Time Perception in NDN: Towards Better Understanding of NDN Operational Time Scales (Challenges and Solutions)

Karim Habak, Dave Oran, and Mark Stapp

Agenda

- Introduction
- Motivations and Problem Statement
- Using Per-Hop Error Detection
- Using Interest Acknowledgement
- Two Phase Architecture (Thunks)
- Evaluation
- Conclusion

Agenda

- **Introduction**
- Motivations and Problem Statement
- Using Per-Hop Error Detection
- Using Interest Acknowledgement
- Two Phase Architecture (Thunks)
- Evaluation
- Conclusion

Introduction

- What is Named Data Networking (NDN)?
- Named Data Networking Communication Paradigm
  - ✦ Consumer asks for data (Interest Packets)
  - ✦ Routers forward interests leaving breadcrumbs (Pending Interest Table) used to forward the content/data when it comes back.
  - ✦ Producers prepare the data and send it in response to the consumers

Agenda

- ✓ Introduction
- **Motivations and Problem Statement**
- Using Per-Hop Error Detection
- Using Interest Acknowledgement
- Two Phase Architecture (Thunks)
- Evaluation
- Conclusion

Motivations

- Timeouts in NDN
  - ✦ Why timeouts (Distributed System live lock and dead lock)
  - ✦ Interest satisfaction time?
- Interest satisfaction time components
  - ✦ Interest/Data delivery time (O(ms))
  - ✦ Application response time (Can be much higher)
  - ✦ Loss recovery time (Interest vs. Data)
- Drawbacks of using interest satisfaction time
  - ✦ Application response time knowledge and slow recovery?

Motivations

Network Time Scale	Application Time Scale
✓ Fast recovery ✓ New interests can pass	✓ Low overhead ✓ Regular Bandwidth allocation
- Huge overhead - Challenging Bandwidth allocation	- Slow Recovery - Masking New/Other interests - Requires a lot of knowledge

- Why don’t we have it in current IP world (HTTP like traffic)?
  - Decoupling network round trip time and application response time

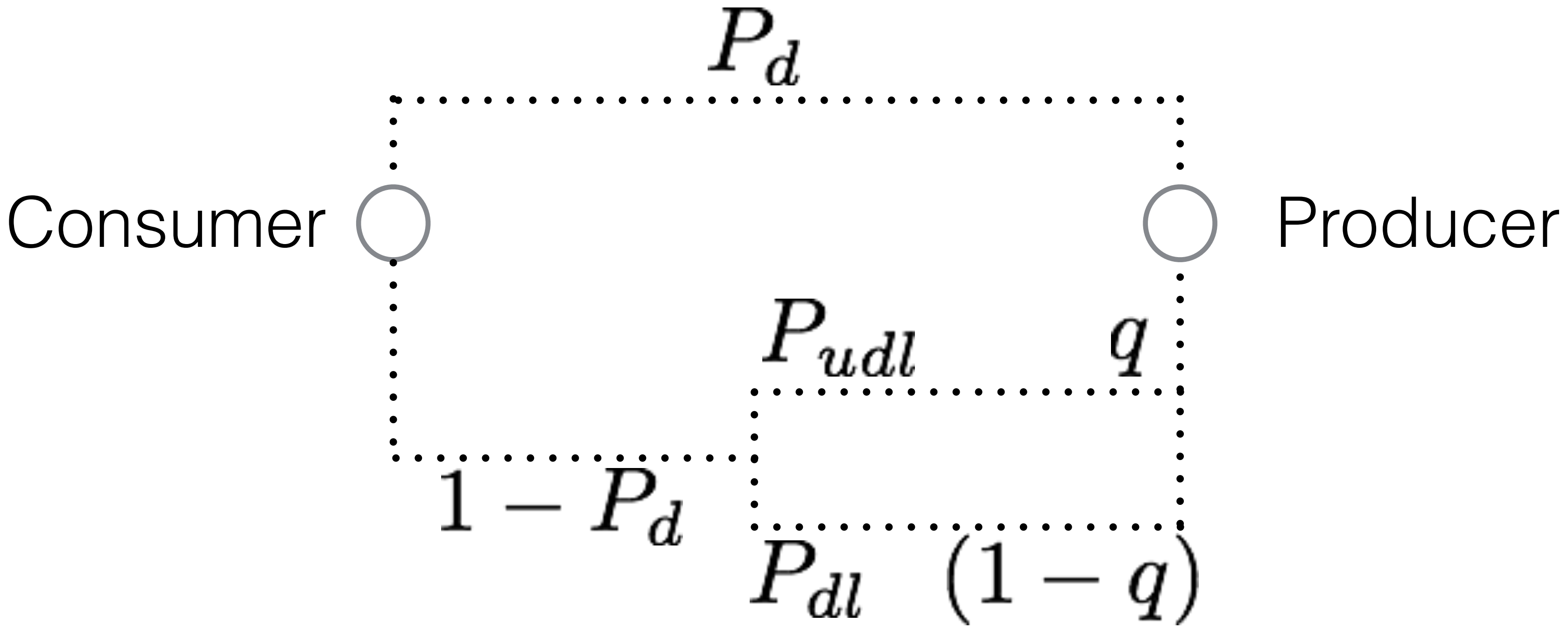
Problem Statement

- Goals
  - ✦ Minimize the Interest Satisfaction time
    - ➡ Insuring that the producer starts preparing the data as soon as possible
    - ➡ Getting the data as soon as they become available
  - ✦ Minimize the overhead

Agenda

- ✓ Introduction
- ✓ Motivations and Problem Statement
- **Using Per-Hop Error Detection**
- Using Interest Acknowledgement
- Two Phase Architecture (Thunks)
- Evaluation
- Conclusion

Using Per Hop Error Detection



$$(1 - P_d) = P_{udl} + P_{dl}$$

$$E_R = \frac{1 - P_d}{Pd}$$

$$q = \frac{P_{udl}}{P_{udl} + P_{dl}}$$

Using Per Hop Error Detection

$$q^2[\alpha T_{net} E_R^2 + (1+\alpha) T_{app} E_r^2] +$$
$$q[T_{net}(E_R^2 + 2\alpha E_R) + 2(1+\alpha) E_R T_{app}] +$$
$$[T_{app} + T_{net}(1 + 2E_R) - T_{i/s}] \geq 0$$

- Objectives
  - ✦ Getting (q)
  - ✦ Given (T_{net} , T_{app} , $T_{i/s}$ & P_d)

Agenda

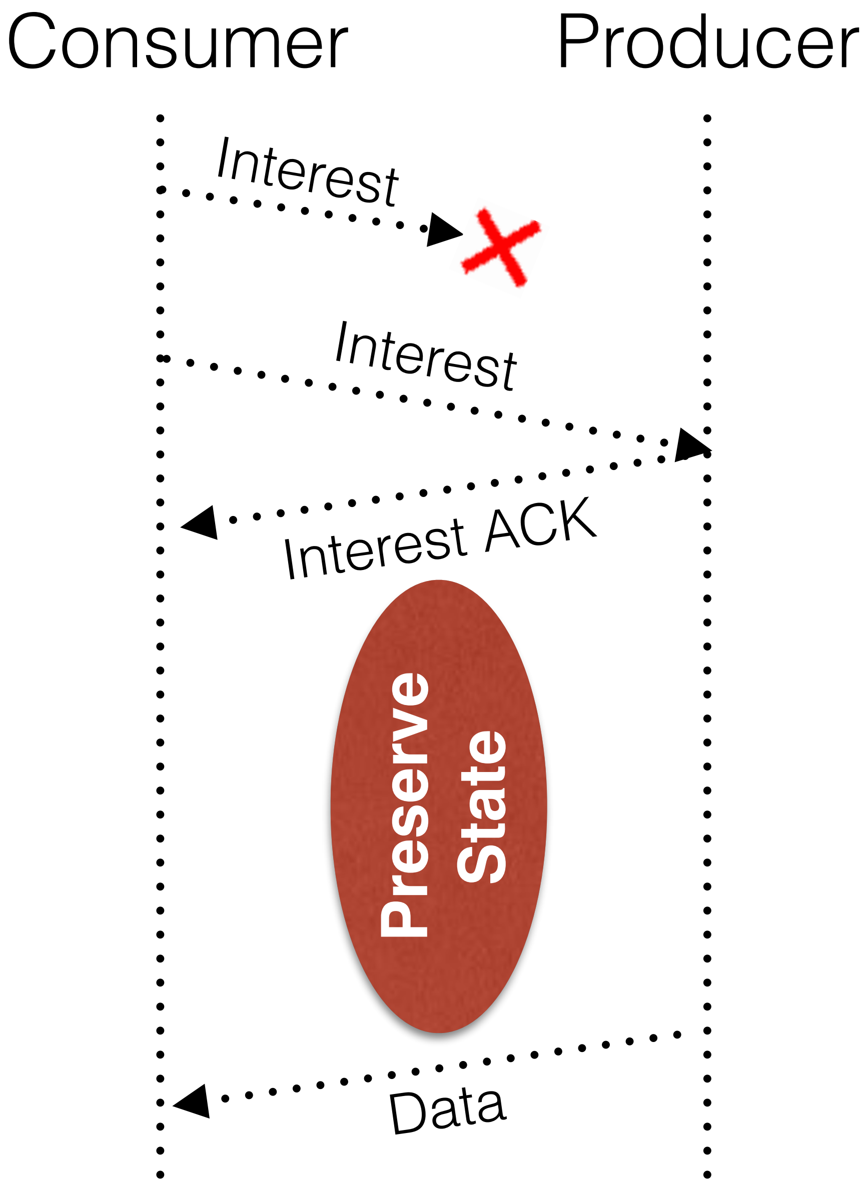
✓ Introduction

✓ Motivations and Problem Statement

✓ Using Per-Hop Error Detection

- **Using Interest Acknowledgement**
- Two Phase Architecture (Thunks)
- Evaluation
- Conclusion

Using Interest Acknowledgement

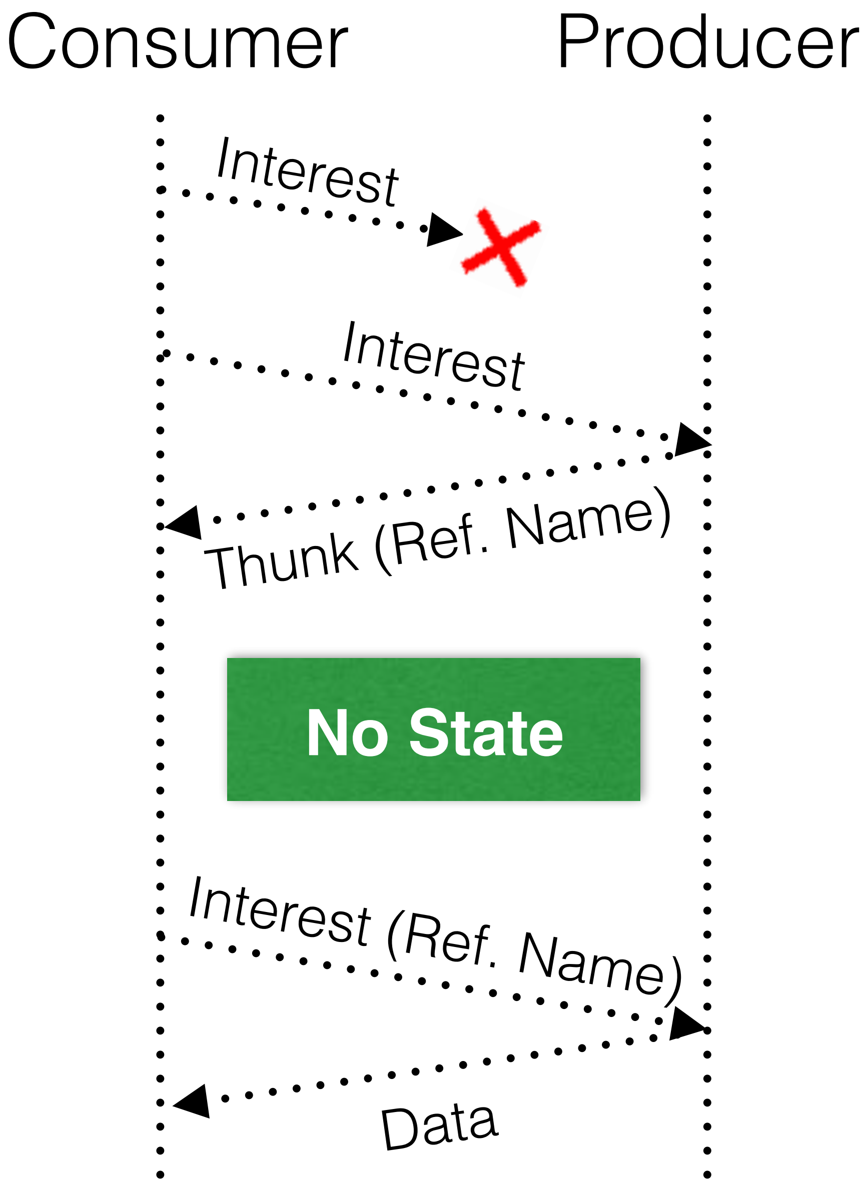


- Interest ACKs are not PIT destroying
- Interest ACKs contains suppression period
- Interest ACKs are cached in the middle nodes

Agenda

- ✓ Introduction
- ✓ Motivations and Problem Statement
- ✓ Using Per-Hop Error Detection
- ✓ Using Interest Acknowledgement
- **Two Phase Architecture (Thunks)**
- Evaluation
- Conclusion

Two Phase Architecture (Thunks)

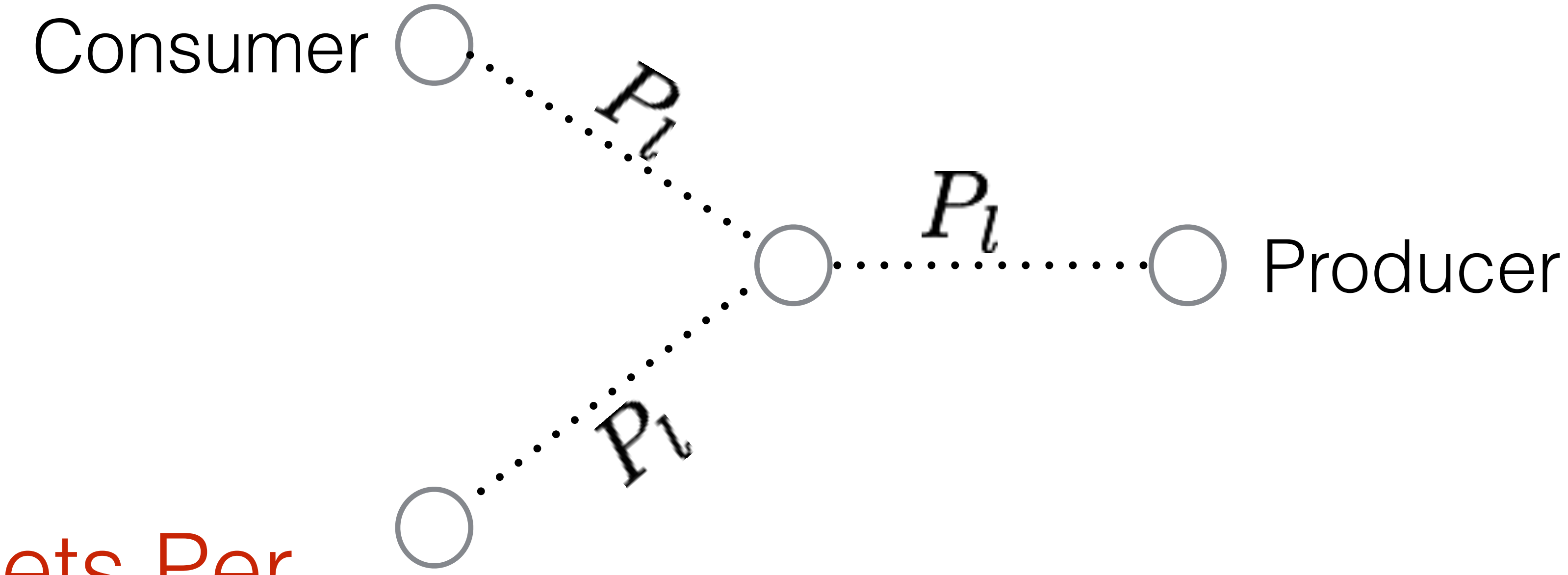


- Thunks are PIT destroying
- Thunks contains waiting period
- Thunks shouldn't be cached
  - ✦ **Why?**

Agenda

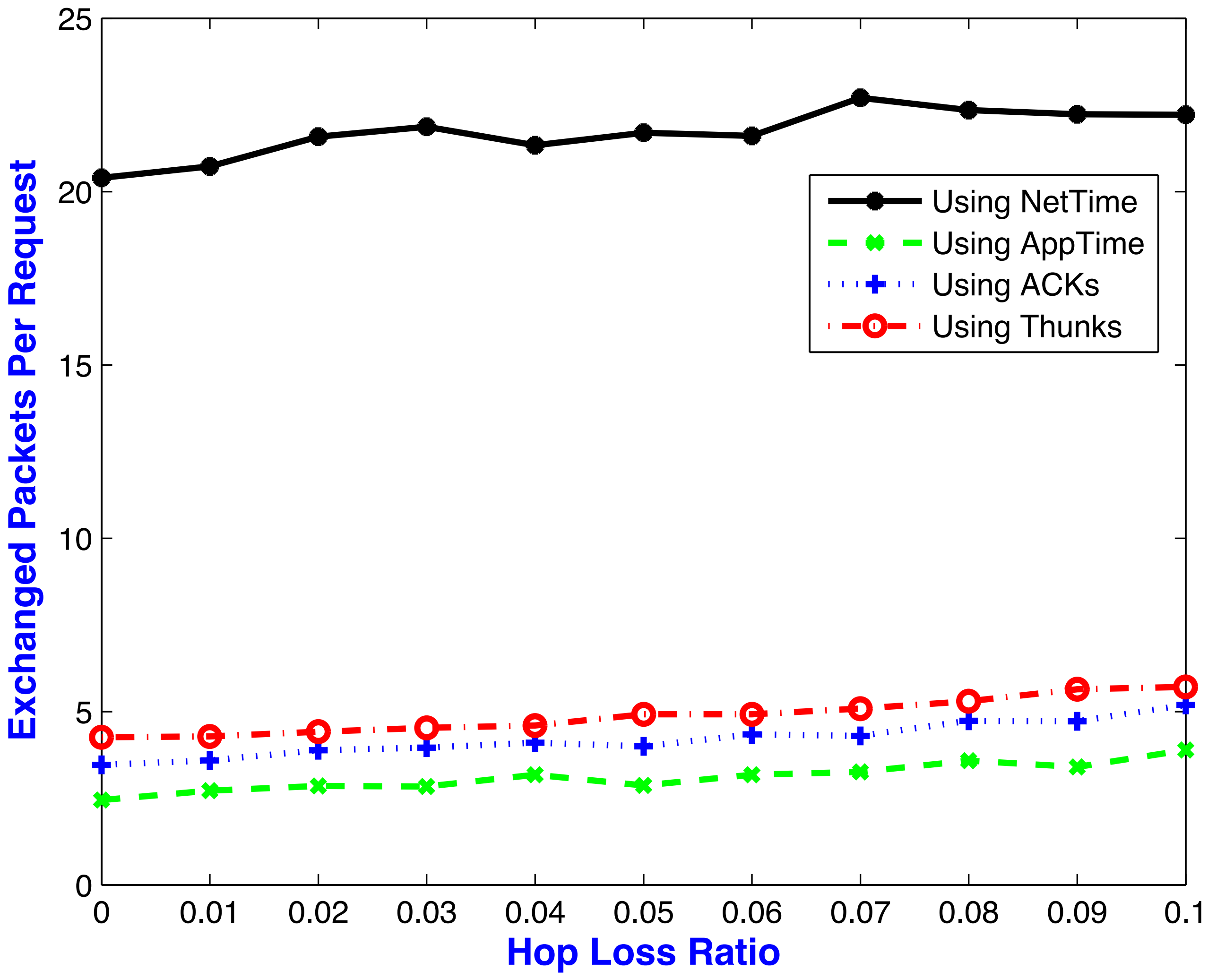
- ✓ Introduction
- ✓ Motivations and Problem Statement
- ✓ Using Per-Hop Error Detection
- ✓ Using Interest Acknowledgement
- ✓ Two Phase Architecture (Thunks)
- **Evaluation**
- Conclusion

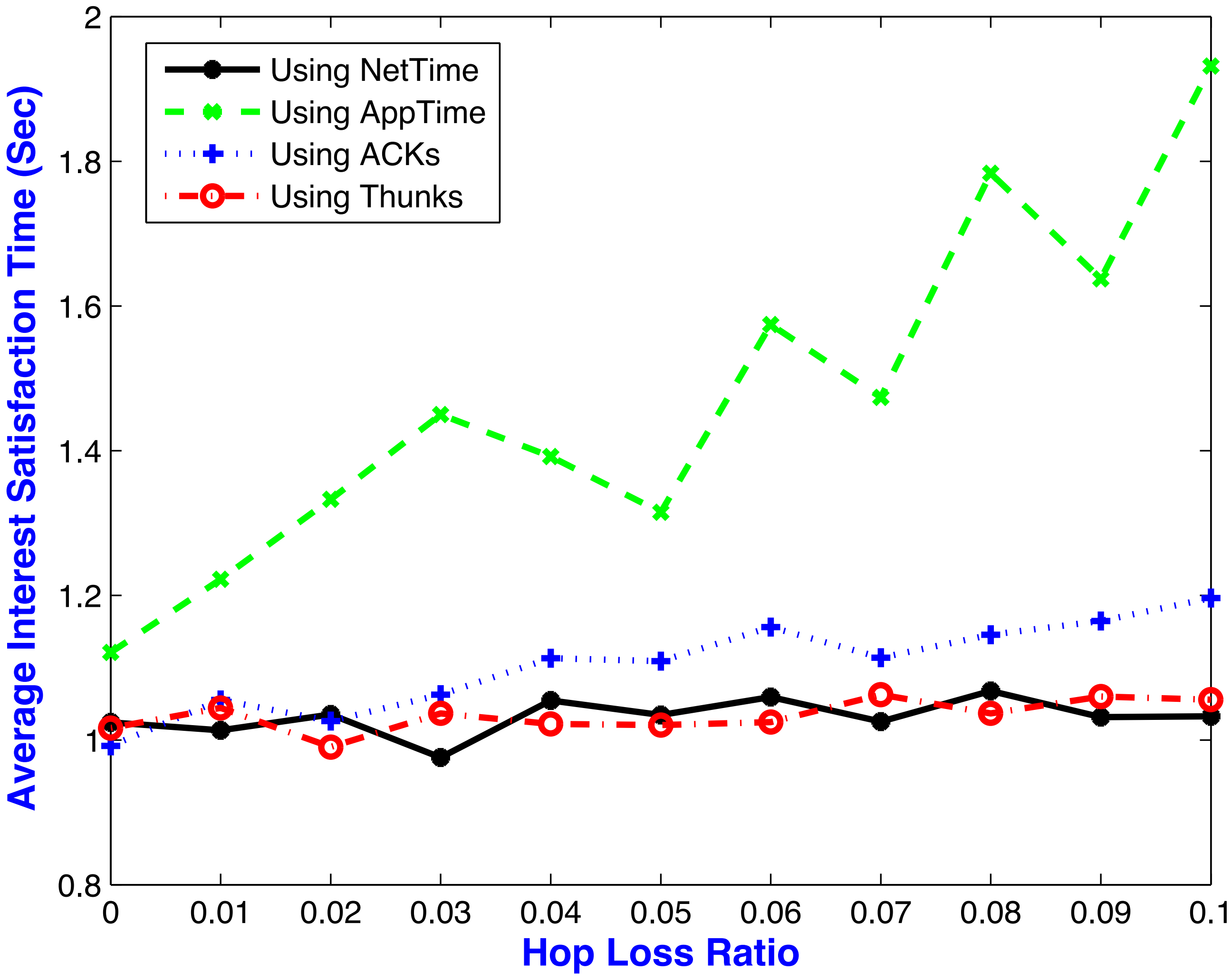
Evaluation



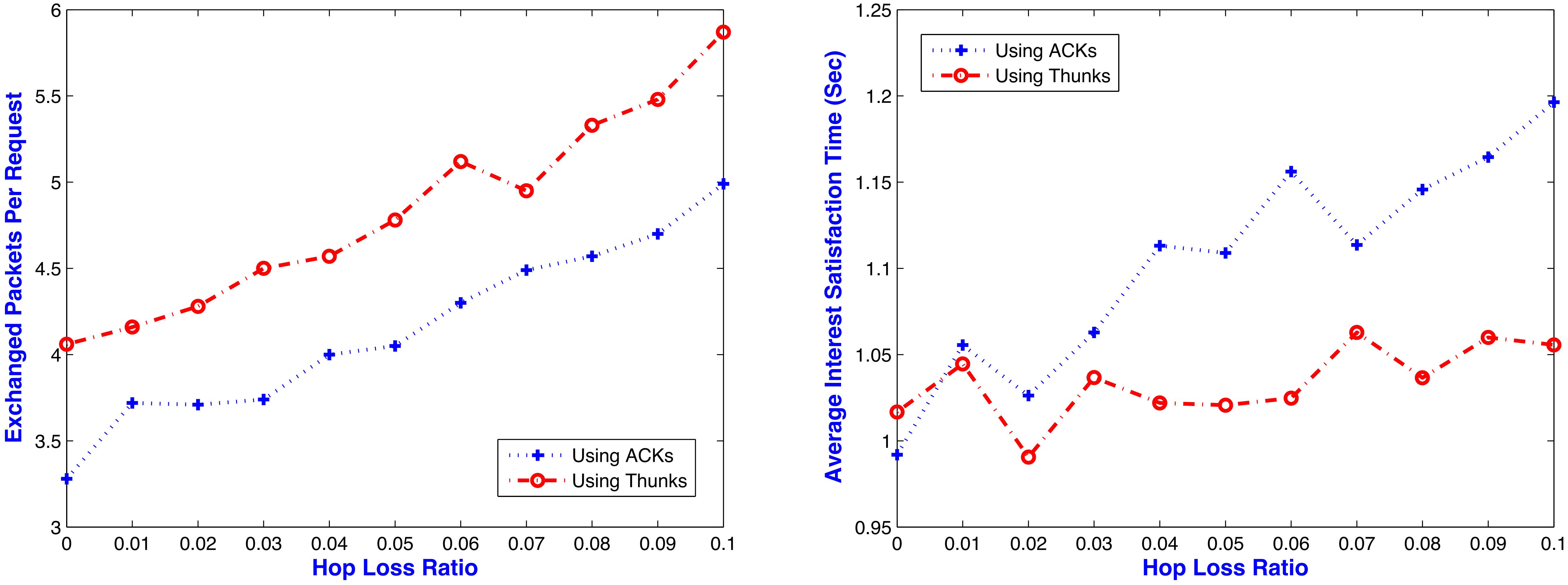
- Metrics
  - ✦ Average Interest Satisfaction Time
  - ✦ Exchanged Packets Per Request
  - ✦ Interest Satisfaction Time CDF

Evaluation
Hop Loss Ratio

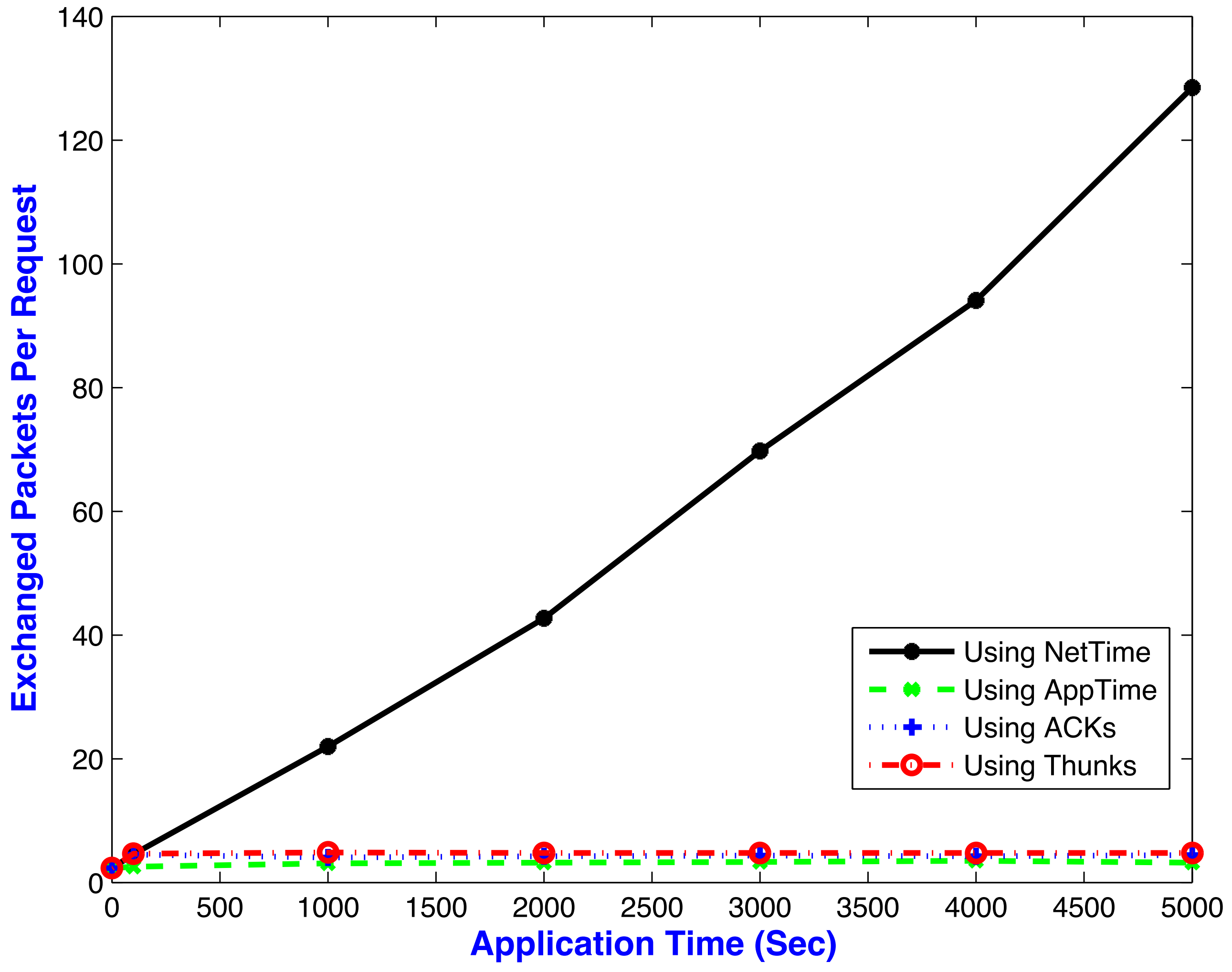


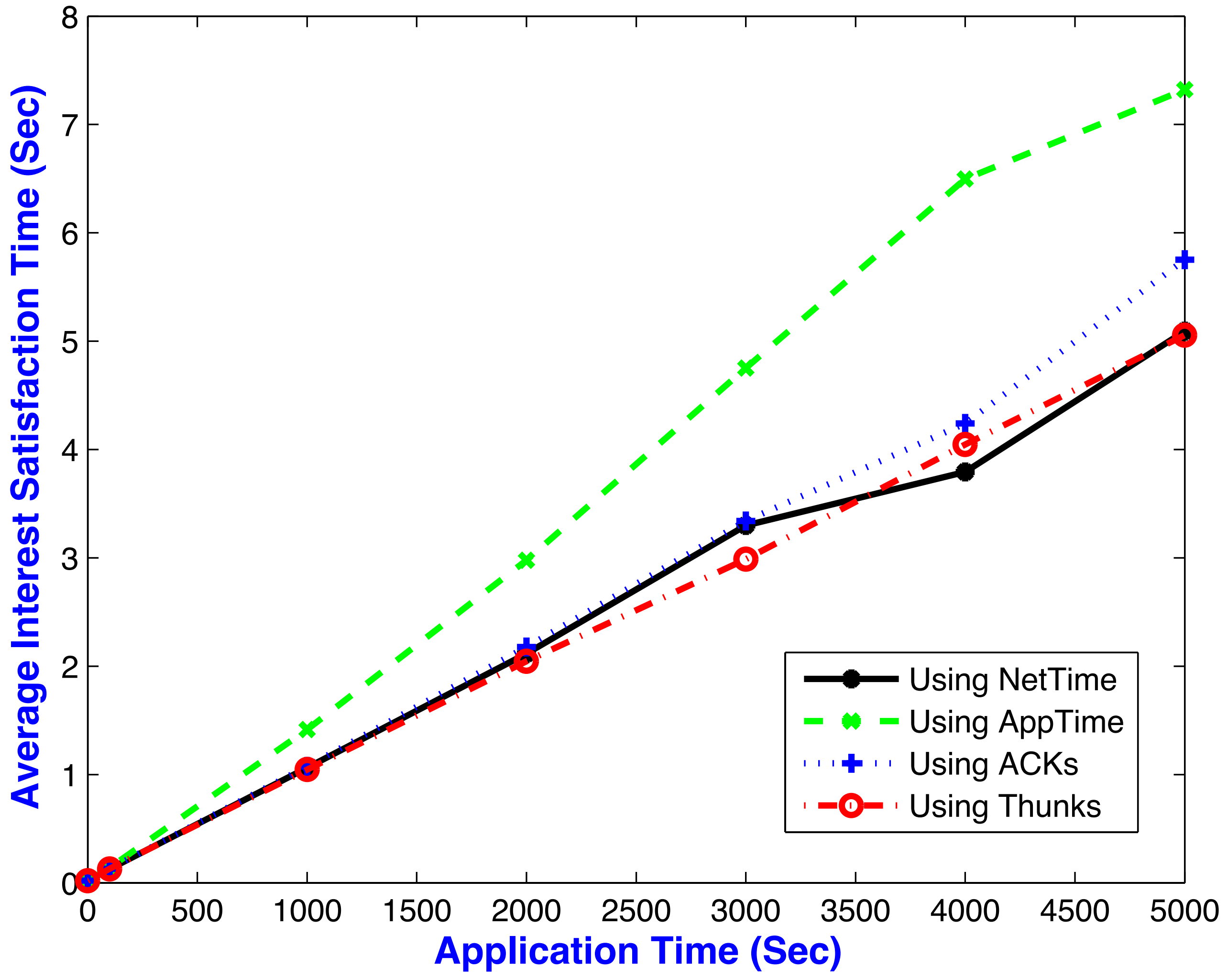


Evaluation
Hop Loss Ratio

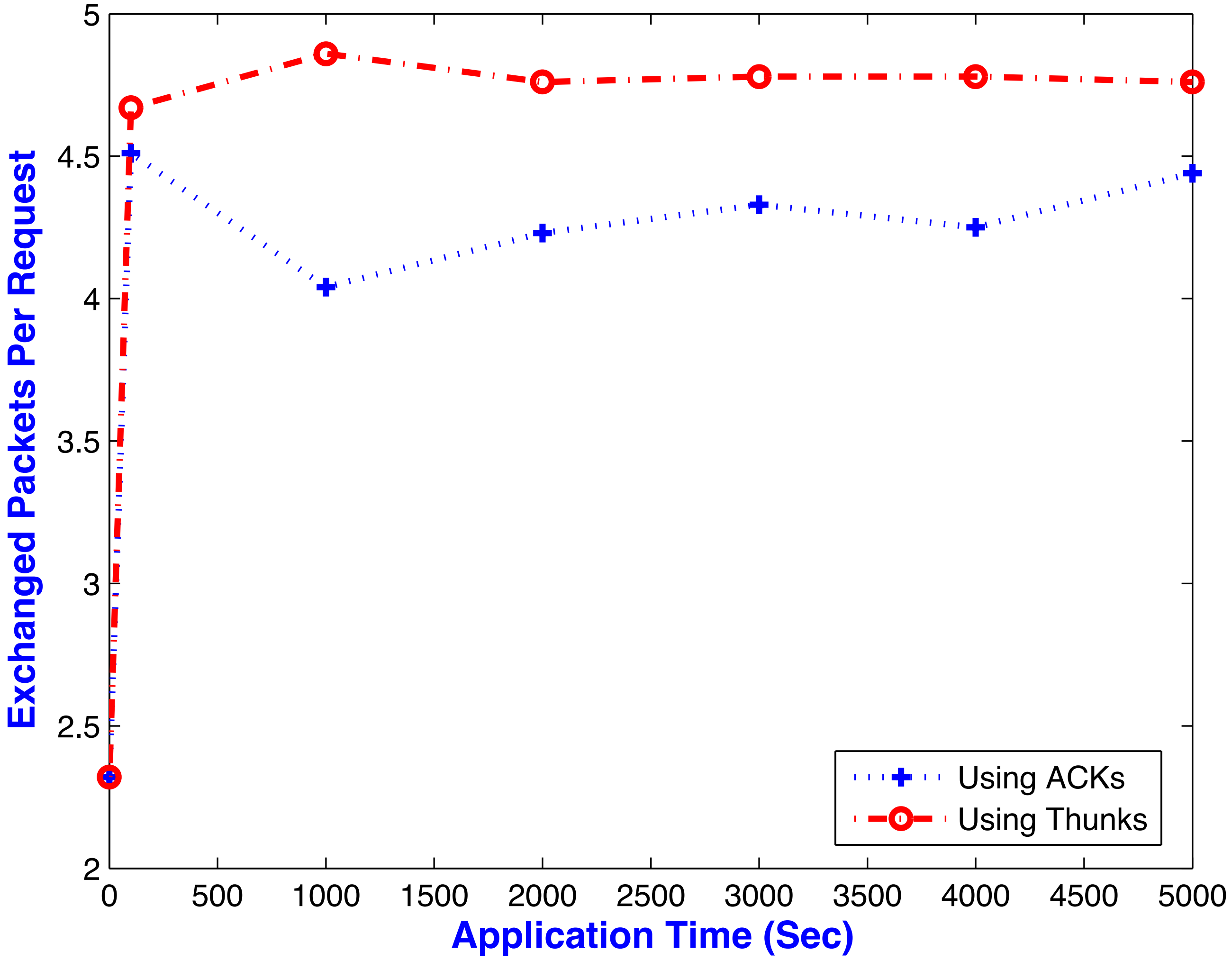


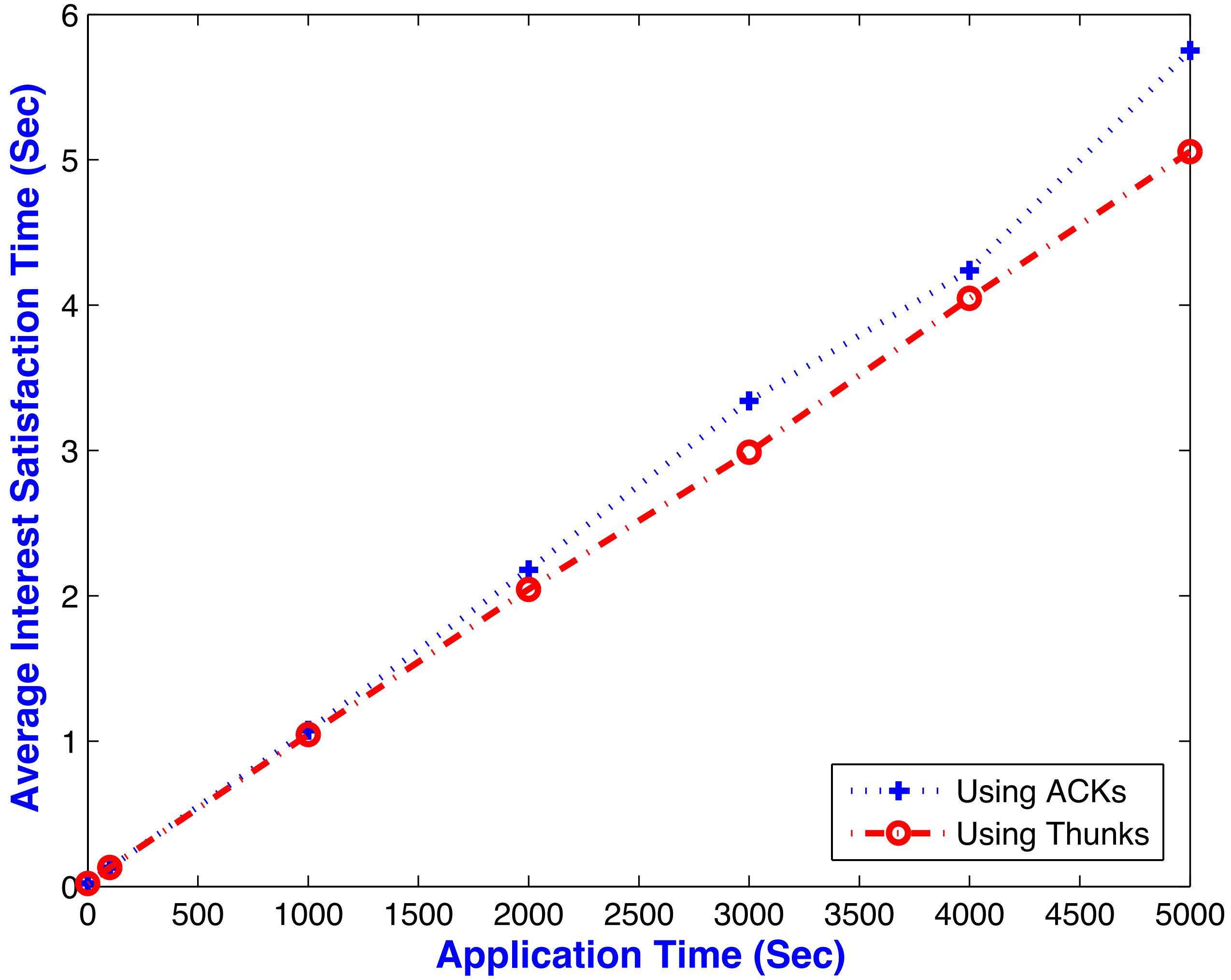
Evaluation
Application Time Effect



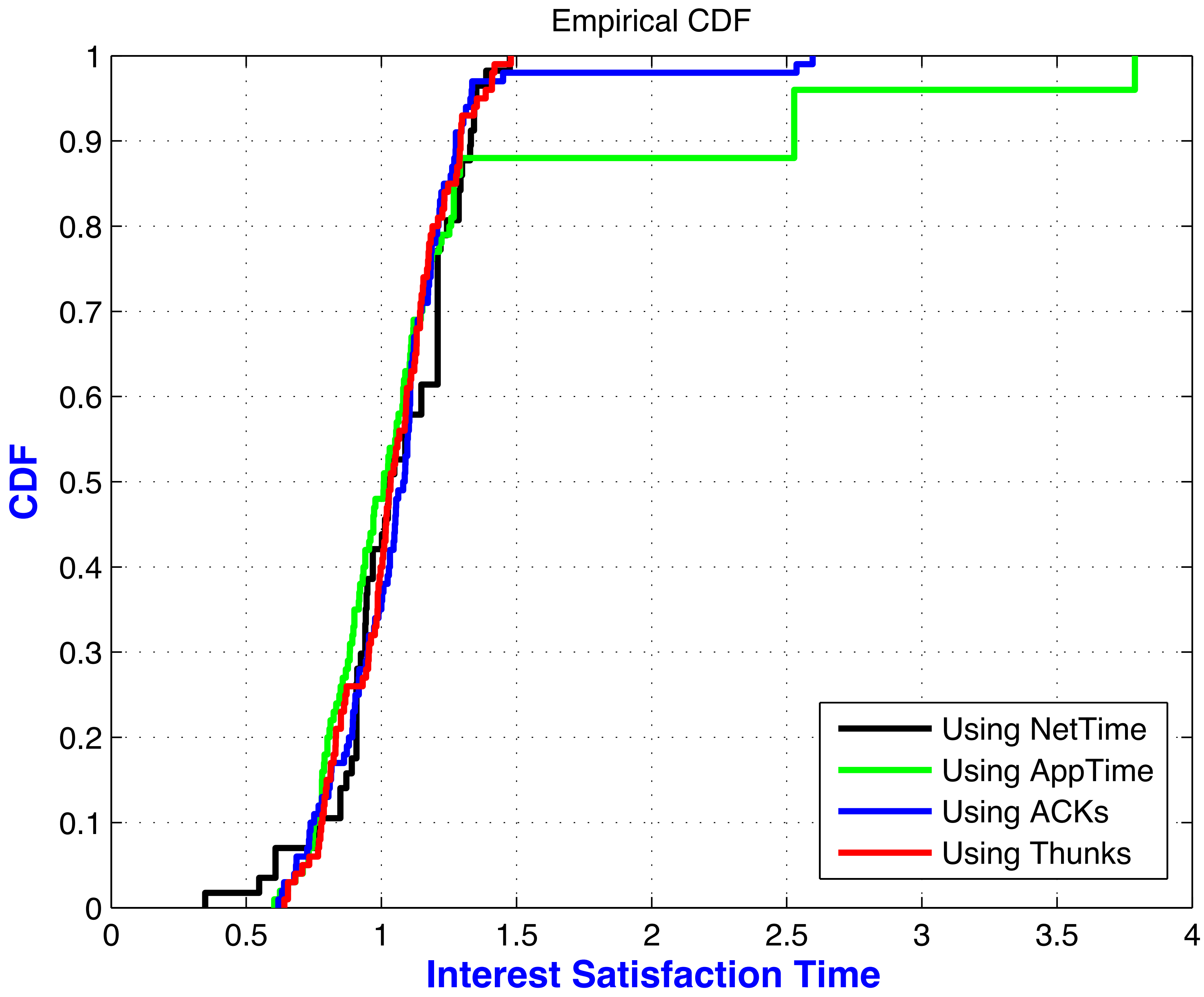


Evaluation
Application Time Effect





Evaluation


Empirical CDF
CDF
Interest Satisfaction Time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.5
1
1.5
2
2.5
3
3.5
4
Using NetTime
Using AppTime
Using ACKs
Using Thunks

Agenda

- ✓ Introduction
- ✓ Motivations and Problem Statement
- ✓ Using Per-Hop Error Detection
- ✓ Using Interest Acknowledgement
- ✓ Two Phase Architecture (Thunks)
- ✓ Evaluation
- **Conclusion**

Concluding

- Time perception in NDN
- Interest ACK VS Thunks
  - ✦ Mobility ?
  - ✦ Scalability ?
- More into explicit signaling