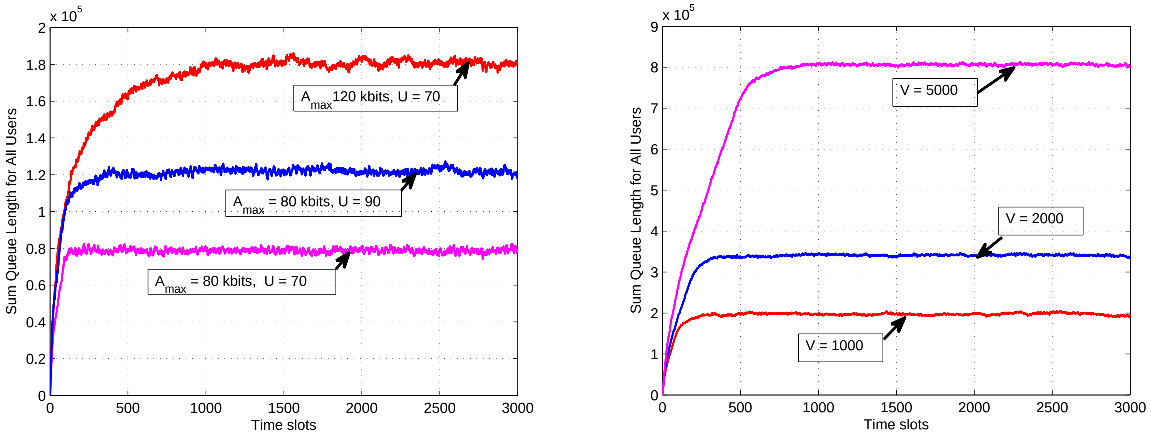
Fig. 7(a) depicts the fairness index of average delay versus the control parameter V. On the whole, the fairness index slightly increases as V increases, which reveals that the delay fairness among the UTs can be improved by increasing V, at expense of large delay per UT experienced. For instance, with



(a) Fairness Index vs. V, U = 100.

(b) Fairness Index vs. UT Amount, V = 2000.

Fig. 7: Fairness index of average delay for FEMOS.



(a) Queue backlog vs. t, V = 2000.

(b) Queue backlog vs. t, $A_{\max}$ = 120 kbits, U = 100.

Fig. 8: Evolution of sum queue length of request buffers under different parameters.

$A_{\max}$ = 100 kbits and U = 100, the fairness index increases from 0.62 to 0.66. In addition, with a given control parameter V and the UT amount U = 100, it can also be observed that the service delay fairness index as $A_{\max}$ increases, which fits the intuition that under high workload, the FCN needs to make more efforts to balance the queue length among the UTs through dynamically adjustment of UT-FAN association. Fig. 7(b) investigates the impacts of UT amount on service delay fairness index. It can be observed that the fairness index increases with the increase of UT amount U. To be more specific, the fairness index increases from 0.59 to 0.66 when U increases from 70 to 100, with the given workload $A_{\max}$ = 120 kbits and control parameter V = 2000.

E. Convergence Time Evaluation of FEMOS

Finally, we conduct numerical experiments regarding to the convergence time under the proposed FEMOS algorithm by tracing the sum queue length of request buffers with different workloads, UT amount and V. Fig. 8(a) depicts the evolution of sum queue length of request buffers under different workload $A_{\max}$ and UT amount U. We can see that all queue backlogs increase at the beginning and are stable eventually for the simulated three cases. Besides, we see that either large UT amount U or high workload $A_{\max}$ leads to a longer convergence time. Despite that, queue backlogs for all three cases are able to be stabilized within approximately 1000 time slots, which reveals that the proposed FEMOS algorithm is adaptable to the scenarios with either reasonable amount of UTs or workload.

In Fig. 8(b), We illustrate the evolution of network queue backlogs under different control parameter V. We observe that the stable queue backlogs increase as V increases, which again confirms that V indeed affects the service delay. Besides, network with large V requires more time slots to be stabilized. Thus, a proper V needs to be chosen in practice.

VII. Conclusions

In this paper, we investigated the online multi-tier operations scheduling in fog-enabled network architecture with heterogeneous node capabilities and dynamic wireless network conditions. A low-complexity online algorithm based on Lyapunov optimization was proposed, which achieves at least $\frac{1}{2}$ of the optimal value. Performance analysis, as well as simulations, explicitly characterized the tradeoffs between average network throughput and service delay, and confirmed the benefit of centralized assignment of access node and dynamic online bandwidth allocation. For future work, we would like to investigate the proactive FAN assignment and resource management problem given the availability of predictive information about UT behaviors, content requests, traffic distributions, fading channel statistics, and network dynamics.

References

[1] S. Zhao, Z. Shao, H. Qian, and Y. Yang, "Online User-AP association with predictive scheduling in wireless caching networks," in *proceedings of 2017 IEEE Global communication conference(GLOBECOM)*, December 2017.

[2] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020*, 2016.

[3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[4] Y. Shi, J. Zhang, and K. B. Letaief, "Group sparse beamforming for green cloud-ran," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2809–2823, 2014.

[5] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 77–83, August 2016.

[6] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, and X. Shen, "An efficient and fine-grained big data access control scheme with privacy-preserving policy," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 563–571, 2017.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *proceedings of the 1st edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[8] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, October 2014.

[9] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, December 2016.

[10] N. Chen, Y. Yang, T. Zhang, X. Luo, and J. Zao, "FA^{2}ST: Fog as a service technology," *under review*, May 2017.

[11] P. Yang, N. Zhang, S. Zhang, K. Yang, L. Yu, and X. Shen, "Identifying the most valuable workers in fog-assisted spatial crowdsourcing," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1193–1203, 2017.

[12] E. BaştuÇğ, M. Bennis, M. Kountouris, and M. Debbah, "Cache-enabled small cell networks: Modeling and tradeoffs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 41, 2015.

[13] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, "Distributed caching for data dissemination in the downlink of heterogeneous networks," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3553–3568, 2015.

[14] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in fog-rans: From centralized to distributed algorithms," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7039–7051, 2017.

[15] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, 2017.

[16] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, June 2017.

[17] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[18] Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, and H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Network*, vol. 31, no. 1, pp. 52–58, 2017.

[19] A.-C. Pang, W.-H. Chung, T.-C. Chiu, and J. Zhang, "Latency-driven cooperative task computing in multi-user fog-radio access networks," in *proceeding of 2017 IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 615–624.

[20] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of YouTube network traffic at a campus network–measurements, models, and implications," *Computer Networks*, vol. 53, no. 4, pp. 501–514, 2009.

[21] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[22] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2619–2692, 1998.

[23] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtilä, "IEEE 802.11 ac: Enhancements for very high throughput WLANs," in *proceeding of the 22nd IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2011, pp. 849–853.

[24] L. Huang, S. Zhang, M. Chen, and X. Liu, "When backpressure meets predictive scheduling," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2237–2250, August 2016.

[25] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[26] H. Ahlehagh and S. Dey, "Video caching in radio access network: impact on delay and capacity," in *proceeding of 2012 IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 2276–2281.

[27] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[28] J. Edmonds, "Matroids and the greedy algorithm," *Mathematical programming*, vol. 1, no. 1, pp. 127–136, 1971.

[29] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, April 2006.

[30] L. Georgiadis, M. J. Neely, L. Tassiulas *et al.*, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends® in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[31] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Scaling behavior for device-to-device communications with distributed caching," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4286–4298, July 2014.

[32] N. Khan and C. Oestges, "Impact of transmit antenna beamwidth for fixed relay links using ray-tracing and winner ii channel models," in *proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*, 2011, pp. 2938–2941.

[33] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 5, pp. 1444–1462, 2014.

[34] H. Shi, R. V. Prasad, E. Onur, and I. Niemegeers, "Fairness in wireless networks: Issues, measures and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 5–24, 2014.

[35] Y. Yang and T. S. P. Yum, "Multicode multirate compact assignment of ovsf codes for qos differentiated terminals," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 6, pp. 2114–2124, November 2005.