Robust and lightweight routing with attractor selection

Naotaka Ozuka, Naoki Wakamiya, Masayuki Murata
Osaka University, Japan

Overview of our routing mechanism
Packet reception or generation
Routing table maintenance

Routing Information for destination d
- Activity (a_d): goodness of current next-hop selection for destination d
- State value (A_d): goodness of neighbors as next-hop for destination d

Routing table maintenance process
- Collect path quality information
- Update routing information based on attractor selection

Learning from biology
- Attractor selection model [3]
  - \( f(x, y, z) \) \( = \nabla f(x) + \eta \nabla 
\) (potential function, \( \eta \) (Gaussian noise))
- Activity (a): goodness of current condition
- State value (A): state of system

Objectives
- Realize adaptive and robust routing in wired networks by adopting attractor selection model
- Selection of nutrient to synthesis
- Selection of next-hop to forward packets

Collection of path quality information
- Measure path quality (one-way delay) to destination d
  - Node sends route control message to destination d and d sends back feedback message
  - Calculate one-way delay as many as possible by using single pair of control and feedback messages

Background
- Rapid growth of information networks makes traditional mechanisms unsuccessful and unfeasible
- Traditional routing suffers from increased computational complexity and continuous change

Objective
- Realize adaptive and robust routing in wired networks by adopting attractor selection model
- Selection of nutrient to synthesis
- Selection of next-hop to forward packets

Collection of path quality information
- Measure path quality (one-way delay) to destination d
  - Node sends route control message to destination d and d sends back feedback message
  - Calculate one-way delay as many as possible by using single pair of control and feedback messages

Collection of path quality information
- Measure path quality (one-way delay) to destination d
  - Node sends route control message to destination d and d sends back feedback message
  - Calculate one-way delay as many as possible by using single pair of control and feedback messages
Updating routing information

- Routing information is updated based on calculated delay
  \[ \alpha = \frac{d_{\text{min}}}{d_{\text{min}} + \eta} \]
  \[ \alpha = \text{goodness of path} \]
  \[ m_i = \text{goodness of next-hop} \]
  \[ \eta_i = \text{Gaussian noise} \]
  \[ x = \text{number of neighbors} \]
  \[ \phi = \text{constant value} \]

- \( \alpha \) is low
  - \( \alpha = 0 \)
  - \( m_i = 0.1 \)
  - \( \eta_i = 0.2 \)
  - Random selection by \( i \)

- \( \alpha \) is high
  - \( \alpha = 1 \)
  - \( m_i = 0.3 \)
  - \( \eta_i = 0.3 \)
  - Stable selection by \( f(x) \)

Potential problem of hop-by-hop routing

- Independent selection of next-hop by each node may create looping path

- Loop detection and resolution
  - Check existence of loop and if exits find non-looping path

Simulation settings

- Simulator
  - Omnet++ [7]

- Network model
  - Waxman model [8]
  - Number of nodes: 50–400
  - Number of links: 2x the number of nodes

Parameters

- Proposal
  - Control interval: 100 [s]
  - OSPF
  - Default parameters
  - LSA exchange and Dijkstra computation per \( T \) [s]

Scenario 1: Evaluation of overhead

- Scenario
  - No change in topology and traffic
  - Number of nodes: 50–400

- Evaluation metrics
  - Processing time
    - Total of realistic processing time in updating routing information during \( T \) [s] (Intel Core i7-2600 3.4 [GHz], the memory of 16 [Gbyte])
  - Control overhead
    - Total size of forwarded messages during \( T \) [s]
      - Proposal: node control messages and feedback messages
      - OSPF: DD, LSR, LSU, and LSU message

Scenario 2: Evaluation of robustness

- Scenario
  - Number of nodes: 100
  - Removal of one randomly selected node on one randomly selected path

- Evaluation metrics
  - Reachability
    - Probability that message reaches destination
    - \( \text{Reachability} \) = \( \frac{1}{N} \) of \( S \) to \( D \)

- Average path length

Processing time and control overhead

- Processing time
  - OSPF: \( O(N^2) \)
  - Control overhead
  - OSPF: \( O(N^2) \)

- Higher efficiency and scalability

2014/2/21
Reachability and average path length

- **Reachability**
  - OSPF: Drop to zero right after a failure
  - Proposal: Nearly one right after a failure
  - Slightly decrease by stochastic searches

- **Difference in average path length**
  - Gradually improved from 0.82 (0 s) to 0.56 (3000 s)

Conclusion and future work

- **Conclusion**
  - We propose a robust, scalable, and adaptive routing mechanism for wired networks.
  - Shorter processing time and lower control overhead than OSPF.
  - But, proposal suffers from slightly lower reachability.
  - Disadvantage of stochastic behavior of proposal.

- **Future work**
  - Extension to multi-path routing.
  - Each node adaptively selects one of alternative paths by using attractor selection model.
  - 100% reachability is expected.