Ownership-aware software-defined backhauls in next-generation cellular networks

Ownership-Aware Software-Defined Backhauls in Next-Generation Cellular Networks

Francesco Malandrino
The Hebrew University of Jerusalem
Email: francesco@cs.huji.ac.il

David Hay
The Hebrew University of Jerusalem
Email: dhay@cs.huji.ac.il

***Abstract*—Future cellular networks will be owned by multiple parties, e.g., two mobile operators, each of which controls some elements of the access and backhaul infrastructure. In this context, it is important that as much traffic as possible is processed by the same party that generates it, i.e., that the *coupling* between traffic and network ownership is maximized. Software-defined backhaul networks can attain this goal; however, existing management schemes ignore ownership altogether. We fill this gap by presenting an ownership-aware network management scheme, maximizing the amount of traffic that is processed by the same party it belongs to.**

I. Introduction

Next-generation cellular networks will differ from present-day ones in many ways, one of the most perspicuous being *ownership*. Today's networks are designed, deployed and managed by a single mobile network operator; in future networks, different elements of the access and backhaul infrastructure is expected [1] to belong to different parties.

This change is already happening, under the guise of *network sharing* [2], [3], where mobile operators integrate their access infrastructure to reduce costs, especially in sparsely-populated areas [3]. Future cellular networks are most likey to continue on this direction, becoming increasingly *virtualized* [1], [4]. Users, mobile operators and over-the-top service providers will be able to obtain each segment of the telecommunications value chain [1] (e.g., spectrum, access and backhaul infrastructure, value-added processing) from a different source. Generalizing these trends, we can envision a network where each element (e.g., base station, switch, backhaul server) belongs to a different entity, which may or may not coincide with the owner of the data that should be processed.

Managing such a complex network is possible as a software-defined network (SDN), that has easy and scalable *traffic steering* capabilities at the flow level, allowing network managers to decide how, and by whom, individual traffic flows are processed. Figure 1 depicts the structure of future backhaul networks, as envisioned in [5]: traffic originates at a base station, then traverses a sequence of switches and middleboxes, and finally reaches the Internet. In most cases, there are multiple ways to serve the same traffic flow, i.e., multiple sequences of middleboxes (and switches) it can traverse. Thus, network administrators face two challenges: first, they have to decide which sequences (corresponding to colored lines in Figure 1) to activate, subject to the limited number of sequences a switch can support; second, they have to split the traffic among active sequences. SIMPLE [6] solves both challenges, considering a single-owner network and using load-balancing as an objective.

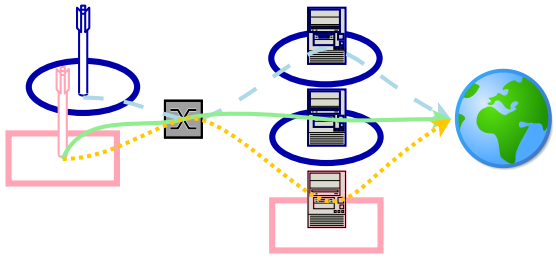


Fig. 1. A cellular network, wherein some elements belong to the blue/circle provider and others to the pink/square one. Colored lines represent physical sequences, i.e., ways in which the traffic can be served. Suppose that the switch can only support two such sequences, and the SDN controller is in charge of choosing which ones. Ownership-unaware algorithms may decide to drop the yellow, dotted sequence, thus having all the pink traffic processed by blue boxes. Ownership-aware solutions would drop the green, solid one, maximizing the amount of traffic each party can process using its own resources.

In multi-owner scenarios such as the one presented in Figure 1, steering decisions are made jointly (e.g., by a third-party entity appointed by the operators) and have a different objective, i.e., maximizing the *coupling* between the origin of traffic and the ownership of the network elements processing it. For example, in Figure 1 we would like the traffic originating from the pink base station to be processed at the pink middlebox, instead of one of the blue ones, for both technical (the pink middlebox may be better-suited to the task) and non-technical reasons.

The remainder of this paper is organized as follows. We begin by reviewing related work in Section II. Then, in Section III, we present our model of multi-owned, SDN-based cellular backhaul networks, along with three metrics formalizing the coupling concept we introduced above. In Section IV we discuss our ownership-aware offline/online decomposition strategy. We compare its performance against the optimal case and the ownership-oblivious approach taken in [6] in Section V. Finally, we conclude the paper in Section VI.

II. Related work

Our work lies between two research areas: next-generation cellular networking and software-defined networking.

Next-generation cellular networking: There is wide consensus on the fact that next-generation cellular networks will not be just a faster, higher-capacity version of present-day ones. One reason is technical: new physical-layer technologies such as massive MIMO [7] and new types of infrastructure such as millimiter-wave base stations [8] can support altogether new applications, such as proximity [9] and machine-to-machine services. The other reason is economical: network operators (which are often private companies, not public bodies) have limited resources and little interest in deploying next-generation networks, which will greatly benefit over-the-top content providers such as Google and Netflix.

One way of coping with this issue is transferring some revenue from over-the-top providers to mobile operators, either directly as in the Netflix/Comcast deal [10] or indirectly, as in Facebook Zero [11] and similar agreements by Twitter [12]. At the same time, over-the-top providers are increasingly willing to deploy their own infrastructure, as Google is doing with their Wi-Fi hotspots [13]. This will lead to increasingly heterogeneous and *virtualized* networks [1], [14], where network resources coming from different providers are viewed as a single pool, that can be managed in a centralized fashion.

We base on these works to build our system model, which aims at representing those aspects of next-generation networks that are most relevant to us – most notably, ownership issues.

Software-defined networking: As a result of the aforementioned trends, next-generation cellular networks will be complex entities, requiring fine-grained, centralized, scalable management. Software-defined networking (SDN) provides all these properties. As early as 2007, the authors of [15] shown that a single controller can manage large-scale networks, including thousands of Ethernet switches. Later works introduced the OpenFlow protocol in both wired [16] and wireless [17] SDN networks.

More recent works advocate the adoption of SDN in present- and next-generation cellular networks, both based on the 3GPP-defined evolved packet core (EPC). In particular, [5] envisions to replace the components of traditional backhaul networks – serving gateways and packet data network gateways – with *middleboxes*. Introduced in [18], middleboxes are specialized appliances, each performing a network task – e.g., firewall or intrusion detection. They run on commodity hardware, making backhaul networks cheaper and more flexible.

A convergent trend is represented by network function virtualization (NFV) [19]–[21]. The core idea is to implement middleboxes using virtual machines, thus obtaining higher performance and better scalability. SDN thus has to steer traffic between (virtual) middleboxes, also accounting for the (physical) host machine they run into.

Using SDN to steer traffic between middleboxes is not, however, an easy task. Works such as [6] aim at simplifying SDN policy enforcement, by providing a framework to map high-level steering decisions into switch-level rules. A particularly important aspect to consider is that switches can only support a limited amount of rules, determined by the size of their flow table.

We use these works, especially [5], [6] as a reference and benchmark. Essentially, our purpose is (i) to establish whether the existing schemes are fit for next-generation cellular networks as they are envisioned, and (ii) propose a way to improve their performance.

Traditional routing: In spite of the fundamental differences between the two, it is interesting to mention that traffic steering in multi-owned cellular core networks is similar, in principle, to inter-AS routing: in both cases, a set of independent entities cooperate to manage a network. Coupling, in particular, can be seen as the equivalent of the valley-free property [22] in BGP routing – we aim at avoiding to needlessly involve additional entities in the routing (and processing) of traffic.

III. System model

Our system model is summarized in Figure 2. It describes a virtualized, SDN-based cellular backhaul network such as the one depicted in Figure 1. We draw the terminology and solution concept from [6], which we also use as a benchmark to compare against.

System elements: We account for five elements in our sytem: policy chains, physical sequences, middleboxes, switches, and owners.

Policy *chains* $c \in \mathcal{C}$ are sequences of *types* of middleboxes. A flow has to go through a specific chain, e.g., "first an IDS, then a firewall, then a proxy". Notice that policy chains do not specify *which* individual boxes should be used. Network resources, i.e., middleboxes and switches, correspond to elements $b \in \mathcal{B}$ and $r \in \mathcal{R}$, respectively. We do not consider the capacity of links between switches and middleboxes; this corresponds to implicitly assuming that said links are never a bottleneck, as in [5], [6].

Physical *sequences* $s \in \mathcal{S}$ are sequences of middleboxes and switches that implement a policy chain. A policy chain can be implemented by multiple physical sequences; similarly, the

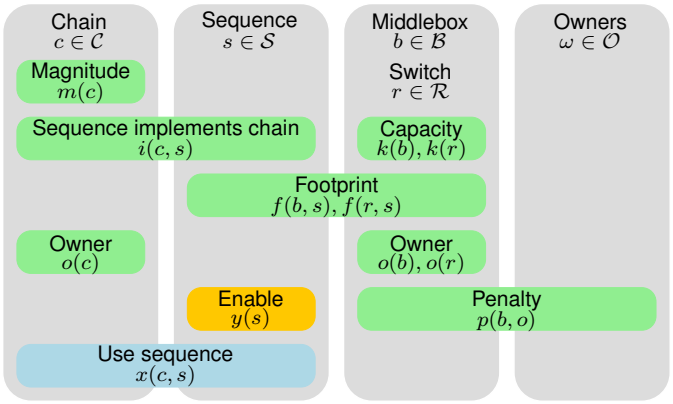


Fig. 2. System model. Gray, vertical boxes represent sets of elements our model accounts for. Green boxes represent known parameters; blue boxes represent continuous (i.e., real) variables; orange boxes represent discrete (binary, in our case) variables. Horizontal and vertical boxes cross when the parameter/variable identified by the horizontal box is indexed by the element represented by the vertical box.

same middlebox or switch can appear in multiple sequences, possibly implementing different policy chains. Each sequence implements exactly one policy chain. Physical sequences are normally computed offline, namely by brute force (as in [6]) or with more sophisticate embedding algorithms. From the viewpoint of our model, however, they are given as an input.

Owners $\omega \in \mathcal{O}$ are the parties that can own network elements.

Parameters: For each policy chain $c \in \mathcal{C}$ we know its *magnitude* $m(c) \in \mathbb{R}^+$, i.e., how much traffic will have to traverse that chain.

For each chain $c \in \mathcal{C}$ and physical sequence $s \in \mathcal{S}$ we know a binary parameter $i(c,s) \in \{0,1\}$, expressing whether sequence s *implements* chain c. Because each sequence implements exactly one chain, we have $\sum_{c\in\mathcal{C}} i(c,s) = 1, \forall s \in \mathcal{S}$. For each sequence and middlebox we know a *footprint* $f(b,s) \in \mathbb{R}^+$, i.e., how much of b's computational capabilities (e.g., CPU cycles) are used to process a unit of traffic of sequence s. We also define a footprint $f(r,s) \in \mathbb{R}$ for each switch $r \in \mathcal{R}$, representing the amount of memory needed at switch r to enable sequence s.

The total *capacity* of box $b \in \mathcal{B}$ and switch $r \in \mathcal{R}$ is expressed by parameters $k(b) \in \mathbb{R}^+$ and $k(r) \in \mathbb{N}$ respectively. Notice that by "capacity" of a middlebox we refer to its computational capabilities, while by "capacity" of a switch we indicate the size of its flow table. Indeed, as we will see later, switches and middleboxes are subject to distinct sets of constraints, accounting for their different role.

Finally, chains, middleboxes and switches have an *owner*, represented by parameters $o(c) \in \mathcal{O}$, $o(b) \in \mathcal{O}$ and $o(r) \in \mathcal{O}$ repsectively. To streamline the notation, we will also indicate with $o(s)$ the owner of a sequence $s \in \mathcal{S}$, i.e., the owner of the (single) chain it implements.

Decision variables: Similar to [6], we have two main decisions to make. The first one is which physical sequences $s \in \mathcal{S}$ to enable, and we model that through a set of binary variables $y(s) \in \{0,1\}$. The second decision is through which physical sequences, serve the traffic of each chain: this is accounted for by variables $x(c,s) \in [0,1]$, representing the fraction of traffic belonging to chain $c \in \mathcal{C}$ served through sequence $s \in \mathcal{S}$.

Constraints: To begin with, we have to ensure that all traffic is served:

$$\sum_{s\in\mathcal{S}} x(c,s) = 1, \quad \forall c \in \mathcal{C}. \tag{1}$$

Furthermore, we want to serve traffic of chain c only through sequences that (i) are enabled, and (ii) do implement c. We enforce both conditions through the following constraint:

$$x(c,s) \leq i(c,s)y(s), \quad \forall c \in \mathcal{C}, s \in \mathcal{S}. \tag{2}$$

The right-hand side of (2) is zero if sequence s is disabled, *or* if it does not implement chain c. In either case, s can serve no traffic belonging to chain c.

The second group of constraints concerns the utilization of middleboxes and switches. We have to ensure that none faces more load than its capacity allows. For switches $r \in \mathcal{R}$, we have to impose:

$$\sum_{s\in\mathcal{S}} y(s)f(r,s) \leq k(r), \quad \forall r \in \mathcal{R}. \tag{3}$$

(3) is valid if forwarding happens in a *hop-by-hop* fashion. For sake of simplicity, we leave tunnel-based optimizations [6] out of the scope of our work.

For middleboxes, we need to account for the magnitude of chains and the footprint of the sequences implementing it:

$$\sum_{c\in\mathcal{C}} m(c) \sum_{s\in\mathcal{S}} x(c,s)f(b,s) \leq k(b), \quad \forall b \in \mathcal{B}. \tag{4}$$

The left-hand side in (4) is the total load over middlebox $b \in \mathcal{B}$, given by the sum over all sequences of the product between the magnitude of the chain the sequence implements (m-value), the fraction of said load using each sequence (x-value) and the footprint of the latter over box b (f-value).

Objective: As discussed earlier, we want to maximize the coupling between the origin of traffic (i.e., the owner $o(c)$ of the chain) and the ownership $o(b)$ of the boxes processing it. Higher coupling values translate, for all parties operating the network, into technical (e.g., higher performance) and economical benefits, e.g., lower fees. Because the actual way fees will be computed will change across different scenarios, we present three definitions of coupling, and study how our ownership-aware pruning influences them.

A simple way to express such coupling is considering the fraction of total load over middleboxes that is due to traffic belonging to the same owner as the middlebox itself, i.e, the following *coupling index*:

$$\max_{x,y} \frac{\sum_{b\in\mathcal{B}} \sum_{c\in\mathcal{C}:\ o(c)=o(b)} m(c) \sum_{s\in\mathcal{S}} x(c,s)f(b,s)}{\sum_{b\in\mathcal{B}} \sum_{c\in\mathcal{C}} m(c) \sum_{s\in\mathcal{S}} x(c,s)f(b,s)}. \tag{5}$$

The maximum value that the quantity in (5) can take is one, corresponding to the case where all traffic is processed by boxes with the same owner as the traffic itself.

An alternative definition of coupling considers the *penalty* $p(b,o)$ experienced by party $o \in \mathcal{O}$ when its traffic is served by middlebox $b \in \mathcal{B}$. Such a penalty may be monetary (i.e., a fee) or technical, e.g., slower processing or the need to synchronize between different servers. Typically (but not necessarily), we have no penalty associated to one's boxes, i.e., $p(b,o(b)) = 0$. Our objective is to minimize the total penalties experienced by all parties, i.e.,

$$\min_{x,y} \sum_{b\in\mathcal{B}} \sum_{c\in\mathcal{C}} p(b,o(c))m(c) \sum_{s\in\mathcal{S}} x(c,s)f(b,s). \tag{6}$$

A more fair variant would be minimizing the maximum penalty experienced by any party, i.e.,

$$\min_{x,y} \max_{\omega\in\mathcal{O}} \sum_{b\in\mathcal{B}} \sum_{c\in\mathcal{C}:\ o(c)=\omega} p(b,o(c))m(c) \sum_{s\in\mathcal{S}} x(c,s)f(b,s). \tag{7}$$

IV. Online/offline decomposition

Directly optimizing any of objectives (5)–(7) subject to constraints (1)–(4) would mean solving an ILP problem, with the well-known associated scalability problems. Solving such a problem in real time may be possible for some individual instances thereof, but does not represent a generally viable strategy.

An alternative approach, used in [6], is online/offline decomposition. In the offline stage, we *prune* the set $\mathcal{S}$ of physical sequences, setting the y-values in such a way that constraint (3), accounting for the limited capacity of switches, is always respected. In the online phase, we use those y-values as a parameter, and set the x-values in order to optimize our objective. The advantage of such a decomposition is that the online problem is a simple LP, with no binary variables – hence easy to solve even in real-time. On the negative side, the offline stage cannot account for real-time traffic conditions, and may therefore make suboptimal decisions.

In this section, we first recap the decomposition used in [6], which does not account for ownership issues, and then introduce an alternative, ownership-aware one.

A. Ownership-unaware pruning

The pruning stage used in [6] is the following:

$$\min_{y} \qquad \max_{b\in\mathcal{B}} \sum_{s\in\mathcal{S}} f(r,s) > 0 \tag{8}$$

$$\text{s.t.} \qquad \sum_{s\in\mathcal{S}} y(s) f(r,s) \le k(r) \qquad \forall r \in \mathcal{R} \tag{9}$$

$$\sum_{s\in\mathcal{S}} y(c,s) \ge C \qquad \forall c \in \mathcal{C} \tag{10}$$

The objective is to minimize the number of sequences in which the most used middlebox appears ((8)), subject to switch capacity limits ((9), corresponding to (3)), and to the fact that for each chain we have at least C active sequences ((10)).

The problem (8)–(10) is solved multiple times, using binary search to find the highest value of C for which it is feasible. The final result is a set of y-values that correspond to the highest possible value of C and maximize the objective (8). By construction, such a solution also respects constraint (3) of the original problem.

B. Ownership-aware pruning

The problem (8)–(10) does not account for the ownership of chains and resources. As pointed out in Figure 1, it may yield solutions with insufficient coupling, i.e., suboptimal values of objectives (5)–(7).

To cope with this problem, we propose an alternative pruning stage. The core idea is to give a higher value to those sequences where the chain and the boxes implementing it have the same owner. More formally, we associate to each sequence $s \in \mathcal{S}$ the following *score*:

$$\sigma(s) = 1 + w\frac{\sum_{r\in\mathcal{B}} f(b,s)\mathbb{1}_{[o(b)=o(s)]}}{\sum_{r\in\mathcal{B}} f(b,s)}, \tag{11}$$

where $w \ge 0$ is the *weight* that ownership has in our decisions. When $w = 0$, we revert back to ownership-unaware pruning; higher values of w mean a stronger tendency to enable same-ownership sequences. The score defined in (11) is $1 + w$ if all the processing is done with middleboxes belonging to the owner of the traffic, and decreases towards 1 as other boxes are used. The w parameter can be fine-tuned for best performance if need be; as we will see in Section V, the default value $w = 1$ guarantees very good performance in most practical cases.

Once we have the scores, we replace (10) in the ownership-unaware pruning stage with the following:

$$\sum_{s\in\mathcal{S}} y(c,s)\sigma(s) \ge C, \quad \forall c \in \mathcal{C}. \tag{12}$$

Using (12) *in lieu* of (10) has the effect of giving a higher value to sequences that include middleboxes owned by the same owner as the traffic they serve. This means that such sequences are more likely to be activated, which in turn will increase the coupling we are able to attain. Also notice that the change has no impact on compuational complexity.

V. Performance evaluation

A. Reference scenarios

We evaluate our solution through simulation, using the same three-layer network topology employed in [5]. Specifically, given a parameter $k \in \mathbb{N}$:

- the access layer consists of groups of ten base stations, connected in a ring;
- the aggregation layer consists of k *pods*, each composed by k switches connected in full-mesh;
- the core layer consists of k^2 switches connected in full-mesh.

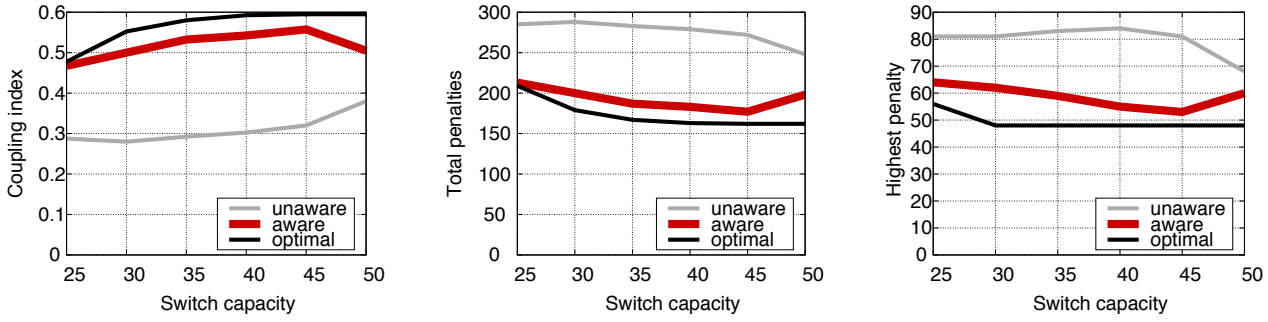
In each aggregation-layer pod, $k/2$ switches are connected to one of the rings of base stations, and $k/2$ to one of the core switches. The $2k^2$ switches from our set $\mathcal{R}$. For our simulations, we set $k = 8$, the same value used in [5].

As in [5], we consider k different types of middleboxes, and randomly connect two instances of each type to switches in each pod of the aggregation layer, and two more to switches in the core layer. We also associate a service chain in $\mathcal{C}$ to each base station. Each service chain is implemented by ten physical sequences in $\mathcal{S}$, each including five randomly-selected middlebox instances. For simplicity, we assign magnitude $m(c) = 1$ to all chains $c \in \mathcal{C}$, and capacity $k(b) = 10$ to all middleboxes $b \in \mathcal{B}$. The weight used to compute the scores defined in (11) is $w = 1$. Also, the penalty $p(b, o)$ is defined as follows:

$$p(b,\omega) = \begin{cases} 0 & \text{if } \omega = o(b) \\ 1 & \text{otherwise.} \end{cases}$$

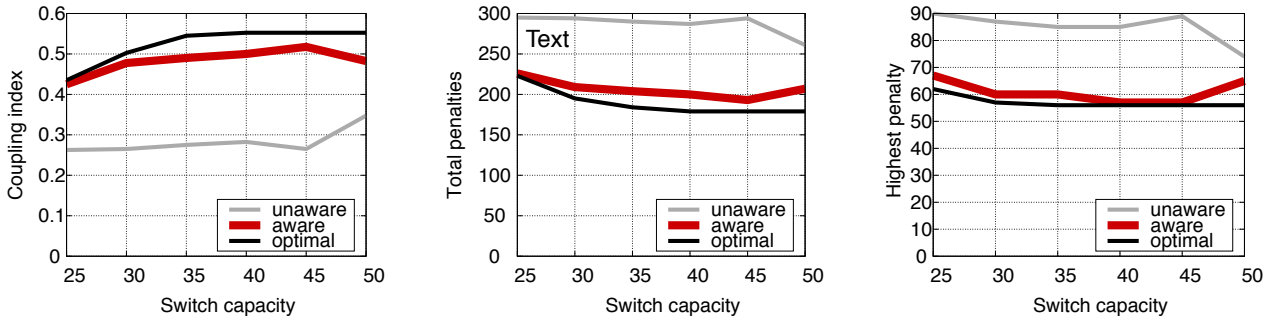
There are a total of four parties that can own switches and middleboxes. We consider two scenarios:

- a *clustered* scenario, where each aggregation-layer pod, all middleboxes and all traffic coming from base stations connected to it belong to the same owner;



(a) (b) (c)

Fig. 3. Clustered scenario: coupling index as defined in (5), (a), total penalties as defined in (6) (b), maximum penalty as defined in (7) (c) as a function of the switch capacity, when no pruning ("optimal"), ownership-aware ("aware") and ownership-unaware ("unaware") pruning are in place.



(a) (b) (c)

Fig. 4. Uniform scenario: coupling index as defined in (5), (a), total penalties as defined in (6) (b), maximum penalty as defined in (7) (c) as a function of the switch capacity, when no pruning ("optimal"), ownership-aware ("aware") and ownership-unaware ("unaware") pruning are in place.

- a *uniform* scenario, where the owner of each switch and middlebox is assigned randomly;

The clustered scenario represents such cases as network sharing between mobile operators, where pre-existing networks are integrated. The uniform scenario describes a *native* next-generation network, designed from the beginning with virtualization in mind.

B. Results

We begin by studying the performance, i.e., the value of objective functions (5)–(7) in three cases:

- when there is no pruning, i.e., when x- and y-values are set jointly;
- with the ownership-unaware pruning introduced in [6] and summarized in Section IV-A;
- with the ownership-aware pruning described in Section IV-B.

No pruning means that we solve the problems defined in Section III directly. Doing so yields the optimal solution but is not feasible in practice; we present those results as a benchmark. We are more interested in quantifying the improvement brought by moving from ownership-unaware to ownership-aware pruning.

Figure 3 refers to the clustered scenario. For all objectives, results are encouraging: moving to ownership-aware pruning i.e., from the gray to the red line in the figures, substantially improves coupling. Indeed, the performance we can obtain is very close to the optimum, represented by black lines. This is especially important, as a higher coupling index and lower penalties directly translate into better-functioning networks and/or lower fees owed to other parties.

It is worth stressing that, irrespective of the pruning, the online stage (objective functions included) is exactly the same: the difference in performance is entirely due to the pruning stages. Also notice that even the ownership-aware pruning yields suboptimal performance: this is because pruning happens offline, i.e., without accounting for the actual traffic.

In Figure 4, we move to the uniform scenario. For all objectives, we can observe a slight decay in performance with respect to Figure 3; intuitively, this is connected to the fact that the middleboxes belonging to a party tend to be away from the base stations where its traffic originates.

More importantly, ownership-aware pruning still yields a performance that is substantially better than ownership-unaware pruning and very close to the optimum.

Next, we want to understand how different pruning stages work, i.e., how they choose the sequences to enable. For each enabled sequence, we count how many of the boxes it includes have the same owner as the sequence itself, i.e.,

$$\sum_{s\in\mathcal{S}} y(s) \sum_{b\in\mathcal{B}} \mathbb{1}_{[f(b,s)>0]} \mathbb{1}_{[o(b)=o(s)]}. \tag{13}$$

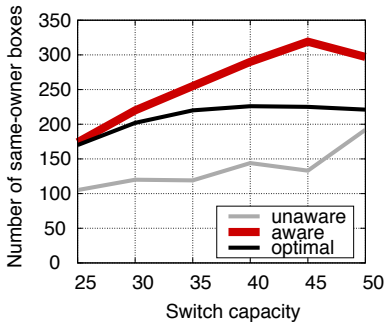


Fig. 5. Uniform scenario: number of same-owner boxes, as defined in (13), as a function of the switch capacity when ownership-aware pruning ("aware") and ownership-unaware pruning ("unaware") is applied. The "optimal" line refers to the y-values set with no pruning.

A higher value of the quantity in (13) means that the pruning scheme tends to enable sequences with many same-owner boxes.

Figure 5 refers to the uniform scenario, but we had similar results for the clustered one. Unsurprisingly, it shows that ownership-aware pruning corresponds to a much higher value of the quantity in (13). It is more interesting to look at the black line, showing how the y-values are set in the optimal case, with no pruning: the value is higher than in the ownership-unaware case, but lower than in the ownership-aware one. We can conclude that the ownership-aware pruning enables *more* sequences with same-owner boxes than needed for optimal performance. Tweaking the weight w we use in (11) could improve this situation; however, as no pruning strategy can account for the actual traffic, the resulting decisions are never guaranteed to be optimal.

VI. Concluding remarks

Next-generation cellular networks will differ from present-day ones in many ways, including *ownership* – different parts of the access and backhaul infrastructure will belong to different parties, as will the traffic flows being served. We modeled such networks in Section III, and argued that they will be managed in such a way to maximize the coupling between the parties originating the traffic and the parties processing it. We presented three alternative definitions of coupling, expressed by the metrics (5)–(7).

Optimizing these metrics directly would be too complex; therefore, in Section IV we presented two offline pruning strategies. One is ownership-unaware and was originally introduced in [6]; the other, described in Section IV-B, aims at enabling physical sequences that can yield a good coupling.

Our evaluation, summarized in Section V, takes into account three different coupling metrics and two different scenarios, and we consistently find that ownwership-aware pruning performs substantially better than state-of-the-art alternatives – almost doubling the coupling between the parties generating traffic and the ones processing it – and always very close to the optimum.

Acknowledgement

This work was supported by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11) and a grant from the U.S.-Israel Binational Science Foundation.

References

[1] L. Doyle, J. Kibiłda, T. Forde, and L. A. DaSilva, "Spectrum without Bounds, Networks without Borders," *Proceedings of the IEEE*, vol. 102, no. 3, 2014.

[2] G. Middleton, K. Hooli, A. Tolli, and J. Lilleberg, "Inter-operator spectrum sharing in a broadband cellular network," in *IEEE Symposium on Spread Spectrum Techniques and Applications*, Aug 2006.

[3] J. Kibilda and L. DaSilva, "Efficient coverage through inter-operator infrastructure sharing in mobile networks," in *Wireless Days (WD), 2013 IFIP*, Nov 2013.

[4] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *Communications Magazine, IEEE*, vol. 51, no. 7, July 2013.

[5] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," in *ACM conference on Emerging networking experiments and technologies*. ACM, 2013.

[6] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "Simple-fying middlebox policy enforcement using sdn," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013.

[7] S. Sun, T. Rappaport, R. Heath, A. Nix, and S. Rangan, "Mimo for millimeter-wave wireless communications: beamforming, spatial multiplexing, or both?" *Communications Magazine, IEEE*, vol. 52, no. 12, December 2014.

[8] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimiter Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE*, 2014.

[9] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3gpp device-to-device proximity services," *Communications Magazine, IEEE*, vol. 52, no. 4, 2014.

[10] E. Wyatt and N. Cohen, "Comcast and Netflix reach deal on service," *New York Times*, 2014.

[11] "Fast and free facebook mobile access with 0.facebook.com," 2015.

[12] S. Datoo, "Twitter's latest deal points to ambitions in emerging markets," *The Guardian*, 2013.

[13] B. Chen, "Google Confirms Plans for Wireless Service," *New York Times*, 2015.

[14] R. Riggio, K. Gomez, L. Goratti, R. Fedrizzi, and T. Rasheed, "V-cell: Going beyond the cell abstraction in 5g mobile networks," in *NOMS, 2014 IEEE*. IEEE, 2014.

[15] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: taking control of the enterprise," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007.

[16] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, 2008.

[17] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, "Blueprint for introducing innovation into wireless mobile networks," in *ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*. ACM, 2010.

[18] P. Srisuresh, J. Kuthan, A. Rayhan, J. Rosenberg, and A. Molitor, "Middlebox communication architecture and framework," 2002.

[19] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, "Clickos and the art of network function virtualization," in *USENIX NSDI*, 2014.

[20] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *Communications Magazine, IEEE*, vol. 51, no. 11, 2013.

[21] J. Batalle, J. Ferrer Riera, E. Escalona, and J. Garcia-Espin, "On the implementation of nfv over an openflow infrastructure: Routing function virtualization," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*. IEEE, 2013.

[22] N. Feamster, H. Balakrishnan, and J. Rexford, "Some foundational problems in interdomain routing," in *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*. Citeseer, 2004, pp. 41–46.