

Bachelor Thesis – Final Presentation

Multipath Transmission for CCN in Vehicular ad-hoc Networks

Thomas Kolonko
Prof. Dr. Torsten Braun

Communication and Distributed Systems
University of Bern

Table of Contents

- > Introduction to Named Data Networking (NDN)
- > Simulation Framework: ns-3 and ndnSIM
- > Current Implementation in ndnSIM v2.0
- > Goals and Implementation
- > Conclusion and Future Works
- > Future Works implemented
- > Questions

Introduction to Named Data Networking (NDN)

- > Current network is dominated by content distribution
- > It is based on host-to-host communication

Introduction to Named Data Networking (NDN)

- > Current network is dominated by content distribution
- > It is based on host-to-host communication

Problems:

- > Huge amount of data that needs to be re-distributed

Introduction to Named Data Networking (NDN)

- > Current network is dominated by content distribution
- > It is based on host-to-host communication

Problems:

- > Huge amount of data that needs to be re-distributed
- > Mobility and the Internet of Things (IoT)

Introduction to Named Data Networking (NDN)

- > Current network is dominated by content distribution
- > It is based on host-to-host communication

Problems:

- > Huge amount of data that needs to be re-distributed
- > Mobility and the Internet of Things (IoT)
- > Security Considerations

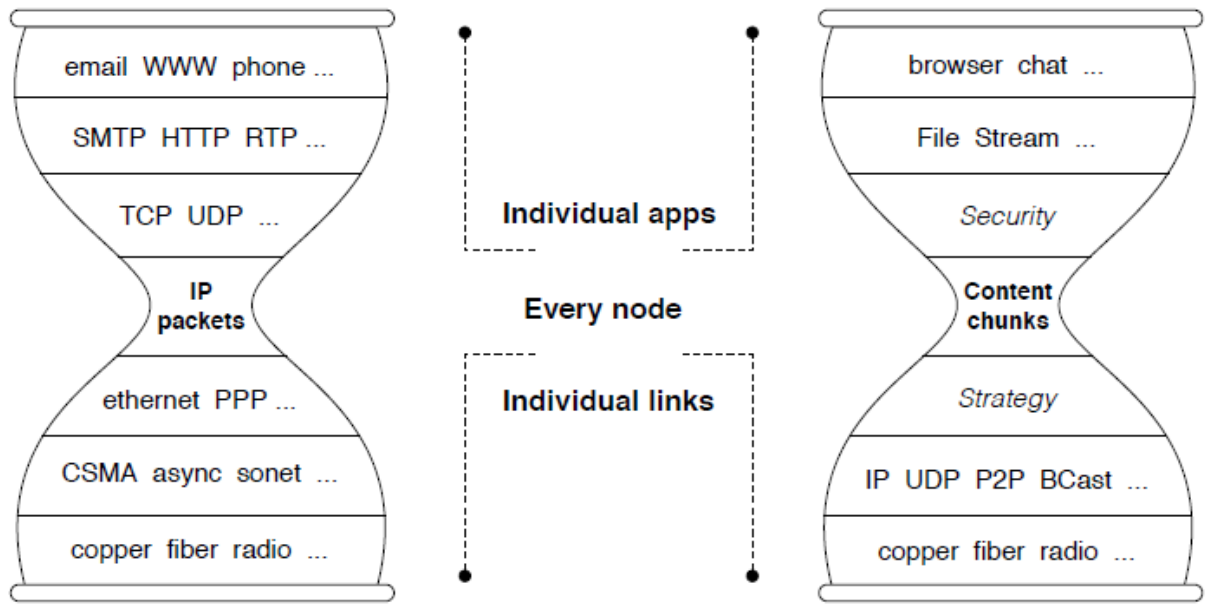
Introduction to Named Data Networking (NDN)

- > Current network is dominated by content distribution
- > It is based on host-to-host communication

Problems:

- > Huge amount of data that need to be re-distributed
- > Mobility and the Internet of Things (IoT)
- > Security Considerations
- > “HTTPS by default” still not used by 79% of top websites ([Google Transparency Report](#))

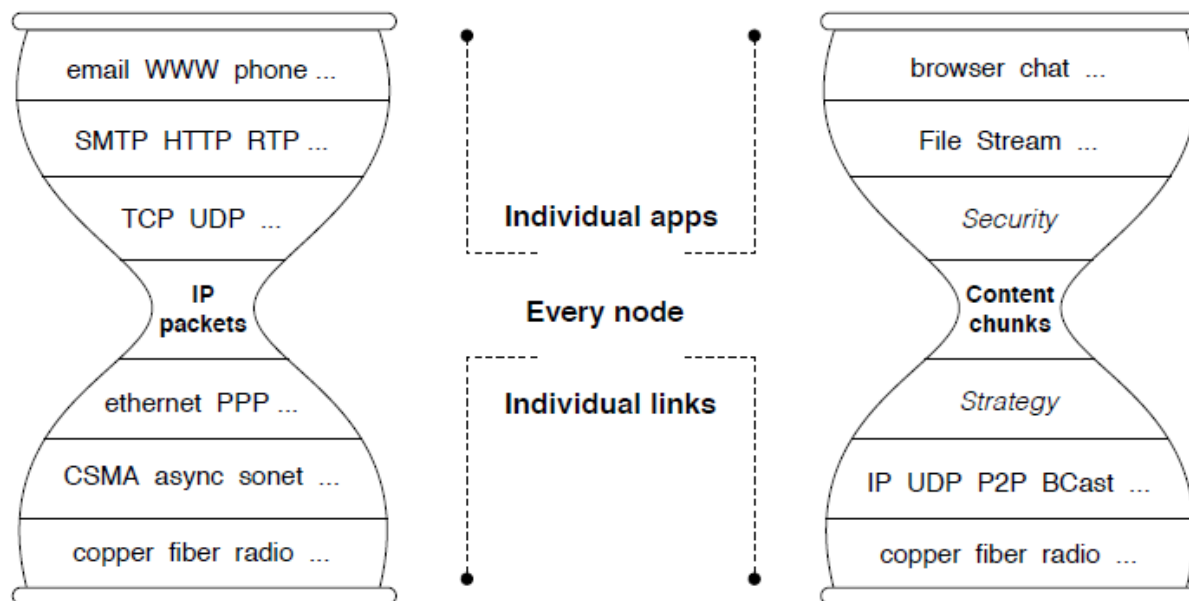
Components of NDN



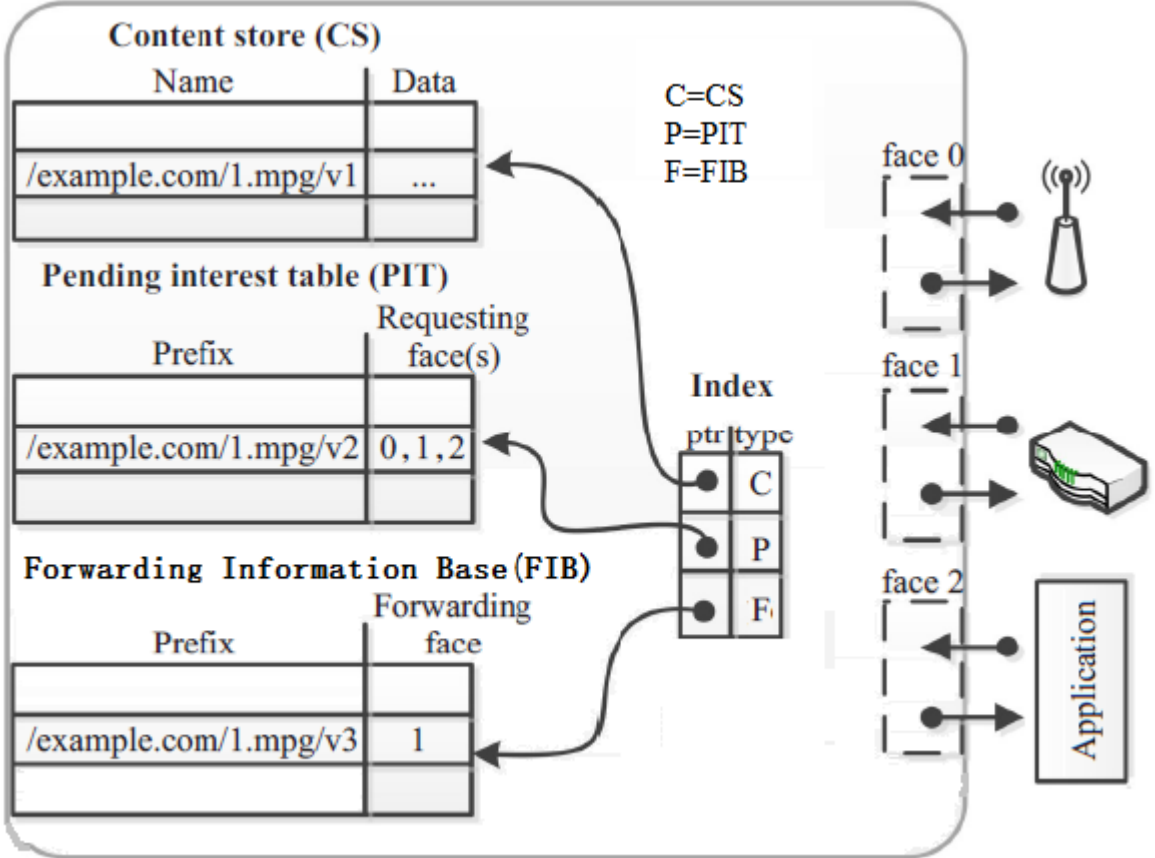
Components of NDN

- > Content
- > Consumer
- > Producer
- > Interest
- > Data
- > Node
- > Interface
- > Router

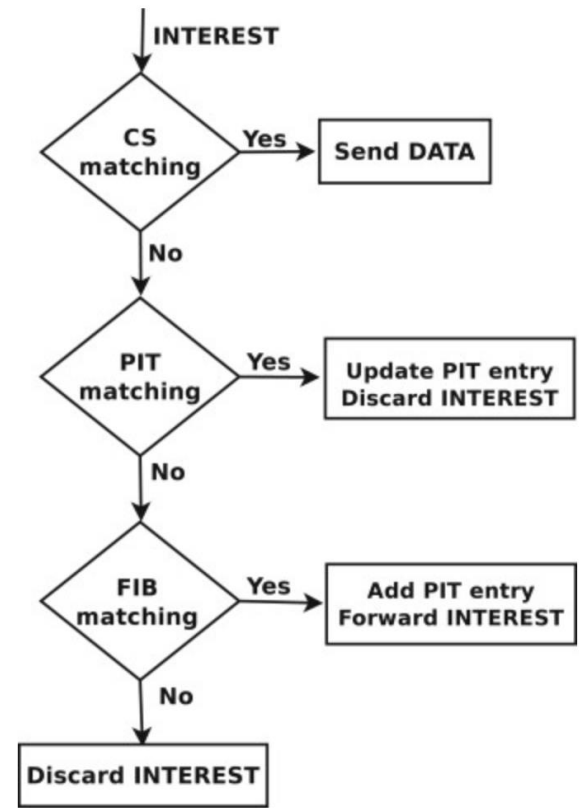
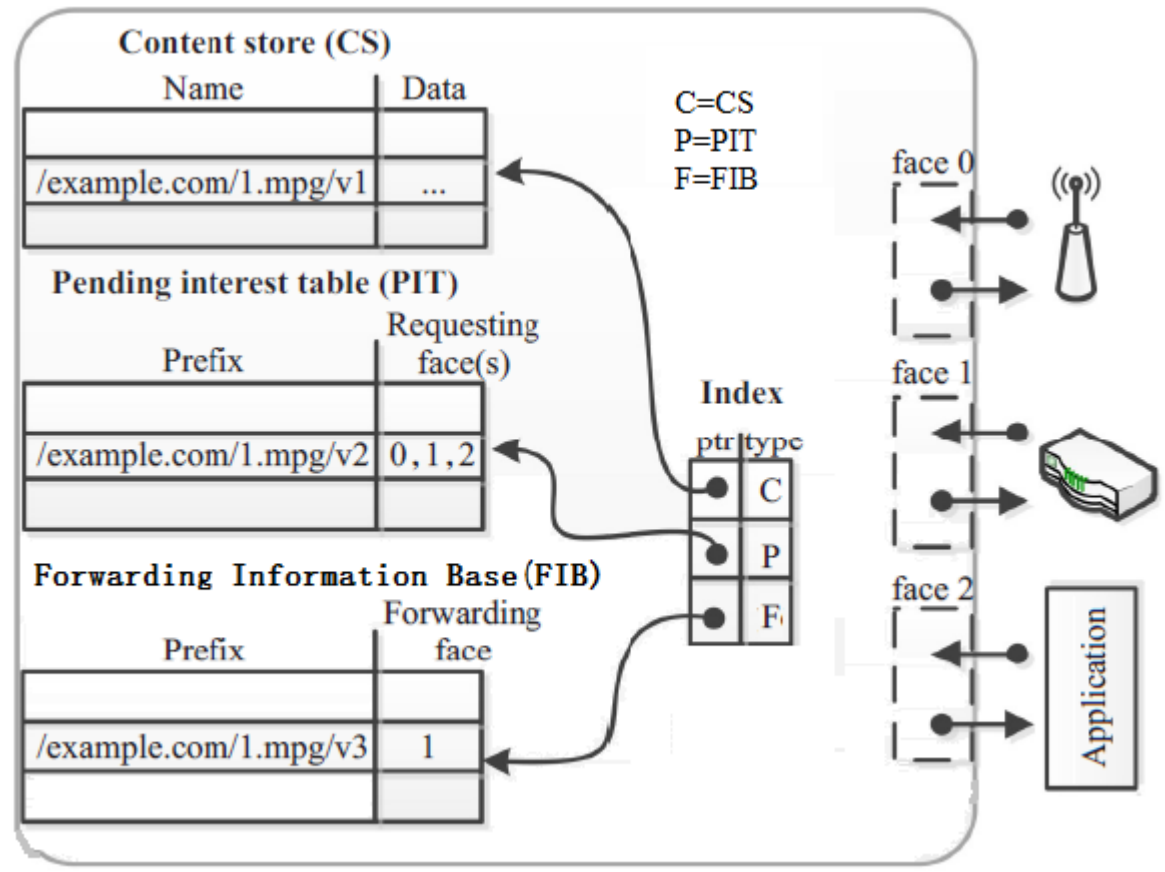
- > FIB
- > CS
- > PIT



The main tables of NDN



The main tables of NDN



Simulation framework: ns-3 and ndnSIM

- > ns-3 is a network simulator
 - 1) Topology definition
 - 2) Model development
 - 3) Node and link configuration
 - 4) Execution
 - 5) Performance analysis
 - 6) Graphical Visualization

- > ndnSIM is an implementation of the basic components of a NDN network.

Current implementation in ndnSIM v2.0

- > Used default strategy: broadcast strategy
- > Every Interest is being broadcast to all upstream faces
- > Data is broadcast to all downstream faces „following“ the PIT entries left by the Interests

Problems:

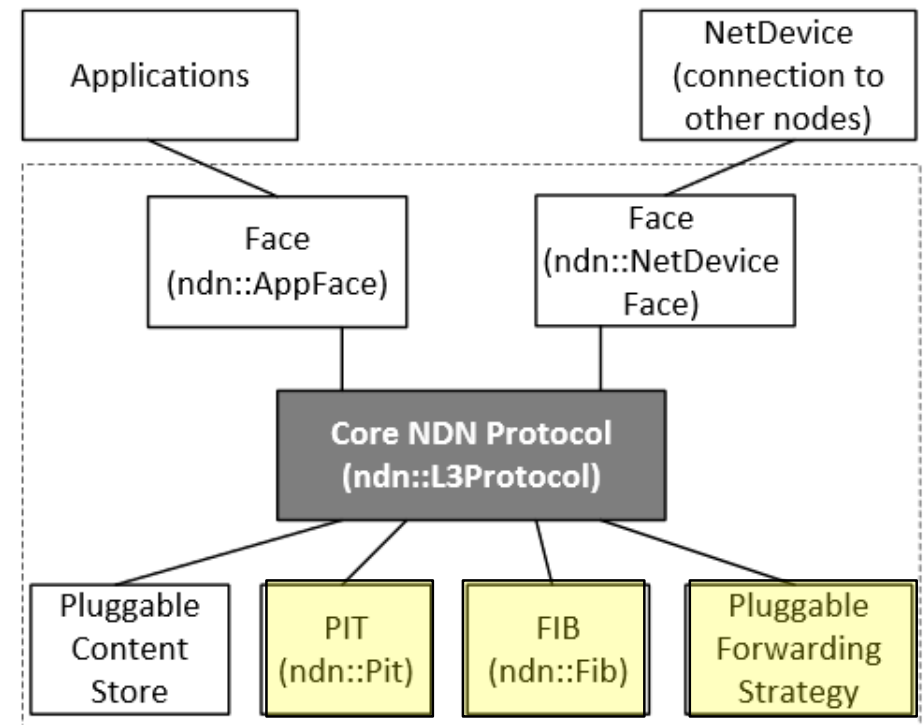
- > Broadcasting slows down the network and creates unnecessary traffic which leads to congestions
- > Multihop currently (v2.0) not possible with wireless communication
- > Forwarding through faces not feasible for wireless connections

Goals and implementation

- > Change several datastructures in order to support origin and target node identification
- > Change the NDN headers for Interest and Data packets
- > Implement new forwarding strategy that is more efficient than the current default implementation

Goals and implementation

- > Change several datastructures in order to support origin and target node identification
- > Change the NDN headers for Interest and Data packets
- > Implement new forwarding strategy that is more efficient than the current default implementation



NDN Headers for Interest and Data Packets

- > Add needed information to Interest and Data packets

NDN Headers for Interest and Data Packets

- > Add needed information to Interest and Data packets

- > Interest packet:
 - MAC address field: `std::string m_interestOriginMacAddress`
 - MAC address field: `std::string m_interestTargetMacAddress`
 - Route taken so far: `std::string m_macInterestRoute`

NDN Headers for Interest and Data Packets

- > Add needed information to Interest and Data packets

- > Interest packet:
 - MAC address field: std::string m_interestOriginMacAddress
 - MAC address field: std::string m_interestTargetMacAddress
 - Route taken so far: std::string m_macInterestRoute

- > Data packet:
 - MAC address field: std::string m_dataOriginMacAddress
 - MAC address field: std::string m_dataTargetMacAddress
 - Route taken so far: std::string m_macDataRoute

PIT / FIB entries

- > Make the Information from the packets persist within the nodes

PIT / FIB entries

- > Make the Information from the packets persist within the nodes

- > The PIT entries were extended:
 - List of origin MAC addresses
 - Latency
 - Setters and getters

PIT / FIB entries

- > Make the Information from the packets persist within the nodes

- > The PIT entries were extended:
 - List of origin MAC addresses
 - Latency
 - Setters and getters

- > The FIB entries were extended:
 - List of target MAC addresses
 - Setters and getters

Forwarding Strategy

> Flooding of the Network

- 1) Interest arrives at some node
- 2) PIT is updated with Interest name and origin MAC address
- 3) Current node's MAC address is added to the Interest
- 4) FIB is checked and found empty → Broadcast Interest with empty target MAC
- 5) Neighboring nodes receive the broadcast and check if a target MAC is present on Interest
 - 5.1) no target MAC was added to Interest → accepted!

Forwarding Strategy

- > Flooding of the Network
 - 1) Interest arrives at some node
 - 2) PIT is updated with Interest name and origin MAC address
 - 3) Current node's MAC address is added to the Interest
 - 4) FIB is checked and found empty → Broadcast Interest with empty target MAC
 - 5) Neighboring nodes receive the broadcast and check if a target MAC is present on Interest
 - 6.1) no target MAC was added to Interest → accepted!
 - 6) Repeat from 2) until some Producer is reached

Forwarding Strategy

> Flooding of the Network

- 1) Interest arrives at some node
- 2) PIT is updated with Interest name and origin MAC address
- 3) Current node's MAC address is added to the Interest
- 4) FIB is checked and found empty → Broadcast Interest with empty target MAC
- 5) Neighboring nodes receive the broadcast and check if a target MAC is present on Interest
 - 5.1) no target MAC was added to Interest → accepted!
- 6) Repeat from 2) until some Producer is reached

- 7) Data arrives at some node
- 8) PIT is checked if Data is solicited
- 9) If it is solicited add current node's MAC address and target MAC from PIT entry to Data.
- 10) Add Next Hop to FIB entry
- 11) Broadcast data.

Forwarding Strategy

> Flooding of the Network

- 1) Interest arrives at some node
- 2) PIT is updated with Interest name and origin MAC address
- 3) Current node's MAC address is added to the Interest
- 4) FIB is checked and found empty → Broadcast Interest with empty target MAC
- 5) Neighboring nodes receive the broadcast and check if a target MAC is present on Interest
 - 5.1) no target MAC was added to Interest → accepted!
- 6) Repeat from 2) until some Producer is reached

- 7) Data arrives at some node
- 8) PIT is checked if Data is solicited
- 9) If it is solicited add current node's MAC address and target MAC from PIT entry to Data.
- 10) Broadcast data.
- 11) Add Next Hop to FIB entry
- 12) Repeat from 7) until the Consumer is reached.

Forwarding Strategy

Algorithm 1 Interest forwarding from requester node

```
1: procedure CHECK FIB
2:   if  $FIBEntry = \emptyset$  then
3:      $OriginMAC \leftarrow MyMAC$ 
4:      $TargetMAC \leftarrow NULL$ 
5:      $forward(Interest, nexthop)$ 
6:   else
7:      $OriginMAC \leftarrow MyMac$ 
8:      $TargetMAC \leftarrow Select(nexthop)$ 
9:      $forward(Interest, nexthop)$ 
10:  end if
11: end procedure
```

Forwarding Strategy

Algorithm 1 Interest forwarding from requester node

```

1: procedure CHECK FIB
2:   if  $FIBEntry = \emptyset$  then
3:      $OriginMAC \leftarrow MyMAC$ 
4:      $TargetMAC \leftarrow NULL$ 
5:      $forward(Interest, nexthop)$ 
6:   else
7:      $OriginMAC \leftarrow MyMac$ 
8:      $TargetMAC \leftarrow Select(nexthop)$ 
9:      $forward(Interest, nexthop)$ 
10:  end if
11: end procedure

```

Algorithm 2 Data forwarding from intermediate node

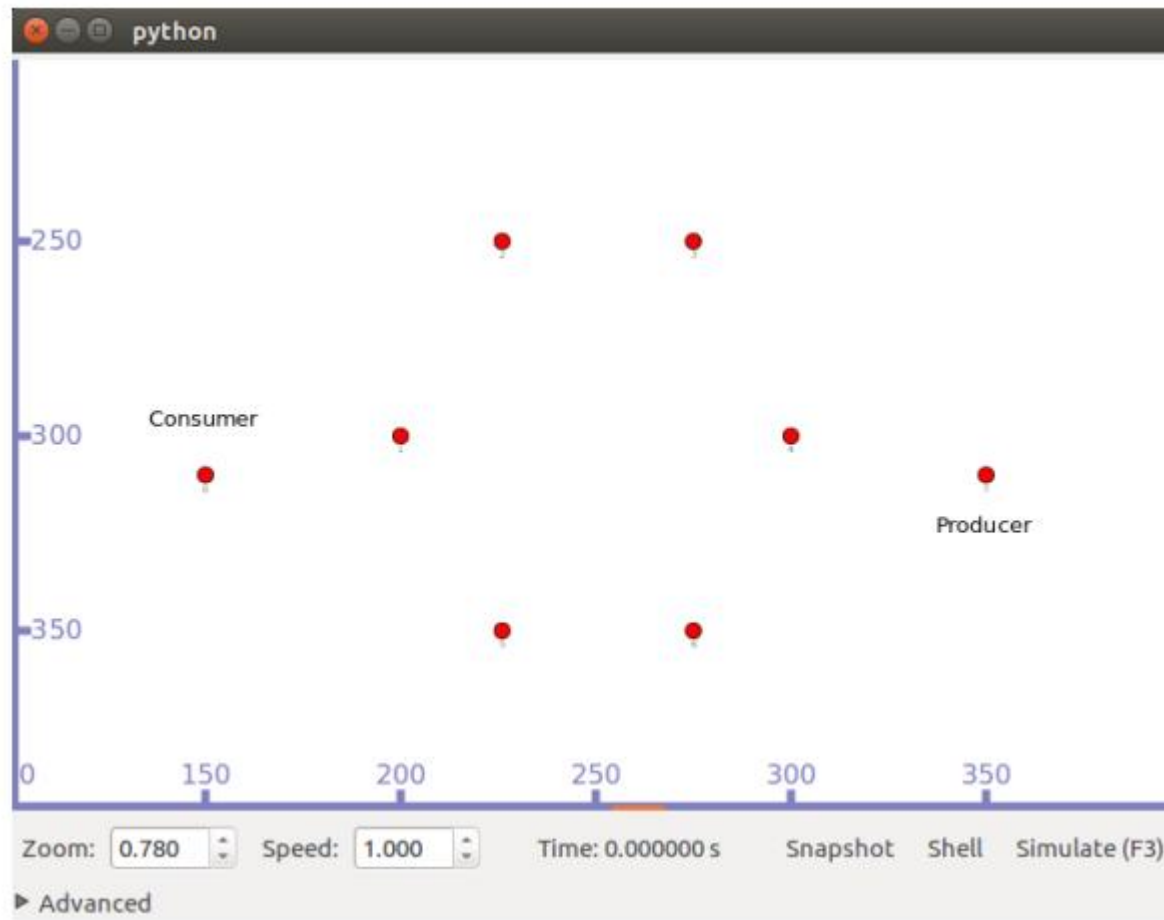
```

1: procedure RECEPTION OF DATA
2:   if  $PITEntry \neq \emptyset$  then
3:     if  $TargetMAC \neq MyMAC$  then
4:        $Create/Update(FIBEntry, OriginMac)$ 
5:     else
6:        $Create/Update(FIBEntry, OriginMac)$ 
7:        $OriginMAC \leftarrow MyMAC$ 
8:        $TargetMAC \leftarrow PITEntry(nexthop)$ 
9:        $forward(Data, nexthop)$ 
10:    end if
11:  end if
12: end procedure

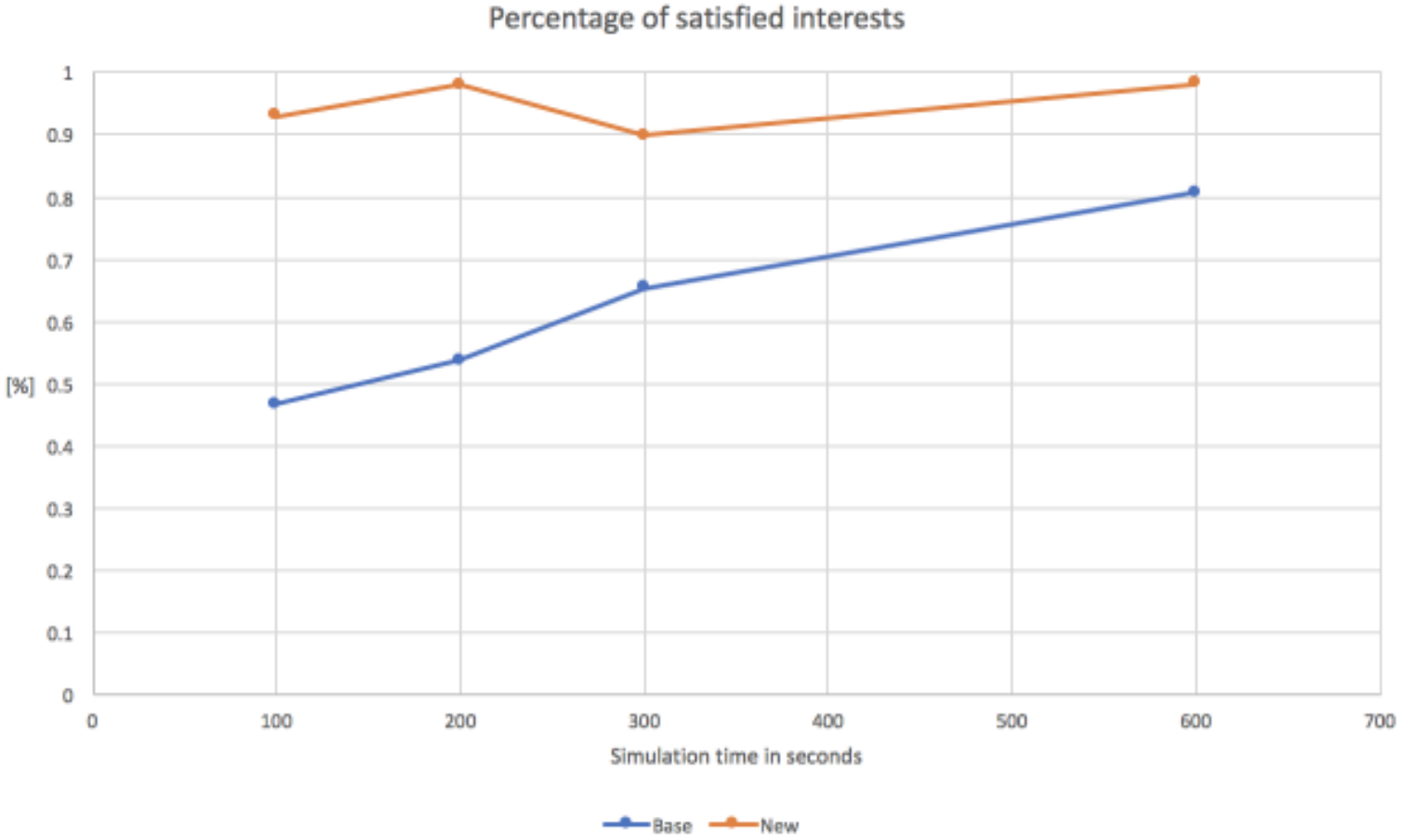
```

Results for the static 8 nodes scenario

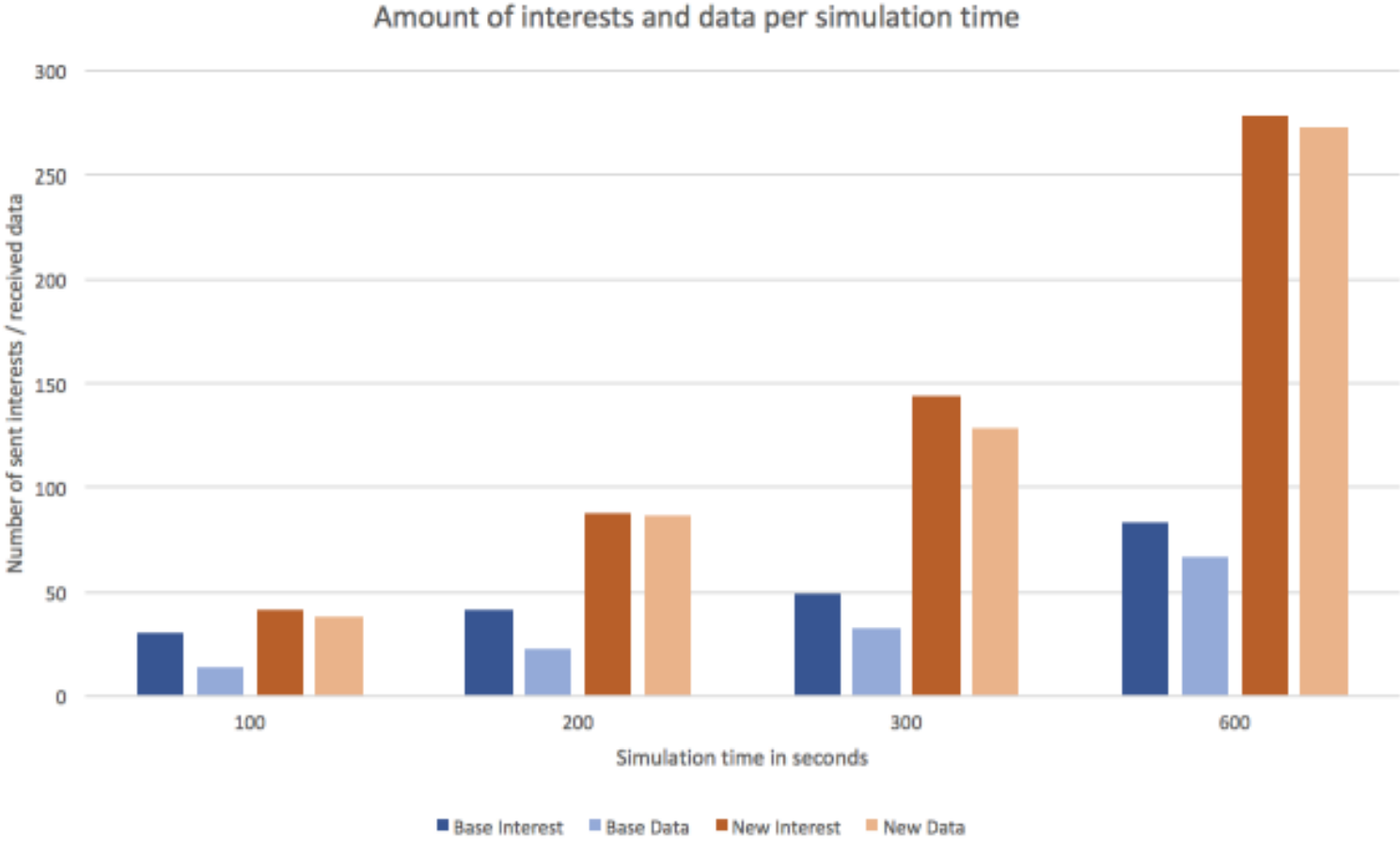
- > Static scenario with 8 nodes
- > Simulation for 100s, 200s, 300s and 600s
- > 3 network interfaces
- > Interest Lifetime of 4 seconds
- > Retransmission timer of 500ms



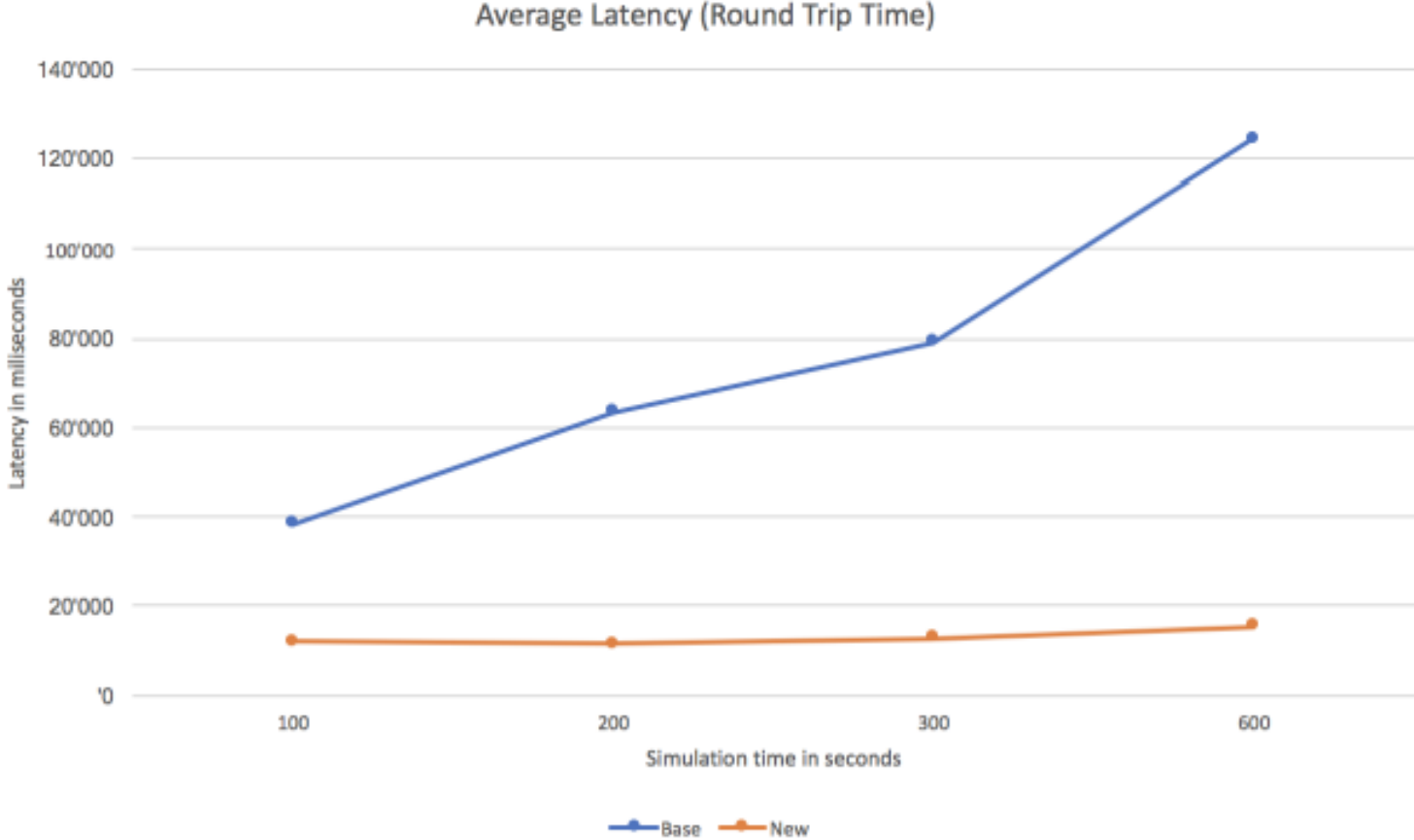
Results for the static 8 nodes scenario



Results for the static 8 nodes scenario

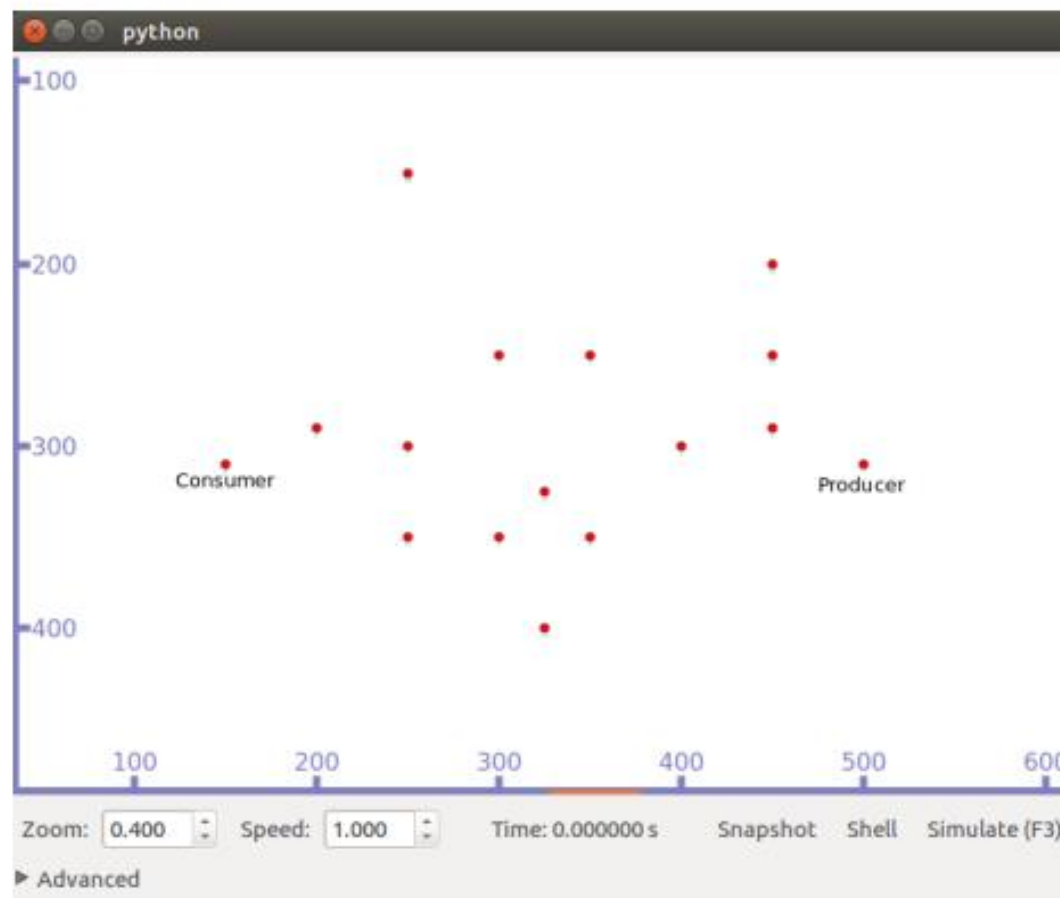


Results for the static 8 nodes scenario

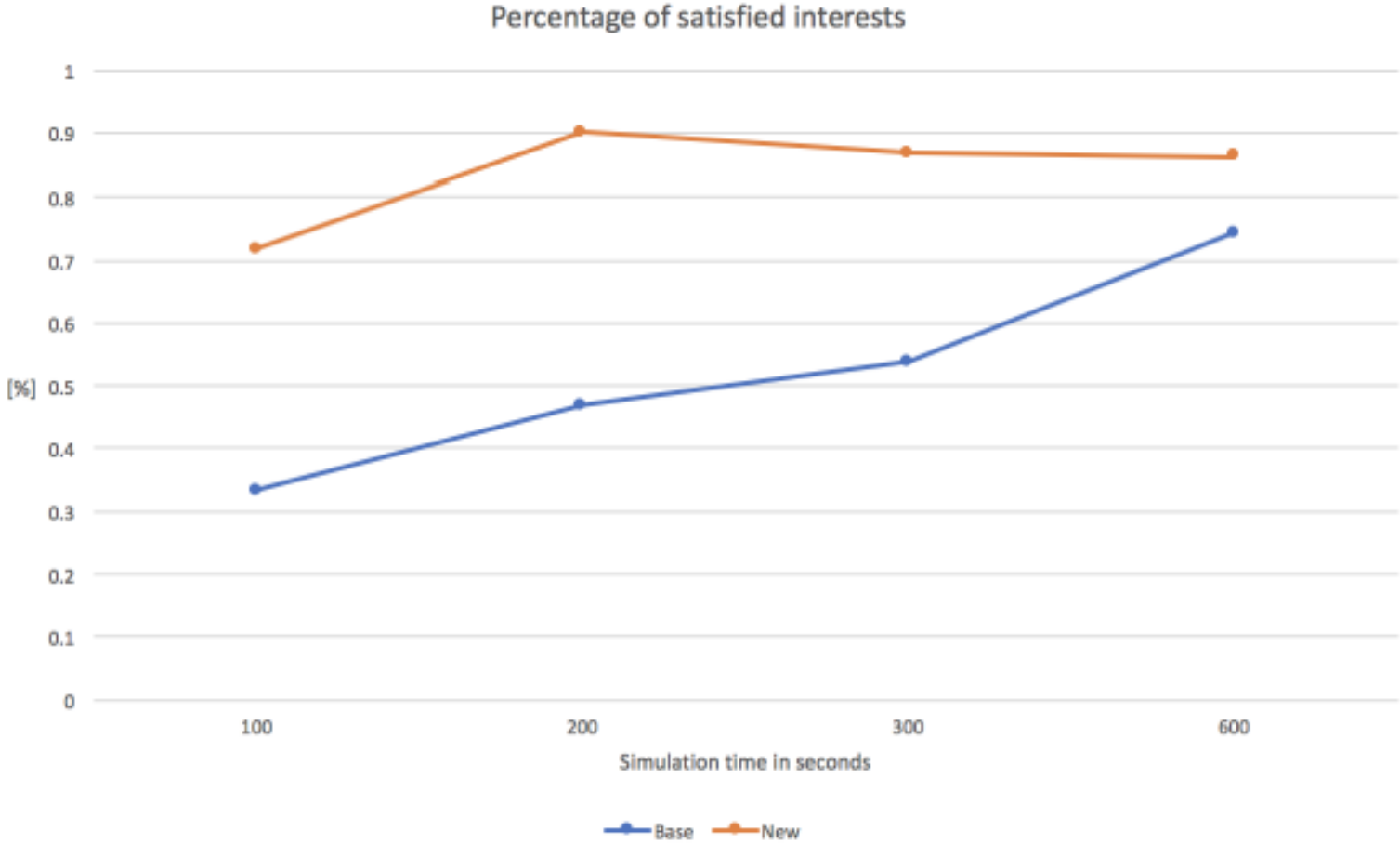


Results for the dynamic 16 nodes scenario

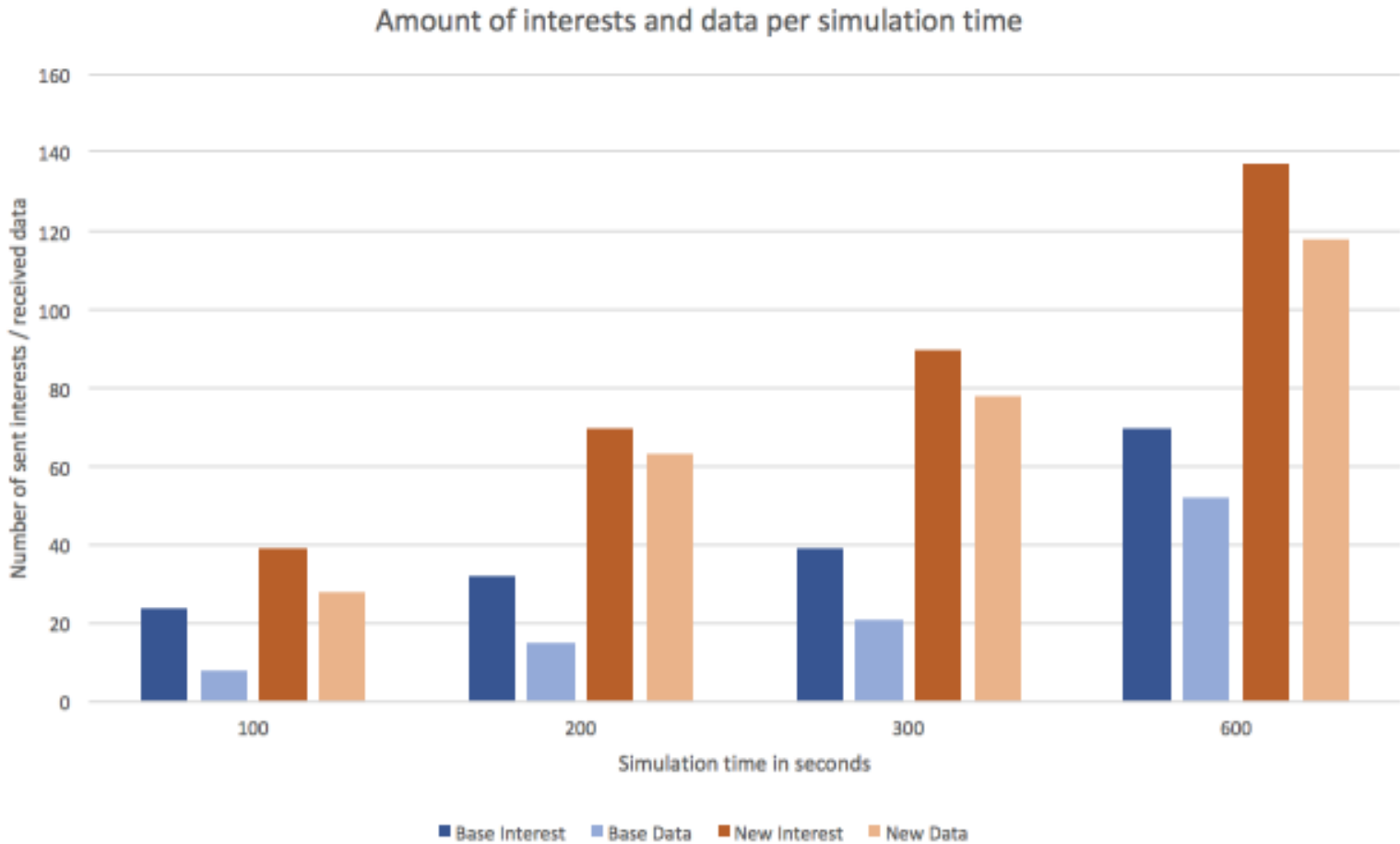
- > Dynamic scenario with 16 moving nodes
- > Simulation for 100s, 200s, 300s and 600s
- > 3 network interfaces
- > Interest Lifetime of 4 seconds
- > Retransmission timer of 500ms



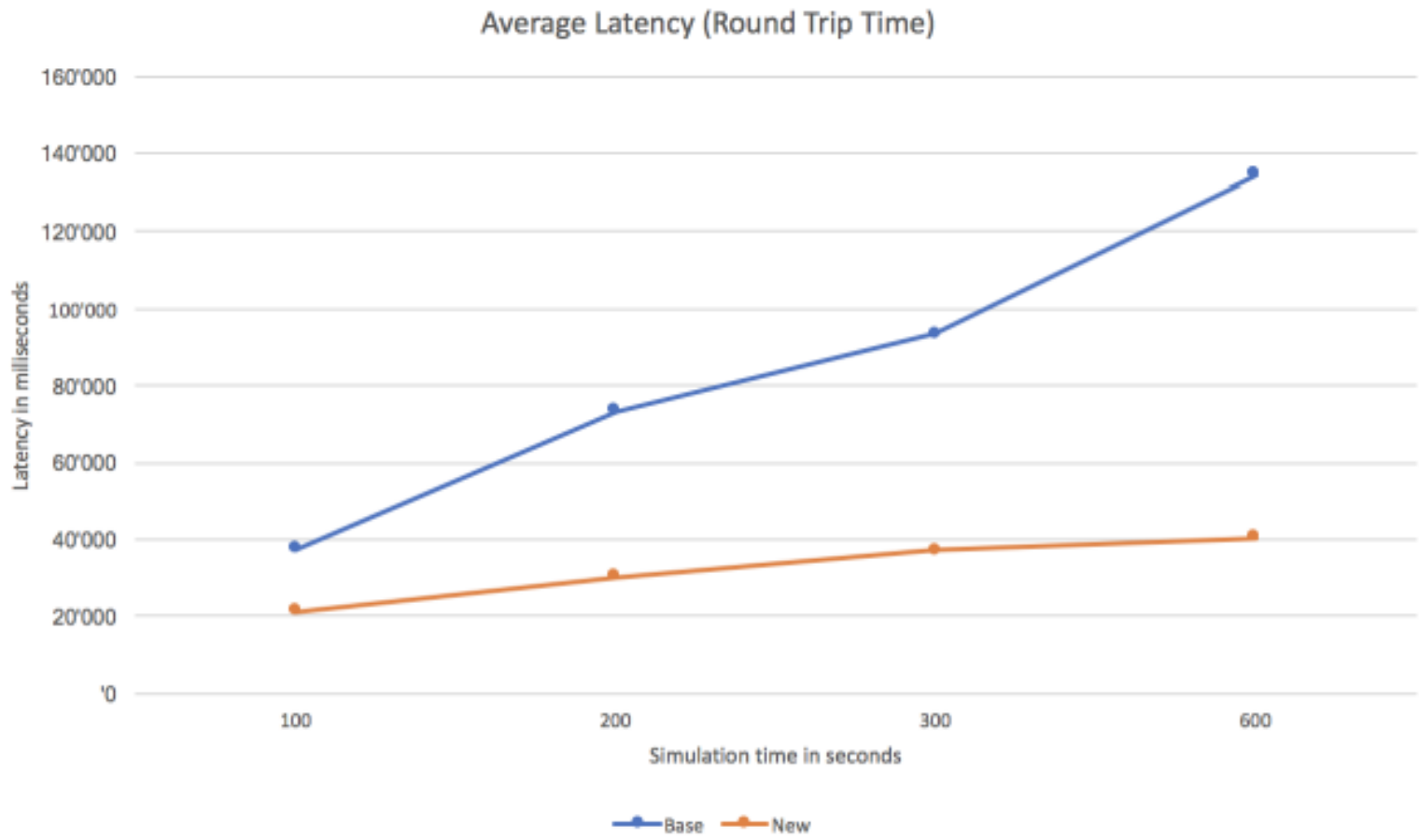
Results for the dynamic 16 nodes scenario



Results for the dynamic 16 nodes scenario



Results for the dynamic 16 nodes scenario



Future Work

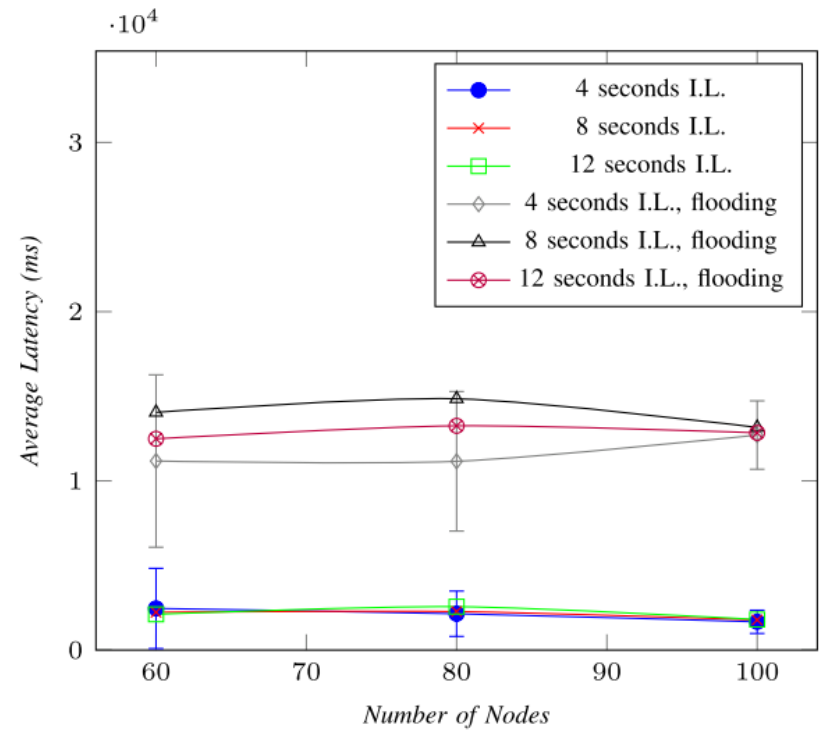
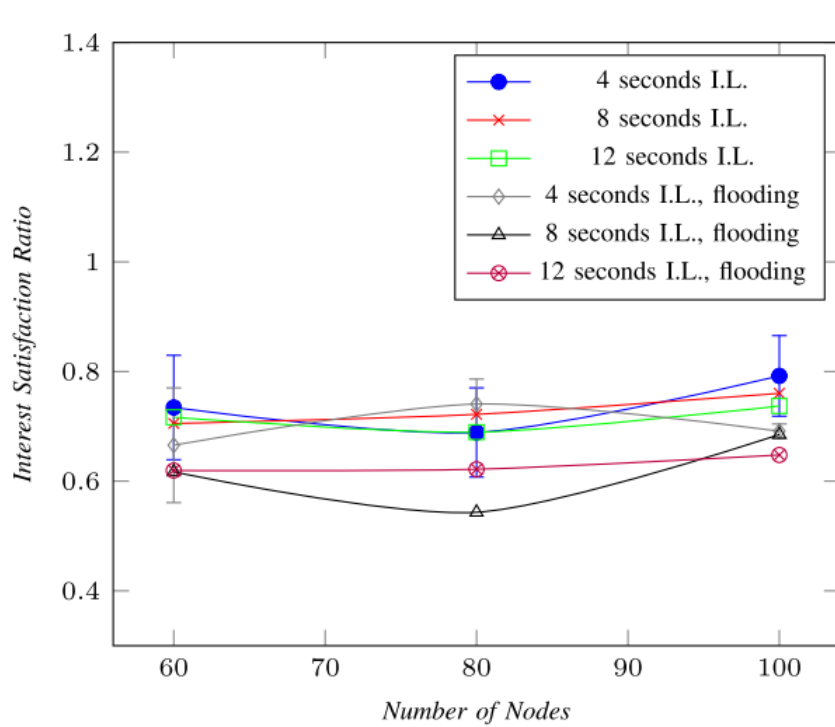
- > Improve on dynamic route selection
- > Incrementing/Decrementing the cost of the FIB next hop according to some network parameters
- > Adding NACKs for better retransmission decisions
- > Introduce CS and more nodes (consumers/producers)
- > Timer for FIB entries (deletion of it forces flooding)

Future Works implemented

- > Implemented
 - per node latency for better route detection
 - Incrementing cost for more distinct routes
 - Improved strategy to choose from several parameters
 - Real MANHATTEN scenario with 60 / 80 and 100 moving nodes
 - Tweaking with parameters like Interest Lifetime

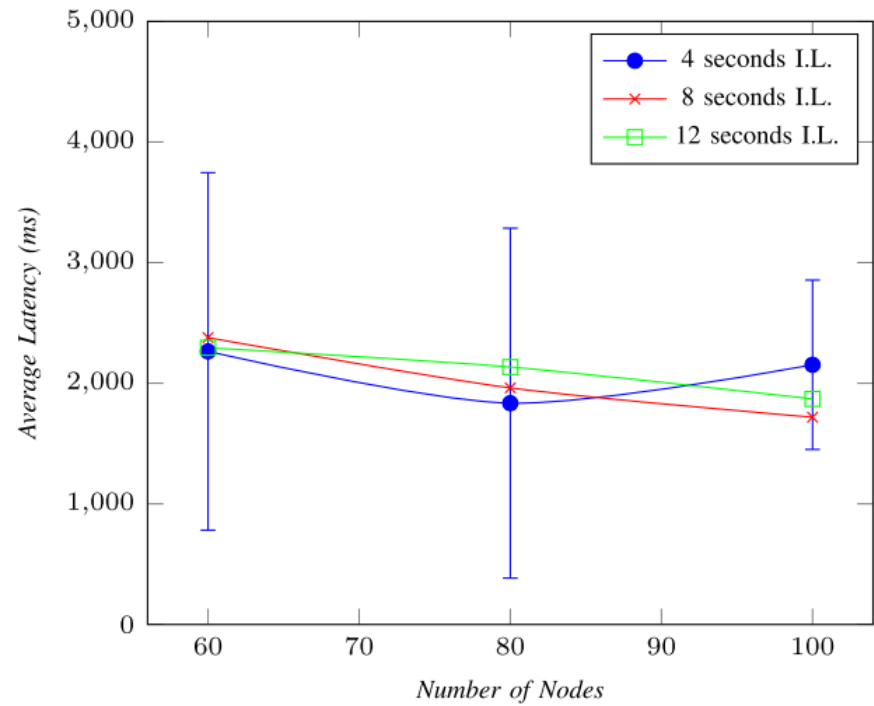
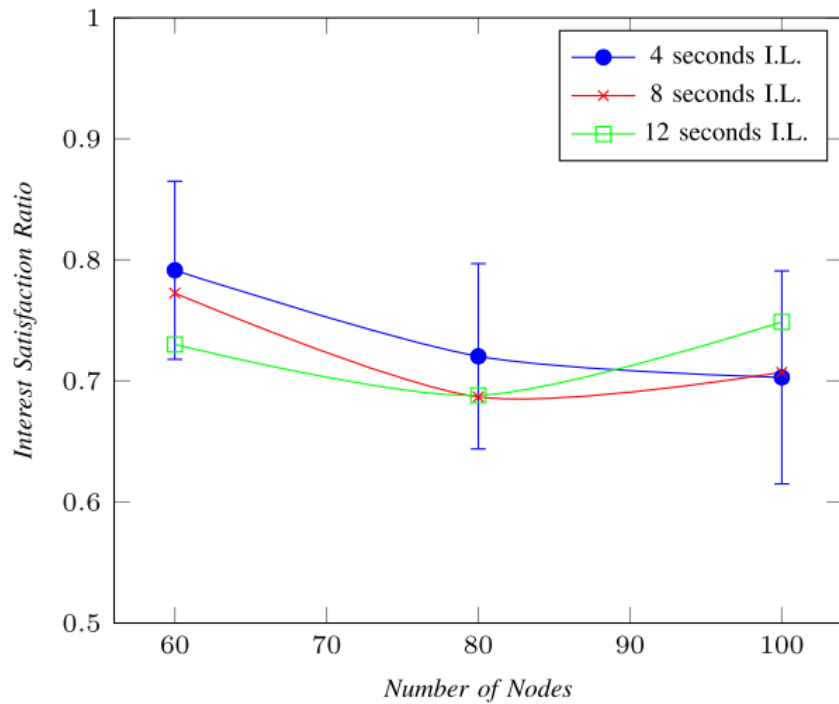
New Results (from Paper)

> First Scenario where the FIB next hops are chosen by the counter variable



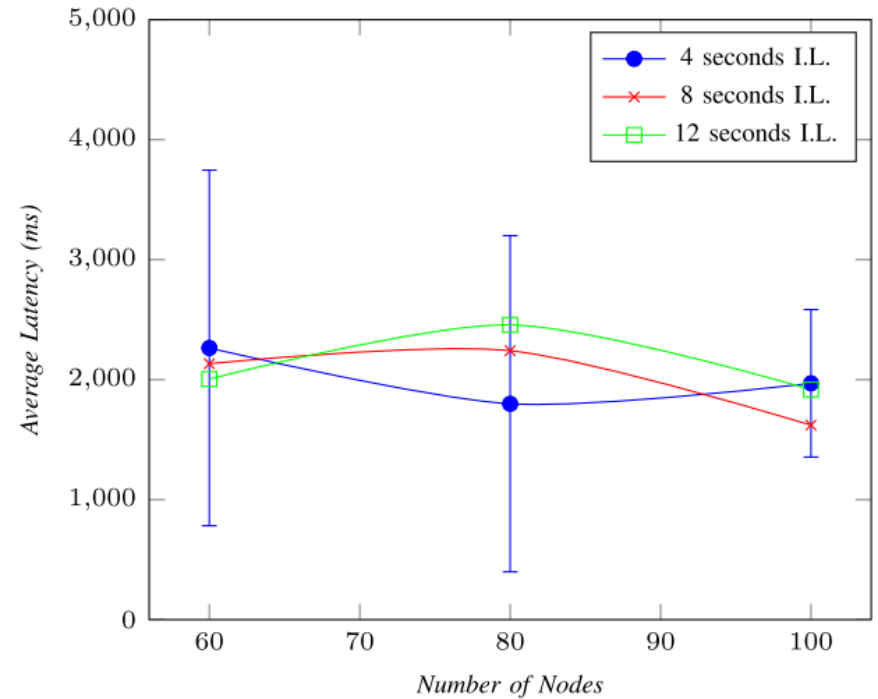
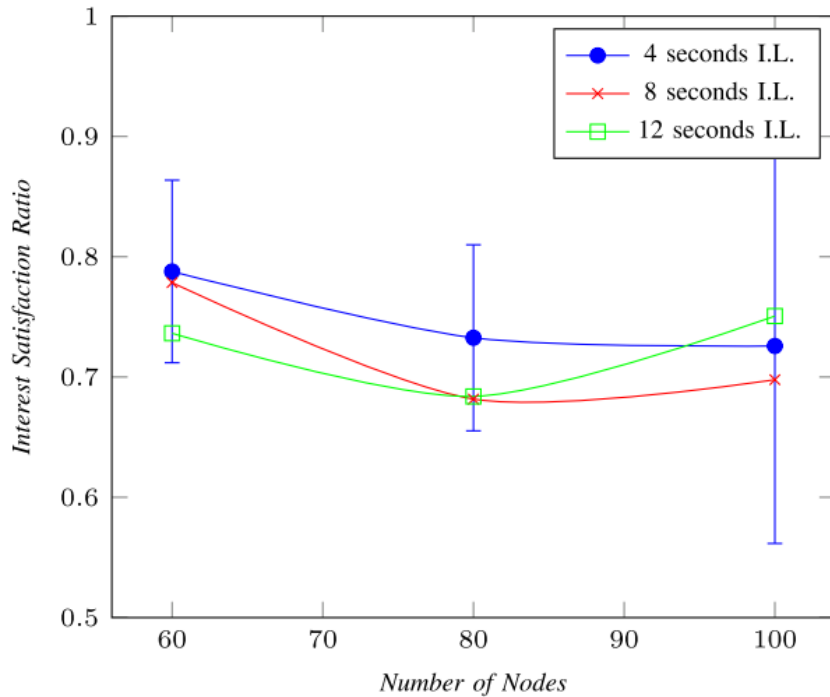
New Results (from Paper)

- > Second Scenario where the FIB next hops are chosen by the latency information in each node



New Results (from Paper)

- > Third Scenario where the FIB next hops are chosen by the counter variable and the latency information.



Questions?



References

- > www.named-data.net (September 2016)
- > www.nsnam.org/ (September 2016)
- > www.ndnsim.net (September 2016)
- > <https://www.google.com/transparencyreport/https/grid/> (September 2016)
- > Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, Rebecca L. Braynard, „Networking Named Content“ (Palo Alto Research Center, Palo Alto, CA, USA)
- > Dnyanada P. Arjunwadkar, „Introduction of NDN with Comparison to Current Internet Architecture based on TCP/IP“, University of Pune Maharashtra, India