

A Flow Rule Timeout Assignment Algorithm for SDN-Assisted Network MIMO Systems

Rui Zou Wenye Wang

Department of Electrical and Computer Engineering

North Carolina State University, Raleigh, NC 27606

Email: {rzou, wwang}@ncsu.edu

Abstract—Network Multiple Input Multiple Output (MIMO) is able to increase spectrum efficiency and mitigate inter-cell interference. These benefits mainly result from one salient feature of network MIMO, the centralized control enabling coordinated scheduling. As Software Defined Networking (SDN) provides the innate centralized control, we propose a new cellular system architecture featuring SDN network MIMO. In this setting, the coordination of network MIMO downlink transmission is achieved in the SDN fashion. Two performance metrics, Cluster Average flow rule Storage Load (CASL) and Cluster Average flow table Miss Rate (CAMR) of Base Stations (BSs) are solved based on a Multivariate Markov Chain (MMC) model for flow rule dynamics of the BSs in a cluster. According to the analysis, we design an algorithm for the controller to assign flow rule timeout values to minimize the CASL for a given CAMR. Simulation results are presented to show the accuracy of the analysis, and the insight into the trade-off between the two metrics.

I. INTRODUCTION

Compared to former generations, cellular wireless systems are providing higher data rate which translates to more efficient radio spectrum usage as exploitable spectrum becoming scarce [1]. To increase spectrum efficiency, many technologies require certain levels of cooperation among nearby cells, such as Coordinated Multi-Point (CoMP) [2], Heterogeneous Networks (HetNet) [3], and network MIMO. Among these technologies, network MIMO [4] downlink transmission demands even tighter cooperation of cells to the extent that a central controller is needed to do MIMO precoding for all the cells. Though there exist works on various aspects of network MIMO, how to achieve the centralized control of cells in a cluster has not received adequate research attention. In [5], simulations show how the size of a coordinated cluster affects throughput in network MIMO system. The authors of [6] discover that different levels of BS cooperation result in different capacity improvements. The quantitative analysis of the benefits brought by cell cooperation is conducted in [7] under ideal conditions. In these works [4]-[7], it is either explicitly or implicitly assumed that the centralized control of BSs is available to facilitate the downlink MIMO broadcast of network MIMO, but how to centrally control network MIMO BSs is left as a future research direction by [4].

In terms of centralized control, it is an innate feature of SDN where controllers inform switches of forwarding interface of each flow using flow rule replies upon receiving the requests, which has been shown to be successful in production networks [8]. Thus, it is worthwhile to explore *what the network MIMO*

cellular system architecture will be if the SDN centralized control is adopted, and how to model the flow rule dynamics and characterize the SDN performance in this environment.

To study the performance of SDN in network MIMO analytically, two challenges need to be conquered. Firstly, the model needs to capture the SDN flow rule dynamics, such as requests, replies, usage, and storage of flow rules, which is the key difference between SDN and traditional routing. This is a challenge because classical tools for network performance studies, such as network calculus and queueing theory, cannot be used to study metrics of flow rule dynamics. Network calculus is applied to analyze delay bounds, queue length of SDN switches and controllers in [9]. However, these performance metrics are not related to SDN flow rules. The authors of [10] analyze the queue length of SDN switches using M/M/1 queue. This model is achieved by abstracting the dynamics of SDN flow rule into a constant flow table miss rate which is considered to be a fixed measurement value. This modeling technique is also used in [11] and [12] for SDN performance studies to bring the system model into a queueing framework, but the flow rule related parameters cannot be captured. Secondly, the network MIMO setting changes the common hardware constraint assumptions of SDN. Regarding the flow table size, many SDN based applications consider the flow rule storage load to be a physical constraint because the Ternary Content Addressable Memory (TCAM) for flow rule storage and lookup is space limited, costly and energy hungry [13]. For the flow rule reply from the controller, the usual assumption is that instantaneous flow rule feedback and installation are reasonable with a powerful controller, sufficient bandwidth, and infrequent flow rule requests such as in cache system [14]. In network MIMO, however, these two popular assumptions are reversed. Though TCAM has limited space, it is still enough for SDN network MIMO flow rules that time out fast because they are dependent on channel states which are ephemeral. SDN network MIMO controller cannot instantaneously reply to flow rule requests because scheduling tasks for a cluster of BSs are nontrivial for the controller in terms of processing [15] and backhaul capability [4].

In order to address the challenges, we model the flow rule dynamics using MMC [16]. Based on the transition probabilities of the MMC, recursive functions are derived to solve two metrics, CAMR and CASL. They characterize two important aspects of flow rule dynamics, the flow table miss

rate and the flow rule storage load in the SDN network MIMO setting respectively. Equipped with the solutions, an algorithm for the flow rule timeout assignment is proposed to achieve the minimum CASL for a target CAMR. The performance metrics and the algorithm are meaningful for the planning and the deployment of SDN network MIMO systems and the design of the corresponding hardware, such as the processing capability of the controller and the flow table size of BSs.

The rest of the paper is organized as follows. Section II describes the system model, including the architecture, the network components, the downlink data transmission steps, and the modeling of the flow rule dynamics for the downlink data transmission in the access network. In Section III, two performance metrics CAMR and CASL are defined, and then solved based on the recursive functions derived from the MMC model. Evaluations of the analysis and the insight into the trade-off between the two performance metrics are presented in Section IV, and the paper is concluded in Section V.

II. SYSTEM MODEL

A. SDN network MIMO architecture for cellular system

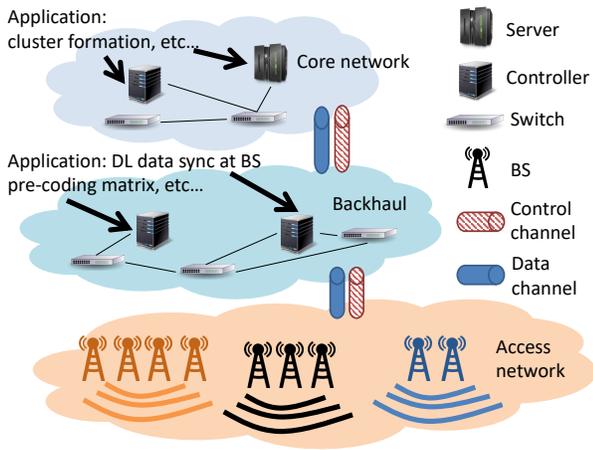


Fig. 1. Cellular system architecture with SDN network MIMO

The cellular system architecture featuring SDN network MIMO is shown in Fig. 1. It is an SDN based cellular network where nearby cells form centrally controlled clusters. The system adopts the SDN paradigm in the core network, the backhaul, and the access network where the control plane and the forwarding plane are separated. Though this is similar to the SDN cellular network architectures in the existing literature, such as the one proposed in [17], special controller applications and servers for network MIMO are required in the core network and the backhaul to accommodate network MIMO in the access network. In the core network, controller applications and special servers supporting high level network MIMO functions are developed, such as forming and maintaining the clusters. In the backhaul, controllers are equipped with applications for switches to assure data synchronization at BSs, and the applications for BS scheduling, including network MIMO precoding matrix generation.

B. Components in the network

This subsection describes the network components and their functions directly related to the analysis of the flow rule dynamics in the access network of SDN network MIMO.

1) *Cluster formation server*: In the core network, there are servers responsible for the formation and the maintenance of the SDN network MIMO cluster. We assume that a server configures the cluster of interest to be formed by one controller C , and N cells, $\{c_i | i \in [1, N]\}$ each of which serves $m_i \in \mathbb{N}^+$ users, adding to $\sum_{i=1}^N m_i = M$ users in total. Another assumption is that the cluster changes very slowly so the formation is considered to be time invariant.

2) *Network MIMO cluster controller*: The controller C processes the flow rule requests from all the N BSs in the cluster of interest. The flow rule replies indicate how to transmit the downlink data, i.e. the network MIMO precoding matrix and the other physical layer resources. As the flow rules incorporate the BS scheduling information, they are generated and configured to be effective in units of scheduling intervals, the period of time for the controller to make flow rule replies. For flow rule i , the timeout parameter is T_i , meaning that flow rule i will become ineffective after T_i number of scheduling intervals. In each scheduling interval, the controller C can only handle up to H flow rule requests. If the number of BS requesting flow rules in a scheduling interval is greater than H , the controller C randomly replies to H requests.

3) *Mobile user*: Each of the M users has certain probability of receiving downlink data in every scheduling interval. Let $A_i(t)$ be the event that user i has downlink data arrival at the scheduling interval t , and $A_i(t)$ are mutually independent among all i and t . As the flow rules specify the network MIMO precoding matrix that are different for every combination of active users, M users map to $2^M - 1$ flow rules.

4) *SDN network MIMO BS*: When transmitting downlink data in a scheduling interval, BSs first look for the corresponding flow rule locally. If the flow rule is still effective in the flow rule table of BSs, they transmit the downlink data using the locally available flow rule; otherwise, the BSs will request the flow rule from the controller C . The freshness of the flow rules stored in all the BSs are defined by the random process

$$F(t) = \{f_{ij}(t)\}_{(2^M - 1) \times N}, \quad (1)$$

where $f_{ij}(t) \in [0, T_i]$ is the number of scheduling intervals for the flow rule i to remain effective in BS j . Each row i corresponds to a flow rule mapped from the combination of mobile devices that are active at a scheduling interval with probability p_i . $f_{ij}(t) = 0$ means that the flow rule is not installed or has become ineffective due to timeout. When a BS b_j receives a flow rule for user combination i from the controller with a timeout parameter T_i at scheduling interval t , then $f_{ij}(t) = T_i$. After that, the f_{ij} value decreases by one every scheduling interval until it freezes at zero.

C. Downlink data transmission process in access network

The downlink data transmission steps in the access network of SDN network MIMO are illustrated in Fig. 2 which shows

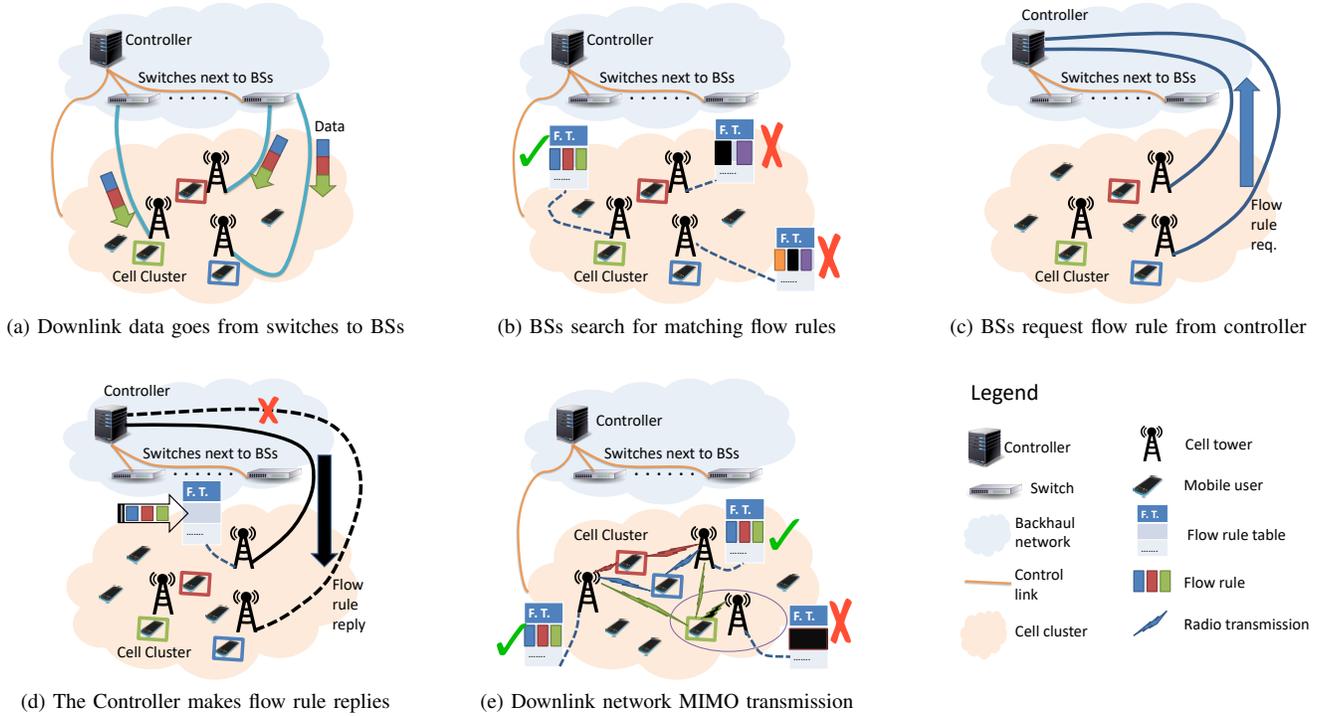


Fig. 2. Downlink data transmission steps

a small scale example of one SDN network MIMO cluster comprised of three cells. In this example, three adjacent BSs are grouped into a cluster governed by a controller. At one scheduling interval, there are downlink data to three users, the mobiles enclosed in color rectangles in Fig. 2. The data are transmitted from the switches in the backhaul to all the BSs in the cluster as shown in Fig. 2(a) to ensure that all the BSs have the same data so they can collaborate on the downlink MIMO transmission. Then, as depicted in Fig. 2(b), the BSs look for the corresponding flow rule that matches the three users. The BSs finding the flow rule stored in their own flow rule tables transmit the data in the next scheduling interval, but the BSs that do not have the flow rule at hand need to request the flow rule from the controller, which is shown in Fig. 2(c). Table miss events happen at the BSs without locally available flow rules. Though two flow rule requests are sent to the controller, only one reply is made before the end of the current scheduling interval as shown in Fig. 2(d), because the processing capacity of the controller is limited.

D. MMC model of the flow rule timeout dynamics

To analyze CAMR and CASL which are defined in the next section, the random process $\mathcal{S}_i(t)$ is defined based on the flow rule freshness process $F(t)$ in Eq. (1). As we will see, the two metrics CAMR and CASL depend on the steady state expectation of one variate, $S_i^0(t)$ in $\mathcal{S}_i(t)$. Recursive functions are derived from the transition probabilities of $\mathcal{S}_i(t)$, resulting in a more convenient way to solve the two metrics than finding the steady state distributions for $\mathcal{S}_i(t)$ whose state space is so large that the steady state distribution defies analysis. Define

the multivariate random process $\mathcal{S}_i(t) = (S_i^0(t), \dots, S_i^{T_i}(t))$ where each element $S_i^k(t)$ is the number of BSs with $f_{ij}(t) = k$ at scheduling interval t . $S_i^k(t) = \sum_{j=1}^N \mathbb{1}_{\{f_{ij}(t)=k\}}$ for $k \in [0, T_i]$, and $\mathbb{1}_{\{\cdot\}}$ is the indicator function.

For flow rule i , the transition probabilities of $\mathcal{S}_i(t)$ are

$$S_i^0(t+1) = \begin{cases} S_i^1(t) + (S_i^0(t) - H)^+, & \text{w.p. } p_i, \\ S_i^1(t) + S_i^0(t), & \text{w.p. } 1 - p_i, \end{cases} \quad (2)$$

$$S_i^j(t+1) = S_i^{j+1}(t), \quad j \in [1, T_i - 1], \quad (3)$$

$$S_i^{T_i}(t+1) = \begin{cases} H - (H - S_i^0(t))^+, & \text{w.p. } p_i, \\ 0, & \text{w.p. } 1 - p_i. \end{cases} \quad (4)$$

The function $(x)^+ = \max\{x, 0\}$. As shown in Eq. (2)-(4), $\mathcal{S}_i(t)$ is Markovian because the future state of any variate is independent of the history states given the current states of all the variates, so $\mathcal{S}_i(t)$ is an MMC. Define $R(t)$ as the total number of flow rule requests from all the BSs in the cluster at t th scheduling interval, so it can be written as

$$R(t) = S_i^0(t), \quad \text{w.p. } p_i, \quad (5)$$

the number of BSs that have no effective flow rule i . Since the MMC $\mathcal{S}_i(t)$ applies to any flow rule i , the index i will be suppressed when no confusion is caused.

III. PERFORMANCE METRIC STUDY

In this section, two performance metrics in SDN network MIMO, CAMR and CASL, are first defined, and then solved based on the recursive functions obtained from the transition probabilities of $\mathcal{S}(t)$. It is also shown that the solutions derived

from the recursive functions are equal to the results achieved by solving the steady state distribution of $\mathbf{S}(t)$.

A. Definitions of the metrics

1) *Definition of CAMR*: Flow table miss rate appears in many SDN performance analysis papers as a measured value, and it is considered to be very influential on the performance of SDN switches [11][12]. In SDN network MIMO setting, the flow table miss rate of the BSs in a cluster is characterized by CAMR which is denoted as ρ and defined as follows.

$$\rho = \frac{\lim_{t \rightarrow \infty} \mathbb{E}(R(t))}{N} = \frac{\sum_{i=1}^{2^M-1} p_i \mathbb{E}(S_i^0(\infty))}{N}. \quad (6)$$

$\mathbb{E}(S_i^0(\infty))$ denotes $\lim_{t \rightarrow \infty} \mathbb{E}(S_i^0(t))$ and CAMR is the ratio between the expectation of the number of BS with no matching flow rule in the steady state and the total number of BS.

2) *Definition of CASL*: Currently, the storage of SDN flow rules in switches is limited by the hardware capability and considered as a major design constraint [13][18], so it is meaningful to analyze the flow rule storage load on the BSs in SDN network MIMO. Define CASL, denoted by ω , as the average flow rule storage load of all the N BSs in the cluster in the steady state of $\mathbf{S}(t)$, so it can be written as

$$\omega = \sum_{i=1}^{2^M-1} \frac{N - \mathbb{E}(S_i^0(\infty))}{N}, \quad (7)$$

which is the expectation of the number of effective flow rules in all the BSs in a cluster in the steady state of $\mathbf{S}(t)$.

B. Solving the metrics

The key to solving the metrics is to find $\mathbb{E}(S^0(\infty))$, the expectation of the number of BSs with no matching flow rule in the steady state of $\mathbf{S}(t)$. The existence of the steady state of the MMC is guaranteed by that fact that it has finite state space, and it is irreducible and aperiodic. The proof of the existence of the steady state is not presented in the paper due to space limitation. Since the metrics of concern depend only on the average value of one variate of the MMC in steady state, we derive a set of recursive functions to solve $\mathbb{E}(S^0(\infty))$. The following group of recursive functions can be obtained from Eq. (3) and Eq. (4) by taking expectations on both sides.

$$\mathbb{E}(S^j(t+1)) = \mathbb{E}(S^{j+1}(t)), \quad j \in [1, T_i - 1]. \quad (8)$$

$$\mathbb{E}(S^T(t+1)) = p(H - (1 - r(t))(H - \mathbb{E}(S^0(t)))). \quad (9)$$

In (9), $r(t) = \mathbb{P}(S^0(t) > H)$. As t goes to infinity, all the expectations exist and converge, and so does $r(t)$. Taking the limit of Eq. (8) and Eq. (9) as t goes to infinity gives

$$\mathbb{E}(S^j(\infty)) = \mathbb{E}(S^{j+1}(\infty)), \quad 1 \leq j \leq T - 1, \quad (10)$$

$$\mathbb{E}(S^T(\infty)) = p(H - (1 - r)(H - \mathbb{E}(S^0(\infty))), \quad (11)$$

where $r = \lim_{t \rightarrow \infty} r(t)$. Together with the condition $\sum_{j=0}^T \mathbb{E}(S^j(\infty)) = N$, we can solve the equations of steady state expectations in Eq. (10) and Eq. (11), and $\mathbb{E}(S^0(\infty))$ is

$$\mathbb{E}(S^0(\infty)) = \frac{N - rpHT}{1 + p(1 - r)T}. \quad (12)$$

To show that $\mathbb{E}(S^0(\infty))$ is equal to the expectation of $S^0(t)$ when t approaches infinity, suppose that $\mathbf{S}^* = \lim_{t \rightarrow \infty} \mathbb{E}(\mathbf{S}(t)) = \mathbb{E}(\mathbf{S}(k))$ and $\mathbb{E}(\mathbf{S}_{n+1} | \mathbf{S}_n = \mathbf{a}) = h(\mathbf{a})$, so $\mathbb{E}(S^0(\infty))$ is in the solution of $h(\mathbf{a}) = \mathbf{a}$ where \mathbf{a} is a constant vector. Thus, \mathbf{S}^* can be written as follows.

$$\mathbf{S}^* = \mathbb{E}(\mathbf{S}(k+1)) = \mathbb{E}(\mathbb{E}(\mathbf{S}(k+1) | \mathbf{S}(k))) = \mathbb{E}(h(\mathbf{S}(k))). \quad (13)$$

In the steady state of $\mathbf{S}(t)$, the function $h(\cdot)$ is linear according to (10) and (11), so we can change the order of the expectation operator and the linear function, which results in

$$\mathbf{S}^* = \mathbb{E}(h(\mathbf{S}(k))) = h(\mathbb{E}(\mathbf{S}(k))) = h(\mathbf{S}^*). \quad (14)$$

According to Eq. (14), the solution of $h(\mathbf{a}) = \mathbf{a}$ equals to \mathbf{S}^* , justifying that the results obtained using the recursive functions are the same with the expectations of steady state distribution.

Thus, CAMR can be further derived as

$$\rho = \frac{1}{N} \sum_{i=1}^{2^M-1} \frac{p_i(N - r_i p_i H T_i)}{1 + p_i(1 - r_i)T_i}, \quad (15)$$

and CASL is obtained by plugging (12) in (7)

$$\omega = \frac{1}{N} \sum_{i=1}^{2^M-1} \frac{N p_i(1 - r_i)T_i + r_i p_i H T_i}{1 + p_i(1 - r_i)T_i}. \quad (16)$$

Note that $r \in [0, 1]$ is an unknown probability, so the solutions for $\mathbb{E}(S^0(\infty))$, CAMR, and CASL involving r are ranges of values depending on r . The range of r can be further narrowed down by applying Markov inequality,

$$r = \mathbb{P}(S^0(\infty) > H) \leq \frac{\mathbb{E}(S^0(\infty))}{H}. \quad (17)$$

The accuracy of the results is discussed in section IV.

C. Algorithm for timeout assignment

Assigning timeout values for flow rules is part of the controller design, and one important goal is to achieve the target table miss rate with as low flow rule storage as possible. Using the above results, a greedy algorithm can be designed to achieve the target CAMR with the lowest CASL. As shown in Alg.1, the inputs of the algorithm are the probabilities of the flow rule usage at a scheduling interval, the bounds of timeout values, the target CAMR, and the output is the timeout assignment. It is worth noting that the while loop starting from line eight can be regarded as a linear search which can be easily changed to an equivalent binary search for lower time complexity to adapt for large inputs. It is shown here as a linear search for easy presentation. If binary search algorithm is used, the time complexity of the algorithm is $O(M \log(|\{TO_{min}, \dots, TO_{max}\}|))$. Since the complexity increases linearly with the number of users in the cluster, the scalability of the algorithm is demonstrated thereby.

IV. EVALUATION

In this section, simulation results are presented to demonstrate the accuracy of the analysis and the insight into the trade-off between the two performance metrics.

Algorithm 1 Achieve target CAMR with the lowest CASL

Input: $p[]$, TO_{min} , TO_{max} , $targetMissRate$,

Output: $T[]$

```

1:  $p[] = \text{sort } p[]$  from large to small
2:  $T[] = [TO_{max}, \dots, TO_{max}]$ 
3:  $missRate = \text{calculate using (12)-(15)}$ 
4: if ( $missRate > targetRate$ ) then
5:   no such  $T$  assignment exists, return empty
6: else
7:    $i = \text{length of } T[]$ 
8:   while  $i \neq 0$  do
9:     while  $T[i] > TO_{min}$  do
10:       $missRate = \text{calculate using (12)-(15)}$ 
11:      if  $missRate \geq targetRate$  then
12:        break
13:      end if
14:       $T[i] = T[i] - 1$ 
15:    end while
16:     $missRate = \text{calculate using (12)-(15)}$ 
17:    if  $missRate \geq targetRate$  then
18:      break
19:    end if
20:     $i = i - 1$ 
21:  end while
22: end if
23: return  $T[]$ 

```

TABLE I

PARAMETERS OF THE THREE SCENARIOS IN ALL THE CASES FOR FIG. 3

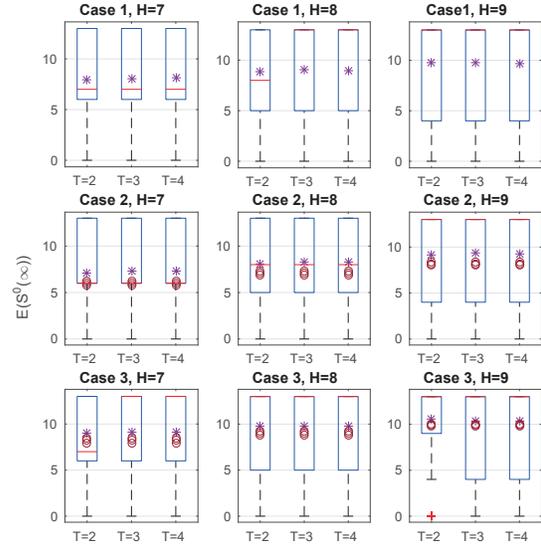
Scenario	H	N	T
1	7	13	2, 3, 4
2	8	13	2, 3, 4
3	9	13	2, 3, 4

A. Accuracy of $\mathbb{E}(S^0(t))$

The accuracy of the analysis for $\mathbb{E}(S^0(t))$ is examined because it is an important intermediate result on which the two metrics depend and its solution given by Eq. (12) and (17) is a range of values. Depending on the form of Eq. (12), three cases are considered based on the value of $\Delta = N - H - pHT$. When $\Delta = 0$, $\mathbb{E}(S^0(\infty))$ equals to H and does not depend on r ; when $\Delta < 0$, the range of Eq. (12) is restricted by Eq. (17); when $\Delta > 0$, Eq. (17) has no effect on the range of $\mathbb{E}(S^0(t))$ given by Eq. (12) when r is in $[0, 1]$. For each case, simulations are conducted in three scenarios and the parameters are summarized in table I, where the parameters remain the same for comparability in all the cases except p for which the computation is explained in each case below.

1) *Case 1*, $\Delta = 0$: In this case, $\mathbb{E}(S^0(\infty)) = H$ according to Eq. (12), so r does not affect the result and the p values are computed as $(N - H)/HT$ in this case.

The simulation results for the three scenarios are illustrated in the three figures in the first row of Fig. 3. The simulation results within the 5th and the 95th percentiles are enclosed in the blue boxes. The red bars show the medians and black


 Fig. 3. Simulation and analysis results of $\mathbb{E}(S^0(\infty))$ in 3 cases

bars are the extreme values. Red crosses represent the outliers excluded from the statistics, and the purple stars are the mean values. As we can see, the simulation mean for three scenarios stays almost the same regardless of the changes of p and T as long as $\Delta = 0$, which is in accordance with our analysis. The analysis results of $\mathbb{E}(S^0(\infty))$ are not plotted in the figures because they are equal to H in this case. They are consistently below the simulation results by no more than one.

2) *Case 2*, $\Delta < 0$: In this case, the Markov inequality constraint in Eq. (17) further restricts the range of $\mathbb{E}(S^0(\infty))$. The p values here are achieved by increasing the p values in case 1 by 25 percent to bring Δ below zero. The simulation results and the analysis are shown in the second row of Fig. 3. The ranges of analysis results are represented by three brown circles indicating the bounds and the center of the range. The range is small compared to the range of simulation results and the analysis results are near the simulation mean.

3) *Case 3*, $\Delta > 0$: In this case, Eq. (17) does not restrict the solution of Eq. (12), and the p values are obtained by reducing the p values in case 1 by 25 percent to bring Δ above zero. The simulation and analysis results are shown in row three of Fig. 3. Similar to the previous case, the ranges of the analysis are also small and close to the simulation results.

B. Accuracy of CAMR and CASL

In this subsection, the simulation parameters are $H = 3$, $N = 4$, $M = 5$, and the profile of user traffic arrival probability in a scheduling interval is given in the first row in table II. Timeout assignment is the same for all the $2^5 - 1 = 31$ flow rules. According to the figure on the left of Fig. 4, CAMR decreases as timeout values increases. The simulation result, shown as Sim in the legend of Fig. 4, is very close to the analysis upper bound, AUB in the legend of Fig. 4.

Presented on the right in Fig. 4, the simulation results for CASL are not far below the analysis lower bound, put as ALB in the legend in Fig. 4. Contrary to the trend of CAMR versus timeout growth, CASL increases as timeout value grows.

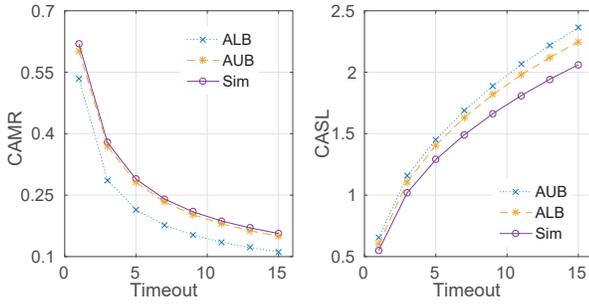


Fig. 4. Analysis and simulation results of CAMR and CASL

TABLE II
PROBABILITY PROFILES OF USER TRAFFIC ARRIVAL

Profile	p_1	p_2	p_3	p_4	p_5	σ
1	0.1833	0.5999	0.6131	0.3764	0.3987	0.1782
2	0.3230	0.1942	0.1171	0.8518	0.4784	0.2908
3	0.9434	0.7154	0.6545	0.5098	0.0922	0.3157
4	0.0912	0.9648	0.0081	0.9771	0.0018	0.5146

C. Trade-off between CAMR and CASL

When Alg.1 is adopted, the relation between CASL and CAMR is studied using simulation in which the parameters are $H = 3$, $N = 4$, $M = 5$, and four different profiles (shown in table II) of user traffic arrival probabilities in a scheduling interval are considered. As shown in Fig. 5, CAMR decreases as CASL increases and the speed of decreasing slows down as CAMR becomes small, so it is beneficial to stop trying to obtain small CAMR by increasing CASL after a certain point. This phenomenon is more obvious in the probability profiles with a large standard deviation, since profile 4 has the steepest decrease, followed by profile 3 and 2 which have smaller and similar standard deviation. The standard deviation of profile 1 is the smallest and its curve descends most mildly.

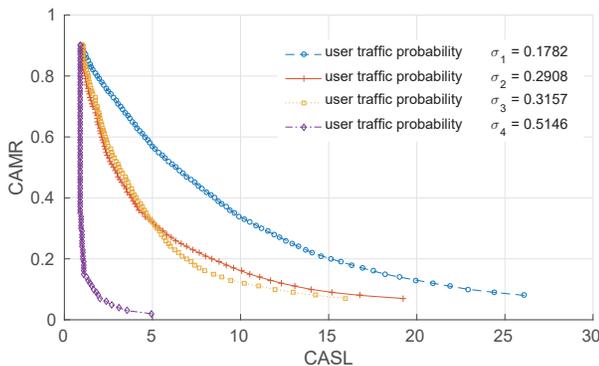


Fig. 5. CASL versus CAMR when Alg.1 is used to assign timeout

V. CONCLUSION

In this paper, we propose the SDN network MIMO architecture to facilitate the centralized control in network MIMO. The SDN flow rule dynamics is analyzed in this setting using an MMC mode based on which CASL and CAMR are defined

and solved. Observing that the solution of the two metrics requires only the expectation of one variate in the MMC in the steady state, recursive functions are derived from the transition probabilities to solve the metrics. Armed with the solutions of the metrics, an algorithm is designed for flow rule timeout assignment to minimize CASL for a given CAMR. Simulation results corroborate the accuracy of the analysis and provide insight into the trade-off between the two metrics.

REFERENCES

- [1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 6, pp. 1065–1082, 2014.
- [2] Q. Wang, D. Jiang, G. Liu, and Z. Yan, "Coordinated multiple points transmission for LTE-advanced systems," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*. IEEE, 2009, pp. 1–4.
- [3] T. Zhang, J. Zhao, L. An, and D. Liu, "Energy efficiency of base station deployment in ultra dense HetNets: A stochastic geometry analysis," *IEEE Wireless Communications Letters*, vol. 5, no. 2, pp. 184–187, 2016.
- [4] D. Gesbert, S. Hanly, H. Huang, S. S. Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1380–1408, 2010.
- [5] H. Huang, M. Trivellato, A. Hottinen, M. Shafi, P. J. Smith, and R. Valenzuela, "Increasing downlink cellular throughput with limited network MIMO coordination," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 2983–2989, June 2009.
- [6] S. Jing, D. N. Tse, J. B. Soriaga, J. Hou, J. E. Smeed, and R. Padovani, "Multicell downlink capacity with coordinated processing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2008, p. 18, 2008.
- [7] K. Hosseini, W. Yu, and R. S. Adve, "A stochastic analysis of network MIMO systems," *IEEE Trans. Signal Processing*, vol. 64, no. 16, pp. 4113–4126, 2016.
- [8] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined WAN," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 3–14.
- [9] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, and D. Simeonidou, "An analytical model for software defined networking: A network calculus-based approach," in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 1397–1402.
- [10] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an OpenFlow architecture," in *Proceedings of the 23rd international teletraffic congress*. International Teletraffic Congress, 2011, pp. 1–7.
- [11] K. Mahmood, A. Chilwan, O. Østerbø, and M. Jarschel, "Modelling of OpenFlow-based software-defined networks: the multiple node case," *IET Networks*, vol. 4, no. 5, pp. 278–284, 2015.
- [12] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of OpenFlow-based software-defined networks based on queueing model," *Computer Networks*, vol. 102, pp. 172–185, 2016.
- [13] M. Huang, W. Liang, Z. Xu, W. Xu, S. Guo, and Y. Xu, "Dynamic routing for network throughput maximization in software-defined networks," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.
- [14] D. S. Berger, P. Gland, S. Singla, and F. Ciucu, "Exact analysis of TTL cache networks," *Performance Evaluation*, vol. 79, pp. 2–23, 2014.
- [15] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On controller performance in software-defined networks," *Hot-ICE*, vol. 12, pp. 1–6, 2012.
- [16] M. K. N. Wai-Ki Ching, *Multivariate Markov Chains*. Boston, MA: Springer US, 2006, pp. 141–169.
- [17] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software defined mobile networks: concept, survey, and research directions," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 126–133, 2015.
- [18] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "On the effect of forwarding table size on SDN network utilization," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1734–1742.