A Utility-Driven Multi-Queue Admission Control Solution for Network Slicing

- Bin Han, Vincenzo Sciancalepore, Di Feng, Xavier Costa-Perez and Hans D. Schotten

What this paper aims to do?

1. to study the slicing admission control problem by means of a multi-queuing system for heterogeneous tenant requests,
2. to derive its statistical behavior model, and
3. to provide a utility-based admission control optimization

Did I like the paper?

- I liked the paper
  - As it gave a mathematical model for resource allocation using queuing
- I disliked the paper
  - As it was more abstract and didn't talk about the real 5G scenario

Background (Resources & Slices)

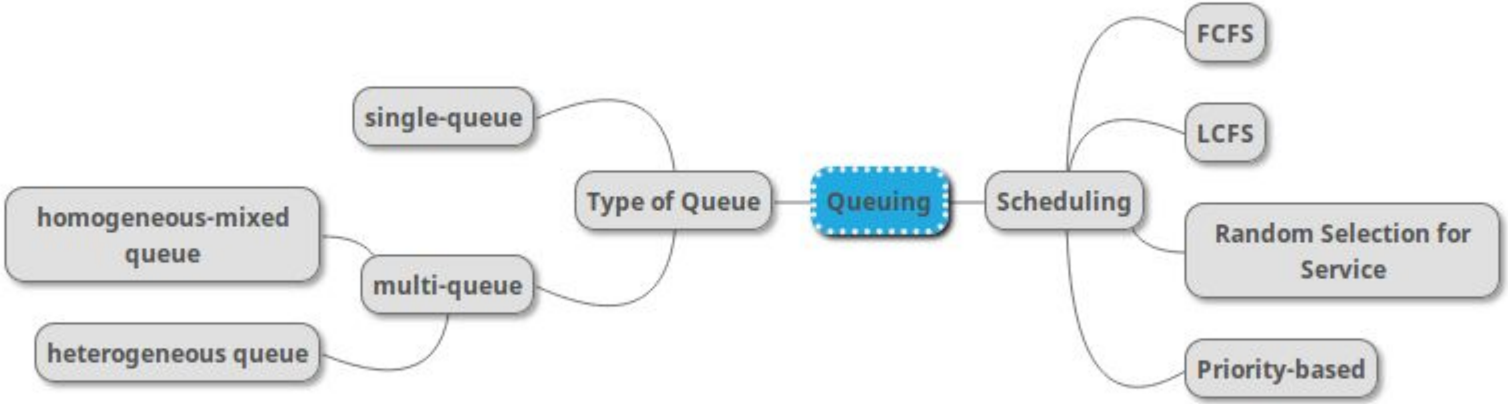
- Resource pool and slice type
  - The resources taken by the slices can be denoted by
    - N is the number of type of slices
    - M is the different types of resource in the resource pool
    - $S = \{s | r_m - a_m \geq 0, \forall \ 1 \leq m \geq M\}$
    - s → set of slices that are active
    - r → resource pool
    - a → assigned resources
    - $a = C \times s$, $C = [c_1, c_2, \ldots, c_N]$,
    - c_n → resource bundle required to maintain the slice of type n

Background (Slices & Admission Control)

- Slice Admission:
  - Inter-arrival time between 2 requests is exponential
  - Request arrival is i.i.d
  - Binary decision for accepting or rejecting the slice
- If MNO resource pool tends to be saturated, then additional slice requests are not accepted (this introduces Admissibility region)
- Lifetime of a slice is i.i.d exponentially distributed variable and expected lifetime depends on the slice type
- A rejected slice can be sent again for reconsideration after some delay

Background (Queuing)

- Among several scheduling algorithm, FCFS is taken into consideration
- Single queues: one queue is implemented for all declined requests that need to wait for the next acceptance opportunity
- Homogeneous-mixed queues: each queue consists of requests for slices of different types
- Heterogeneous queues: each queue is specified for only one unique slice type



Background (Queuing contd.)

- N FCFS Heterogeneous queues are considered

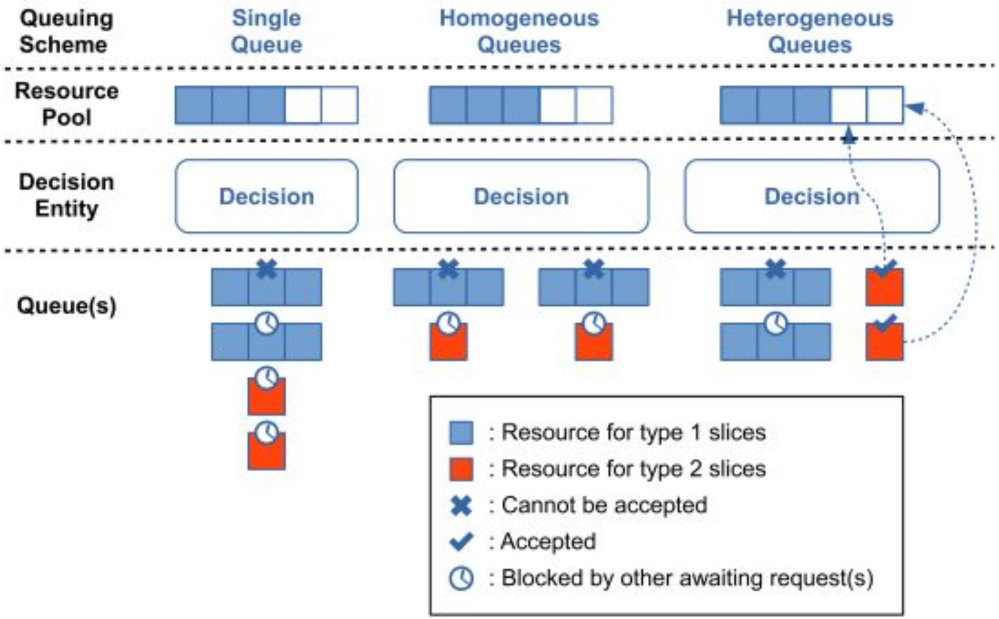
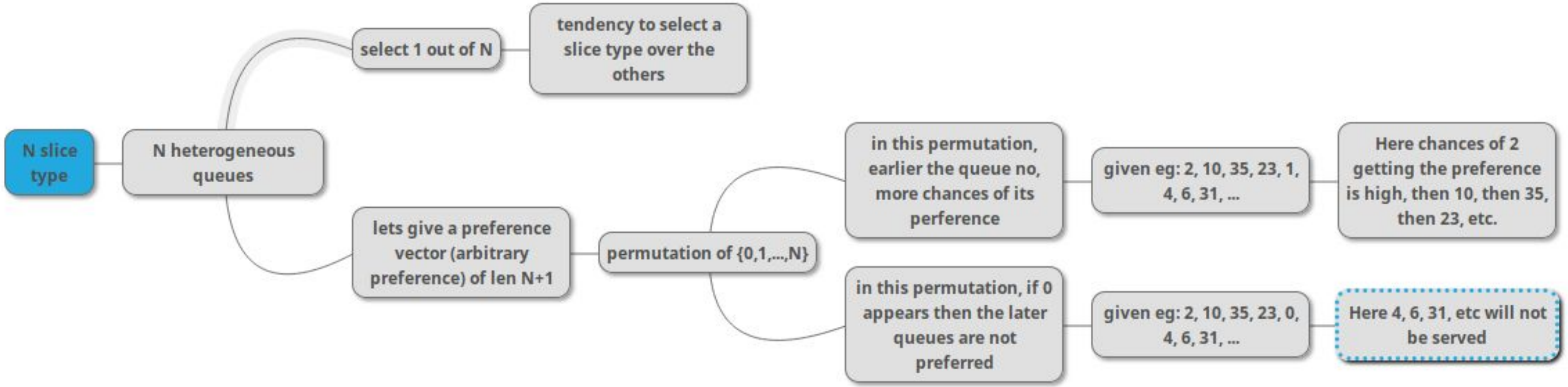


Fig. 1: A simple case study on different queuing schemes.

Core Idea (Slice Admission Control)

- Simultaneous requests might come at the same time and MNO might have to select one and reject others or reject all the requests
- Preference vector: $\Phi = [\phi_1, \phi_2, \ldots, \phi_{N+1}]$



Core Idea (Admission preference matrix)

- The requests are accepted in admissibility region, then the preference matrix will look like
- $\boldsymbol{\Phi} = [\Phi_1, \Phi_2, \ldots, \Phi_{|A|}]$
-
- $= [\varphi_{1,1} \quad \varphi_{1,2} \quad \ldots \quad \varphi_{1,|A|}]$
- $[\varphi_{2,1} \quad \varphi_{2,2} \quad \ldots \quad \varphi_{2,|A|}]$
- $[\;.\quad .\quad .\quad .\;]$
- $[\;.\quad .\quad .\quad .\;]$
- $[\;.\quad .\quad .\quad .\;]$
- $[\varphi_{N+1,1} \; \varphi_{N+1,2} \ldots \varphi_{N+1,|A|}]$

Core Idea (Slice Admission Control Algo)

```
Initialize with certain N, 𝕊, 𝔸, Φ and s;
while True do                                                        Main loop
    Wait for the next incoming tenant issue;
    if Slice of type n released then                                 Releasing a slice
        s ← s − Δs_n;
    else if Slice of type n requested then                           Request arrives
        l_n ← l_n + 1;
    end
    while s ∈ 𝔸 do                   Recursively serving the queues until blocked
        s̃ ← s;
        Find the current preference vector Φ according to Φ and s;
        for 1 ≤ n ≤ N do                                             Serve queues w.r.t. preference
            if φ_n = 0 then                                          Omitting queues after 0
                break;
            else if l_n > 0 AND (s + Δs_n) ∈ 𝕊 then                  Acceptance
                l_n ← l_n − 1;
                s ← s + Δs_n;
            end
        end
        if s̃ = s then                                                Blockage detection
            Break;
        end
    end
end
```

- Multi queue admission control algorithm

Core Idea (Queuing model)

- The acceptance in different queues are mutually independent Poisson processes, if:
  - the arrivals of new requests and releases of active slices are mutually independent Poisson processes for every individual slice type;
  - the arrivals of different slice types are mutually independent from each other, the releases of different slice types are mutually independent from each other
- Queuing-theoretic analysis
  - λ_n → request arrival rate for slice type n
  - L_n → mean length of queue n
  - $\overline{W}_n$ → average waiting time in queue n
  - μ_n → acceptance rate of queue n
  - ρ_n → workload rate of queue n

Core Idea (Queuing model contd.)

$$L_n = \lambda_n \overline{W}_n,$$

→ Little's Formula

$$p_n(l) = (1 - \rho)\rho^l,$$

→ Steady Queue State Probability

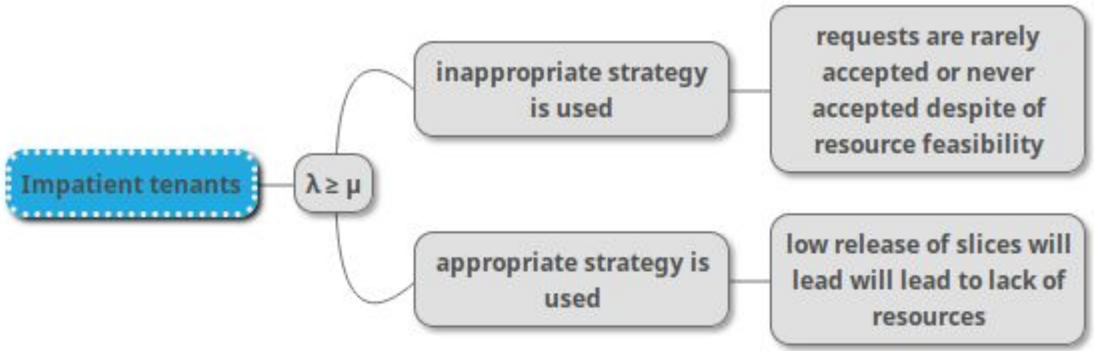
$$f(W_n) = \begin{cases} 0 & W_n < 0 \\ (\mu_n - \lambda_n)e^{-(\mu_n - \lambda_n)W_n} & W_n \geq 0 \end{cases},$$

→ Probability Density Function of Waiting Time Distribution

$$F(W_n) = \begin{cases} 0 & W_n < 0 \\ 1 - e^{-(\mu_n - \lambda_n)W_n} & W_n \geq 0 \end{cases}.$$

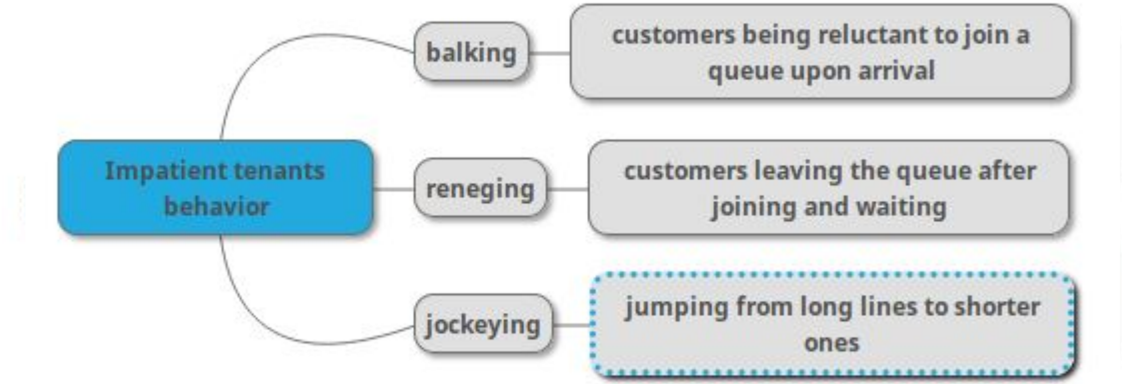
→ Cumulative Density Function of Waiting Time Distribution

Core Idea (Impatient Tenants)



- Reasons for Impatient tenants

Core Idea (Impatient Tenants contd.)



- Impatient Tenant Behavior

Core Idea (Performance Metrics)

$$u_{\Sigma}(t) = \sum_{n=1}^{N} s_n(t) u_n,$$

$$\overline{u}_{\Sigma} = \sum_{n=1}^{N} \frac{\mu_n u_n}{\eta_n},$$

$$\overline{W}_{\mathrm{q}} = \frac{\sum\limits_{n=1}^{N} \overline{W}_{\mathrm{q},n} L_n}{\sum\limits_{n=1}^{N} L_n}.$$

$$\overline{P}(A) = \frac{\sum_{n=1}^{N} \lambda_n P(A_n)}{\sum_{n=1}^{N} \lambda_n}.$$

- Overall network utility
- Average overall network utility
- Average waiting time of all requests in queues
- Overall admission rate

Strength

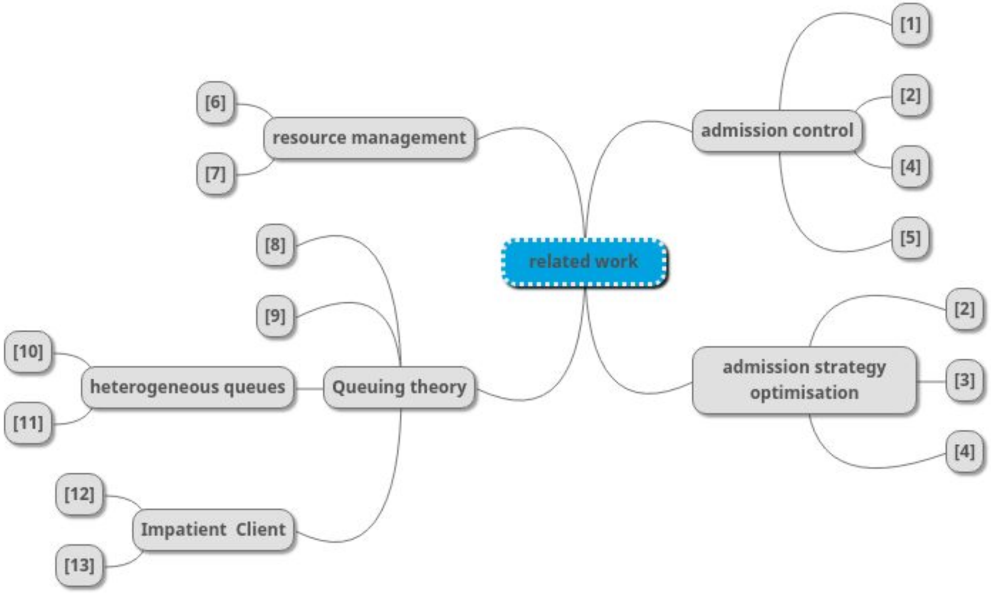
- Gives a good analysis of slice admission problem
- Takes impatient tenants into consideration and tackles it

Weakness

- Takes a static slice allocation approach (Once a slice is instantiation is done, no change in the traffic is considered) [May lead to over allocation]
- No priority of slice considered among the slice requests from the same MNO. (Mentioned about priority scheduling among the MNOs)
- It is quite abstract and doesn’t take 5G scenario into consideration

How to solve it?

Related Work



References

[1] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, “From network sharing to multi-tenancy: The 5G network slice broker,” IEEE Communications Magazine, vol. 54, no. 7, pp. 32–39, 2016.

[2] V. Sciancalepore et al., “Slice as a service (SlaaS): Optimal IoT slice resources orchestration,” in IEEE Global Communications Conference (GLOBECOM), Dec 2017, pp. 1–7

[3] B. Han, L. Ji, and H. D. Schotten, “Slice as an evolutionary service: Genetic optimization for inter-slice resource management in 5G networks,” IEEE Access, vol. 6, no. 1, pp. 33 137–33 147, 2018

[4] D. Bega, M. Gramaglia et al., “Optimising 5G infrastructure markets: The business of network slicing,” in IEEE International Conference on Computer Communications (INFOCOM), 2017

References contd.

[5] V. Sciancalepore, K. Samdanis et al., “Mobile traffic forecasting for maximizing 5G network slicing resource utilization,” in IEEE Conference on Computer Communications (INFOCOM), 2017.

[6] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, “On radio access network slicing from a radio resource management perspective,” IEEE Wireless Communications, vol. 24, no. 5, pp. 166–174, 2017.

[7] P. L. Vo, M. N. Nguyen, T. A. Le, and N. H. Tran, “Slicing the edge: Resource allocation for ran network slicing,” IEEE Wireless Communications Letters, 2018.

[8] J. Vilaplana, F. Solsona, I. Teixidó, J. Mateo, F. Abella, and J. Rius, “A queuing theory model for cloud computing,” The Journal of Supercomputing, vol. 69, no. 1, pp. 492–507, 2014.

References contd.

[9] X. Chang, B. Wang, J. K. Muppala, and J. Liu, “Modeling active virtual machines on IaaS clouds using an M/G/m/m+ K queue,” IEEE Transactions on Services Computing, vol. 9, no. 3, pp. 408–420, 2016.

[10] F. Li, J. Cao, X. Wang, and Y. Sun, “A QoS guaranteed technique for cloud applications based on software defined networking,” IEEE Access, vol. 5, pp. 21 229–21 241, 2017.

[11] M. Guo, Q. Guan, and W. Ke, “Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload,” IEEE Access, vol. 6, pp. 15 178–15 191, 2018.

[12] S. Bocquet, “Queueing theory with reneging,” Defence Science and Technology Organisation, Australia, Tech. Rep., 2005.

[13] De-quan Yue and Yan-ping Sun, “Waiting time of M/M/c/N queuing system with balking, reneging, and multiple synchronous vacations of partial servers,” Systems Engineering-Theory & Practice, vol. 28, no. 2, pp. 89–97, 2008.