TEMPLE UNIVERSITY

中国科学技术大学 University of Science and Technology of China

TOUR: Time-sensitive Opportunistic Utility-based Routing in Delay Tolerant Networks

Mingjun Xiao[a,b], Jie Wu[b], Cong Liu[c], and Liusheng Huang[a]

[a] University of Science and Technology of China, China

[b] Temple University, USA

[c] Sun Yat-sen University , China

Outline

- Introduction on utility-based routing
- Motivation
- Problem
- Solution
- Simulation
- Conclusion

Introduction: utility-based routing

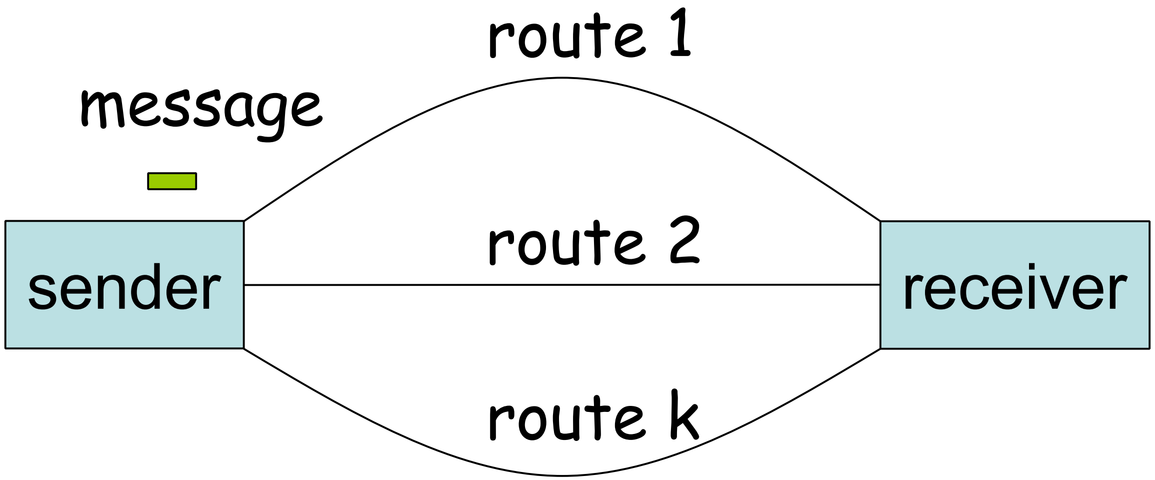
- **Concept** : Utility-based routing [Jiewu 08, 12]
  - **Utility** is a composite metric

    Utility (*u*) = Benefit (*b*) – Cost (*c*)

  - **Benefit** is a reward for a routing
  - **Cost** is the total transmission cost for the routing
  - Benefit and cost are uniformed as the same unit
  - **Objective** is to maximize the utility of a routing

Introduction: utility-based routing

- **Motivation** of Utility-based Routing
  - Valuable message: route (more reliable, costs more)
  - Regular message: route (less reliable, costs less)



Benefit is the successful delivery reward

Motivation

Utility-based routing → **Delay Tolerant Network (DTN)**

delivery delay is an important factor for the routing design

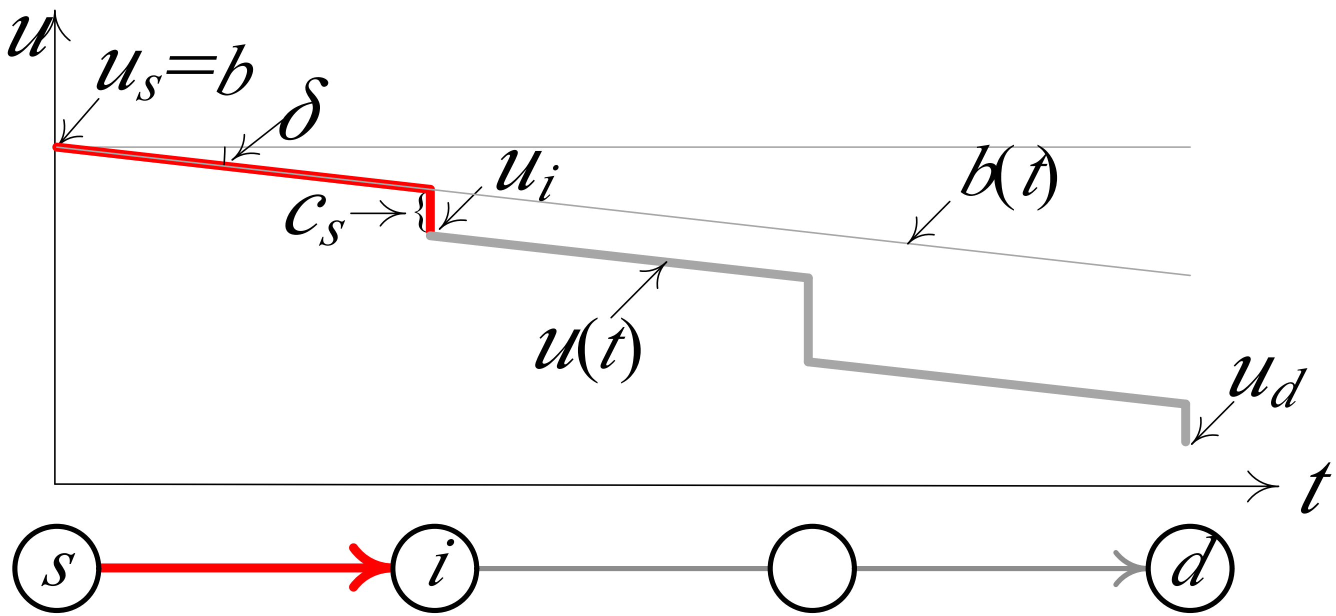
Time-sensitive utility-based routing

Time-sensitive utility model

- **Benefit**: a linearly decreasing reward over time

$$b(t)=\begin{cases} b-t\cdot\delta, & t\le b/\delta \\ 0, & t>b/\delta \end{cases}$$

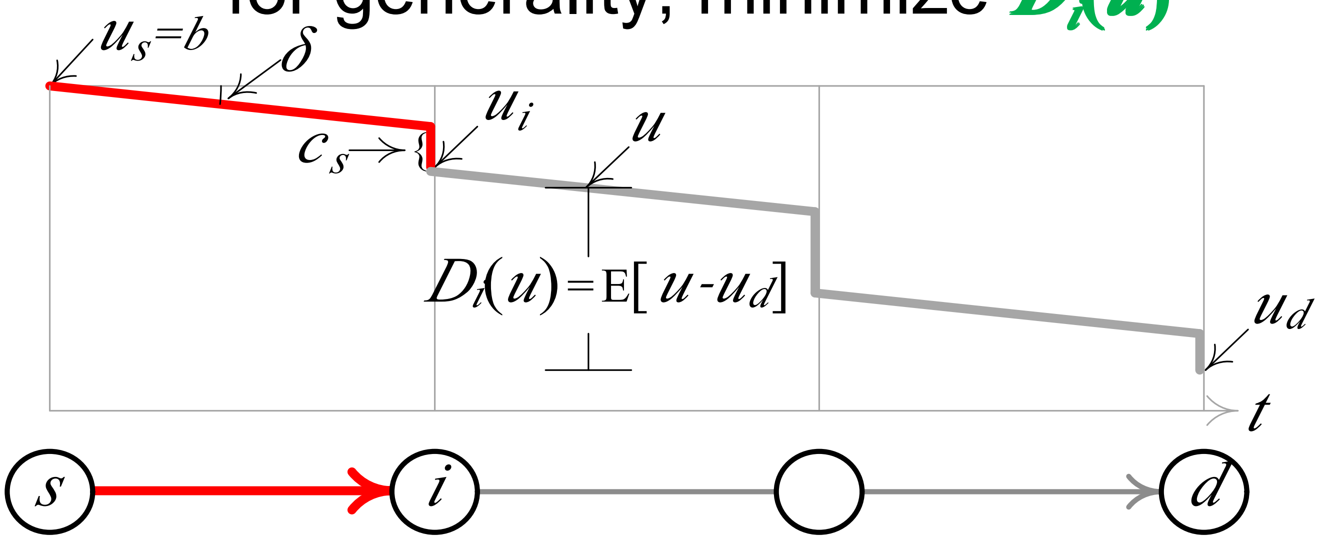
- **Utility**: $u(t)=b(t)-c$



Problem

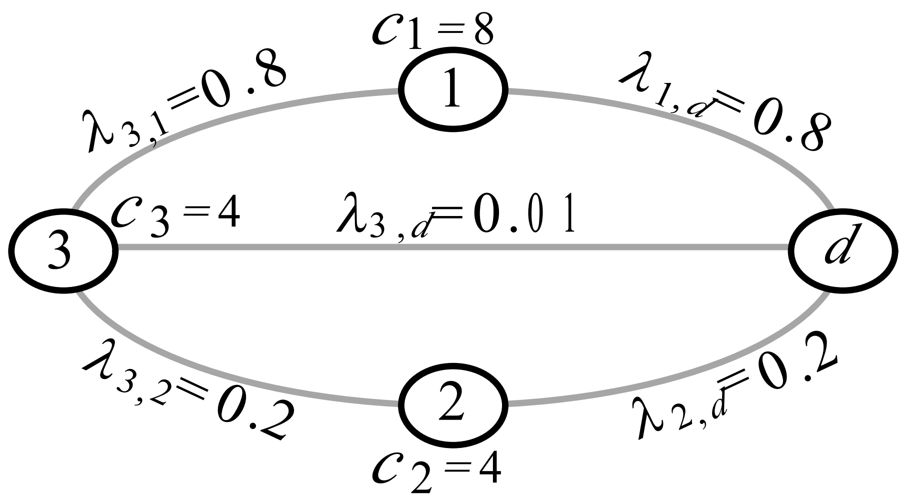
- **Time-sensitive utility-based routing in DTN**
  - DTN: $\mathbf{V=\{1, 2, \ldots\}}$, $\lambda_{i,j}$, c_i $(i, j \in \mathbf{V})$
  - source s, destination d, initial benefit b, benefit decay coefficient δ (**single copy**)
  - **Objective**: maximize $\mathrm{E}[u_d]$ or minimize $D_s(u_s)=b-\mathrm{E}[u_d]$

    for generality, minimize $D_i(u)$



Problem

- **A simple example**
  - DTN: $\mathbf{V=\{1, 2, 3, d\}}$, $\lambda_{i,j}$, c_i $(i, j \in \mathbf{V})$
  - source s=3, destination d, initial benefit b=20, benefit decay coefficient δ=2
  - **Objective**: minimize $D_3(u_3)$



Problem

- **The key problem**
  - **when a node *i* meets another node, whether the node *i* should forward messages to this encountered node, or ignore this forwarding opportunity, so that the node *i* can achieve the minimum $D_i(u)$**

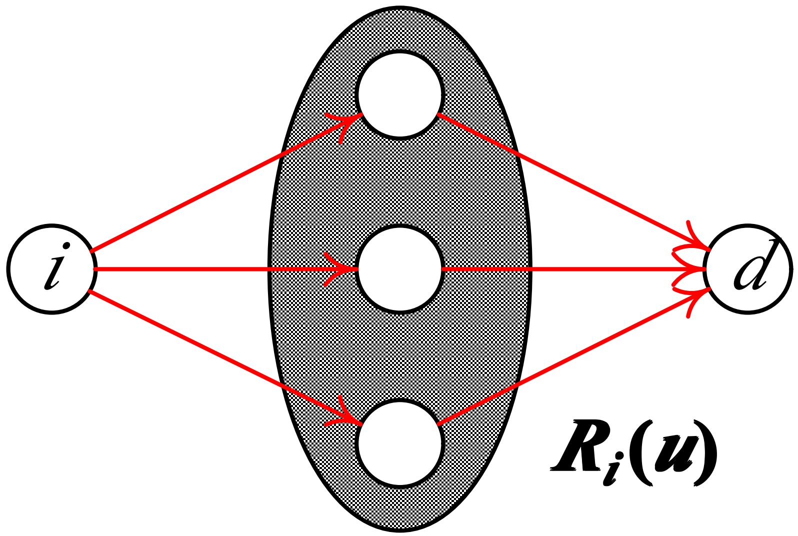
Solution

- **Basic idea:**

  Time-Sensitive Opportunistic Forwarding

  – Dynamically select relays: forwarding set $\boldsymbol{R_i(u)}$

  – Opportunistic forwarding scheme: only forward messages to nodes in forwarding sets; ignore the other nodes outside of the set



$\boldsymbol{R_i(u)}$

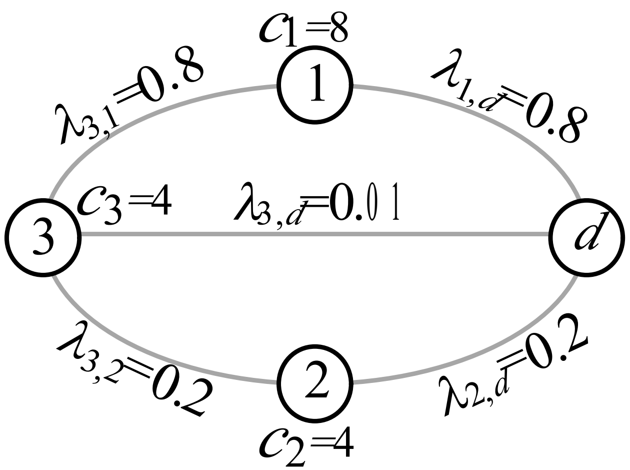
Solution

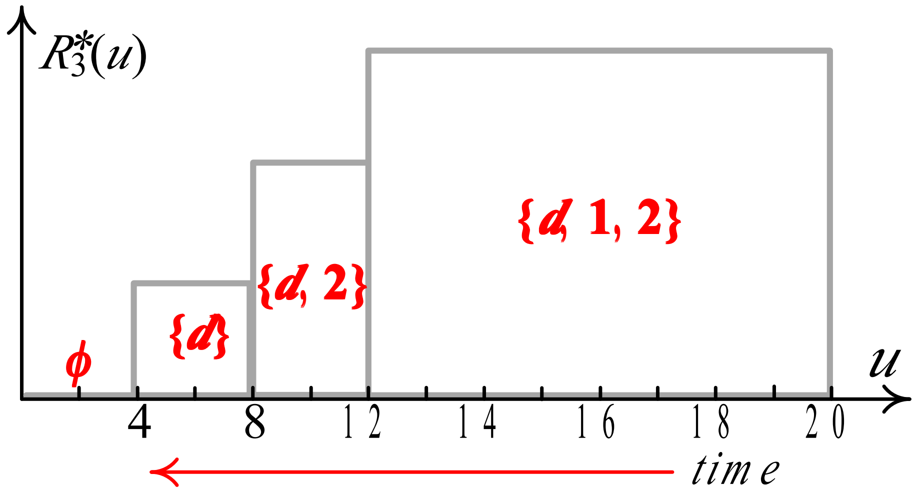
- **Basic idea:**

  Time-Sensitive Opportunistic Forwarding

  – Forwarding set $\boldsymbol{R_i(u)}$ is time-sensitive:

  vary with time, i.e., remaining utility u



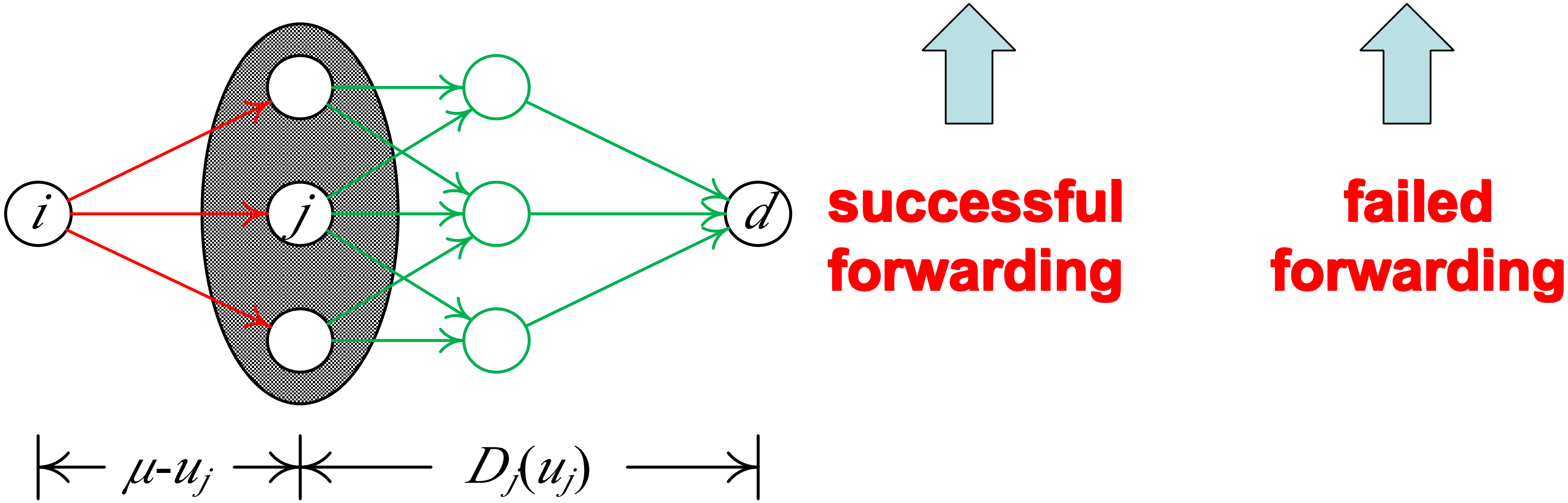


Solution

- **Determine optimal forwarding set**
  - **Computation formula $R_i^*(\mu)$**

$$R_i^*(u) = \arg\min_{R(u)\subseteq N_i} D_i(u)\Big|_{R(u)}$$

$$D_i(u=\mu)\,|_{R(u)} = \int_0^{\mu} \sum_{j\in R(u)} \rho_{i,j}(u)(\mu - u_j + D_j(u_j))\,du + p_f(\mu)\mu$$



Solution

- **Determine optimal forwarding set**

For a single node i: $R_i^*(\mu)$

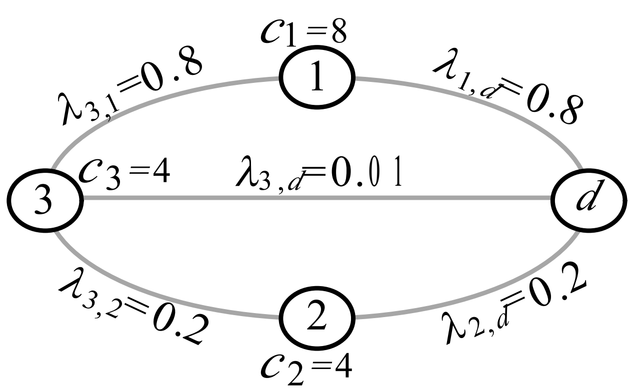
- **Assumption**: $D_j(\mu - c_i) = D_j(u_j)$ are known

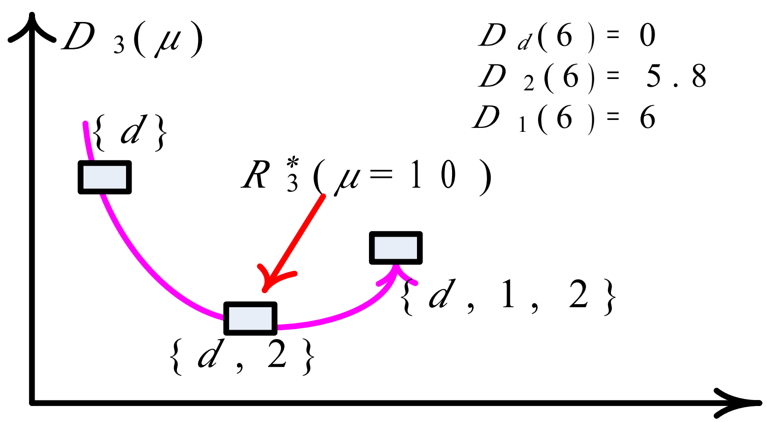
$D_1(\mu - c_1) < D_2(\mu - c_2) < \ldots < D_m(\mu - c_m)$

- **Method**: Greedily compute $R_i^*(\mu)$

$R_i^*(\mu)$: 1, 2, …, k, k+1, …, m

- **Correctness**: Theorem 1





Solution

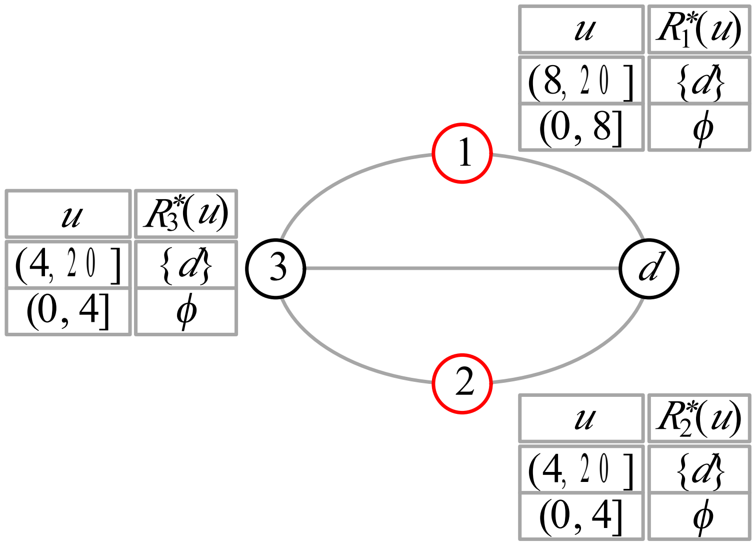
- **Determine optimal forwarding set**

For all nodes $i \in V$: $R_i^*(\mu)$

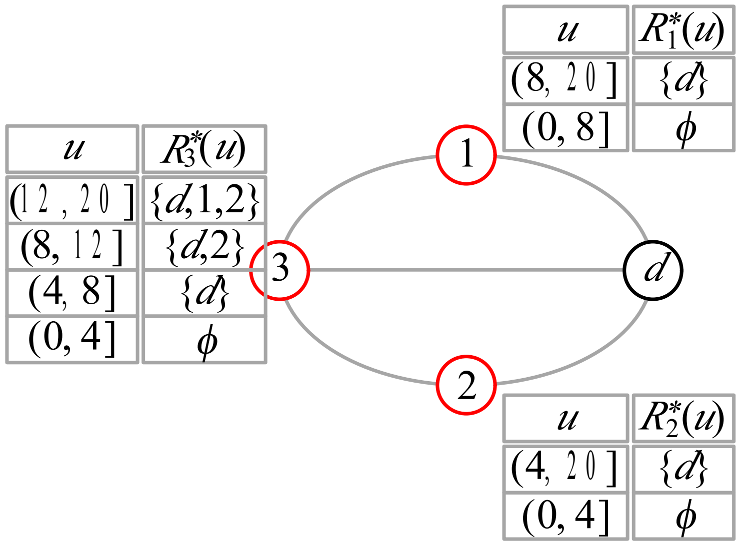
– **Method: iteratively compute $R_i^*(\mu)$ for all $i \in V$**

|V|-1 rounds of computation

– **Convergence: Theorem 2**



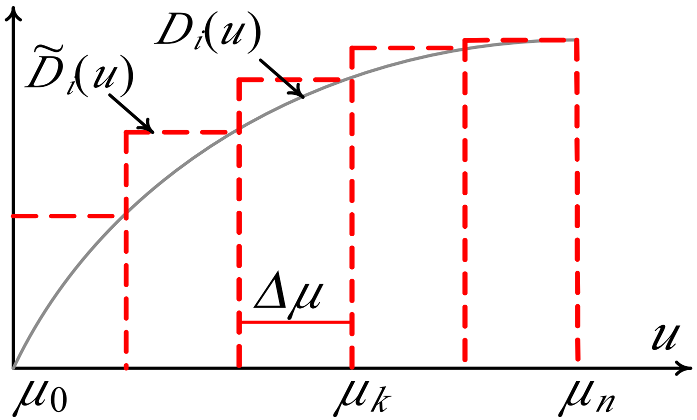
Round 1



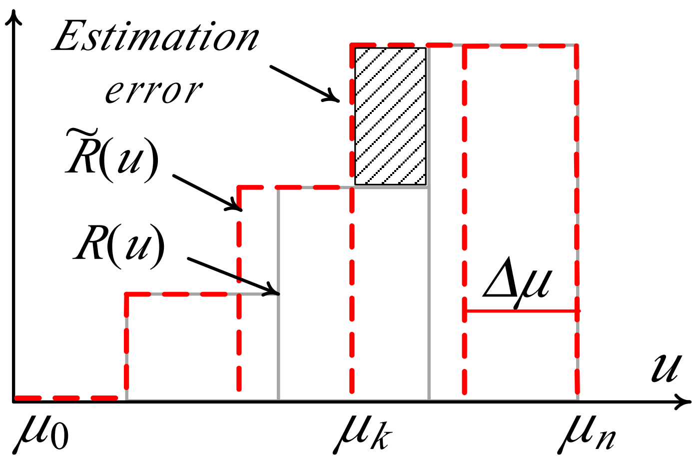
Round 2

Implementation

- **Discrete Process**
  - $D_i(u)$ ⇨ $\widetilde{D}_i(u)$



  - $R_i(u)$ ⇨ $\widetilde{R}_i(u)$



Implementation

- **Discrete Process**

$$D_i(\mu)\,|_{R(u)} = \int_0^{\mu} \sum_{j \in R(u)} \rho_{i,j}(u)(\mu - u_j + D_j(u_j))\,du + p_f(\mu)\mu$$

$$\Downarrow$$

$$\widetilde{D}_i(\mu)\,|_{\widetilde{R}(u)} = \int_0^{\mu} \sum_{j \in \widetilde{R}(u)} \widetilde{\rho}_{i,j}(u)(\mu - u_j + \widetilde{D}_j(u_j))\,du + \widetilde{p}_f(\mu)\mu$$

Theorem 3 gives the upper bound of estimation error of the discrete process

Simulation

- **Real trace used**
  - Cambridge Haggle Trace

Trace	Contacts	Length (d.h:m.s)	Routing nodes	External nodes
Intel	2,766	4.3:48.32	9	128
Cambridge	6,732	6.1:34.2	12	223
infocom	28,216	2.22:52.56	41	264.9

  - UMassDieselNet Trace

    40 buses

    55 days, Spring 2006

Simulation

- **Algorithms in comparison**
  - TOUR (10 discrete sampling points)
  - TOUR-OPT (100 discrete sampling points)
  - SimpleUtility, MinDelay, MinCost
- **Metrics**
  - Remaining utility
  - Derivation
  - Cost

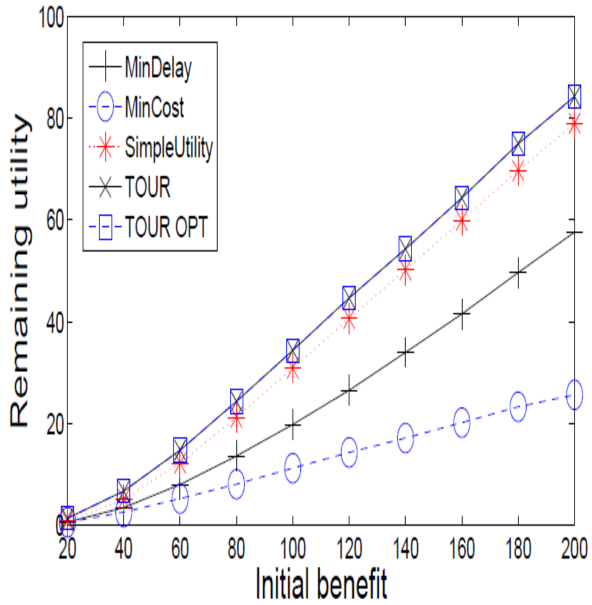
Simulation

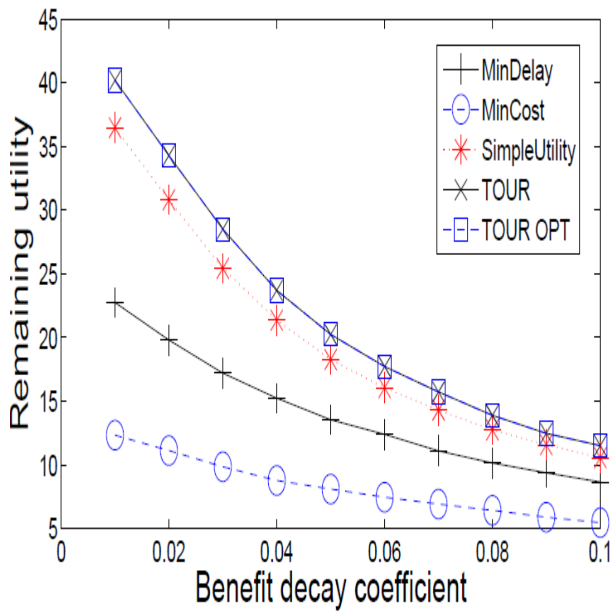
- **Settings**

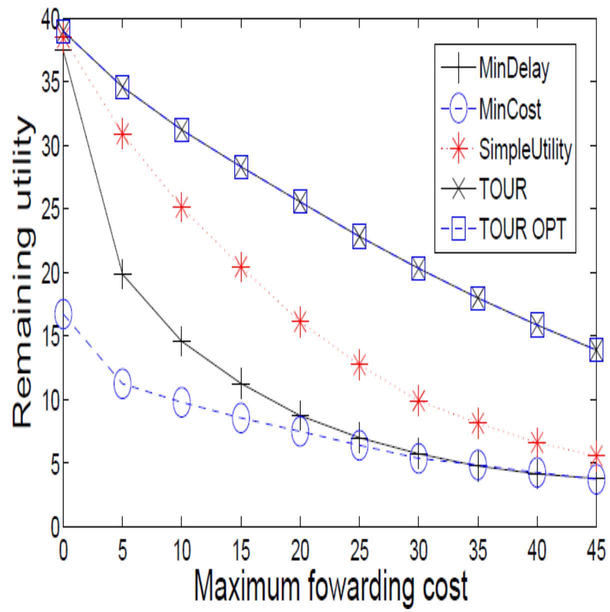
Parameter name	Default	Range
Initial benefit	100	20-200
Maximum forwarding cost	5	0-45
Benefit decay coefficient	0.02	0.01-0.1
Number of messages	30,000	

Simulation

- Results
  - Remaining utility vs. initial benefit, benefit decay coefficient, maximum forwarding cost

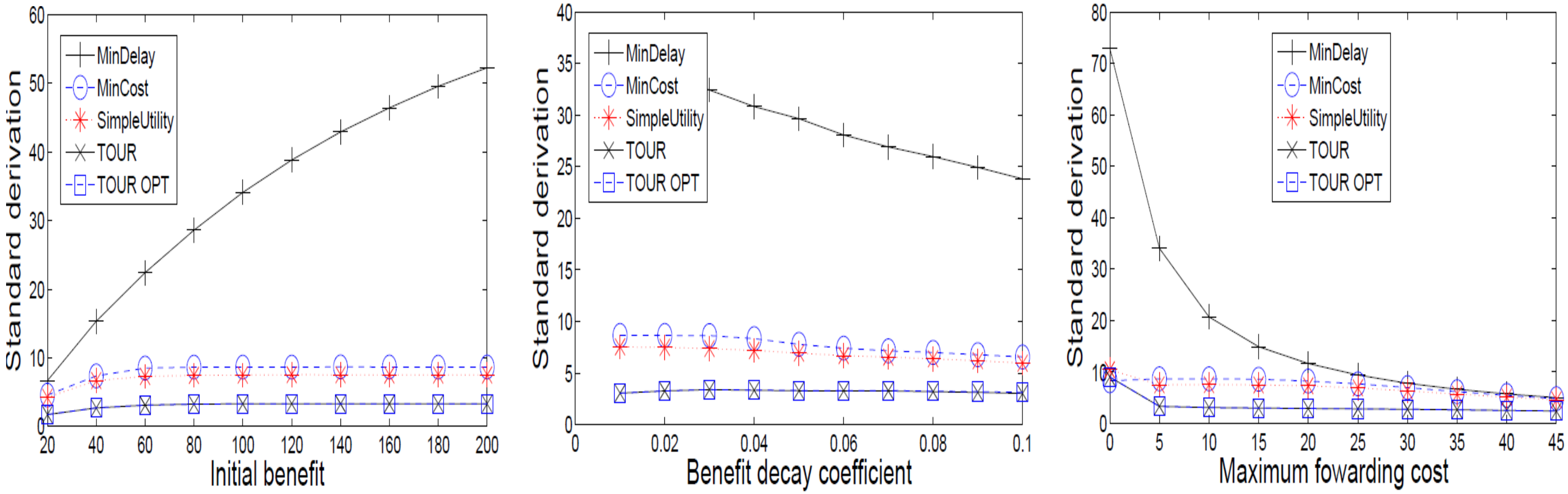






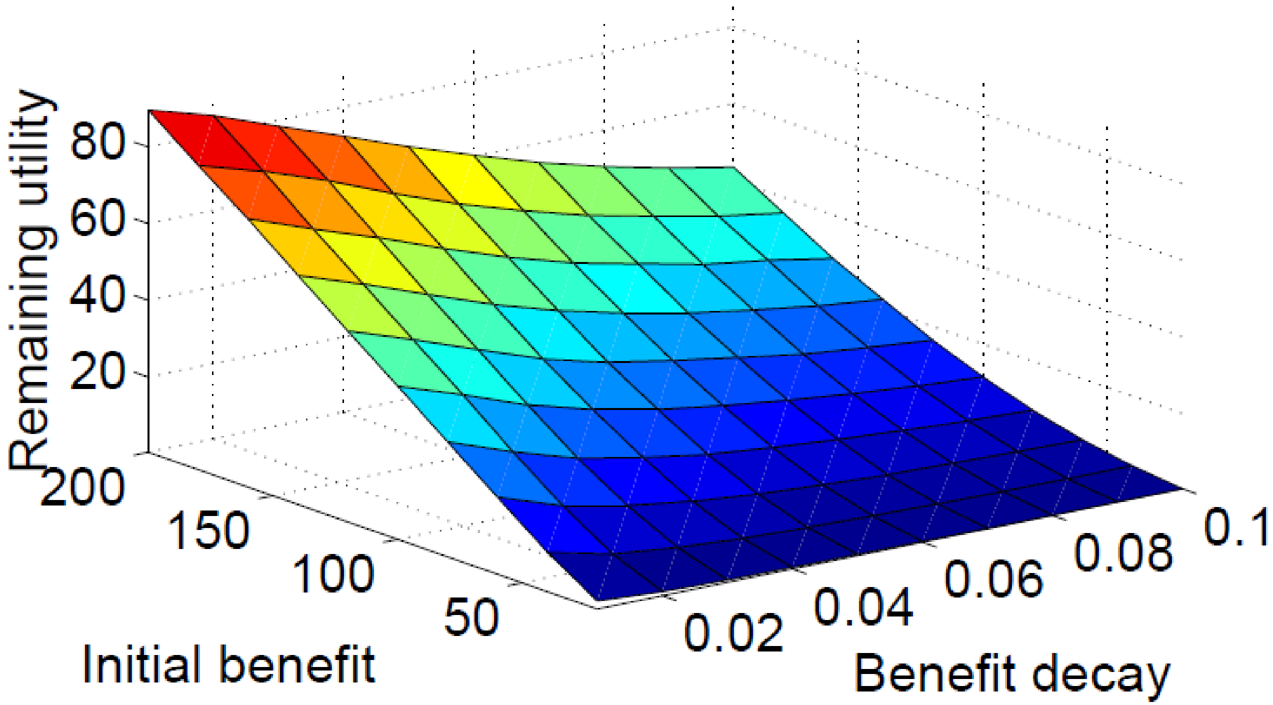
Simulation

- Results
  - Derivation vs. initial benefit, benefit decay coefficient, maximum forwarding cost



Simulation

- Results
  - Remaining utility vs. initial benefit and benefit decay coefficient



Conclusion

- Our proposed algorithm outperforms the other compared algorithms in utility.
- The larger the initial benefit and the smaller the benefit decay coefficient are, the larger the remaining utility would be.
- Our proposed algorithm can schedule different message deliveries to different paths.

Thanks!

Q&A