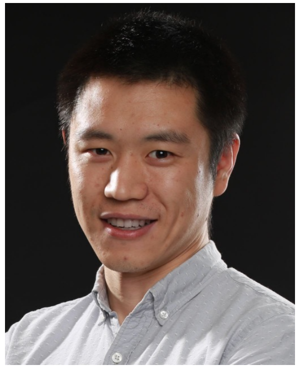
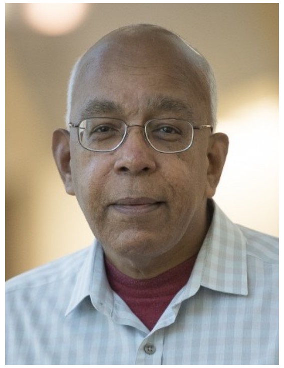
Paper ID: 3944

Delay-aware backpressure routing using Graph Neural Networks




Zhongyuan Zhao*, Bojan Radojičić[§], Gunjan Verma[†], Ananthram Swami[†], Santiago Segarra*

*Rice University, USA

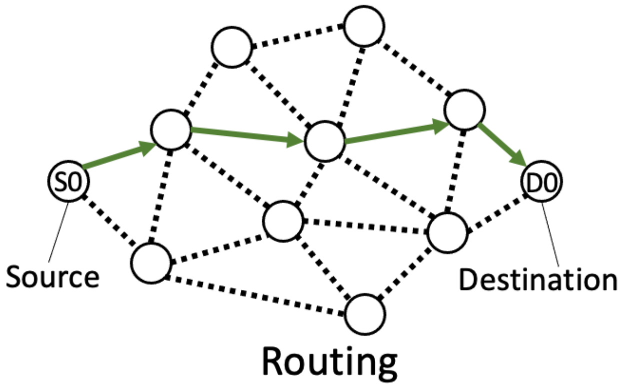
§University of Novi Sad, Serbia

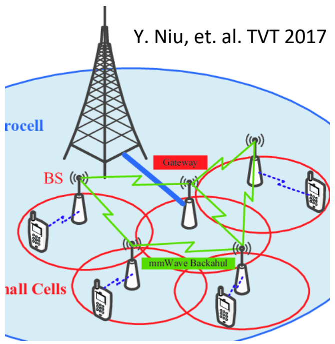
† US Army's DEVCOM Army Research Laboratory, USA

2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Rhodes Island, Greece, June 4-10, 2023

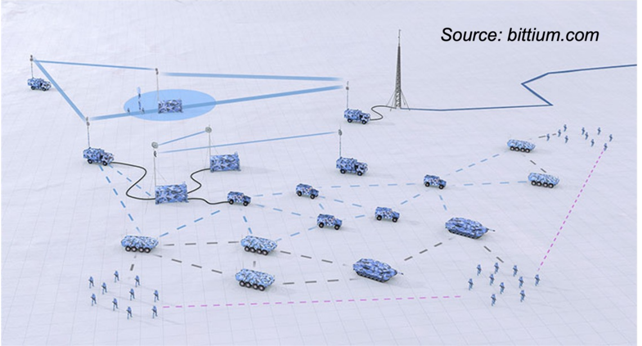
Backpressure (BP) Routing for Multihop Wireless Networks



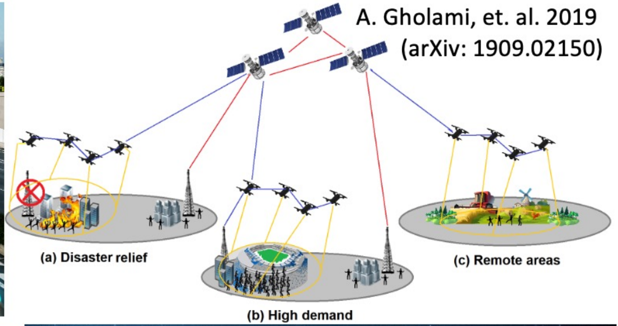


Y. Niu, et. al. TVT 2017

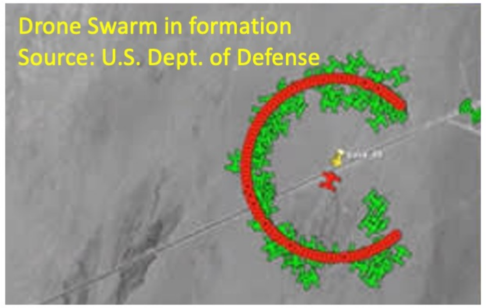
- Wireless Ad-hoc Networks
  - Military
  - Disaster relief
- Wireless Sensor Networks



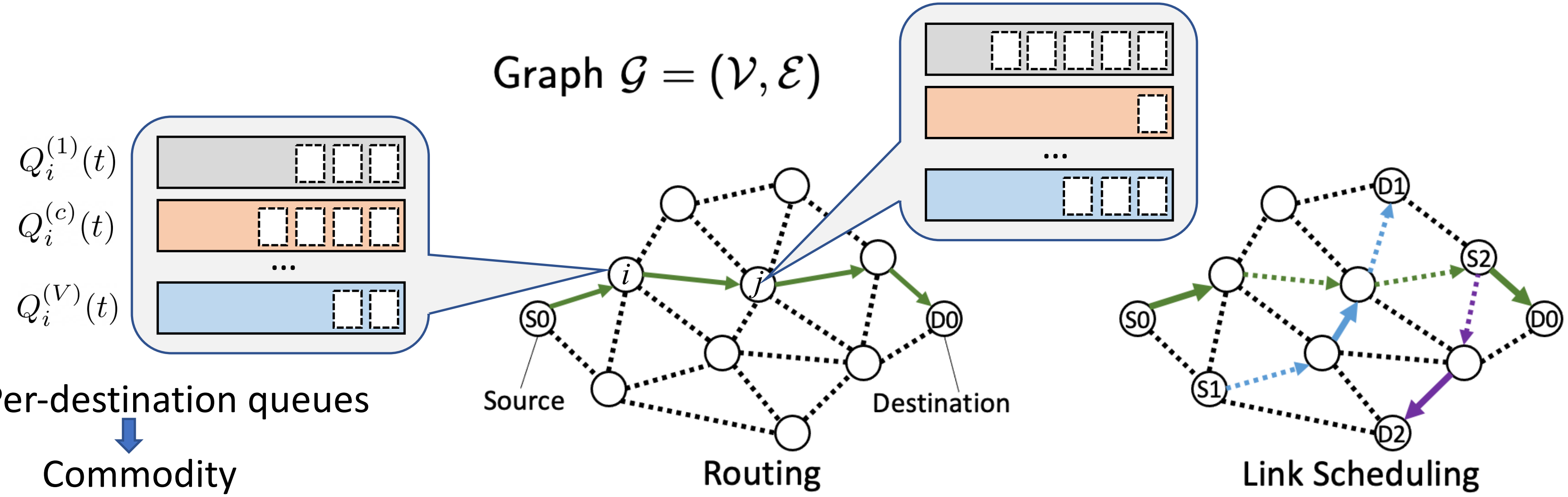
- Wireless Backhaul Networks
  - Small cell backhaul
  - Drone/CubSat-assisted 5G/6G
  - Starlink
  - Rural/Agriculture broadband
- Machine-to-Machine Comm.
  - Internet-of-Things (IoT)
  - Connected vehicles
  - Drone fleet / Robotic Swarm
  - Smart factory



A. Gholami, et. al. 2019 (arXiv: 1909.02150)



Backpressure Routing*



Source: zhongyuanzhao.com

- ✓Distributed Routing
- ✓Congestion prevention
- ✓Throughput optimality
- ✗Poor latency performance
  - ✗Slow start
  - ✗Loop
  - ✗Last-packet problem

1. Select optimal commodity $c_{ij}^*(t) = \underset{c \in \mathcal{V}}{\operatorname{argmax}}\{U_i^{(c)}(t) - U_j^{(c)}(t)\}$

$$U_i^{(c)}(t) = Q_i^{(c)}(t)$$ Vanilla BP

$$U_i^{(c)}(t) = Q_i^{(c)}(t) + B_i^{(c)}$$ Biased BP

Shortest path-aware bias

2. Find link gradient $w_{ij}(t) = \max\{U_i^{(c_{ij}^*(t))}(t) - U_j^{(c_{ij}^*(t))}(t), 0\}$

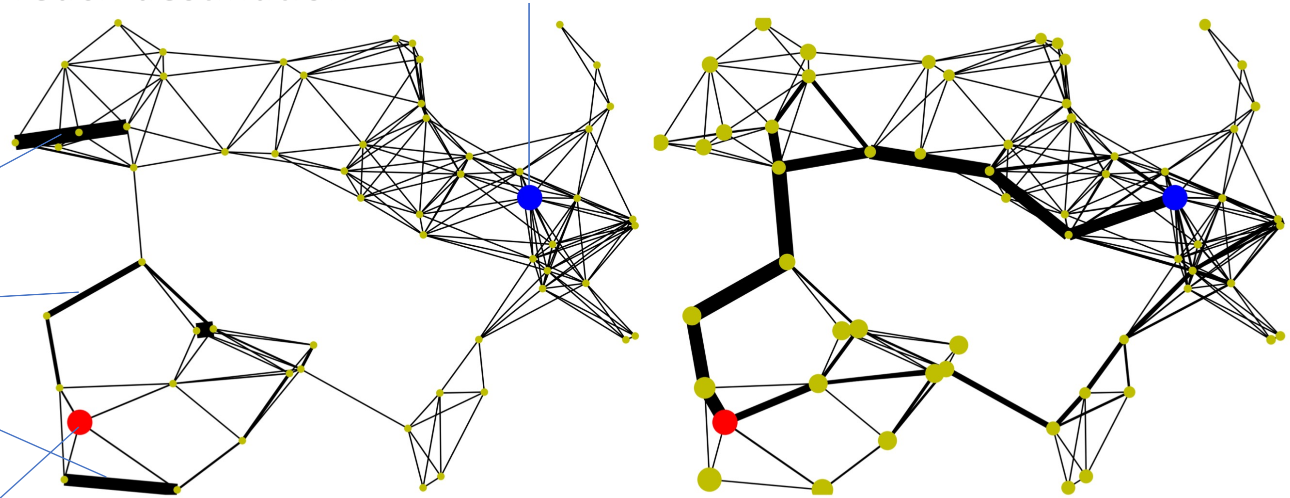
3. MaxWeight scheduling $\mathbf{l}^{BP}(t) = \underset{\mathbf{l}(t) \in \{0,1\}^{|\mathcal{E}|}}{\operatorname{argmax}} \mathbf{l}(t)^\top \cdot [\mathbf{r}(t) \odot \mathbf{w}(t)]$

4. Assign link capacity $\mu_{ij}^{(c)}(t) = \begin{cases} r_{ij}(t), & \text{if } c = c_{ij}^*, w_{ij} > 0 \\ 0, & \text{otherwise} \end{cases}$

* L. Tassiulas, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," IEEE Trans. on Automatic Control, vol. 31, no. 12, 1992.

Vanilla v.s. biased BP routing

- ✓Distributed Routing
- ✓Path finding
- ✓Throughput optimality
- ✓~~Poor~~ latency performance
  - ✓ ~~Slow start~~
  - ✓ ~~Loop~~
  - ✓ ~~Last packet problem~~



(a) (b)

Vanilla BP
Congestion gradient

Biased BP: shortest hop distance*
Congestion gradient + hop distance gradient

Route visualization: Normalized number of packets over links in 500 steps

Can we do better than shortest hop distance bias?

* M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," IEEE J. Sel. Areas Commun., vol. 23, no. 1, pp. 89–103, 2005

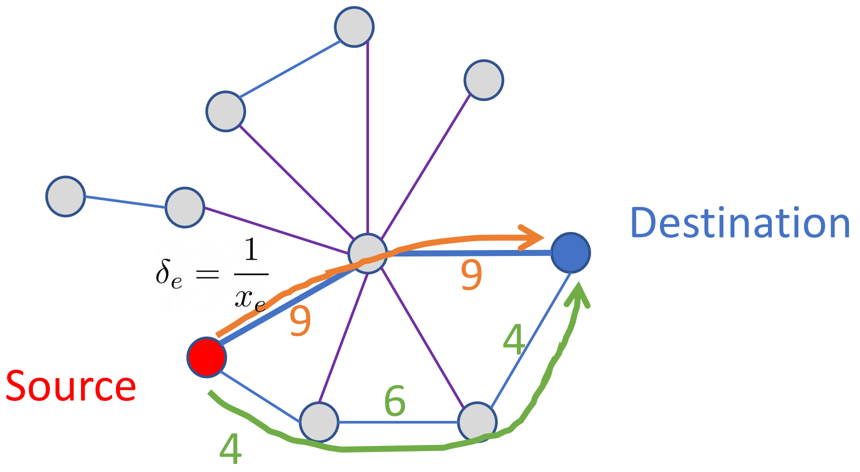
Delay-aware shortest path bias based on link duty cycle

How likely a link is scheduled under current network topology and traffics

Link duty cycle $$0 < x_e \leq 1 \qquad e \in \mathcal{E}$$

Per hop distance $$\delta_e = \frac{1}{x_e}$$

Per hop distance with link rate $$\delta_e = \frac{\bar{r}}{x_e r_e}$$



Link duty cycle estimated by an L-layer graph convolutional neural network (GCNN) $$\mathbf{x} = \Psi_{\mathcal{G}^c}(\mathbf{1}; \boldsymbol{\omega})$$

Fully distributed execution

$$\mathbf{X}^l_{e*} = \sigma_l \left(\mathbf{X}^{l-1}_{e*} \mathbf{\Theta}^l_0 + \left[\mathbf{X}^{l-1}_{e*} - \sum_{u \in \mathcal{N}_{\mathcal{G}^c}(e)} \frac{\mathbf{X}^{l-1}_{u*}}{\sqrt{d(e)d(u)}} \right] \mathbf{\Theta}^l_1 \right)$$

Properties of delay-aware shortest path bias

- Complexity
  - GCNN $\mathcal{O}(L)$
  - Single source shortest path (SSSP)
  - All pairs shortest path (APSP)

Distributed weighted SSSP and APSP

$$\mathcal{O}(V)$$

GCNN and SP algorithms only need to run once a while, when topology changes

- Throughput Optimality
  - Shortest path bias is non-negative and constant
  - Throughput optimality holds

BP algorithm can stabilize the queues in the network as long as the arrival rates of flows are within the network capacity region

Training of GCNN

- Draw a network instance
- Find delay-aware shortest path bias with GCNN and APSP
- Run BP routing and collect schedules of each time slot
- Update parameters of GCNN with loss function

Connectivity graph, Conflict graph, Flows, Flow arrivals, Link rates

$$(\mathcal{G}^n(k), \mathcal{G}^c(k), \mathcal{F}(k), \mathbf{A}(k), \mathbf{R}(k)) \sim \Omega.$$

Per-link distance $\mathbf{x}(k) = \Psi_{\mathcal{G}^c(k)}(\mathbf{1};\boldsymbol{\omega}),$

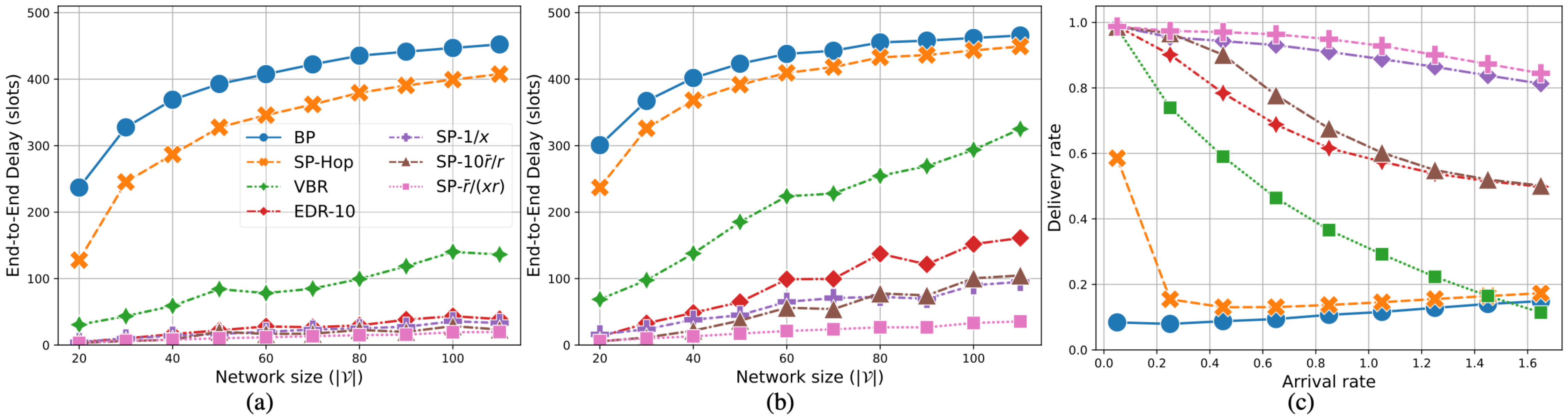
Shortest path bias $\mathcal{B}(k)$

Empirical schedule $\mathbf{s}^k(t) \in \{0,1\}^{|\mathcal{E}|}$

$$\ell(\boldsymbol{\omega}) = \mathbb{E}_{\Omega}\left[MSE(\mathbf{x}(k), \mathbb{E}_t(\mathbf{s}^k(t)))\right]$$

Performance

On 100 random graphs from **2D point process model**
T=1000



Interface conflict model, e.g., mmWave networks

Unit-disk conflict model, e.g., wireless sensor/ad-hoc networks

Delivery rate under unit-disk conflict model with 100 nodes

Conclusion & Future directions

- Delay-aware per-hop distance for biased backpressure routing
  - Link duty cycle
  - Conflict-aware (adaptive to network density)
  - Significantly improve end-to-end delay & delivery rate
- Keep advantages of shortest hop distance bias
  - Fully distributed execution
  - Minimal increase in complexity
  - Simplicity
  - Low overhead (update only once a while)
- Apply to other routing schemes
- Improved training method