Adaptive Broadcast Scheduling Scheme for High-Definition Map Tile Dissemination in Vehicular Networks

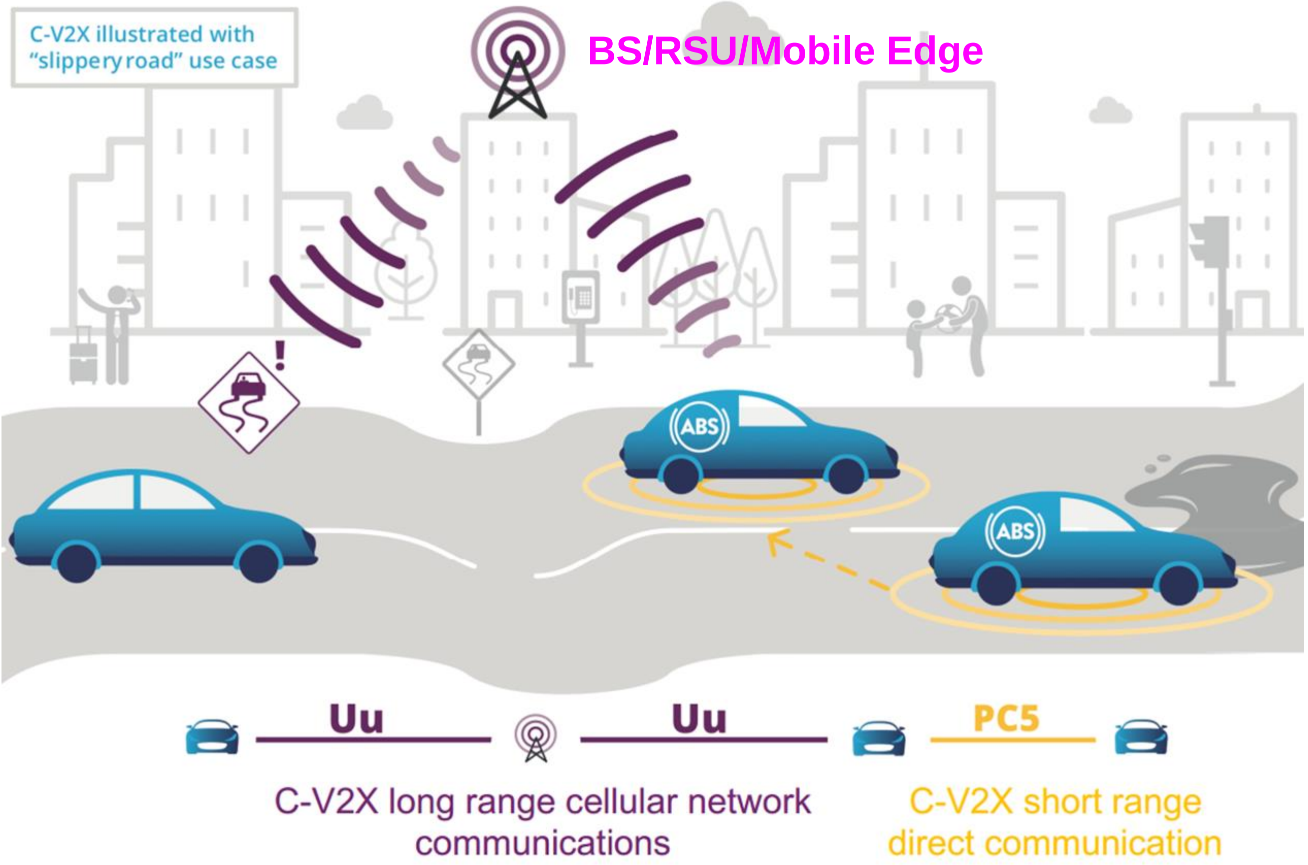
Madhuri Annavazzala, Supriya Dilip Tambe,
Antony Franklin A, and **Bheemarjuna Reddy Tamma**
Department of Computer Science and Engineering

Indian Institute of Technology Hyderabad

Outline

1. Introduction
2. Literature Review
3. Problem Statement
4. System model
5. Optimization Model
6. Proposed Adaptive Broadcast Scheme (ABS)
7. Toy Example
8. Experiment Setup
9. Performance Results

Cellular-V2X for ITS Use cases



HD Maps for ITS Use cases

- High-definition (HD) maps are highly accurate (~ cm) → autonomous vehicles [1][2]
- Terminology: Layer, Grid, Tile, Segment (.osm, .pcd)
- Need for distribution over wireless networks [3]
    - Localization, Path planning
    - Control, Perception, etc
- Network delay ~ 100 millisec
- HD Map is BW-hungry and time-sensitive app

Table 1: Components of a HD Map

Data style	Data content	Update frequency
Static landmark data	Road facilities, surrounding buildings, trees, etc.	Months
Dynamic traffic data	Congestion, temporary speed limit etc.	Seconds or minutes
Static road data	Road and lane details (curvature, slope, etc.)	Days or months
Real-time environment data	Speed, position and direction of pedestrians and vehicles	Seconds

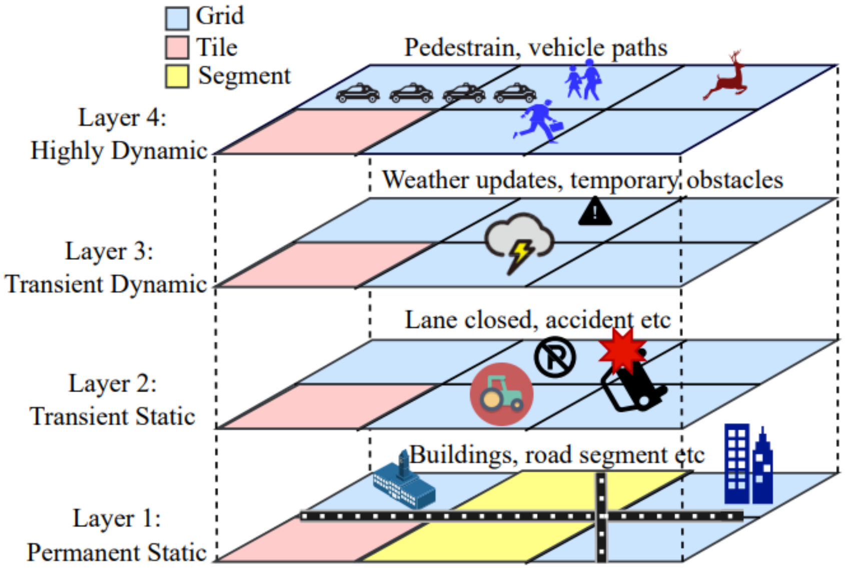


Figure 1: Example of a HD Map

Literature Review and Problem Statement

What is the Academia looking at?

1. Caching based solutions to quickly deliver the maps from RSUs [4][5][6][7][8][14]
2. Request-reply mode (Unicast) [9]

What is the Industry looking at?

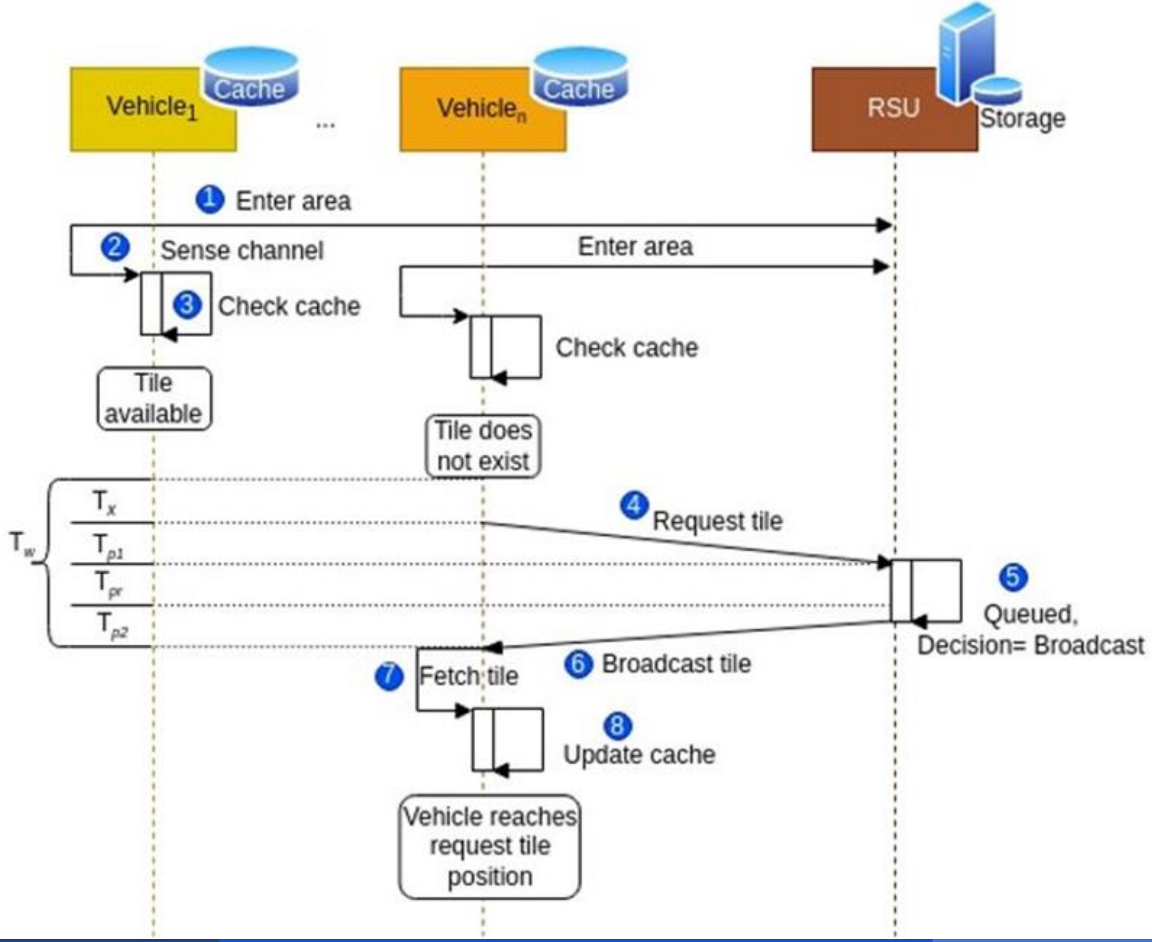
1. AI-based solutions for perception, planning, localization [10][11][12][13]
2. Publish-subscribe mode (Groupcast) [15]

What are they NOT looking at?

1. Properties of HD maps: Time sensitive, Non-private, Customized layer properties (size, validity, importance/priority)
2. Efficient distribution of HD maps wirelessly
    a. Modes of distribution: Unicast vs Groupcast vs Broadcast
        i. Network delay and Bandwidth tradeoff
3. **Problem statement: Adaptive broadcast scheduling that meets delay constraints with minimal bandwidth consumption**

Notation	Description
$\mathcal{V}$	Set of v vehicles.
$\mathcal{S}$	Set of s slots.
$\mathcal{R}$	Set of r tile requests.
N	Number of requests in $\mathcal{R}$
$\mathcal{L}$	Set of l layers.
Q	Maximum priority queue to order tile requests.
O	Ordered set of tile requests.
T_x	Transmission delay of vehicle.
T_{p1}	Propagation delay from vehicle to RSU.
T_{p2}	Propagation delay from RSU to vehicle.
T_{pr}	Queuing and processing delay at RSU.
T_{ch}	Time to acquire channel.
T_{tat}	Turn Around Time (TAT) of the requested tile.
T_{pos}	Time required by the vehicle v to reach the requested tile position.
T_b^r	Time at which a tile request r appears on the broadcast channel.
T	Complete duration.
t	Current time.
t_r^a	Arrival time of the tile request r.
t_r^d	Deadline of the tile request r.
T_P	Broadcast period.
B_R	Physical data rate of RSU.
B_v	Physical data rate of vehicle.
D_{RV}	Distance between RSU and vehicle.
D_{VL}	Distance between vehicle and requested tile.
$speed_v$	Speed of the vehicle V (in km/hr).
$size_r$	Size of tile request packet (in bytes).
c	Speed of light.
$Popularity(r)$	Number of times vehicles requested r.
$Priority(r)$	Relative importance of the layer.

System Model



Optimization Model for Broadcasting

Decision Variable

$X_{r,s}^{t} =$ 1, if request *r* is scheduled in slot *s* at time *t*, otherwise 0.

Objective Function

The objective is to minimize the sum of unscheduled requests across all requests (r) and times (t), taking into account the arrival and deadline times for each request:

$$\min \sum_{r \in \mathcal{R}} \left(1 - \sum_{t=t_r^a}^{t_r^d} \sum_{s \in \mathcal{S}} X_{r,s}^{t} \right)$$

Subject to:

•**Constraint 1: Slot Occupancy Constraint:** Each slot can hold only one tile at a time. This constraint ensures that two requests cannot be scheduled in the same slot at the same time.

$$\forall s \in \mathcal{S}, \forall t : \quad \sum_{r \in \mathcal{R}} X_{r,s}^{t} \leq 1.$$

•**Constraint 2: Request Scheduling Constraint:** Every request needs to be scheduled at least once between its arrival time and deadline.

$$\forall r \in \mathcal{R} : \quad \sum_{t=t_r^a}^{t_r^d} \sum_{s \in \mathcal{S}} X_{r,s}^{t} \geq 1.$$

Working of Proposed Adaptive Broadcast Scheme (ABS)

1. Decision Phase
2. Queuing Phase
3. Scheduling Phase

1. Decision Phase

If the requested tile is going to appear on the broadcast schedule of RSU in the near future, ignore the request.

Else insert the request in a Queue for serving soon.

2. Queuing Phase

Prioritize the requests in the **max-Priority queue** using type, popularity and deadline in that order to break ties among tiles having same priority for arriving at the scheduling order.

3. Scheduling Phase

Find an empty slot that is the nearest but before the deadline of tile's request.

If no such slot is found, replace an existing tile in the broadcast schedule (**preempt**) order based on the priority decided in the previous phase.

Proposed Adaptive Broadcast Scheme: Queuing and Scheduling Phases

Algorithm 1: Queuing Phase

Input: Set of requests $\mathcal{R}$, with priorities, popularity, and deadlines
Output: Set of ordered requests, O
Initialize the max-priority queue, Q.
for r *in* $\mathcal{R}$ **do**
    Enqueue r into Q
Initialize an empty list for ordered requests, O.
while Q *is not empty* **do**
    $r_1 \leftarrow$ Dequeue the top element from Q **if** Q *is not empty* **then**
        $r_2 \leftarrow$ Peek the next top element from Q. **if** *Priority*$(r_1) >$ *Priority*(r_2) **then**
            Append r_1 to O;
        else if *Priority*$(r_1) <$ *Priority*(r_2) **then**
            Append r_2 to O; Reinsert r_1 into Q;
        else
            if *Popularity*$(r_1) >$ *Popularity*(r_2) **then**
                Append r_1 to O;
            else if *Popularity*$(r_1) <$ *Popularity*(r_2) **then**
                Append r_2 to O; Reinsert r_1 into Q;
            else
                if *Deadline*$(r_1) <$ *Deadline*(r_2) **then**
                    Append r_1 to O;
                else
                    Append r_2 to O; Reinsert r_1 into Q;
    else
        Append r_1 to O;

Algorithm 2: Scheduling phase

Input: Request r with its properties
Output: Request r scheduled in slot s at time t
isPcdSlotForOsmFound $\leftarrow$ False
isOsmSlotForOsmFound $\leftarrow$ False
if $Type(r) = osm$ **then**
    for i *in* *range(a,d)* **do**
        if $X^t_{r,s_i} = l_{pcd}$ **then**
            $X^t_{r,s_i} \leftarrow 1$;
            isPcdSlotForOsmFound $\leftarrow$ True
            Break;
    if $isPcdSlotForOsmFound$ = *False* **then**
        for i *in* *range(a,d)* **do**
            if $X^t_{r,s_i} = l_{osm}$ & $Popularity(r_i) < Popularity(r)$ **then**
                $S_i \leftarrow r$; $X^t_{r,s_i} \leftarrow 1$;
                isOsmSlotForOsmFound $\leftarrow$ True
                Break;
    if $isOsmSlotForOsmFound$ = *False* & $isPcdSlotForOsmFound$ = *False* **then**
        Schedule miss for r.
if $Type(r) = pcd$ **then**
    for i *in* *range(a,d)* **do**
        if $X^t_{r,s_i} = l_{pcd}$ & $Popularity(r_i) < Popularity(r)$ **then**
            $S_i \leftarrow r$; $X^t_{r,s_i} \leftarrow 1$;
            isPcdSlotForOsmFound $\leftarrow$ True
            Break;

Toy example for tile distribution (Unicast Request & Broadcast Reply Mode)

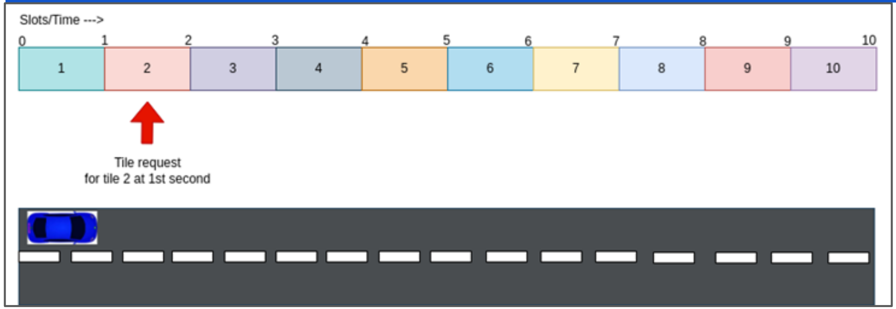


Figure A : Best-case wait time in Static Broadcast Scheduling

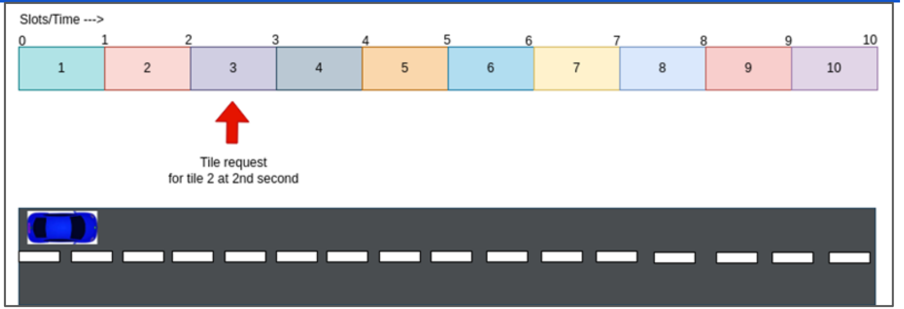


Figure B: Worst-case wait time in Static Broadcast Scheduling

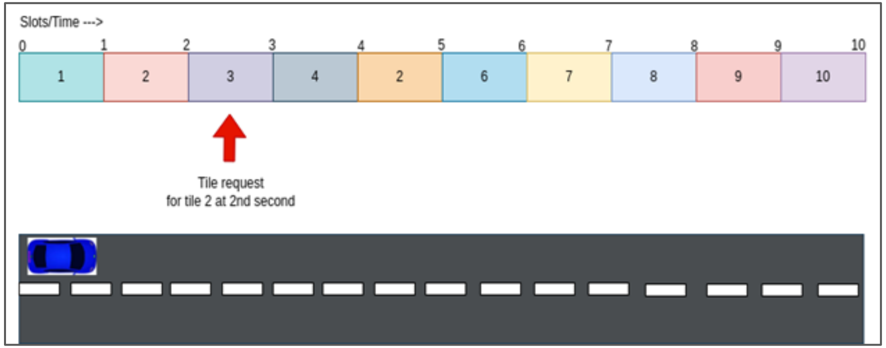


Figure C: Increment frequency of broadcast of the required tile in case of Adaptive Scheduling

Wait time = [Slot in which tile published next - Request Time slot]

Case 1: Best-case wait time in Static scheme= 0

Case 2: Worst case wait time in Static scheme= 9 slots

Case 3: Wait time in Adaptive scheme (our work) = 2 slots

Comparison of Various Broadcasting Schemes

- Broadcast Scheduling Schemes studied:
  - Static (Basic) Scheduling Scheme
  - Adaptive Scheduling Scheme (ABS)
  - Optimal Scheduling Scheme

Table: Delay comparison between various approaches

	Basic	Adaptive	Optimal
Direct Hit	$T_x+T_{p1}+T_{p2}$	$T_x+T_{p1}+T_{p2}$	$T_x+T_{p1}+T_{p2}$
Hit with wait	$T_x+T_{p1}+T_{p2}+T_b$	$T_x+T_{p1}+T_{p2}+T_b$	NA
Scheduled Hit	NA	$T_x+T_{p1}+T_{p2}+T_{pr}$	NA
Miss	$T_x+T_{p1}+T_{p2}+T_l$	$T_x+T_{p1}+T_{p2}+T_l$	$T_x+T_{p1}+T_{p2}$

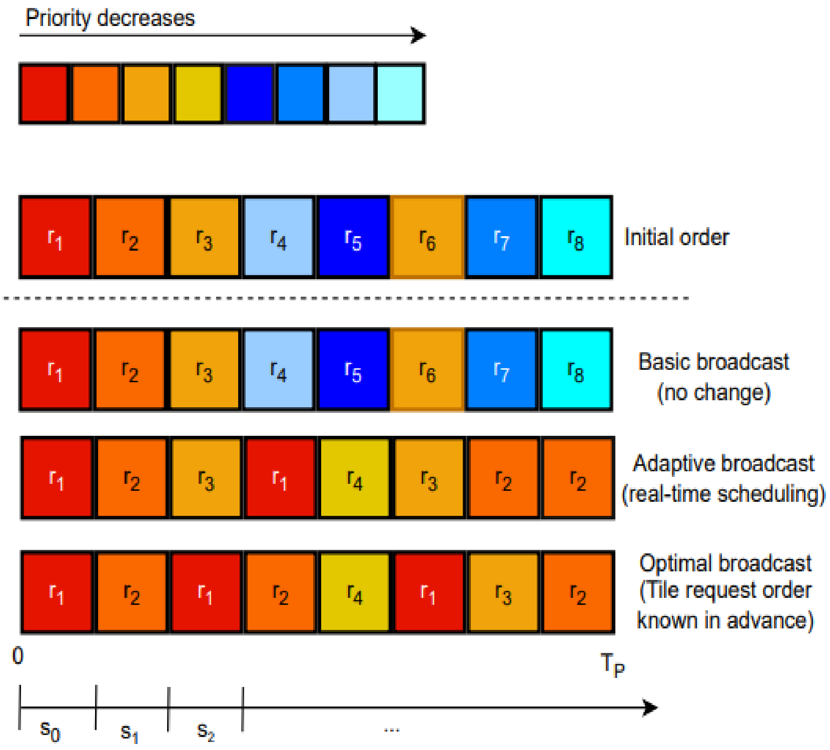
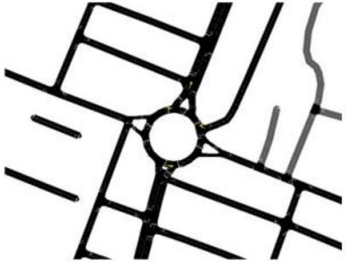


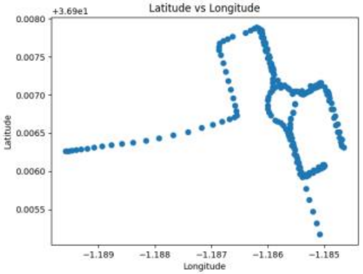
Figure: Demonstration of considered approaches

Experiment Setup

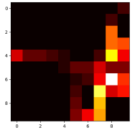
Figure A: SUMO Setup for HD map tile request generation by vehicles in urban road segment



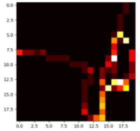
(a) An urban road segment scenario created in SUMO.



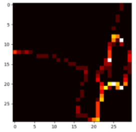
(b) Mapping of the vehicles trajectories.



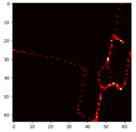
(a) (10x10)



(b) (20x20)



(c) (30x30)



(d) (64x64)

Figure B: Heat map generated for different tile sizes

Table: Experimental parameters

Parameter	Description
System Type	NR-V2X Mode 1
Scenario	Urban intersection
Number of Vehicles	10
Total tile requests	1500
Speed of vehicles	30 – 80 KMPH
RSU data rate	100Mbps
Grid area considered	1000×1000 m^2 divided into 10×10, 20×20, 30×30, and 64×64 tiles.
Number of layers	2 layers each (.osm and .pcd)
Size of .osm layer	2KB
Size of .pcd layer	10KB
Size of tile request	2KB
Vehicle data rate	3 – 5 Mbps
Mobility model	SUMO based location
Broadcast cycle	200ms (10×10), 800ms (20×20), 1800ms (30×30) and 8192ms (64×64)

Performance Results: The Hit Rate Saga!

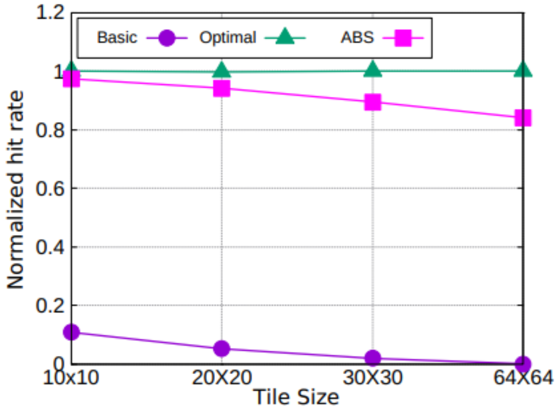


Figure: Impact on Hit rate

Hit rate for Basic vs ABS vs Optimal

1. Tile size ∝ (1/ number of tiles), broadcast period increases with increasing tiles.
2. ABS performs better for the same tile size.
3. Optimal beforms the best, but pattern to be known in advance.

Speed vs Hit rate

1. ABS supports <10% miss for speeds up to 80kmph.
2. For speeds >100kmph, 1GB/mile data; additional bandwidth required

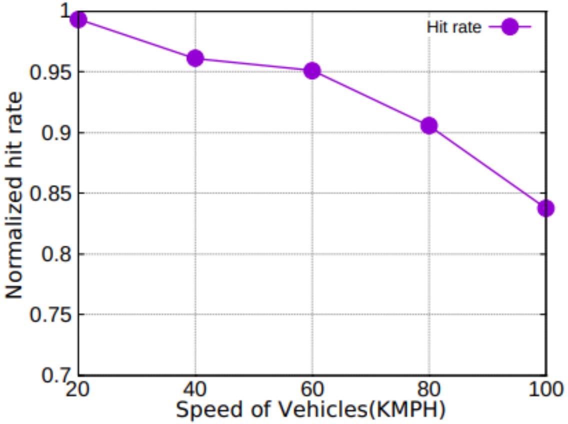


Figure:Variation in hit rate with speed of vehicles

Performance Results: Impact on Turn Around Time (TAT)

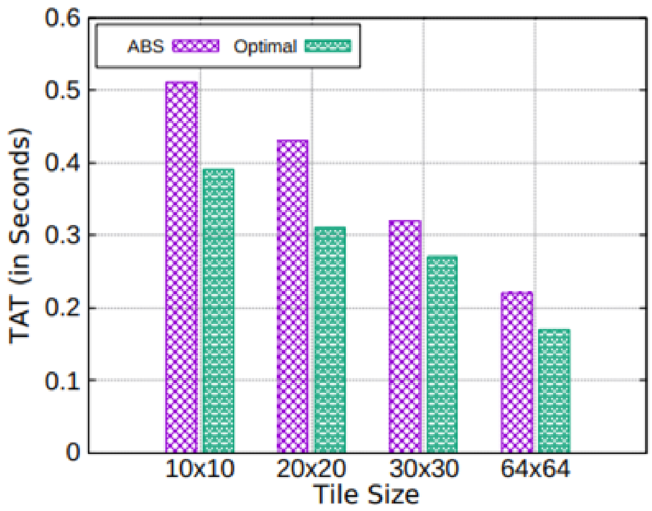


Figure:Impact on TAT by varying tile sizes

Tile size vs TAT

1. Optimal needs advance knowledge of tile requests which might not be viable in real-world scenarios.
2. ABS has an average TAT of ~0.5 sec per request for all tile sizes.
3. Suitable for critical V2X applications.

Conclusion and Future Directions

Conclusions

1. Proposed ABS scheme for effective dissemination of HD map data in dense NR V2X environments.
2. Optimization model that elevates the problem to an NP-Hard challenge, paralleling multi-objective scheduling scheme.
3. Emulation results of ABS scheme show promise in reducing schedule misses as much as by 50% compared to the conventional periodic broadcast and minimizing turn around time per request.

Future Work

1. Future work includes enhancing ABS scheme using mini-slot scheduling to address challenges related to request starvation for lower priority requests
2. This technique involves distributing parts of different tiles within a single slot, offering a potential solution to improve request handling efficiency.

References

[1] ABI Research and HERE Technologies, “THE FUTURE OF MAPS: TECHNOLOGIES, PROCESSES, AND ECOSYSTEM", 2018. https://www.here.com/file/7766/download?token=dwOqPUix

[2] WebCite, “Why We’re Mapping Down to 20 cm Accuracy on Roads", 2014. http://www.webcitation.org/72UbZ4QJh

[3] Society of Automotive Engineers International, “SAE Standards News: J3016 automated-driving graphic update”, 2019. https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic.

[4] H. Khelifi, S. Luo, B. Nour and H. Moungla, "A QoS-Aware Cache Replacement Policy for Vehicular Named Data Networks," 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9013461

[5] H. Khelifi, S. Luo, B. Nour and H. Moungla, "A QoS-Aware Cache Replacement Policy for Vehicular Named Data Networks," 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9013461.

[6] Sato, K., Koita, T. & Fukuda, A. Broadcasted Location-Aware Data Cache for Vehicular Application. J Embedded Systems 2007, 029391 (2007). https://doi.org/10.1155/2007/29391

[7] J. Chen, H. Wu, P. Yang, F. Lyu and X. Shen, "Cooperative Edge Caching With Location-Based and Popular Contents for Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 69, no. 9, pp. 10291-10305, Sept. 2020, doi: 10.1109/TVT.2020.3004720.

[8] X. Xu, S. Gao and M. Tao, "Distributed Online Caching for High-Definition Maps in Autonomous Driving Systems," in IEEE Wireless Communications Letters, vol. 10, no. 7, pp. 1390-1394, July 2021, doi: 10.1109/LWC.2021.3068498.

[10] F. Wu, W. Yang, J. Lu, F. Lyu, J. Ren and Y. Zhang, "RLSS: A Reinforcement Learning Scheme for HD Map Data Source Selection in Vehicular NDN," in IEEE Internet of Things Journal, vol. 9, no. 13, pp. 10777-10791, 1 July1, 2022, doi: 10.1109/JIOT.2021.3128636.

[11] Qixia Hao, Jiaxin Zeng, Xiaobo Zhou, and Tie Qiu. 2022. Freshness-Aware High-Definition Map Caching with Distributed MAMAB in Internet of Vehicles. In Wireless Algorithms, Systems, and Applications: 17th International Conference, WASA 2022, Dalian, China, November 24–26, 2022, Proceedings, Part III. Springer-Verlag, Berlin, Heidelberg, 273–284. https://doi.org/10.1007/978-3-031-19211-1_23

[12] L. Toka, Mark Konr, Istvan Pelle, Bal Sonkoly, Marcell Szab, Bhavishya Sharma, Shashwat Kumar, Madhuri Annavazzala, Sree Teja Deekshitula, Antony Franklin A, "5G on the roads: optimizing the latency of federated analysis in vehicular edge networks," NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 2023, pp. 1-5, doi: 10.1109/NOMS56928.2023.10154348.

[13] Qiang Liu, Tao Han, Jiang Linda Xie, and BaekGyu Kim. 2021. LiveMap: Real-Time Dynamic Map in Automotive Edge Computing. In IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. IEEE Press, 1–10. https://doi.org/10.1109/INFOCOM42981.2021.9488872

[14] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu and Y. Jia, "An Edge Caching Scheme to Distribute Content in Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 67, no. 6, pp. 5346-5356, June 2018, doi: 10.1109/TVT.2018.2824345.

[15] https://prod.ucwe.capgemini.com/wp-content/uploads/2022/11/CV2X-white-paper_Nov-2022.pdf

Acknowledgments

This work was supported by DST National Mission Interdisciplinary Cyber-Physical Systems (NM-ICPS)'s Technology Innovation Hub on Autonomous Navigation and Data Acquisition Systems (TiHAN), Indian Institute of Technology (IIT) Hyderabad

THANK YOU!

tbr@cse.iith.ac.in

Visit NeWS Lab @ IIT Hyderabad for more details: https://newslab.iith.ac.in/