Duplicate Packet Suppression in Multi-Access NDN Network using Reinforcement Learning

Bidhya Shrestha
Saurab Dulal
Lan Wang

INTRODUCTION

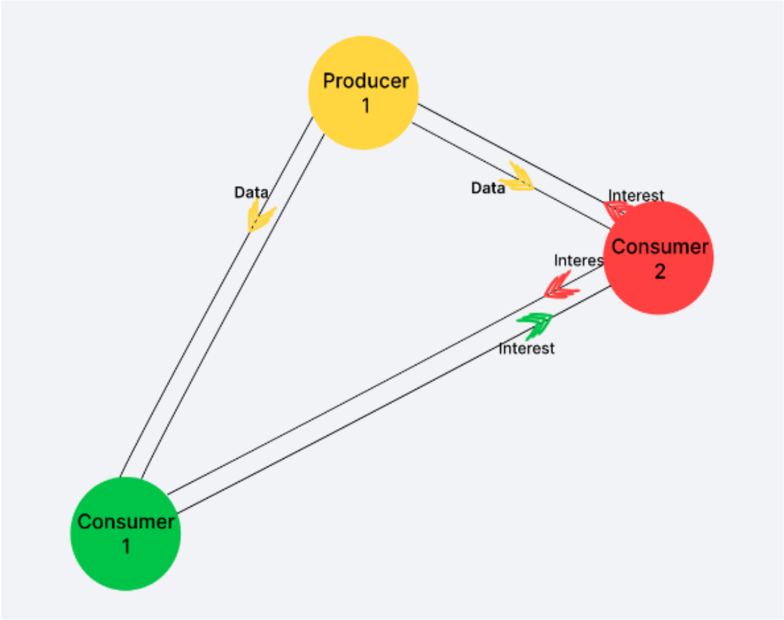
- Data-centric communication is more suitable for today's applications than host-centric communication.
  - vehicular networks
  - video conferencing
  - social networking
- Named Data Networking is a promising Internet architecture that focuses on data rather than location of host.
  - Built-in data multicast
  - In-network caching
  - Multipath forwarding
  - Secured data

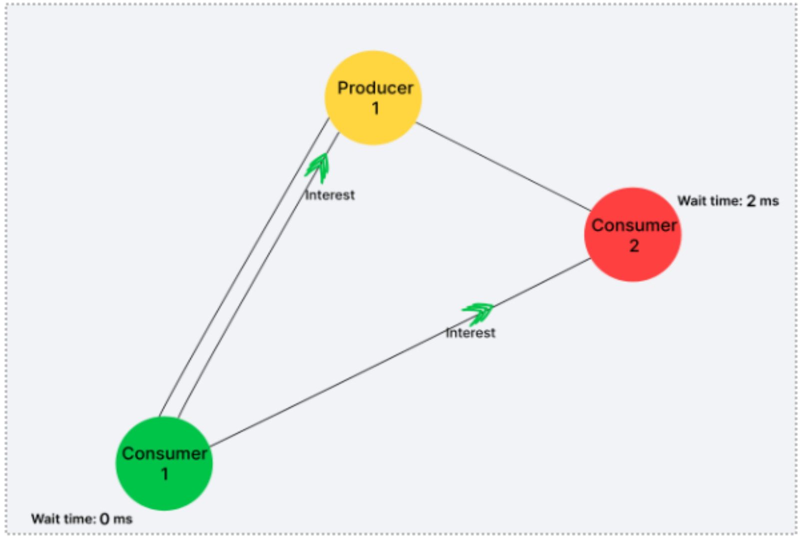
Problem Statement

- Multicasting in multi-access networks (e.g. Wifi) supports group communication and reduces unnecessary transmission

- Without proper coordination, many Interest and Data packets may flow across the network
- Even with just **two consumers**, **multiple Interest packets** can be generated for the **same data**, and increasing the number of nodes can cause **packet flooding**
- The same applies to **Data packets**, leading to further redundancy

- If nodes **briefly wait before forwarding packets**, they may suppress duplicates and **reduce network overhead**





Existing Solution

- Some researchers [1,2, 3] used random wait time between fixed [min, max] interval before sending the Interest/Data packets
  - If interest is overheard during the wait time, the interest transmission is canceled
  - Not adaptive to the number of consumers and producers
- Dulal et. al. introduced Adaptive suppression mechanism [4] that relies solely on observation of duplicate packet counts.
  - Unclear how to tune ADS parameters like duplicate threshold, smoothing factor, multiplicative factor etc. for different networks

Design Goal and Solution

- Goal: Reduce redundant NDN traffic in multi-access networks without negatively impacting data transfer time (one-hop scenario)
- Solution: Reinforcement Learning (RL) is suitable for dynamic environments.
  - Each node experiments with different suppression time and learns a policy that reduces duplicates under varying network conditions.
  - The reward guides the node to increase or decrease the suppression time based on the network state

System Design

- Two main components:
  - NDN Forwarding Daemon
  - Node with RL module
- NFD handles the NDN packets across the network managing FIB, PIT and CS
- Node with RL module acts as **agent** that waits briefly before forwarding packets

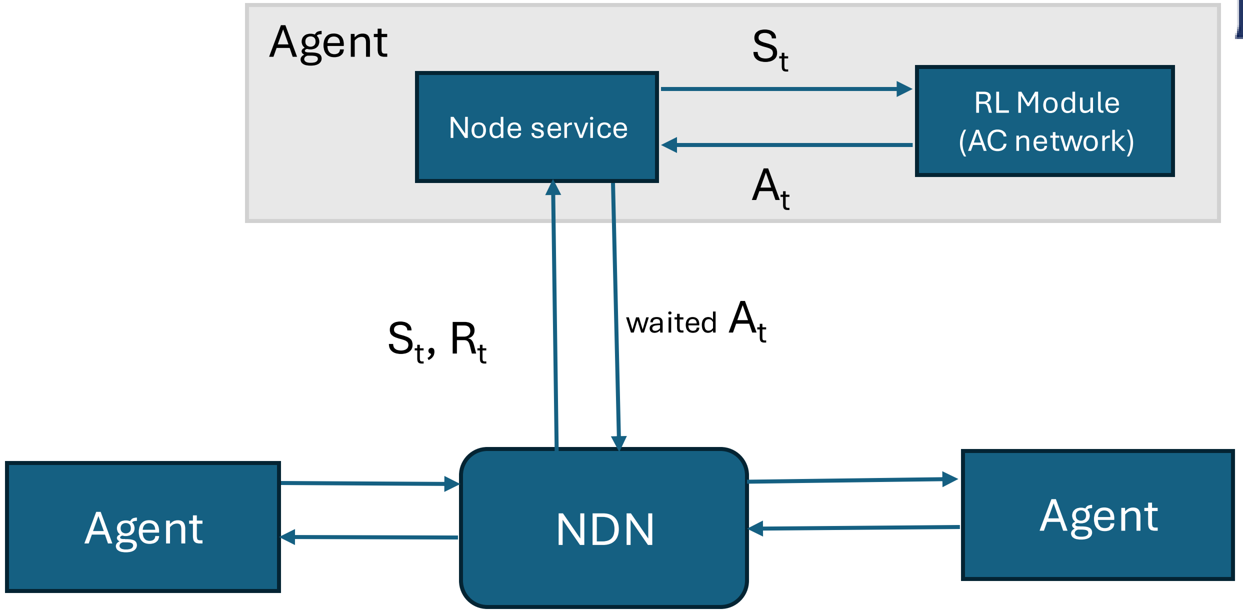


Fig: Computation of Suppression time as RL task

- The Agent observes the current state (S_t)
- Takes the action (A_t) which is to wait for the computed suppression time
- Environment is transitioned to next State (S_{t+1}).
- The Agent receives reward (R_t) based on outcome

RL Module (Actor Critic Network)

- **State** captures important environment information the agent needs to decide how long to wait before forwarding packet
  - **np –** Name Prefix
  - **st –** Suppression Time waited by agent
  - **dc –** Duplicate Count
  - **p –** Packet Type
  - **wf –** Boolean status indicating if the node forwarded the packet
- NDN represents the **Environment**

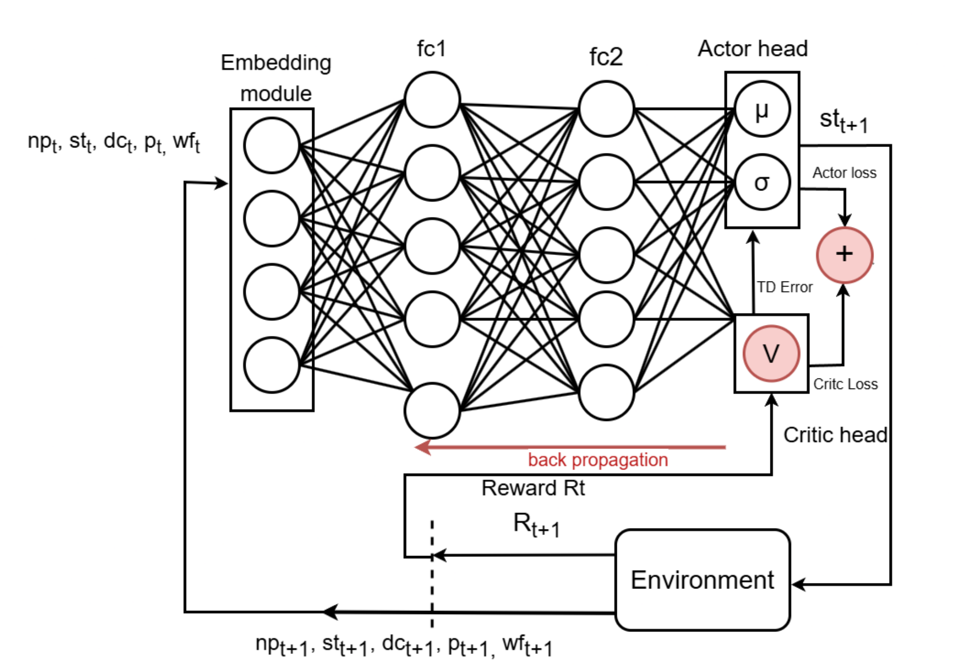


Fig: Actor Critic Network

- Embedding layer converts the input state to dense vector representation
- Fully connected layers extract relevant features from the input
- Actor head computes the suppression time (action to take)
- Critic head evaluates the action by providing feedback to the Actor head, indicating how good the action was.

Model Learning

- Initialize the weights of the neural network
- Collect Experience (sequence of S_t, A_t, R_t, S_{t+1},…)
- After taking action, Critic evaluates action using TD Error:

  TD Error (δ) = Rt + $\gamma V(S_{t+1}) - V(S_t)$

  where V(S) represents how good it is to be in the state S and γ is the discount factor
- Update the Actor's policy using gradient ascent method:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi(a|s;\theta)\delta$$

  where Θ are parameters of actor, α is learning rate, $\nabla_\theta \log \pi(a|s;\theta)$ is the gradient of the log-probability of taking action a in state s.
- Critic is updated to minimize mean squared error between the predicted value ($V(S_t)$) and actual outcome ($R + \gamma V(S_{t+1})$)

$$\phi \leftarrow \phi - \alpha \nabla_\phi(\delta^2)$$

  where ϕ is critic parameter

Measurement Module

- RL module retrieves current state of environment from the Measurement table
- Measurement table stores:
  - Name prefix
  - Number of duplicate packets
  - Forwarded status of the packet
- Suppression time is computed when a measurement table entry is removed
- Data measurement entry is removed when its record expires
- An Interest measurement entry is removed either:
  - When it is satisfied by a Data packet
  - When the record is expired

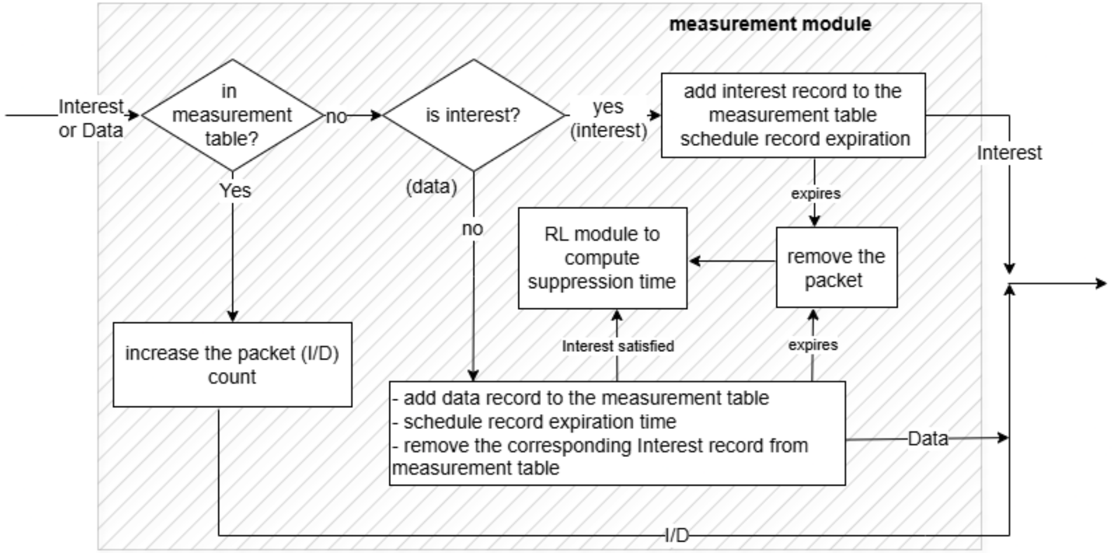


Fig: Measurement Module

Overview of packet processing pipeline

Incoming packet processing pipeline

- The incoming packet is checked if the same packet is scheduled for the transmission
- If yes, cancel the schedule, this decrease duplicate interest packet or data packet

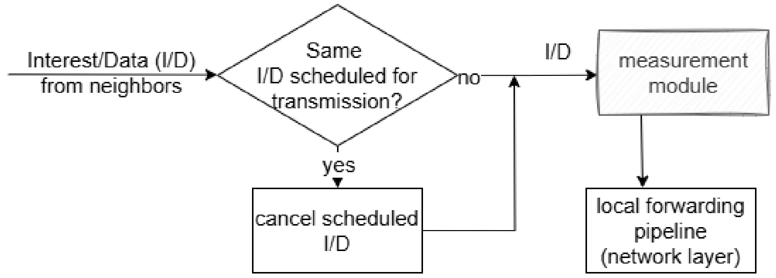


Fig: Incoming Packet Processing Pipeline

Outgoing packet processing pipeline

- Before forwarding the packet, check if the packet is already in measurement table, indicating if it had been recently forwarded (by this node or others)
- If yes, drop the forwarding, decreasing the possible duplicates

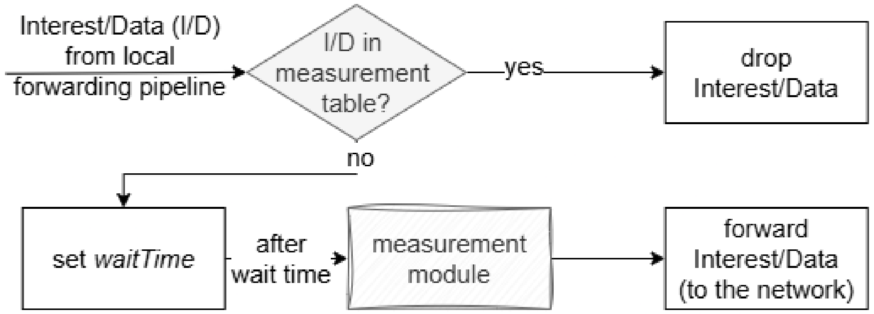


Fig: Outgoing Packet Processing Pipeline

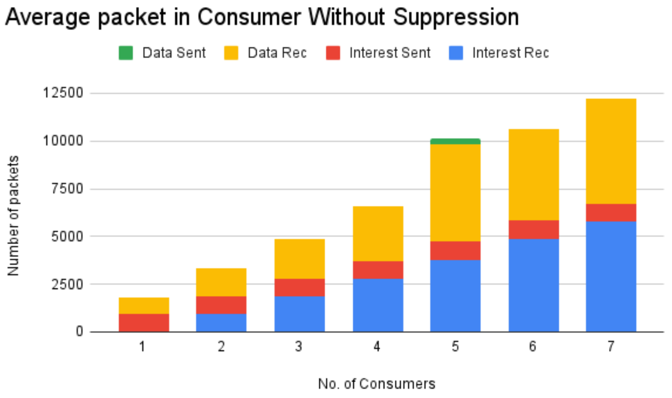
Experiment

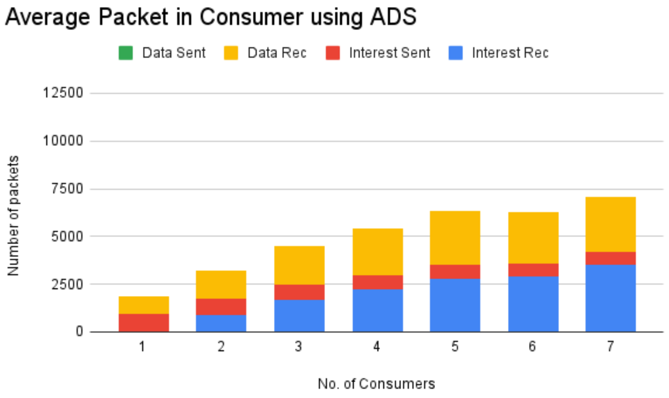
- Topology of the experiment:
  - 1 producer, 1-7 consumer
  - File size: 1 MB
  - One hop scenario, multi-access network

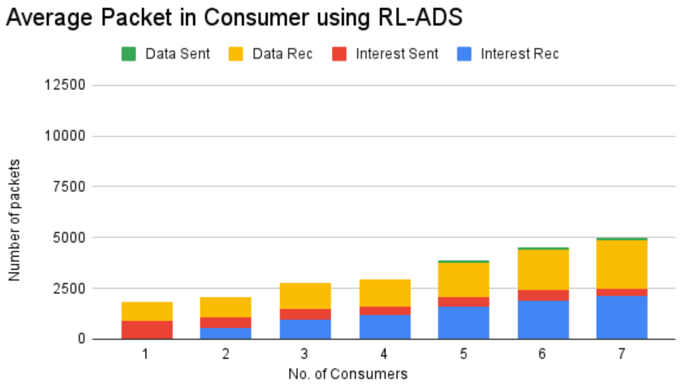
- Emulator: Mini-NDN Wifi
- Use catchunks/putchunks over multi-access link to publish and fetch packets

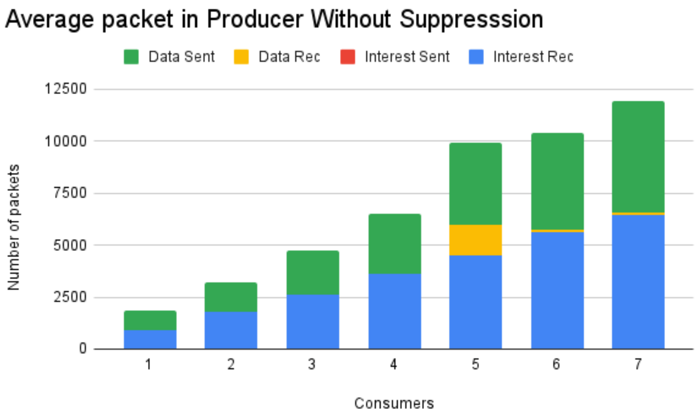
Result

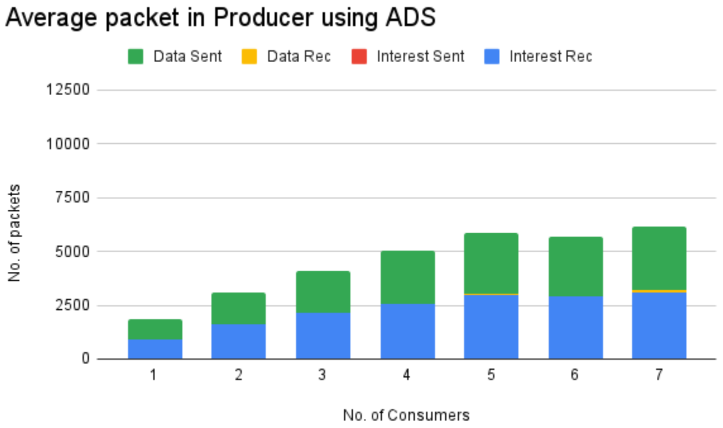
- As the number of nodes in the network increases, RL-ADS reduces the number of packets more effectively.

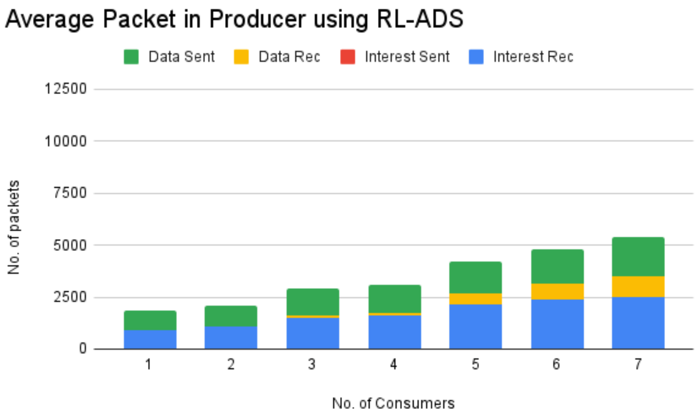






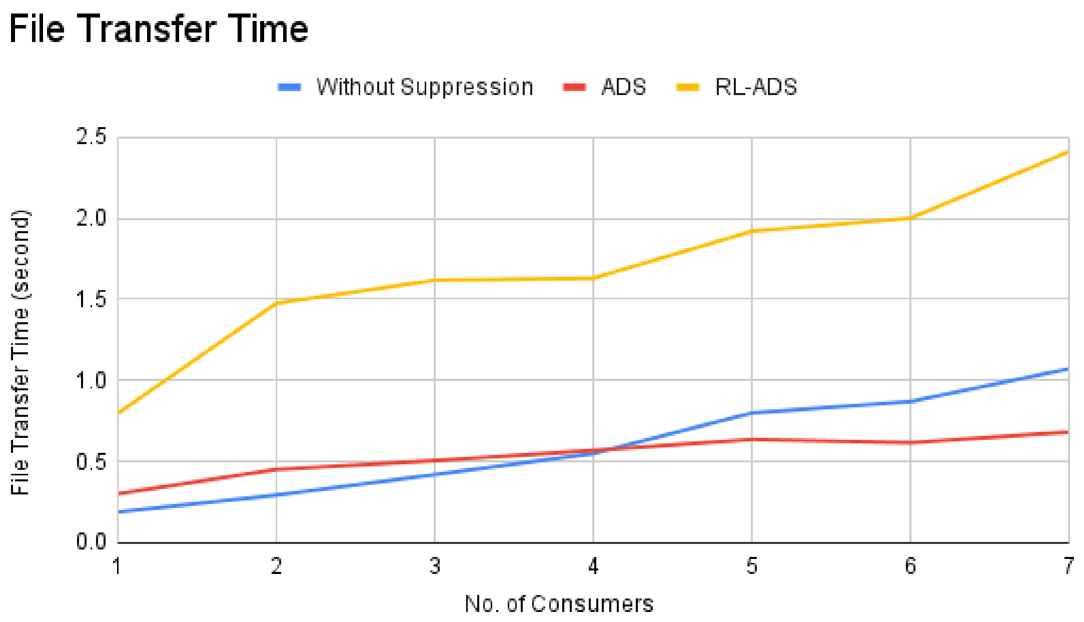






Result

- ADS is the most efficient approach for minimizing file transfer time as the number of consumers increases
- RL-ADS has the highest file transfer time:
  - FIFO-based communication between C++ and Python
  - Neural network computational overhead during decision-making also causes some delay
  - State does not include any observation related to round trip time. As a result, the reward does not file transfer time into account.



Conclusion

- Designed Reinforcement learning based suppression mechanism in NDN forwarding
- Implemented Actor Critic Algorithm to compute the suppression time
- Compared the performance of Actor Critic Algorithm with Adaptive Suppression Mechanism and No suppression Forwarding mechanism

Future work

- Online finetuning of the model to adapt in different network conditions
- Improve inter process communication
- Explore other RL algorithms

References

1. Amadeo, M., Campolo, C., Molinaro, A., and Mitton, N. Named data networking: A natural design for data collection in wireless sensor networks. In 2013 IFIP wireless days (WD) (2013), IEEE, pp. 1–6.
2. Podder, P., Gupta, S. D., Neishaboori, A., and Afanasyev, A. sv2pc: On scaling lte-based vehicle-to-pedestrian communication using ndn. In NDN Community Meeting, NDNComm (2021), NIST.
3. Wang,L.,Afanasyev,A.,Kuntz,R.,Vuyyuru,R.,Wakikawa,R.,and Zhang, L. Rapid traffic information dissemination using named data. In Proceedings of the 1st ACM workshop on emerging name-oriented mobile networking design-architecture, algorithms, and applications (2012), pp. 7–12.
4. Dulal, S., & Wang, L. (2023, October). Reining in redundant traffic through adaptive duplicate suppression in multi-access ndn networks. In Proceedings of the 10th ACM Conference on Information-Centric Networking (pp. 78-87).

Thank you