DEFLECTION ROUTING IN COMPLEX NETWORKS

Ljiljana Trajković

Communication Networks Laboratory

http://www.ensc.sfu.ca/~ljilja/cnl/

Simon Fraser University

Vancouver, British Columbia, Canada

Communication Networks Laboratory

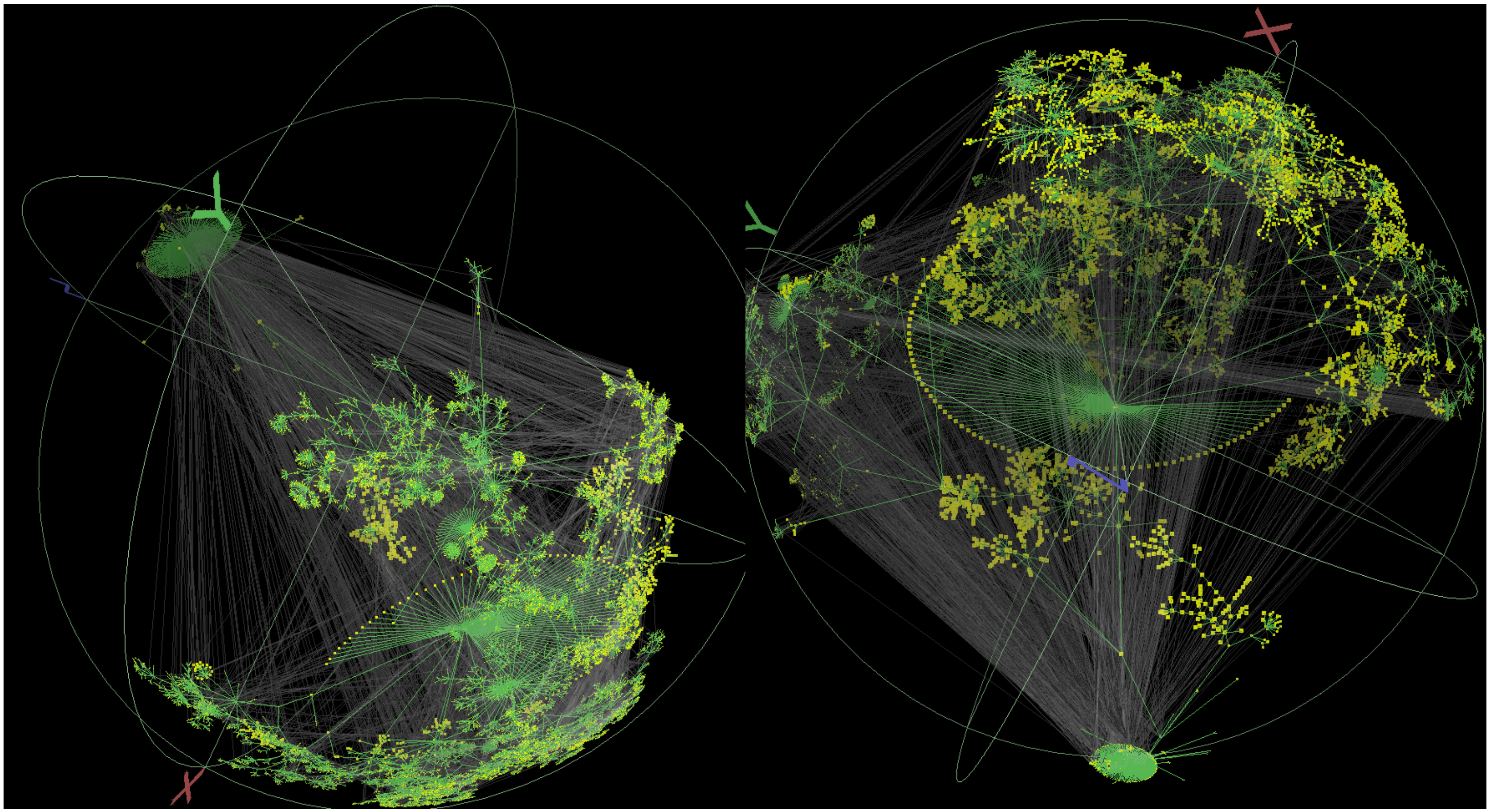
Ph.D. student:

- Soroush Haeri

Outline

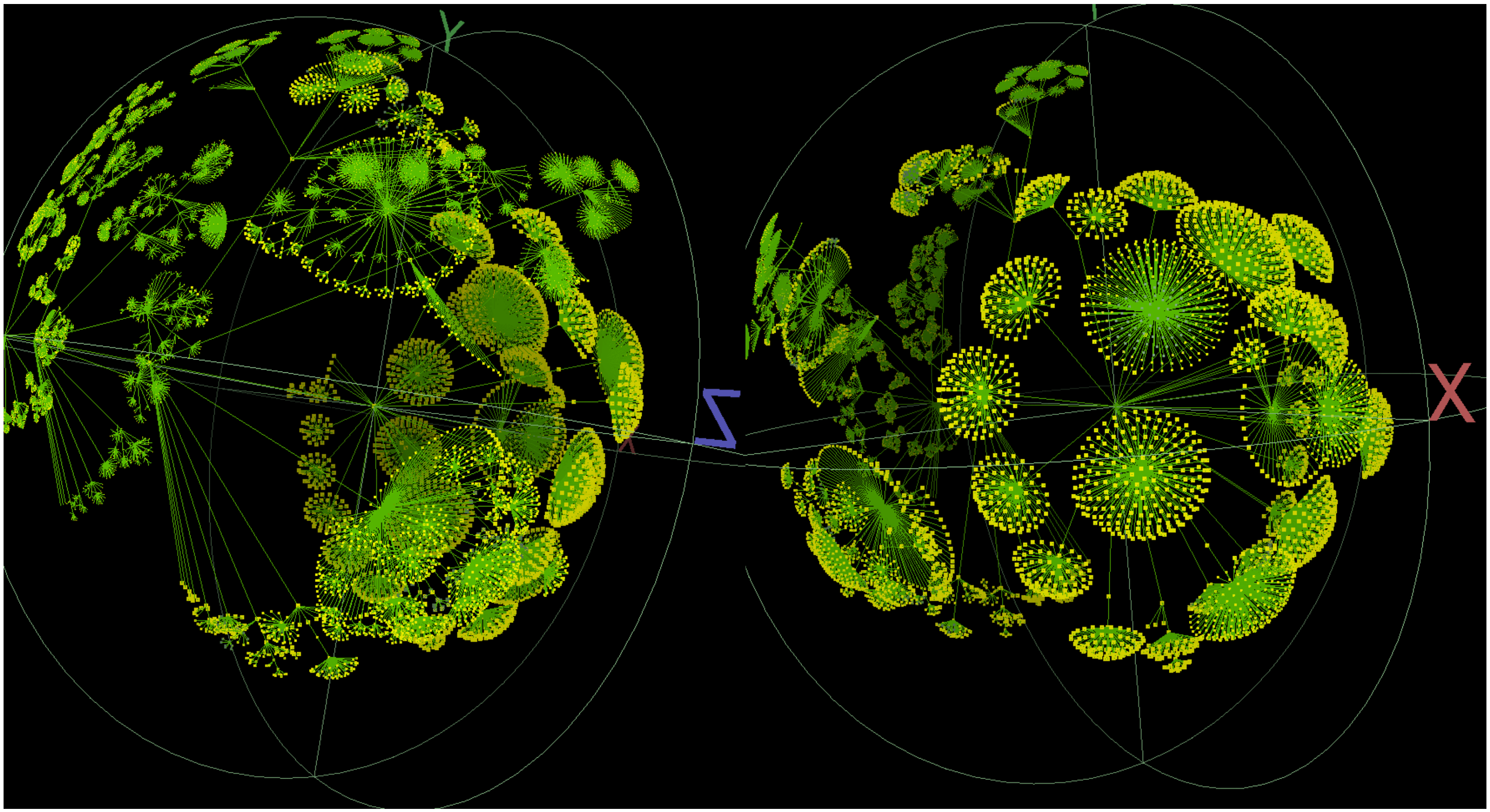
- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Riesling: 54,893 nodes and 79,409 links



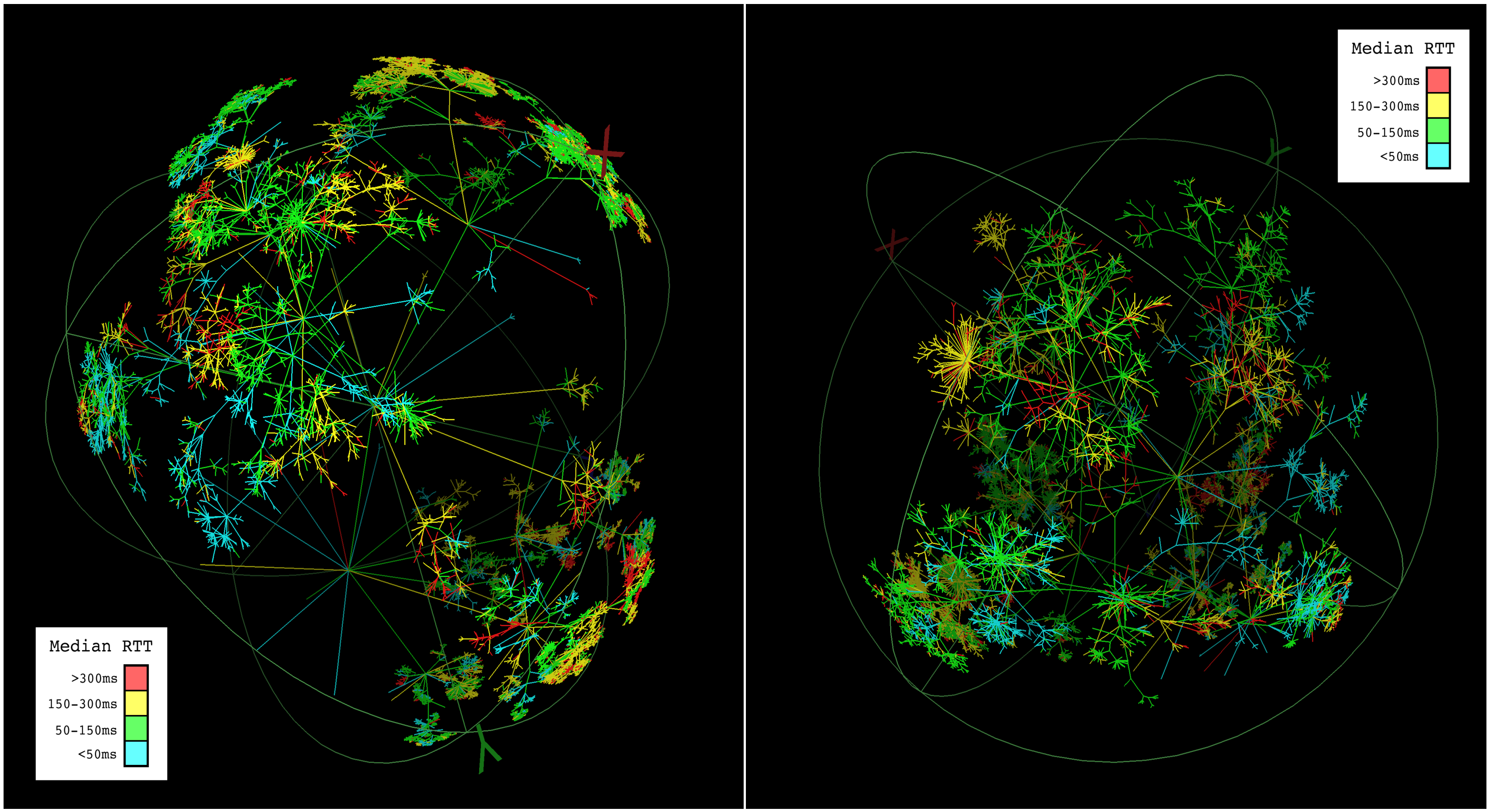
http://www.caida.org/home

CVS Repository:
18,474 nodes and 18,473 links



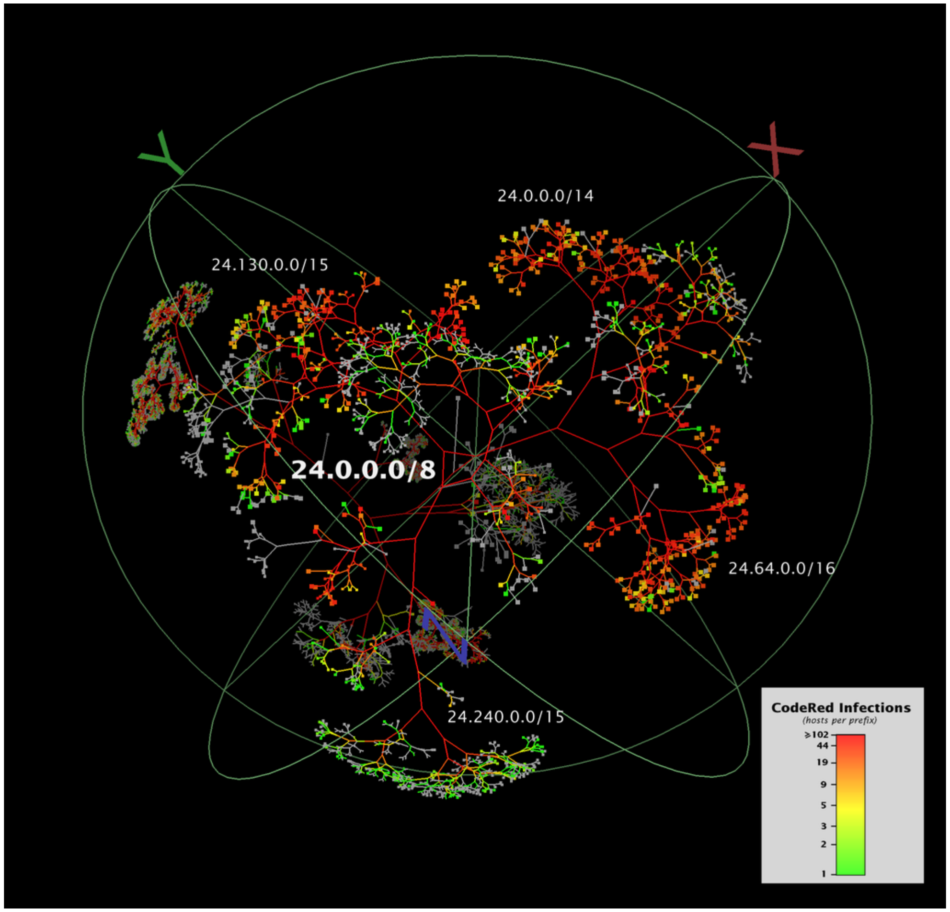
http://www.caida.org/home

Round-Trip Time Measurements: 63,631 nodes and 63,630 links



http://www.caida.org/home

Code Red Infection



http://www.caida.org/home

Introduction

- Design of routing protocols is influenced by:
  - discovery of power-law distribution of node degrees
  - scale-free properties of communication networks
- Power laws are present in the Internet's inter-Autonomous System-level topology
- Waxman and Barabási-Albert algorithms have been widely used to generate Internet-like graphs

Outline

- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Buffer-Less Architecture and Contention

- Buffer-less nodes do not possess first-in-first-out (FIFO) buffers to queue data
- Buffer-less network architectures:
  - optical burst switched networks
  - on-chip networks
- Contention is the main source of packet loss in buffer-less networks

Buffer-Less Architecture and Contention

- Routing protocol selects the optimal path between a source and a destination
- Contention:
  - multiple arriving traffic flows at a node need to be routed through a single outgoing link
- If there is no contention resolution scheme:
  - a flow is routed through the optimal outgoing link defined by the routing table
  - other flows are discarded because a node has no buffer

Deflection Routing

- Deflection routing may be employed as a contention resolution scheme
- A deflection routing algorithm temporarily misroutes packets instead of buffering or discarding
- Deflection routing and the underlying routing protocol co-exist in a network

Deflection routing: Tradeoff

- Tradeoff:
  - quality of deflection routing vs. the number of signaling messages required by the deflection routing algorithm
- The underlying routing protocol generates a significant number of control signals
- Deflection routing protocols should generate few control signals
- Goal:
  - exhibit good performance while transmitting fewer control signals

Complex Networks and Deflection Routing

- High-speed optical links are often used to connect the Internet autonomous systems
- Optical burst switching may be used for inter-Autonomous System communication
- Typical topologies used to test deflection routing algorithms:
  - small-size networks (National Science Foundation network)
  - torus topologies
- These topologies do not resemble structure of the Internet

Outline

- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Reinforcement Learning

- Formalizes trial-and-error-based learning processes
- Four basic elements:
  - agent or decision maker
  - environment of the agent
  - actions performed by agent
  - feedback signals generated by environment
- Objective:
  - maximize/minimize the rewards/penalties that agent receives from environment

Reinforcement Learning: Agent-Environment Interaction

- Agent observes the state of the environment and selects an appropriate action
- Environment generates a reinforcement signal and transmits it to the agent
- Agent employs the reinforcement signal to improve its subsequent decisions

Reinforcement Learning

- Reinforcement learning agent requires:
  - information about state of environment
  - reinforcement signals from environment
  - rule (algorithm) to update its statistics
- Examples of learning algorithms:
  - Q-learning
  - feed-forward networks

Q-Learning

- Q-learning algorithm has been used to design a learning and decision making module
- Q-learning:
  - is a simple reinforcement learning algorithm
  - maintains a Q-value Q(s,a) in a Q-table for every state-action pair

C. J. C. H. Watkins and P. Dayan, "Technical note, Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

Q-Learning

- Q-learning update rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

- s : state of the system
- a : action
- α : learning rate
- r : reward
- γ : discount factor
- t and $t+1$: two consecutive decision instances

Reinforcement Learning for Deflection Routing

- Reinforcement learning algorithms are random in nature
- Randomness enables a deflection routing protocol to make viable decisions while transmitting fewer control signals
- Deflection decision or action:
  - select an alternate outgoing link that a node may use to deflect a traffic flow
- Modules required for reinforcement learning:
  - signaling
  - learning and decision making

Reinforcement Learning for Deflection Routing

- Signaling module:
  - is aware of the state of the system
  - implements the algorithm for generating and delivering feedback signals

Reinforcement Learning for Deflection Routing

- Learning and decision making module:
  - In case of contention:
    - receives the state of the system from the signaling module
    - generates a deflection decision
  - After generating a deflection decision:
    - receives feedback from the signaling module
    - employs feedback to enhance its future decisions

Outline

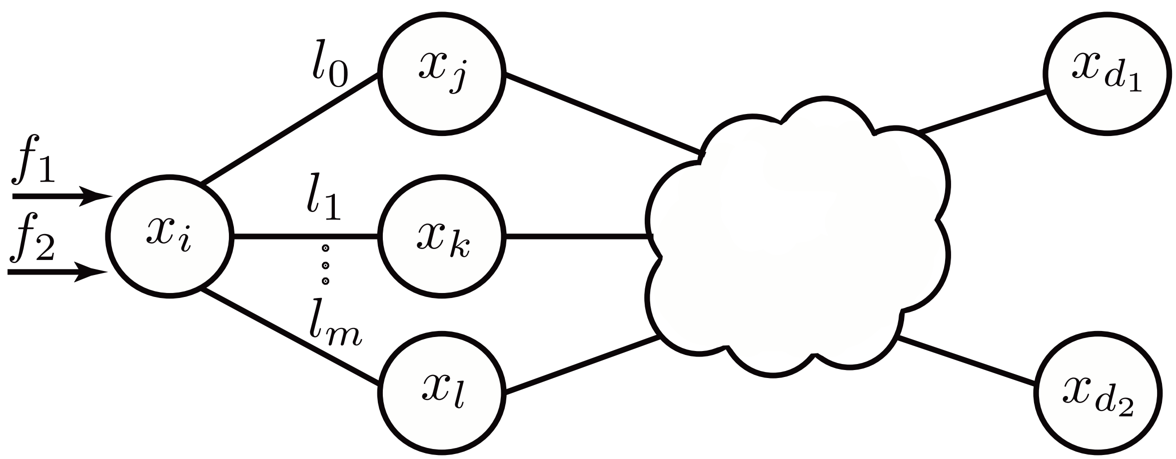
- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Network Model

- Consider a network with n buffer-less nodes:

$$\mathcal{N} = \{x_1, x_2, \ldots, x_n\}$$

- Node x_i is connected to its m neighbors via outgoing links: $\mathcal{L} = \{l_0, l_1, \ldots, l_m\}$



Network Model

- System state:
  - original destination of a traffic flow that needs to be deflected
  - set of states encountered by an arbitrary node x_i:

$$\mathcal{S}_{x_i} = \{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$$

Network Model

- Action:
  - alternate outgoing link used to deflect a traffic flow
  - set of actions available to a node x_i with m outgoing links:

$$\mathcal{A}_{x_i} = \{l_0, l_1, \ldots, l_m\}$$

Network Topology: Waxman Algorithm

- Commonly used model to generate random network topologies
- Probability of a link connecting nodes u and v in Waxman graph:

$$\Pr\left(\{u,v\}\right) = \eta \exp\left(\frac{-d(u,v)}{L\delta}\right)$$

- Graphs generated with larger η and smaller δ have larger number of short edges:
  - bi-component
  - longer hop diameter
  - shorter length diameter

Network Topology: Waxman Algorithm

- Graphs generated using Waxman algorithm:
  - do not resemble the Internet backbone and hierarchy
- The algorithm does not guarantee generating a connected graph

Network Topology: Barábasi-Albert Algorithm

- Generates scale-free graphs
- Power-law distribution of node degrees:
  - incremental growth
  - preferential connectivity
- Begin with connected network of *n* nodes
- New node *i* added to the network connects to existing nodes *j* with probability:

$$\Pr(i, j) = \frac{d_j}{\sum_{k \in N} d_k}$$

Network Topology: Barábasi-Albert Algorithm

- d_j: degree of node j
- N: set of all nodes
- $\sum_{k \in N} d_k$: sum of degrees of all nodes in the network

Outline

- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

iDef Framework

- Facilitates development of reinforcement learning-based deflection routing protocols
- Abstracts a reinforcement learning-based deflection routing algorithm in three modules:
  - mapping
  - decision making
  - signaling

iDef Framework

- Minimized dependency among its modules enables:
  - implementation of portable deflection routing protocols
  - design of modules that can be replaced without changing the entire design
- Example of portability:
  - replacing the decision-making algorithm requires no changes in signaling module

Outline

Reinforcement Learning-Based Deflection Routing Algorithms

- Node Degree Dependent Algorithm with Q-Learning (Q-NDD)
  - scales well in larger networks
  - its complexity depends on the node degree
  - feedback signals are received only if the deflected packet is discarded by another node

Reinforcement Learning-Based Deflection Routing Algorithms

- Predictive Q-learning Deflection Routing Algorithm (PQDR)
  - employs the predictive Q-Routing (PQR)
  - addresses the shortcomings of Q-learning
  - its complexity depends on the network size
  - feedback signals are received for every deflected packet

Reinforcement Learning-Based Deflection Routing Algorithms

- Reinforcement Learning Deflection Routing Scheme (RLDRS)
  - employs the Q-learning algorithm for deflection routing
  - its complexity depends on the network size
  - does not generate optimal routing policies in networks with low loads
  - does not learn new optimal policies in cases when network load decreases

Outline

- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Q-NDD

- Complexity of the Node Degree Dependent (NDD) algorithm depends only on a node degree
- Consider optical burst switched networks network with *N* nodes:
  - each node maintains a Q-table
  - all nodes are NDD compatible
- NDD defines a state of the system by:
  - states of the optical interfaces
  - output port defined by the routing table

Q-NDD

- NDD defines an action as an output port number
- On burst transmission, a node:
  - inspects the routing table for the next hop
  - checks the status of its optical interfaces

Q-NDD

Four cases may occur:

1. desired optical interface is available
2. desired optical interface is busy and the burst has not been deflected earlier by any other node
3. desired optical interface is busy and the burst has been deflected earlier by another node
4. all optical interfaces are busy and the burst has been deflected earlier by another node

Case 1 and Case 2

- Case 1: Desired optical interface is available:
  - optical cross-connects are configured according to the path defined by the routing table
- Case 2: Desired optical interface is busy and the burst has not been deflected earlier by another node:
  - node passes the state of the system to the Q-learning module

Case 2 (cont.)

- Q-learning module then:
  - passes the output port number (action) associated with the maximum Q-value to the NDD module
  - waits for feedback
  - makes no new decisions during the idle interval

Case 2 (cont.)

- NDD module:
  - adds to the burst header:
    - unique ID number
    - address of the node that initiated the deflection
    - deflection hop counter DHC, which increments each time other nodes deflect the burst

Case 2 (cont.)

- records the current time as the deflection time (DfT) with the ID that has been added to the burst
- initiates the drop notification (DN) timer
- records the action selected by the Q-learning module
  - records are used if the node needs to deflect a burst:
    - that has been deflected earlier
    - during an idle interval

Case 2 (cont.)

- waits for a feedback signal for DN_{max} seconds
  - if no feedback is received:
    - assumes that the burst has been successfully delivered
    - returns the maximum reward value to the
      Q-learning module for an update

Case 2 (cont.)

- if feedback is received:
  - calculates a reward value based on the feedback signal
  - returns the reward value to the Q-learning module for an update

Case 3

- Case 3: Desired optical interface is busy and the burst has been deflected earlier by another node:

Case 3 (cont.)

- NDD module:
  - checks the deflection hop counter (DHC) field in the burst header and if:
    - DHC == DHC_{max}: prepares a feedback signal and discards the burst
    - DHC < DHC_{max}: checks the state of the system and performs the latest action that the Q-learning module has generated for the current state

Case 4

- Case 4: All optical interfaces are busy and the burst has been deflected earlier by another node:
    - NDD module prepares a feedback signal and discards the burst

Feedback Signal

- Feedback signal is composed of:
  - burst ID number that was assigned to the burst by the node that initiated the deflection
  - DHC field value:
    - DHC is not equal to DHC_{max} when the burst is discarded because the node if fully congested
  - drop time DrT:
    - time instant when the burst was discarded

Reward

- Reward is generated based on the feedback signal received at the node that has initiated the deflection
  - NDD module:
    - calculates the total travel time:

$$TTT = DrT - DfT$$

    - uses a decreasing function with the global maximum at (0,0) to map TTT and DHC to a real valued reward

Reward

- example:

$$maxR, \alpha, \beta > 0$$

$$\begin{cases} maxR & DHC = 0 \text{ and } TTT = 0 \\ -\alpha \times DHC - \beta \times TTT & \text{otherwise} \end{cases}$$

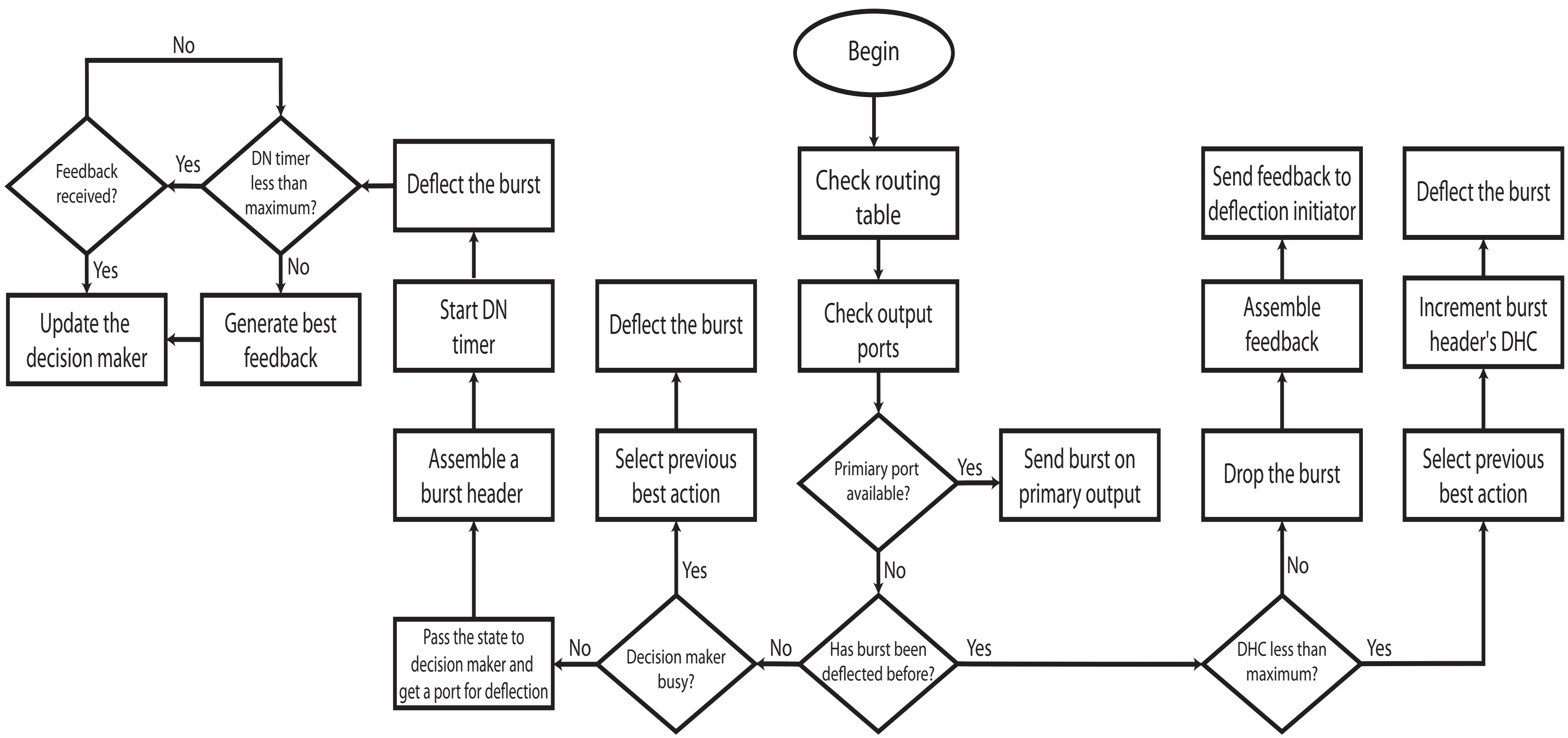
Q-Learning Module Update

- Q-learning module updates the Q-value of the current state and the selected action as:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r - Q(s,a)\big)$$

  - s: state of the system
  - a: action
  - α: learning rate
  - r: reward

NDD: Flow Chart



Outline

- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Predictive Q-learning Deflection Routing: PQDR

- PQDR learning and decision making module of a node x_i maintains four tables:
  - Table 1
    - $Q_{x_i}(x, l)$ stores accumulated rewards that x_i receives for deflecting packets to destinations x via outgoing link l
  - Table 2
    - $B_{x_i}(x, l)$ stores the minimum Q-value that x_i has calculated for deflecting packets to destinations x via outgoing link l

PQDR

- PQDR learning and decision making module of a node x_i maintains four tables:
  - Table 3
    - $R_{x_i}(x, l)$ stores deflection decision recovery rates for deflecting packets for destinations x via outgoing link l
  - Table 4
    - $U_{x_i}(x, l)$ stores the time instant when has last updated the (x, l) entry of its Q-table after receiving a reward

Three steps of PQDR

- Generate deflection decision
- Update the tables
- Generate control signals

Generate the Deflection Decision: ζ

- Calculate Δt

$$\Delta t = t_c - U_{x_i}(x_{d_2}, l_i)$$

  - t_c : current time
  - l_i : original outgoing link
  - x_i: source node
  - x_{d_2}: destination node

- Compute $Q'_{x_i}(x_{d_2}, l_i)$ =

$$\max\left(Q_{x_i}(x_{d_2}, l_i) + \Delta t \times R_{x_i}(x_{d_2}, l_i), B_{x_i}(x_{d_2}, l_i)\right)$$

Generate the Deflection Decision: ζ

- Use Δt and $Q'_{x_i}(x_{d_2}, l_i)$ to generate deflection decision ζ (outgoing link)

$$\zeta \leftarrow \arg\min_{l_i \in \mathcal{L}} \{Q'_{x_i}(x_{d_2}, l_i)\}$$

Algorithm for Table Updates

- When x_i receives a reward r from the network for deflecting a packet on the outgoing link ζ, it updates its four tables:
- Q_{x_i}
- B_{x_i}
- R_{x_i}
- U_{x_i}

Algorithm for Table Updates

- Update Q_{x_i}:
  - calculate

$$\phi = r - Q_{x_i}(x_{d_2}, \zeta)$$

  - update

$$Q_{x_i}(x_{d_2}, \zeta) = Q_{x_i}(x_{d_2}, \zeta) + \alpha \times \phi$$

  - $0 < \alpha \leq 1$: learning rate

- Update B_{x_i}:

$$B_{x_i}(x_{d_2}, \zeta) = \min(B_{x_i}(x_{d_2}, \zeta), Q_{x_i}(x_{d_2}, \zeta))$$

Algorithm for Table Updates

- Update R_{x_i}:

$$R_{x_i}(x_{d_2},\zeta) = \begin{cases} R_{x_i}(x_{d_2},\zeta) + \beta \dfrac{\phi}{t_c - U_{x_i}(x_{d_2},\zeta)} & \phi < 0 \\ \gamma R_{x_i}(x_{d_2},\zeta) & \text{otherwise} \end{cases}$$

  - $0 < \beta \leq 1$: recovery learning rate
  - $0 < \gamma \leq 1$: decay rate

- Update U_{x_i}:

$$U_{x_i}(x_{d_2},\zeta) = t_c$$

Signaling Algorithm

- PQDR signaling module of a node x_i maintains one table:
  - $P_{x_i}(l)$ stores blocking probabilities of outgoing links l attached to node x_i

Signaling Algorithm

- $P_{x_i}(l)$ is updated periodically every τ seconds:

$$P_{x_i}(l_i) = \begin{cases} \dfrac{\omega_{l_i}}{\lambda_{l_i} + \omega_{l_i}} & \lambda_{l_i} + \omega_{l_i} > 0 \\ 0 & \text{otherwise} \end{cases}$$

  - λ_{l_i} : number of packets that were successfully transmitted on link l_i
  - ω_{l_i}: numbers of packets dropped on link l_i

Signaling Algorithm

- When a node x_k receives a deflected packet for the destination x_{d_2} from its neighbor x_i:
  - routes the packet through one of its outgoing links l_{kl} by using its routing table or its PQDR module
  - calculates the feedback value:

$$\nu = Q_{x_k}(x_{d_2}, l_{kl}) \times D(x_k, x_l, x_{d_2})$$

    - $D(x_k, x_l, x_{d_2})$: number of hops from x_k to destination x_{d_2} through its neighbor x_l

Signaling Algorithm

- sends the feedback signal to node x_i that initiated the deflection
- Node x_i receives the feedback ν for its decision ζ from its neighbor x_k and calculates the reward r:

$$r = \frac{\nu(1 - P_{x_i}(\zeta))}{D(x_i, x_k, x_{d_2})}$$

- Reward r is passed to the PQDR's learning and decision making module for table updates

Outline

- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Simulation Scenarios

- Three reinforcement learning based deflection routing algorithms are compared:
  - Q-learning based Node Degree Dependent: Q-NDD
  - Predictive Q-learning based deflection routing: PQDR
  - Reinforcement learning based deflection routing scheme (RLDRS)

Simulation Scenarios

- Network topologies:
  - Waxman
  - Barabási-Albert
  - 10, 20, 50, 100, 200, 500, and 1,000 nodes
  - 24, 48, 120, 240, 480, 1,200, and 2,400 Poisson traffic flows

Simulations: Network Architecture

- Buffer-less optical burst switching architecture:
  - 1 Gbps fiber links
  - single wavelengths
- Traffic flows:
  - network load at 40%
  - Poisson arrivals
  - 0.5 Gbps data rate
  - 50 bursts:
    - each burst carries 12.5 kB payload

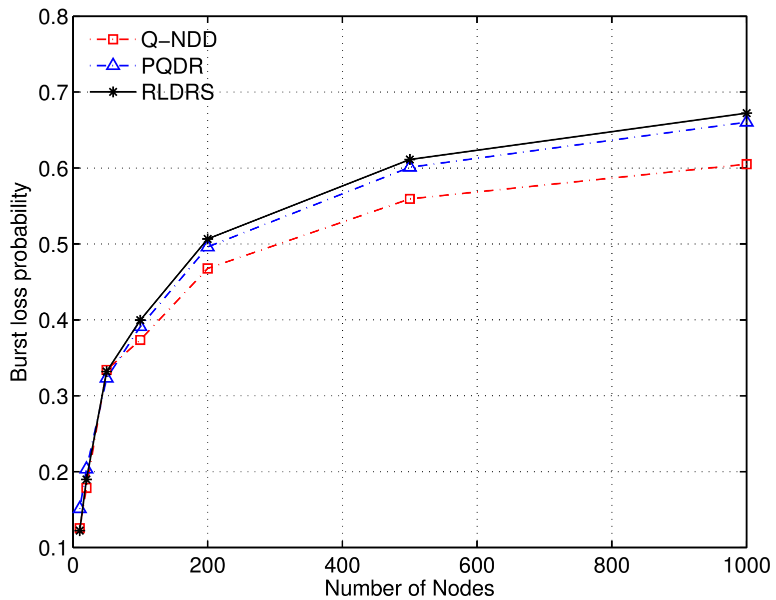
Simulations Parameters

- Deflection routing parameter:
  - Learning rate: $\alpha = 0.1$
  - Learning recovery rate: $\beta = 0.7$
  - Recovery decay rate: $\gamma = 0.9$
  - Burst loss probability calculation window: $\tau = 50\ ms$
- Waxman topology parameters:
  - $\eta = 0.2$
  - $\delta = 0.15$

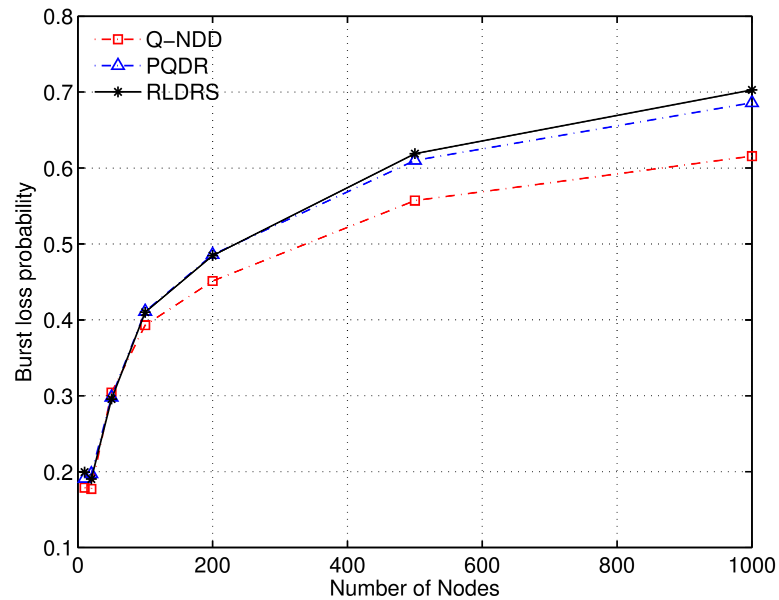
Burst Loss Probability

Comparison of Q-NDD, PQDR, and RLDRS:

- Waxman and Barabási-Albert topologies



Waxman



Barabási-Albert

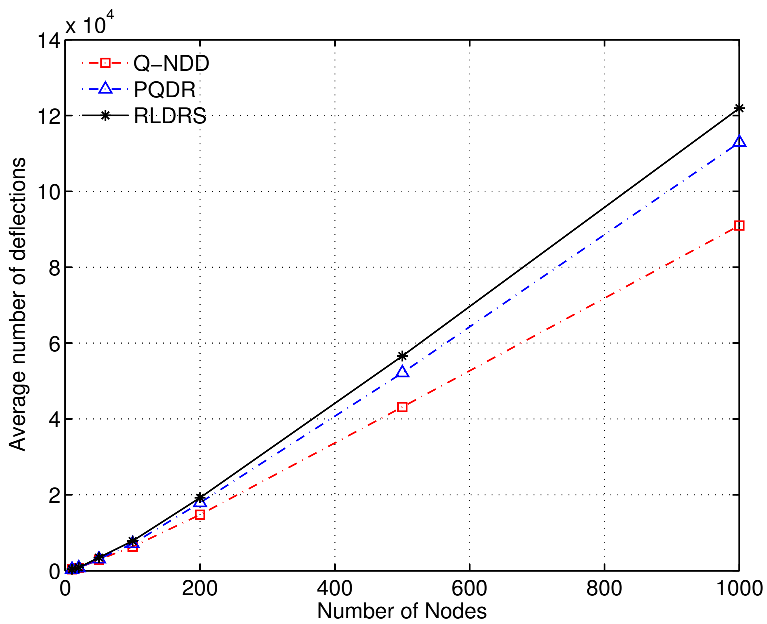
Burst Loss Probability

- Burst-loss probability has a logarithmic trend
- Slightly higher in Barabási-Albert networks
- Q-NDD scales better than PQDR and RLDRS as the size of the network grows

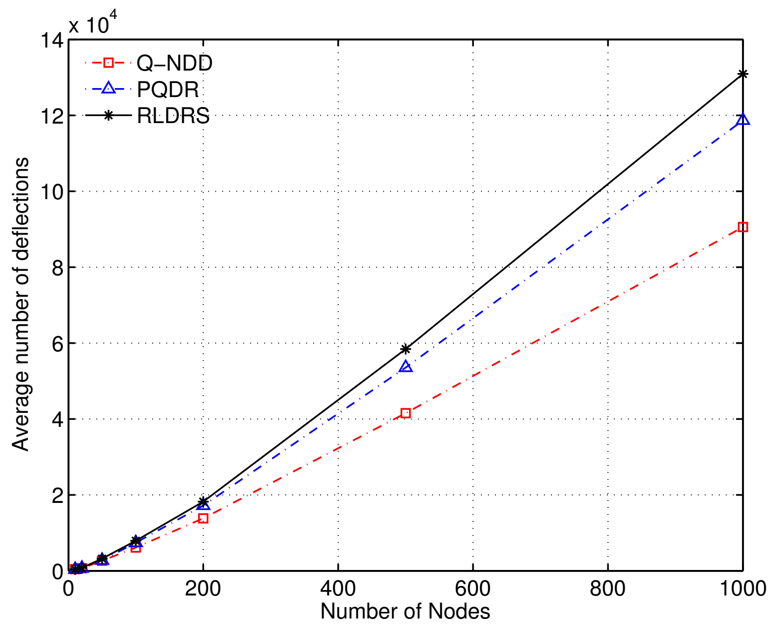
Average Number of Deflections

Comparison of Q-NDD, PQDR, and RLDRS:

- Waxman and Barabási-Albert topologies



Waxman



Barabási-Albert

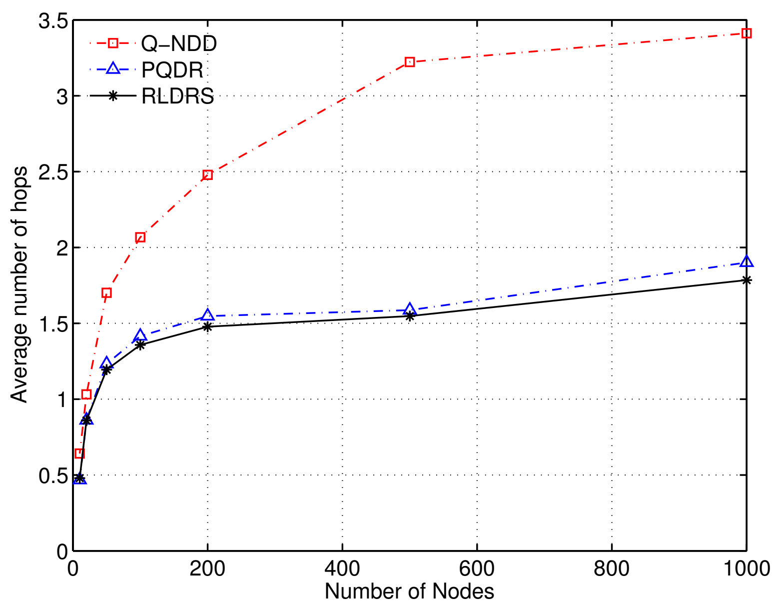
Average Number of Deflections

- Burst deflection reduces the burst-loss probability while introducing excess traffic load to the network
- Waxman and Barabási-Albert network topologies do not show a significant variation in terms of the number of deflections
- Q-NDD deflects fewer number of bursts than PQDR and RLDRS

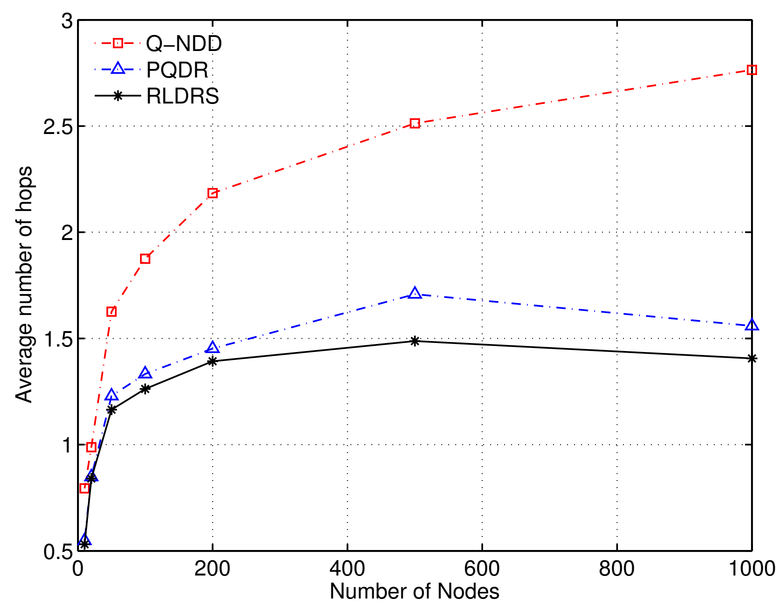
Average Number of Hops

Comparison of Q-NDD, PQDR, and RLDRS:

- Waxman and Barabási-Albert topologies



Waxman



Barabási-Albert

Average Number of Hops

- The underlying topology and nodes connectivity impact the number of hops traveled by bursts
- Bursts travel fewer hops in case of Barabási-Albert networks
- In case of Q-NDD, bursts travel through more hops than PQDR and RLDRS

Memory and CPU Usage

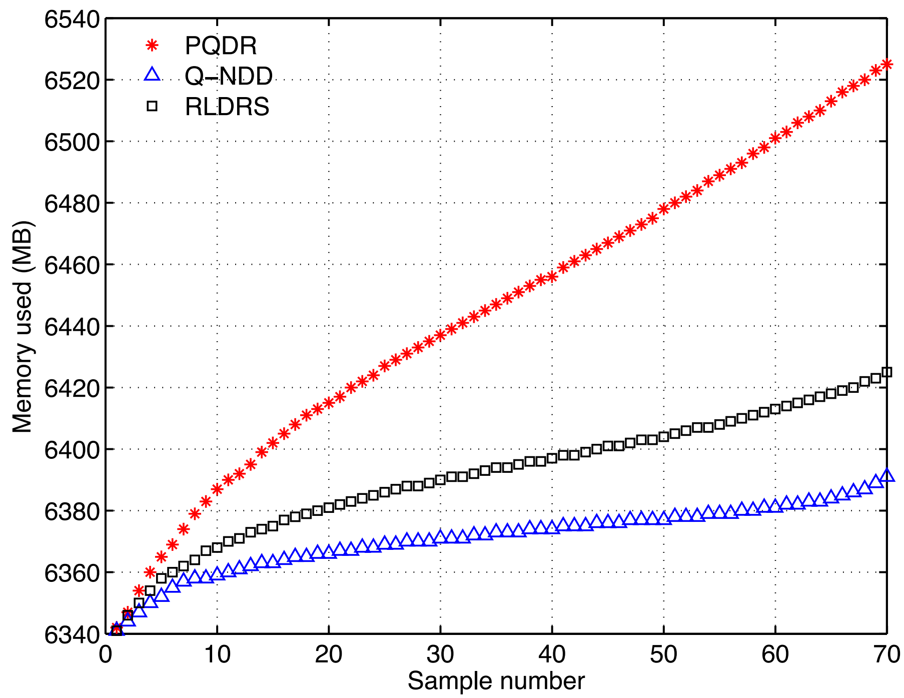
- Memory and CPU usage depends mostly on size of routing and deflection tables of the nodes
- They are functions of number of nodes
- Comparisons are made only for Waxman topologies with 500, 1,000, and 2,000 nodes

Waxman Topologies: Memory and CPU Usage

Algorithm	Number of nodes	Number of links	Number of flows	Min. memory usage (MB)	Max. memory usage (MB)	Total CPU time used (mm:ss)	Total simulation time (s)
PQDR	500	1,500	3,000	566	609	4:00.65	2,857.5
	1,000	3,000	6,000	1,727	1,817	19:09.44	6,613.1
	2,000	6,000	12,000	6,342	6,526	107:33.2	17,770.8
Q-NDD	500	1,500	3,000	561	578	1:25.61	1832.8
	1,000	3,000	6,000	1,727	1,754	16:44.46	4,872.8
	2,000	6,000	12,000	6,341	6,391	94:38.74	13,680.4
RLDRS	500	1,500	3,000	566	588	4:04.57	2,919.5
	1,000	3,000	6,000	1,727	1,769	18:36.04	6,633.7
	2,000	6,000	12,000	6,341	6,424	110:56.7	18,069.7

Waxman Topology: Memory Usage

- 2,000 nodes
- 70 sample points



Memory and CPU Usage

- The three algorithms initially have the same memory requirements
- The memory usage of PQDR grows faster compared to RLDRS and Q-NDD
  - PQDR stores five tables while RLDRS stores two and Q-NDD stores one table
  - Q-NDD requires the least memory space and CPU time
  - Q-NDD space complexity depends on the node degree rather than network size

Outline

- Introduction
- Deflection routing in buffer-less networks
- Reinforcement learning
- Network model and topology
- iDef framework
- Reinforcement learning for deflection routing
- Node degree dependent algorithm with Q-learning
- Predictive Q-learning deflection routing
- Simulation results
- Conclusions and references

Conclusions

- Network topology does not significantly affect number of deflections
- Barabási-Albert topologies:
  - bursts travel through fewer hops
- Q-NDD performs significantly better than PQDR and RLDRS in terms of burst-loss probability and memory usage
- Bursts travel through additional hops in the case of Q-NDD
- Q-NDD scales better with network size

References: Internet Topology

- M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 251–262, Aug. 1999.
- A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509-512, Oct. 1999.
- R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47-97, Jan. 2002.
- Lj. Trajković, "Analysis of Internet topologies," *IEEE Circuits and Systems Mag.*, vol. 10, no. 3, pp. 48–54, Third Quarter 2010.
- L. Subedi and Lj. Trajković, "Spectral analysis of Internet topology graphs," in *Proc. IEEE Int. Symp. Circuits and Systems*, Paris, France, June 2010, pp. 1803–1806.
- M. Najiminaini, L. Subedi, and Lj. Trajković, "Analysis of Internet topologies: a historical view," in *Proc. IEEE Int. Symp. Circuits and Systems*, Taipei, Taiwan, May 2009, pp. 1697–1700.
- B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.
- (2012, Sept.) BRITE topology generator. [Online]. Available: http://www.cs.bu.edu/brite.

References: Reinforcement Learning

- L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: a survey,” *J. of Artificial Intell. Research*, vol. 4, pp. 237–285, 1996.
- R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.
- C. J. C. H. Watkins and P. Dayan, “Technical note Q-learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

References: Reinforcement Learning for Routing

- L. Peshkin and V. Savova, “Reinforcement learning for adaptive routing,” in *Proc. Int. Joint Conf. Neural Netw.*, Honolulu, HI, USA, May 2002, vol. 2, pp. 1825–1830.
- A. Nowe, K. Steenhaut, M. Fakir, and K. Verbeeck, “Q-learning for adaptive load based routing,” in *Proc. IEEE Int. Conf. Syst., Man, and Cybern.*, San Diego, CA, USA, Oct. 1998, vol. 4, pp. 3965–3970.
- J. A. Boyan and M. L. Littman, “Packet routing in dynamically changing networks: a reinforcement learning approach,” in *Advances in Neural Inform. Process. Syst.*, vol. 6, pp. 671–678, 1994.
- S. P. M. Choi and D. Yeung, “Predictive Q-routing: a memory-based reinforcement learning approach to adaptive traffic control,” in *Advances in Neural Inform. Process. Syst.*, vol. 8, pp. 945–951, 1998.
- W. W.-K. Thong, G. Chen, and Lj. Trajković, “RED-f routing protocol for complex networks,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Seoul, Korea, May 2012, pp. 1644–1647.

References: Reinforcement Learning for Deflection Routing

- Y. Kiran, T. Venkatesh, and C. Murthy, “A reinforcement learning framework for path selection and wavelength selection in optical burst switched networks,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 18–26, Dec. 2007.
- A. Belbekkouche, A. Hafid, and M. Gendreau, “Novel reinforcement learning-based approaches to reduce loss probability in buffer-less OBS networks,” *Comput. Netw.*, vol. 53, no. 12, pp. 2091–2105, Aug. 2009.
- S. Haeri, W. W-K. Thong, G. Chen, and Lj. Trajković, “A reinforcement learning-based algorithm for deflection routing in optical burst-switched networks,” in *Proc. IEEE Int. Conf. Inf. Reuse and Integration*, San Francisco, USA, Aug. 2013, pp. 474–481.
- S. Haeri, M. Arianezhad, and Lj. Trajković, “A predictive Q-learning based algorithm for deflection routing in buffer-less networks,” in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics (SMC 2013)*, Manchester, UK, Oct. 2013, pp. 764–769.

References: Optical Burst Switching

- C. Qiao and M. Yoo, "Optical burst switching (OBS)—a new paradigm for an optical Internet," *J. of High Speed Netw.*, vol. 8, no. 1, pp. 69–84, Mar. 1999.
- A. Zalesky, H. Vu, Z. Rosberg, E. W. M. Wong, and M. Zukerman, "Modelling and performance evaluation of optical burst switched networks with deflection routing and wavelength reservation," in *Proc. INFOCOM*, Hong Kong SAR, China, Mar. 2004, vol. 3, pp. 1864–1871.
- X. Yu, J. Li, X. Cao, Y. Chen, and C. Qiao, "Traffic statistics and performance evaluation in optical burst switched networks," *J. Lightw. Technol.*, vol. 22, no. 12, pp. 2722–2738, Dec. 2004.