How Should I Slice My Network?

Hailiang ZHAO @ ZJU-CS
http://hliangzhao.me

May 2, 2020

This slide is a report on paper *How Should I Slice My Network? A Multi-Service Empirical Evaluation of Resource Sharing Efficiency*, published on **MobiCom'18**.

Outline

Outline

Outline

Outline

Outline

Why We Need Network Slicing?

Current mobile services have a strong **diversification** on *Key Performance Indicator (KPI)* and *Quality of Service (QoS)* requirements.

examples

1. massive IoT devices with ultra-low rate communication
2. automotive and tactile applications with millisecond latencies
3. industrial communications with extreme reliability
4. virtual/augmented reality services with very high data rates

Current mobile network architectures lack the **flexibility** to meet the **extreme requirements** imposed by those heterogeneous mobile services!

Network Virtualization is Imperative!

There exists a strong need for **customized network support** with present-day and future traffic. 5G networks achieve this mainly via:

Network Virtualization ($\mathrm{MNO}_{slice} \leftrightarrow \mathrm{SP}_{SLA}$)

creates a set of *logical network instances (i.e. network slices)* on top of the physical infrastructure, each tailored to accommodate fine-tuned *Service Level Agreement (SLA)* reflecting the needs of different *Service Providers* (a.k.a. *Tenant*).

For spectrum mngmt., baseband processing, mobility mngmt., etc:

(i) traditional hardbox paradigm $\rightarrow$ a cloudified architecture
(ii) hardware-based network functions $\rightarrow$ software-based *Virtual Network Functions (VNFs)*

Is **Dynamic** Resource Allocation to Slices always Good?

1. When instantiating a slice, the MNO needs to allocate sufficient computational & communicaitonal resources to ths slice
2. However, the tenants' demand can be time-varying...

$$\Downarrow$$

Dynamic Resource Allocation Algorithms are welcome!

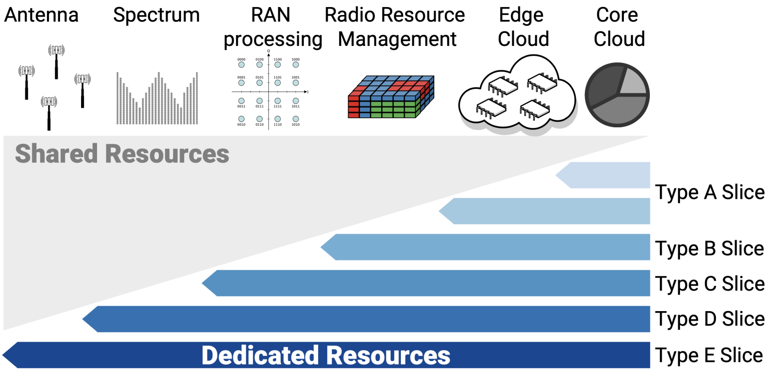
Nevertheless...

It will lead to...

1. additional complexity
2. in some cases hinder *resource isolation*
3. fully customized slices cannot be guaranteed

The **Inherent** Trade-off in Network Slicing

1. **Service Customization** (Core Cloud $\nearrow$ Antenna)
2. **Resource Management Efficiency** (dynamic sharing $\uparrow$)
3. **System Complexity** (dynamic resource allocation $\uparrow$)



Network Slicing Types

High-level Opinions

Slicing strategies at the *higher* network layers provide a *lower* level of customization yet they can *more easily* achieve efficient resource sharing *without* additional complexity.

1. **Public Internet (including Core Network)** $\rightsquigarrow$
   type-A: VM or container resource assignment
2. **Backhaul of RAN** $\rightsquigarrow$
   type-B: radio resource at C-RAN & Multi-access Edge
   type-C: customized baseband processing in BBUs, guaranteed bandwidth in the air
3. **Fronthaul of RAN** $\rightsquigarrow$
   type-D: guaranteed spectrum in Base Stations (BSs)
   type-E: dedicated end-to-end resources down to the antennas

Contribution and Takeaways of this Paper

There already exist

1. **mature** cloud resource orchestrators (Kubernetes)
2. **developing** edge resource orchestrators (KubeEdge)
3. **multifarious** dynamic resource allocate algorithms to slices

However, the implications of network slicing in terms of efficiency of reosource utilization are **still not well understood**.

Contributions

This paper analyzes the trade-off between *customization*, *efficiency*, and *complexity* in network slicing, by evaluating the impact of resource allocation **dynamics** at different network levels (***type-A*** $\rightarrow$ ***type-E***).

Takeaways: The efficiency gains are very high **in the edge**, where emplopying technologies that allow for dynamic resource allocation provides a high reward.

Outline

Hierarchical Mobile Network Architecture

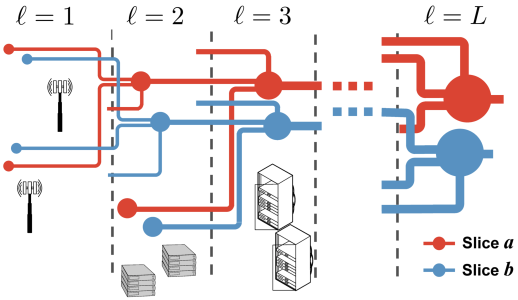
Mobile Network Scenario

We consider a mobile network providing coverage to *a generic geographical region*, where mobile subscribers consume *a variety of heterogeneous services* provided by SPs. The MNO who owns the infrastructure implements slices $s \in \mathcal{S}$, **each dedicated to a subset of services**.

1. The mobile network is modeled as a **hierarchy** composed by a fixed number of levels ($l = 1, ..., L$), ordered from the most distributed ($l = 1$) to the most centralized ($l = L$)
2. Every network level l is composed by a set $\mathcal{C}_l$ of network nodes, each serving a given number of base stations ($|\mathcal{C}_1| > ... > |\mathcal{C}_L|$)
3. $\forall$ node $\in \mathcal{C}_1$, it's bijective mapping to individual antenna
4. $\mathcal{C}_L$ contains a single node, i.e. a fully-centralized datacenter

Hierarchical Mobile Network Architecture (Cont'd)

5. $\forall l$, a node $c \in \mathcal{C}_l$ operates on dataflows that are increasingly aggregated with l
6. from $l = 1$ to $l = L$:
   operating at antenna level $\rightarrow$
   running VNFs in C-RAN datacenters $\rightarrow$
   running VNFs in telco-cloud datacenters $\rightarrow$
   running containers/VMs in a fully-centralized cloud datacenter



Modeling the Network Slices

Slice specifications $z = (f, w)$

A slice specification is established so as to ensure a sufficient service quality for the slice's demands.

1. **guaranteed time fraction (proportion)** $f \in [0, 1]$: during *at least* f of the observation time, the traffic demands of this slice can be *fully served*
2. **window length** w: the traffic demands of this slice *is averaged over* a time slot of length w

For slice s at node c, the averaged load over window k is

$$\overline{o}_{c,s}(k) = \frac{1}{w} \int_k o_{c,s}(t) \mathrm{d}t,$$

where $o_{c,s}(t)$ is the real-time load required for each moment t during the window k.

Modeling the Network Slices (Cont'd)

Reconfiguration period

Reconfiguration period is *the minimum time needed* for resource reallocation, whose length is denoted as τ. Actually, in practice the periodicity of reconfiguration is limited by the adopted slicing strategy and the constraints of the underlying technology. Thus, we assume that $\tau \gg w$.

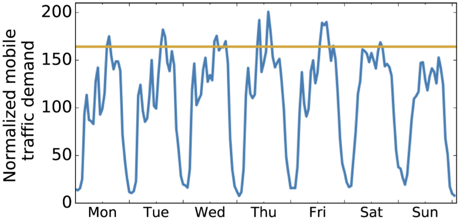
1. **The whole system observation time** is composed by a set $\mathcal{T}$ of all the reconfiguration periods. Let us denote by $r_{c,s}^{z}(k)$ the resources allocated to slice s at node c during window k.
2. Because $r_{c,s}^{z}(k)$ **cannot be changed** during windows of the same reconfiguration period, which means **NO** reassignment of resources is available. Let us use $\hat{r}_{c,s}^{z}(n)$ as the final allocated resources to node c during the reconfiguration period n, then we have $\hat{r}_{c,s}^{z}(n) = \max_{k \in \text{ period } n}\{r_{c,s}^{z}(k)\}$.

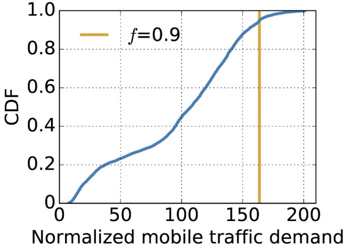
Modeling the Network Slices (Cont'd)

3. $\forall n \in \mathcal{T}, \forall s \in \mathcal{S}$, the following constraint should be satisified[1]:

$$\frac{\sum_{k \in \text{ period } n} \mathbb{1}\{\hat{r}^z_{c,s}(n) \geq \bar{o}_{c,s}(k)\}}{\tau / w} \geq f. \tag{1}$$

4. Let $F^w_{c,s,n}$ as the CDF of the demand for slice s at node c during period n.[2] The satisified $\hat{r}^z_{c,s}(n)$ should be calculated as $\hat{r}^z_{c,s}(n) = (F^w_{c,s,n})^{-1}(f)$.





[1] The authors write a wrong math formula here.

[2] The authors have clerical errors here.

Defining Multiplexing Efficiency

For the whole system observation time, at network level l:

1. **The total amount of resources needed in Network Slicing**:

$$\mathbb{R}^{z}_{l,\tau} = \sum_{s\in\mathcal{S}} \sum_{c\in\mathcal{C}_l} \sum_{n\in\mathcal{T}} \tau \cdot \hat{r}^{z}_{c,s}(n)$$

   (If no reconfiguration is allowed, we can set $|\mathcal{T}| = 1$)

2. **Perfect sharing** (no isolation among different services, and traffic multiplexing is the maximum):

$$\mathbb{P}^{z}_{l,\tau} = \sum_{c\in\mathcal{C}_l} \sum_{n\in\mathcal{T}} \tau \cdot \hat{r}^{z}_{c}(n),$$

   where $\hat{r}^{z}_{c}(n) = \max_{k\in \text{ period } n}\{r^{z}_{c}(k)\}$, and $r^{z}_{c}(k) = \sum_{s\in\mathcal{S}} r^{z}_{c,s}(k)$,[3] with $\hat{r}^{z}_{c,s}(n)$ replaced by $\hat{r}^{z}_{c}(n)$ in eq. (1).

[3] The authors forget to give the calculation of $r^{z}_{c}(k)$.

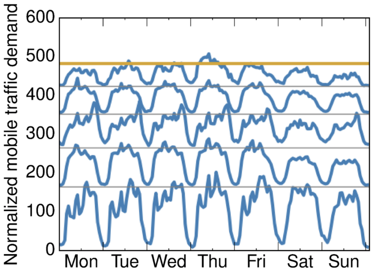
Defining Multiplexing Efficiency (Cont'd)

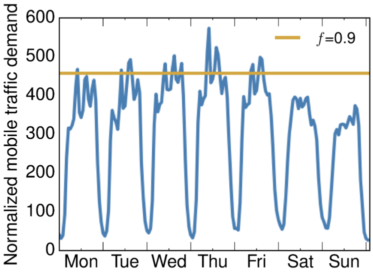
Multiplexing efficiency

Multiplexing efficiency is the ratio between the resources required with network slicing and those needed under perfect sharing, i.e.

$$\mathbb{E}^{z}_{l,\tau} = \mathbb{P}^{z}_{l,\tau} / \mathbb{R}^{z}_{l,\tau} \in [0, 1].$$

As it approaches to 1, the total amount of slice-isolated resources tend to that assured by a perfect sharing.





Outline

Collecting Traffic Demands of Mobile Services

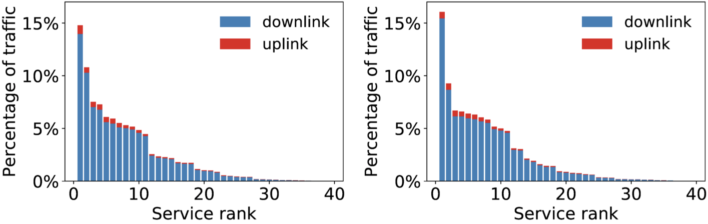
The real-world demands over **38 services** were aggregated *temporally* (over 5-minute time intervals) and *geographically* (per antenna sector) by the MNO, so as to make the data non-personal.

1. Those services are identified by
    1. they generate a substantial offered load (above 0.1% of the total network traffic), sufficient to justify the creation of a dedicated network slice
    2. they entail clearly distinguishable KPIs and QoS requirements
2. Those services cover a wide range of classes with diverse network requirements, including
    1. mobile broadband (e.g., long-lived and short-lived video streaming)
    2. lowlatency (e.g., gaming, messaging)
    3. best effort (e.g., web browsing, social media)
3. The data was collected *during three months* in late 2016

Collecting Traffic Demands of Mobile Services (Cont'd)

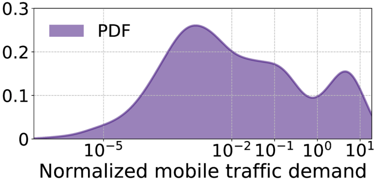
4. Antenna deployments in two target regions:

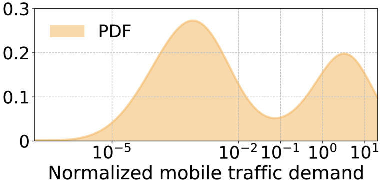
5. Percentage of mobile traffic among 38 services *(spans several orders of magnitude)*:



Collecting Traffic Demands of Mobile Services (Cont'd)

6. the Probability Density Function (pdf) of the total offered load at individual antenna sectors *(spans several orders of magnitude)*:





- The main cause of heterogeneity is the radio access technology
- Compared with 2G and 3G, 4G antennas accommodate much larger fractions of the demand and generate the rightmost bell-shaped lob of the distributions
- 10-time differences in the traffic volume appear even across 4G antenna sectors, implying substantial location-based demand variability

Associating antennas to different network levels

How to get the hierarchy?

We do not have information on the architecture of the mobile networks beyond the radio access, thus we model the network architecture as a *hierarchy* by

associating the level-l nodes with distributed antenna sites

At level l, the MNO deploy a number $N_l = \mathcal{C}_l$ of nodes, **each responsible for a subset of the antenna sites at the radio access level**. The association is created based on two criterias:

1. the offered load should be similar at all nodes
   [ensures basic load balancing]
2. the subset of antennas associated to a same node shall be geographically contiguous
   [reduces capital expenditures for wired connection]

Associating antennas to different network levels (Cont'd)

Balanced graph k-partitioning

The problem of level-l node-to-antenna site association translates into **dividing the graph into N_l sub-graphs**, such that the sum of costs of nodes in each partition is balanced.

1. each vertex $v \in V$ maps to one antenna site
2. an associated cost $c(v)$ equal to the mobile traffic demand recorded at the site
3. an edge $e = \{u, v\} \in E$ connect vertices u and v only if the corresponding antenna sites are geographically *adjacent*

$$e_{uv} = \begin{cases} 1 & \text{if } e \text{ is a cut edge} \\ 0 & \text{otherwise} \end{cases} \qquad \forall e \in E,$$

$$x_{v,k} = \begin{cases} 1 & \text{if } v \text{ is in partition } k \\ 0 & \text{otherwise} \end{cases} \qquad \forall v \in V, \forall k,$$

Associating antennas to different network levels (Cont'd)

The formulated Integer Linear Programming (ILP) problem[4]:

$$
\begin{aligned}
\min \sum_{e_{uv} \in E} & e_{uv} \\
\text{s.t.} \quad & \sum_{v \in V} x_{v,k} c(v) \leq (1+\epsilon) \frac{\sum_{v \in V} c(v)}{N_\ell}, \ \forall k \\
& \sum_{v \in V} x_{v,k} c(v) \geq (1-\epsilon) \frac{\sum_{v \in V} c(v)}{N_\ell}, \ \forall k \\
& \sum_{k} x_{v,k} = 1, \ \forall v \in V. \\
& e_{uv} \geq x_{u,k} - x_{v,k}, \ \forall e \in E, \forall k \\
& e_{uv} \geq x_{v,k} - x_{u,k}, \ \forall e \in E, \forall k
\end{aligned}
$$

The NP-hard problem is solved by the Karlsruhe Fast Flow Partitioner (KaFFPa) heuristic.

[4] Two nodes from different sub-graphs formulate a **cut edge**.

Associating antennas to different network levels (Cont'd)

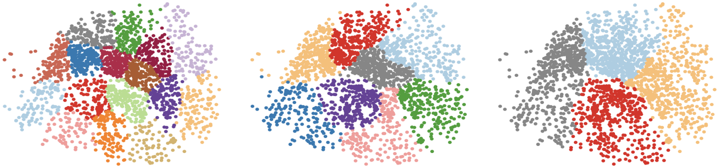


Figure 8: Association of antenna sites to level-ℓ nodes in the large metropolis scenario. The plots refer to $\ell = 8$ (16 nodes, left), $\ell = 9$ (8 nodes, middle) and $\ell = 10$ (4 nodes, right). Figure best viewed in colours.

$\boldsymbol{\ell}$		1	2	3	4	5	6	7	8	9	10	11	12
Traffic per node		5	10	15	30	60	75	100	150	300	600	1167	2334
$\mathcal{N}_\ell$	Metropolis	422	230	160	80	40	32	23	16	8	4	2	1
	City	122	60	40	20	10	8	6	4	2	1		

[5]

[5] At $l = 1$, nodes map to individual 4G antenna sectors.

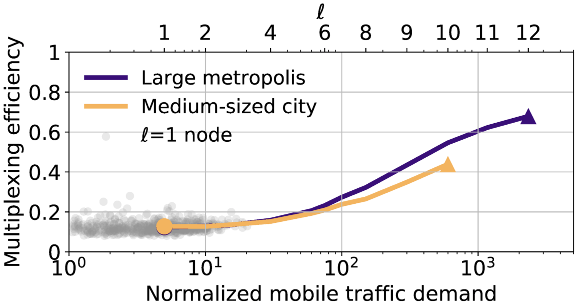
Efficiency Evaluation #1: **in Worst Settings**

Worst case settings

1. strict slice specifications: $f = 1$, $w = 5$ minutes
2. reconfiguration is not allowed: $|\mathcal{T}| = 1$

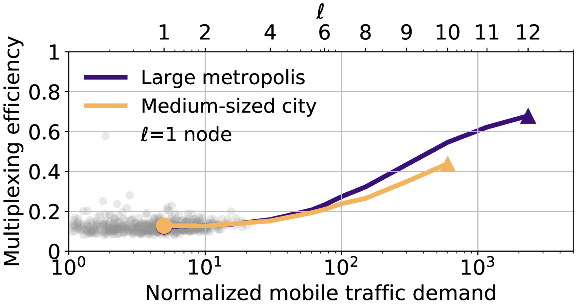
Observation and Analysis:

1. Overall, the efficiency is low $(0.15 \sim 0.65)$



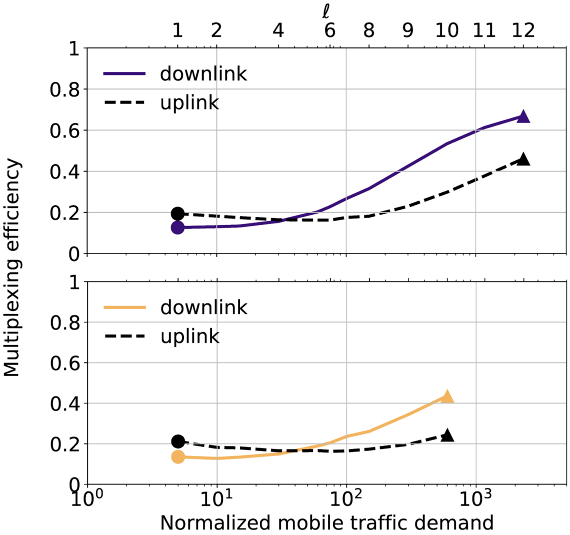
Efficiency Evaluation #1: **in Worst Settings**

2. The efficiency grows as one moves from *the antenna level* to *a fully centralized cloud*
   [Different slices typically peak at different times, and the burstiness of demands associated to each slice is significantly reduced as the network level grows.]
3. Differences are minimal between the two reference cities, and only emerge for high values of l



Efficiency Evaluation #1: **in Worst Settings**

Disaggregated for downlink and uplink traffic:



Efficiency Evaluation #1: **in Worst Settings**

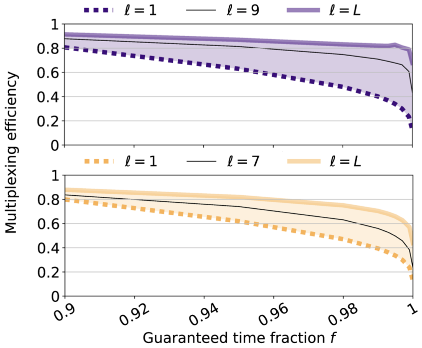
Observation and Analysis (from above figure):

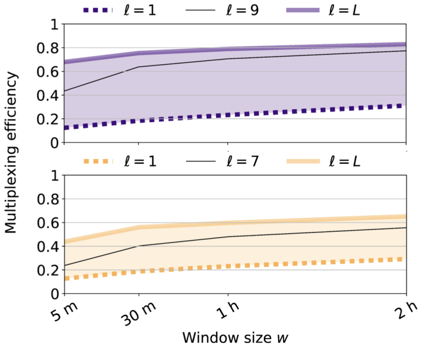
1. Slicing uploads tends to become remarkably (30% to 50%) less efficient as one moves towards more centralized network levels [The reason lies again in the small uplink traffic volume, which results in bursty time series with high peak-to-average ratios, even upon aggregation over multiple antennas.]
2. The overall resource assignment should be driven by the downlink behaviour. However, specific applications, hence slices, heavily rely on uplink traffic are hard to accommodate.

Efficiency Evaluation #2: **in Moderating Settings**

Moderating settings

1. f changes from 0.9 to 1, w changes from 5 minutes to 2 hours
2. reconfiguration is still not allowed: $|\mathcal{T}| = 1$





Efficiency Evaluation #2: **in Moderating Settings**

Observation and Analysis (from above figure):

1. Decreasing f drastically improves the efficiency
2. On the downside, there exists a diminishing returns effect as f is lowered
3. w has a less significant impact on efficiency than f
4. The efficiency gains resulting from decreasing f do not only involve a price in terms of the total time not satisfying the demand, but also in terms of the duration of the corresponding periods
   [demands are not satisfied over periods involving **more than one consecutive time window**]
   (This conclusion is not demonstrated by any figures)

Efficiency Evaluation #3: **Reconfiguration Allowed**

Reconfiguration allowed

We assume the availability of an oracle algorithm that, at the beginning of a reconfiguration interval, has perfect knowledge of the future time series of the demand for each service and for the rest of the interval. **[Oracle exists!]**

The baseline result, in figure below, refers to the case of $\tau = 30$ minutes **[a fairly high resource reconfiguration frequency]**:

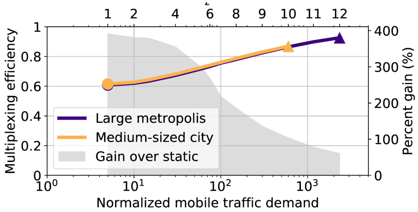


Figure 12: Efficiency of slice multiplexing (left y axis) and percent gain over static assignment (right y axis) versus the normalized mobile traffic served by one node (bottom x axis) at level ℓ (top x axis) in the two reference urban scenarios. Results are for a dynamic resource assignment where re-configurations occur with periodicity $\tau = 30$ minutes, under slice specification $z = (f, w) = (1, 5$ minutes). Dots denote $\ell = 1$ and triangles $\ell = L$ for each scenario.

Efficiency Evaluation #3: **Reconfiguration Allowed**

Observation and Analysis:

1. Dynamic allocation mechanisms and a perfect prediction of the demand over the future 30 minutes can substantially improve the efficiency of slice multiplexing
2. There is a very important difference between efficiency at the radio access and in the network core
   [the gain ranges from 60% to 400%]

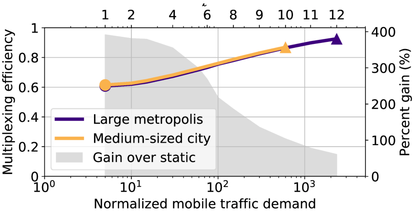
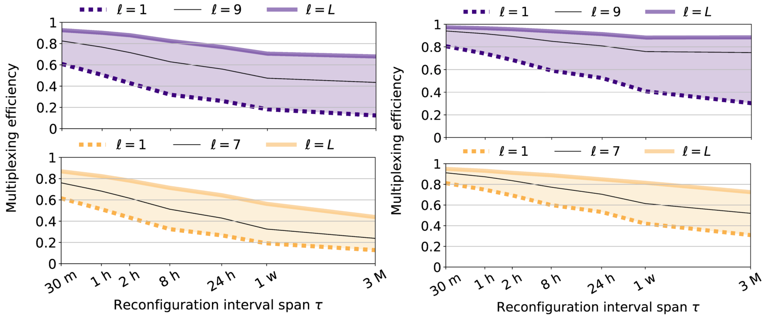


Figure 12: Efficiency of slice multiplexing (left y axis) and percent gain over static assignment (right y axis) versus the normalized mobile traffic served by one node (bottom x axis) at level ℓ (top x axis) in the two reference urban scenarios. Results are for a dynamic resource assignment where re-configurations occur with periodicity $\tau = 30$ minutes, under slice specification $z = (f, w) = (1, 5$ minutes). Dots denote $\ell = 1$ and triangles $\ell = L$ for each scenario.

Efficiency Evaluation #3: **Reconfiguration Allowed**

Observation and Analysis (left figure):

1. The multiplexing efficiency of slices is decreased as τ grows, since the system becomes less flexible
2. The loss of efficiency is most remarkable for low values of τ [30 mins $\sim$ 2 hours: high loss; 2 hours $\sim$ 8 hours: low loss]
3. It is not worth considering dynamic resource allocation at all if the reconfiguration time is in the order of a few hours at most



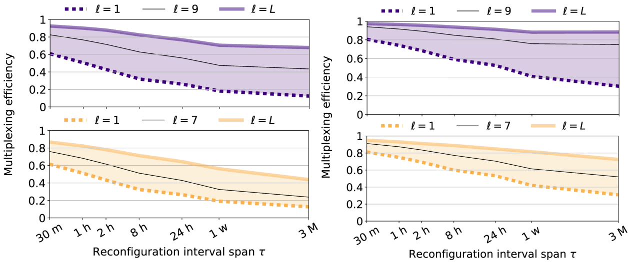
Efficiency Evaluation #4: **Specific Slice Numbers**

Slice configurations #1: **aggregation: 38 Slices → 7 Slices**

The services of a similar type are aggregated together into the same slice, which allows to reduce the 38 slices that we had in the previous experiments down to 7 slices dedicated to streaming, social network, web, cloud, gaming, messaging and miscellaneous services.

Observation and Analysis (right figure):

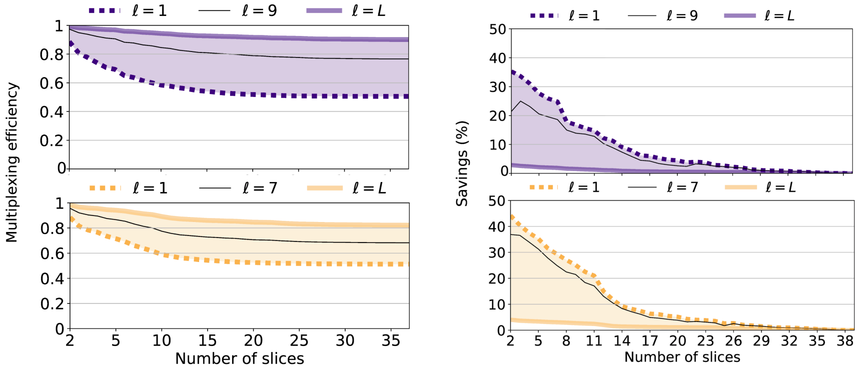
1. Significant gains in efficiency (While customization ↓)
2. Efficiency remains rather low for small l and large τ values



Efficiency Evaluation #4: **Specific Slice Numbers**

Slice configurations #2: **only dedicated to popular services**

Services that generate the highest traffic load acquire a dedicated slice tailored to their service, while the remaining services are aggregated into a common, non-customized, slice.



Efficiency Evaluation #4: **Specific Slice Numbers**

Observation and Analysis (left figure):

1. The efficiency improving trend becomes flat after 15 slices [efficiency is only improved when the services with the largest demands are brought into the common slice]

Slice configurations #3: stricter guarantees for dedicated slices

Inherent from *slice specification #2*, those tenants acquiring dedicated slices are provided a stricter guarantees than the ones in the common slice.

1. For dedicated slices, $f = 1$
2. For common slices, $f = 0.9$

Observation and Analysis (right figure):

1. The savings remain very low in the network core, but can be significant for resources located close to the radio access

Efficiency Evaluation #5: **A New Kind of Efficiency**

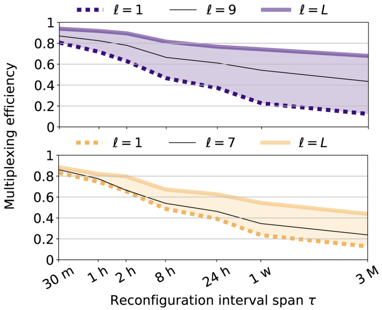
1. The efficiency defined above is appropriate to **evaluate operating expenses (OPEX)**, and can be applied, e.g., to electric power consumption, management overheads, or deterioration of assets with use
2. Another viewpoint on efficiency is interms of equipment to be deployed to meet the instantaneous demand. This relates to **the capital expenditure (CAPEX)** incurred by the mobile network operator, typically hardware and infrastructure costs

$$\mathbb{R}_{l,\tau}^{\star z} = \sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}_l} \max_{n \in \mathcal{T}} \hat{r}_{c,s}^{z}(n), \mathbb{P}_{l,\tau}^{\star z} = \sum_{c \in \mathcal{C}_l} \max_{n \in \mathcal{T}} \hat{r}_{c}^{z}(n)$$

$$\mathbb{E}_{l,\tau}^{\star z} = \mathbb{P}_{l,\tau}^{\star z} / \mathbb{R}_{l,\tau}^{\star z}$$

Efficiency Evaluation #5: **A New Kind of Efficiency**

1. In absence of mechanisms that allow for dynamic reconfiguration, the efficiency is very much comparable to that observed in the previous analysis
2. Flexibility in the orchestration of resources pays off also in terms of equipment deployment efficiency
3. When l is close to 1, a dynamic reconfiguration of resources allows improving deployed infrastructure efficiency much faster than resource usage efficiency
4. In the network core (i.e., for l that tends to L), trends are similar



Outline

Concluding Remarks

Takeaways:

1. Downlink-oriented or uplink-oriented? Traffic direction is a factor. Which is the bottleneck?
2. Dynamic resource assignment must be paid
3. Urban topography has limited impact [What about countrysides?]
4. Aggregating services is beneficial
5. Deployment is more efficient than operation

Reviews:

1. Sufficient empirical evaluation is convictive
2. The modeling is succinct and effective [catch the point!]
3. Good plot, great characterization, enticing illustrations