Constellate: Establishing the opportunity for Distributed Unit pooling in real-world 5G Radio Access Networks

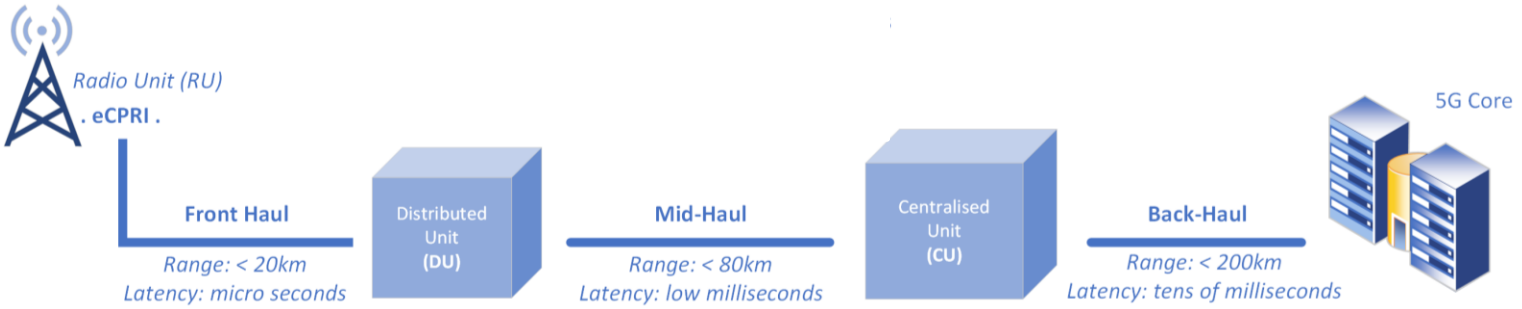
Sri Pramodh Rachuri[1], Anshul Gandhi[1],
Gueyoung Jung[2], Shankaranarayanan P. Narayanan[2], Alex Zelezniak[2]

[1]PACE Lab, Stony Brook University
[2]AT&T Labs

The content in this slide deck is prepared using the information from our paper- Constellate, MASCOTS 2025

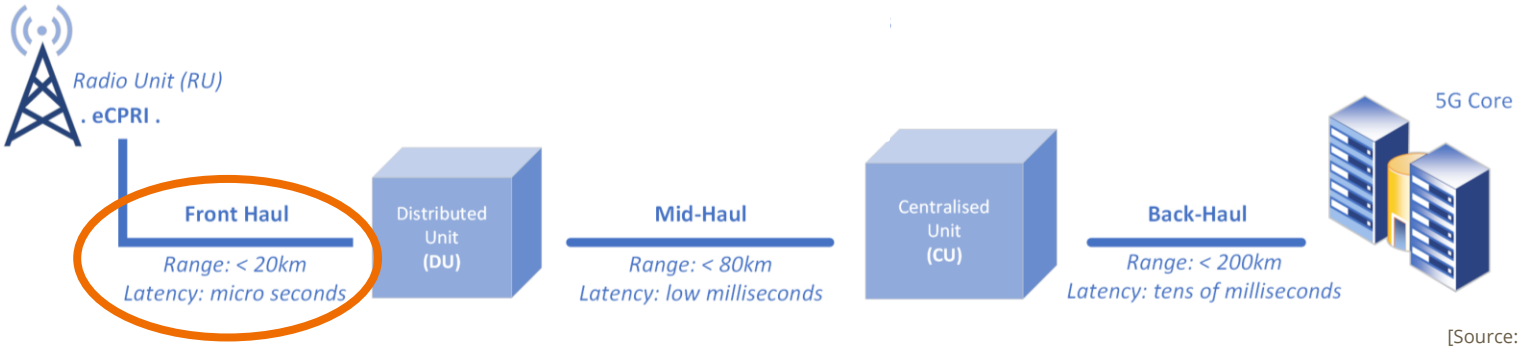
srachuri@cs.stonybrook.edu
pramodh.rachuri.dev

Background



[Source: Stevehegarty]

Background

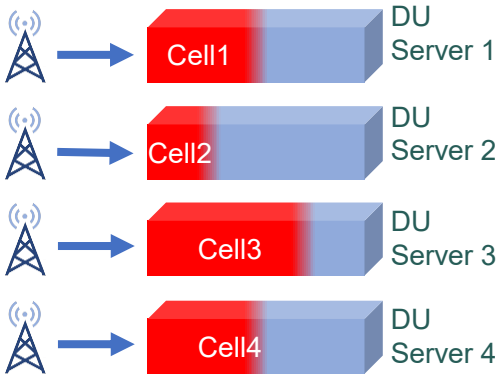


[Source: Stevehegarty]

Background – vDU pooling

Without pooling

- Every RU/Cell mapped to a DU
  - Created once forever
- Need to provision for max spike
- Wastage and Fragmentation of resources



With pooling and virtualization

- DU software is virtualized
  - Shared hardware
  - Reduced fragmentation
- Cell migration but with overhead
- Less provisioning for spikes

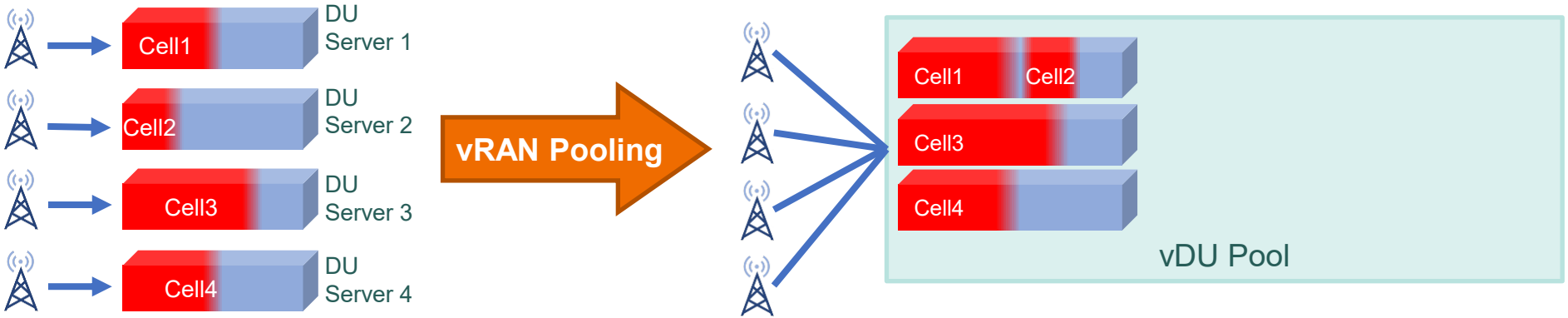
Background – vDU pooling

Without pooling

- Every RU/Cell mapped to a DU
  - Created once forever
- Need to provision for max spike
- Wastage and Fragmentation of resources

With pooling and virtualization

- DU software is virtualized
  - Shared hardware
  - Reduced fragmentation
- Cell migration but with overhead
- Less provisioning for spikes



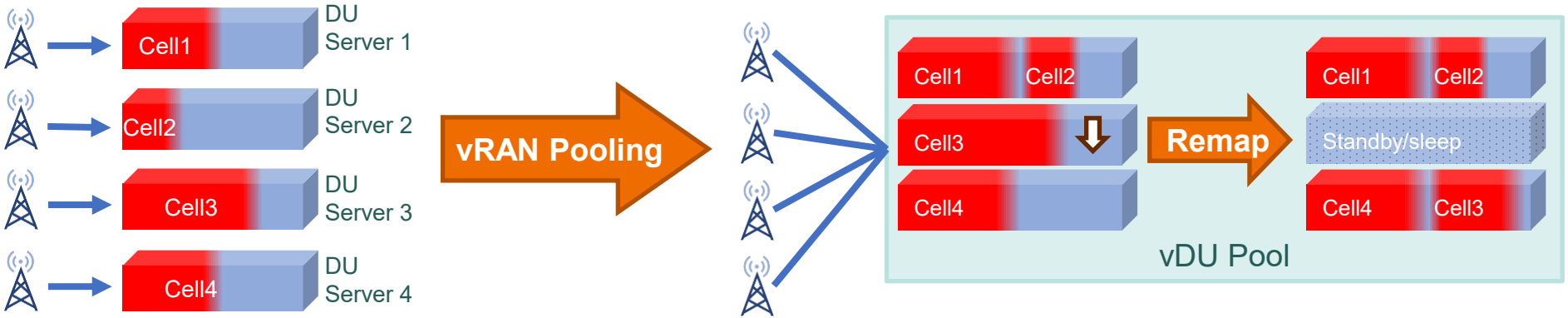
Background – vDU pooling

Without pooling

- Every RU/Cell mapped to a DU
  - Created once forever
- Need to provision for max spike
- Wastage and Fragmentation of resources

With pooling and virtualization

- DU software is virtualized
  - Shared hardware
  - Reduced fragmentation
- Cell migration but with overhead
- Less provisioning for spikes



Prior work – mechanism for vRAN pooling and migration

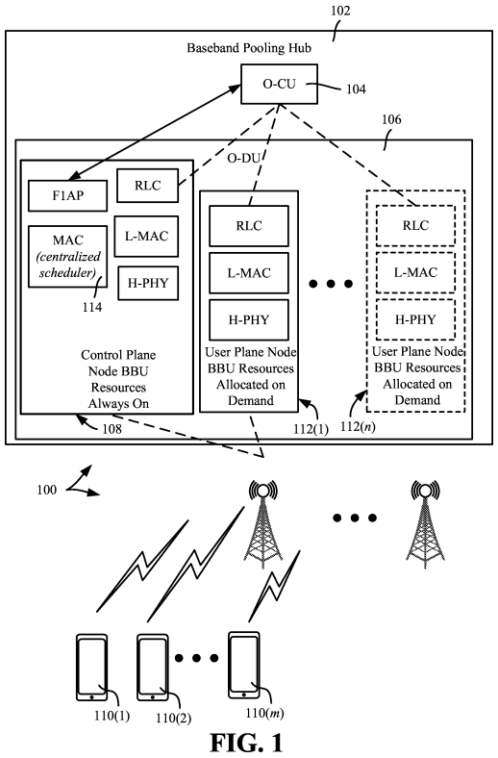


FIG. 1

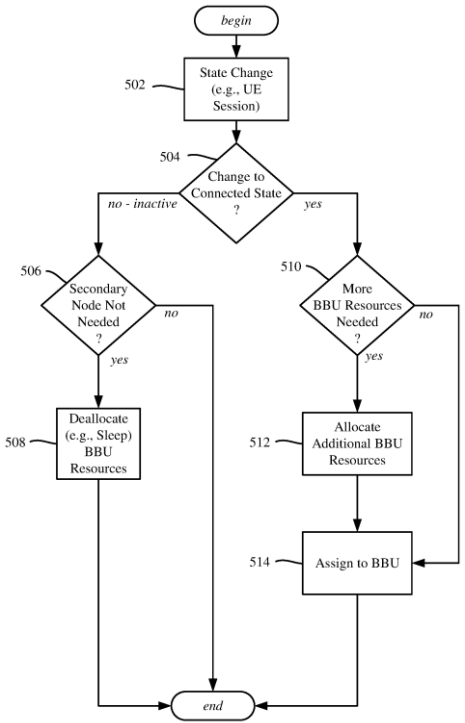
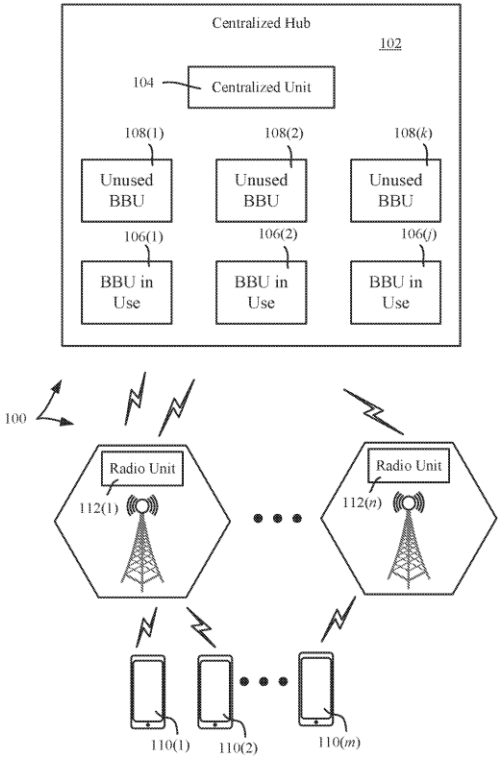
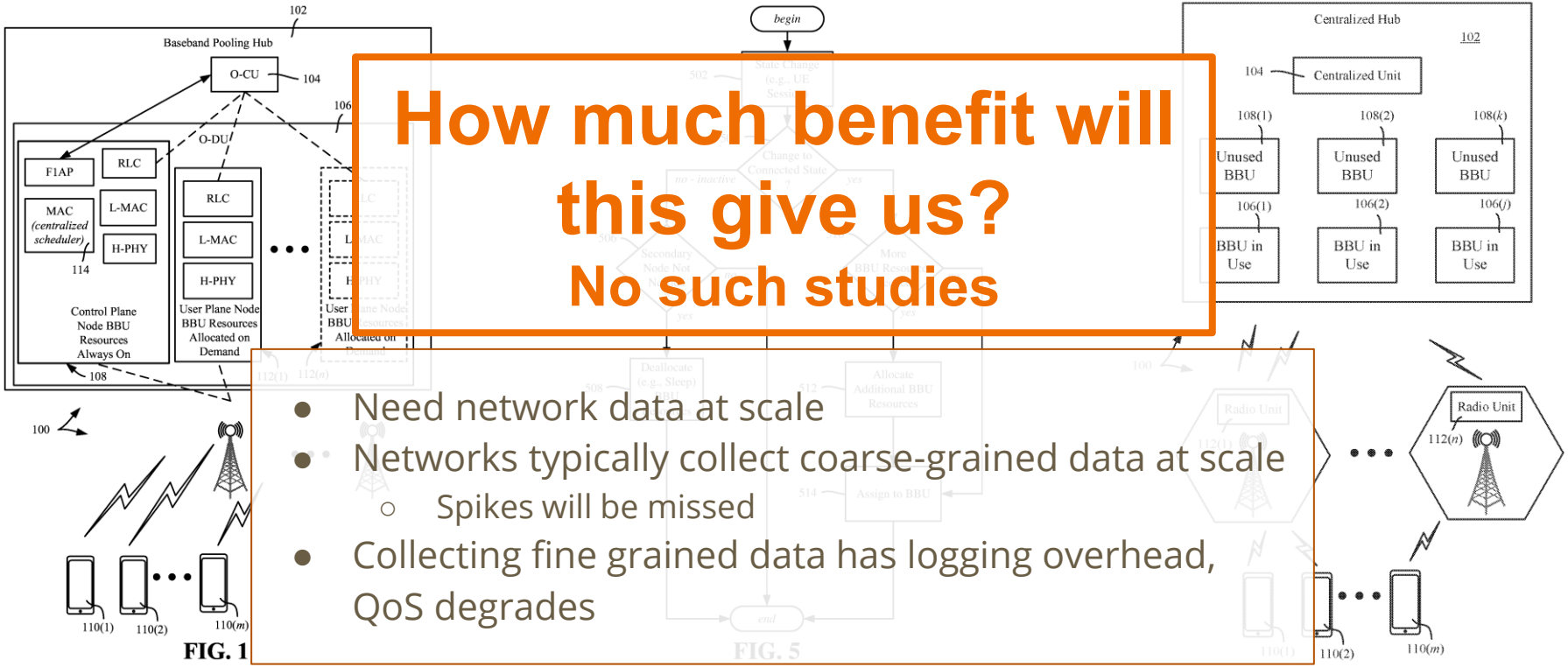


FIG. 5



Patents: US12177871B2, US20230276354A1, US12034603B2

Prior work – mechanism for vRAN pooling and migration



How much benefit will this give us?

No such studies

- Need network data at scale
- Networks typically collect coarse-grained data at scale
  - Spikes will be missed
- Collecting fine grained data has logging overhead, QoS degrades

Patents: US12177871B2, US20230276354A1, US12034603B2

Our Contributions

- Three strategies for pooling
  - Greedy
  - Affinity
  - Dynamic
- Generate fine-grained estimates of the cell workload
- Evaluate pooling benefits using largescale real-world network traces

vDU Pooling

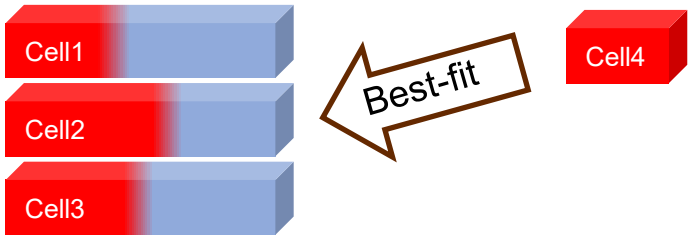
- Placing and moving cells across vDUs servers
  - Resource utilization of cells
  - Max vDU server capacity

  - Bin packing algorithms
    E.g. First-fit, Best-fit etc

vDU Pooling

- Placing and moving cells across vDUs servers
  - Resource utilization of cells
  - Max vDU server capacity

  - Bin packing algorithms
    E.g. First-fit, Best-fit etc



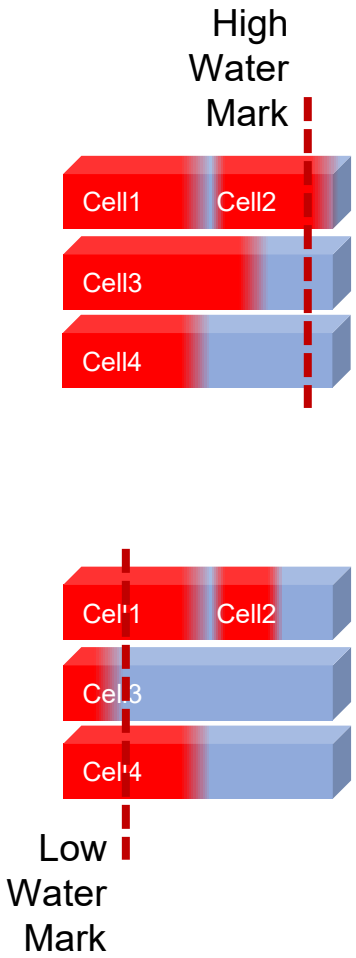
- Greedy - Best fit
  - Sort cells by util (decending)
  - Iterate over cells
  - Find the tightest bin the cell fits
  - If none, spin-up new bin (server)

- Reacting to change in utilization
  - Re-pooling frequency?
  - Keep running bin packing?
    - No affinity
    - Overhead of remapping

vDU Pooling

Affinity pooling

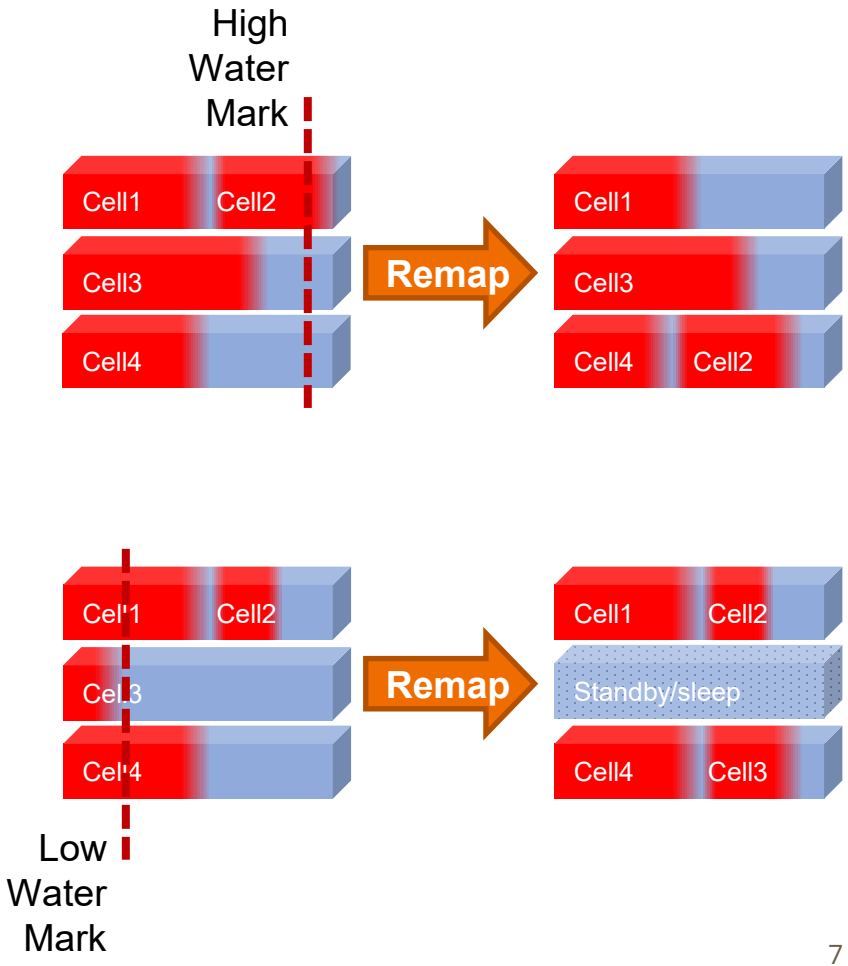
- High and low water mark detection
- High detected on a vDU
  - Move busiest cell away
    - Try best fit (tightest fit)
    - Else, new vDU
  - Repeat till below high water mark
- Low detected on a vDU
  - Move all cell away to other vDUs
    - Best fit
  - Emptied vDU set to sleep



vDU Pooling

Affinity pooling

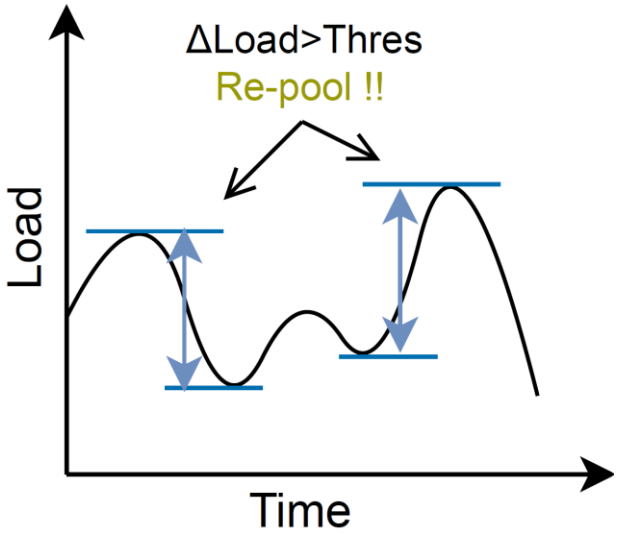
- High and low water mark detection
- High detected on a vDU
  - Move busiest cell away
    - Try best fit (tightest fit)
    - Else, new vDU
  - Repeat till below high water mark
- Low detected on a vDU
  - Move all cell away to other vDUs
    - Best fit
  - Emptied vDU set to sleep



vDU Pooling

Dynamic pooling

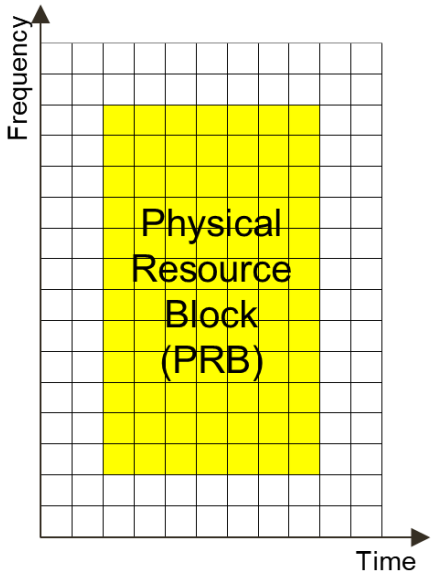
- Keep monitoring total load across all cells
- Re-pool only when difference exceeds a threshold



Measuring load of a cell

- Cell load processing - for pooling and migration

- Pick existing metric as proxy
  - Physical Resource Blocks (PRBs)
  - Smallest allocation unit in RF
    - Both time, freq
  - Prior work show high correlation with CPU processing load
  - Bytes does not work well
    - Channel quality decided ratio



Evaluation Methodology

Anonymized 5G network traces

- No UE/user identifiable information
- Tier 1 network operator- 2.9k cells
- 3 different freq ranges
  - Low, Mid and mmWave bands
- 3 different types of locations
  - Urban (cluster 1)
  - Suburban (cluster 2)
  - Rural (cluster 3)
  - <10 KMs

	Low Band			Mid Band			mmWave		
Cluster ID	1	2	3	1	2	3	1	2	3
# Cells	1328	158	63	1262	59	3	37	-	-
Band Total			**1549**			**1324**			**37**

Evaluation metrics

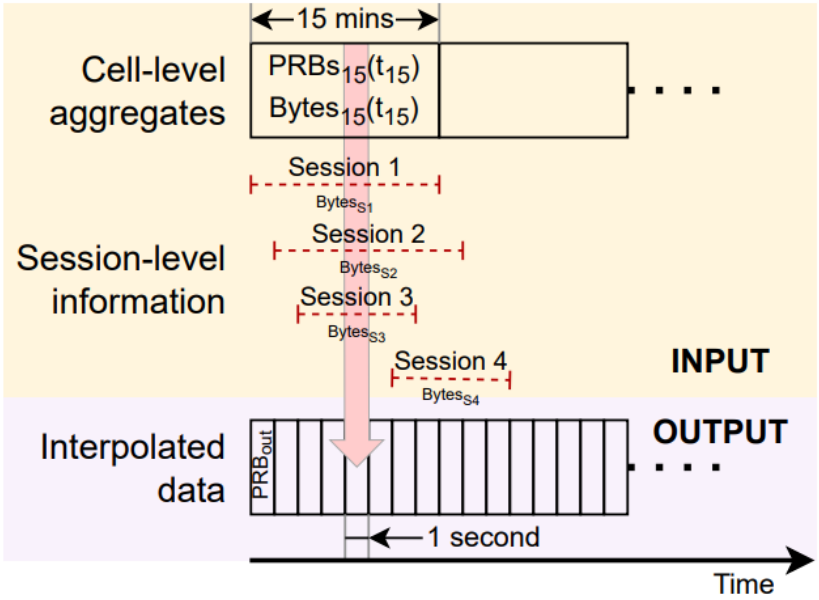
- Capital Expenditure – **Capex**
  - One-time expenditure on buying equipment
  - Captures 'max' number of servers
- Operational Expenditure – **Opex**
  - Continuous expenditure on running equipment
    - For example- electricity
  - Captures 'mean' number of servers
- Number of remaps - **#Remaps**
  - Num of times a cell has been remapped
  - Captures 'overhead' of re-pooling
  - Effective overhead is heavily dependent on implementation

Obtaining fine-grained PRB usage

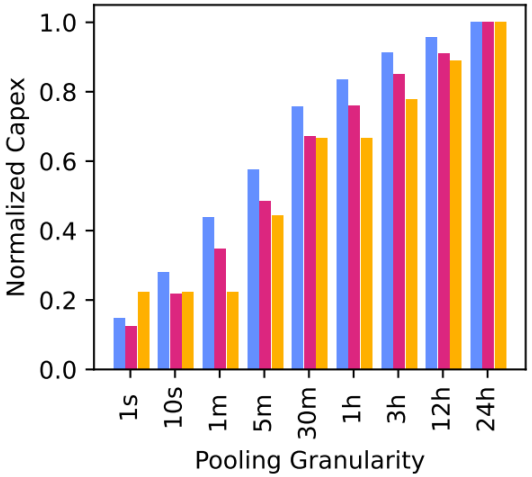
Two different types of datasets

- Coarse grained cell aggregates
  - Collected at order of 10s of mins (e.g. 15 mins)
  - Both PRBs and Bytes
- Fine grained session info
  - Collected at order of seconds (e.g. 4 seconds)
  - Only has bytes. No PRBs

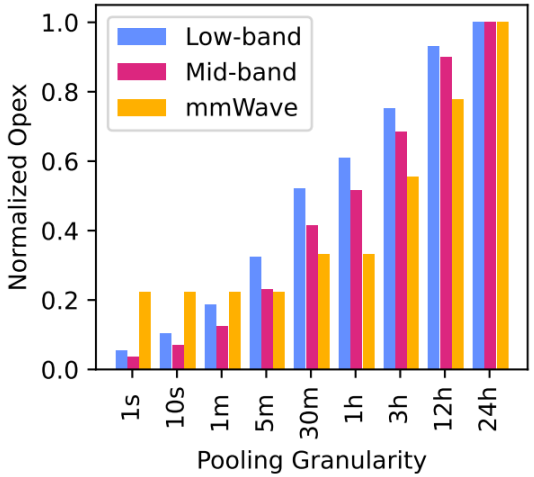
- Information from both datasets combined
  - Sum up bytes/sec across active sessions in a cell
  - Multiply with PRBs/Bytes ratio from cell aggregates



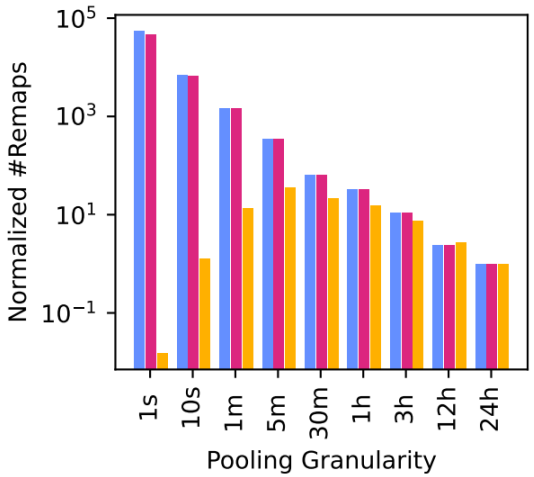
Evaluation Results – Different Bands



(a) Capex

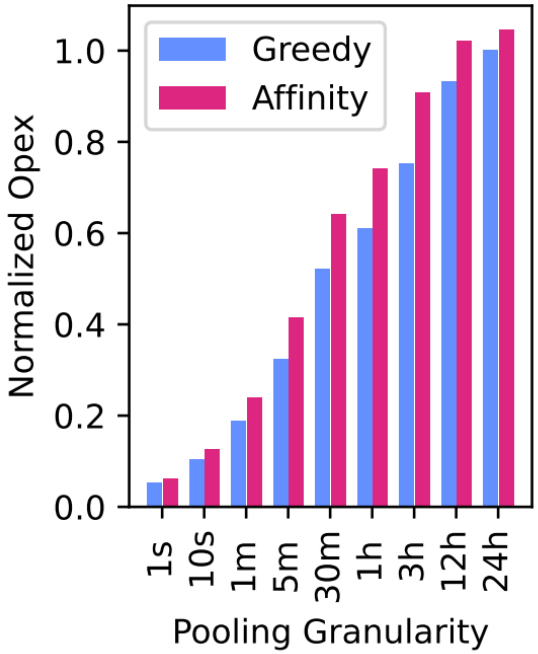


(b) Opex

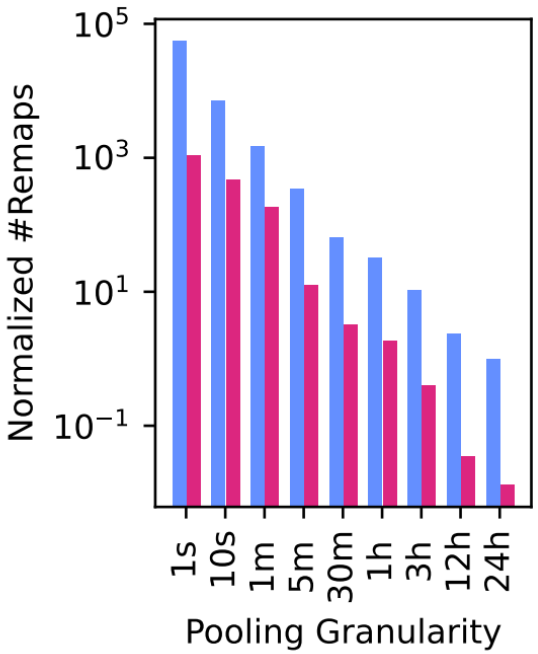


(c) Remaps

Evaluation Results – Greedy vs Affinity

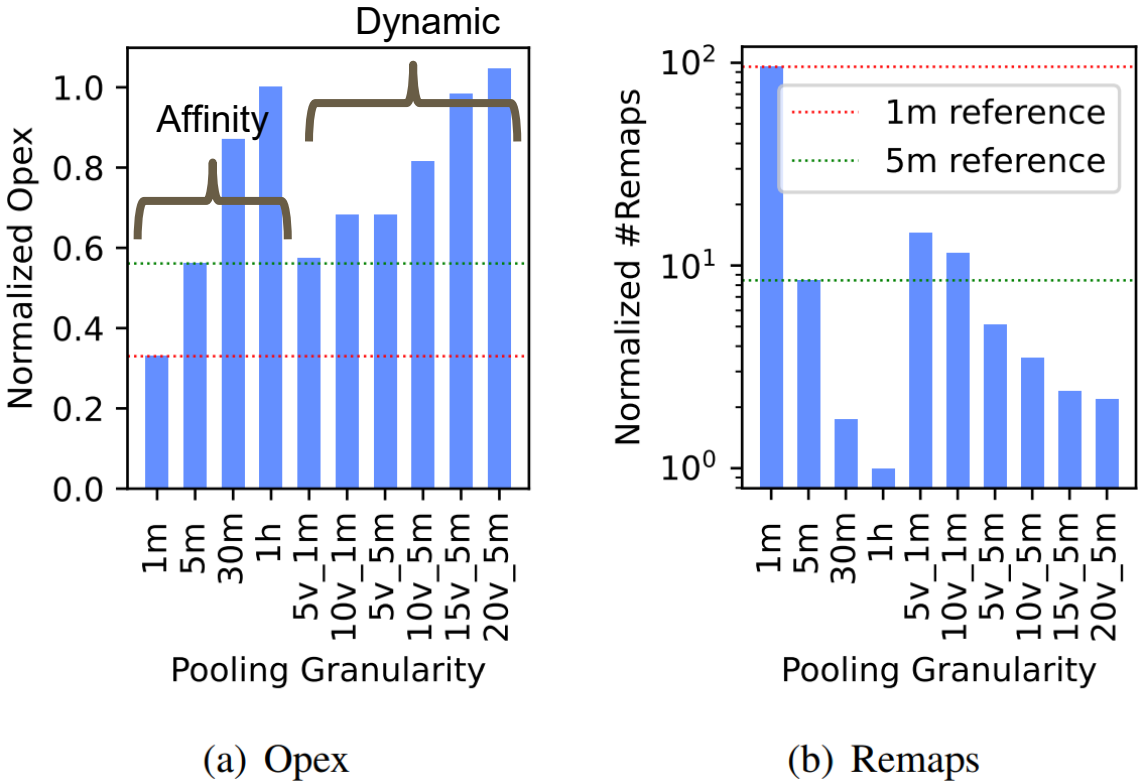


(a) Opex



(b) #Remaps

Evaluation Results – Dynamic Pooling



Conclusion

- **First paper** studying the gains with **DU pooling in 5G**
- Used 5G network **traces from real-world** deployments
- Proposed **interpolation techniques** to get estimated fine-grained data
- Proposed **greedy, affinity** and **dynamic pooling** algos
  - Showed **84% and 94% gains** in Capex and Opex [greedy]
  - Achieved **40% reduction in pooling overheads** while only incurring 22% less in the pooling benefits [affinity and dynamic]
- **Integrating** our strategies by providers should be **possible**
  - Only **feasible-to-collect metrics** are used

Thanks for your attention

Any Questions?

Paper (slides coming soon)

Key Points-

- Study of DU pooling benefits using real-world 5G traces
- Interpolation techniques
- Affinity & Dynamic pooling
  - 84% and 94% gains in Capex & Opex
  - 40% reduction in pooling overheads
- Feasible-to-collect metrics are used