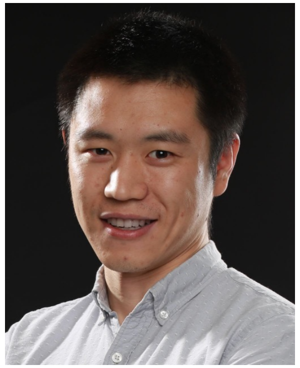
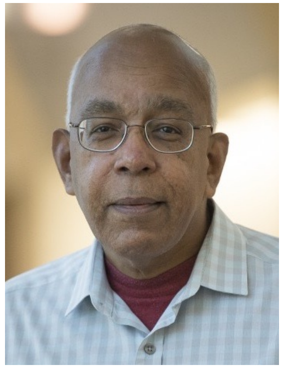
Distributed Link Sparsification for Scalable Scheduling using Graph Neural Networks


Zhongyuan Zhao*, *Ananthram Swami* †, *Santiago Segarra**

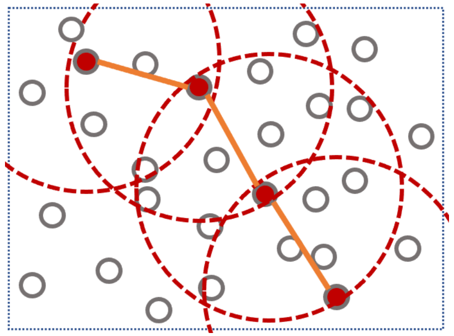
*Rice University, , USA

† US Army's DEVCOM Army Research Laboratory, USA

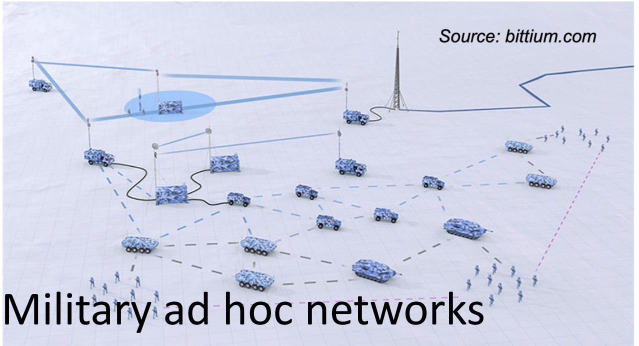
IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Singapore, 22-27 May 2022

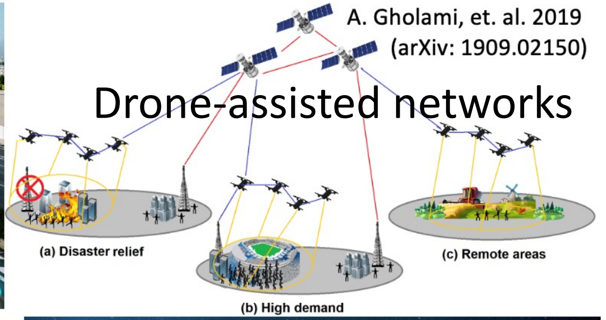
Wireless Multihop Networks

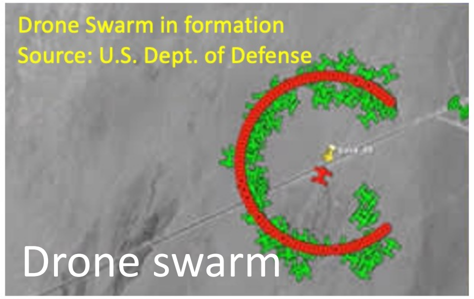


- Autonomous, self-organizing networks
- *No base-stations!*
- Wide applications
  - Mobile Ad-Hoc Networks
  - Internet-of-Things, wireless sensor networks
  - Wireless backhaul (drone + satellites, small cells)
  - Traffic offloading in 5G and beyond (D2D)



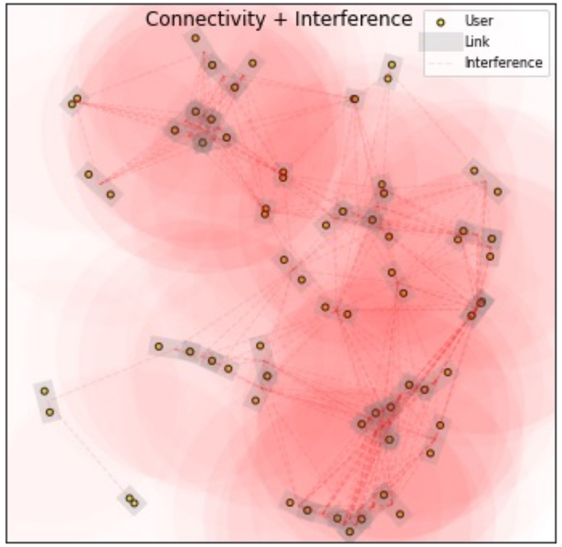
Internet of Things

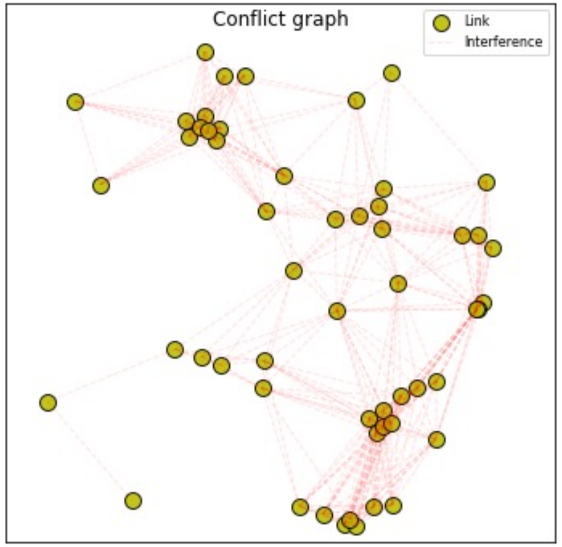


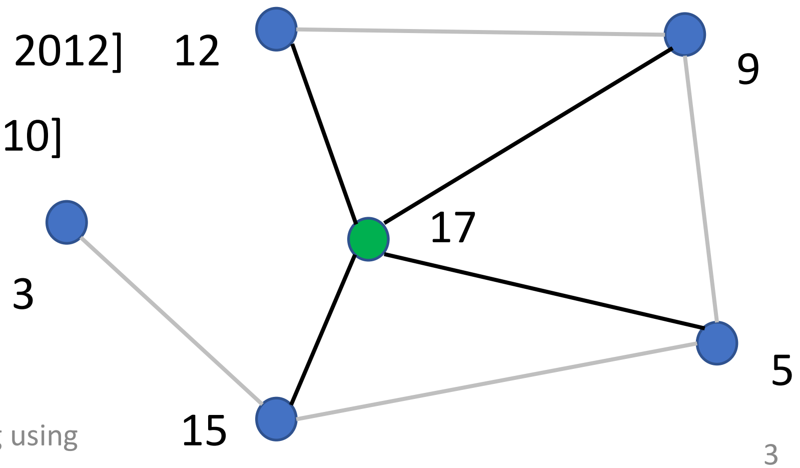


Distributed Scheduling Basics

- Conflict graph
  - Vertices → links
  - Edges → Conflict (interface and interference) between links
- Orthogonal access
- Contention-based scheduling
  - Distributed greedy scheduler (synchronized) [Joo 2012]
  - Weighted CSMA (random access) [Ni 2010, Jiang 2010]
- Scheduling overhead increases by neighborhood size



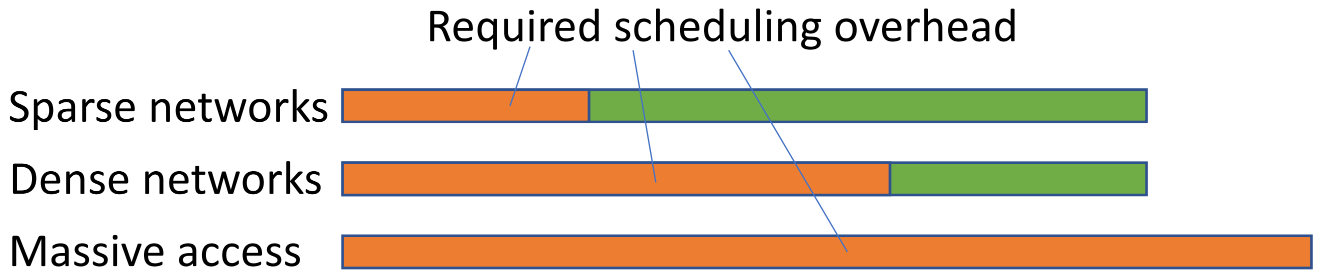


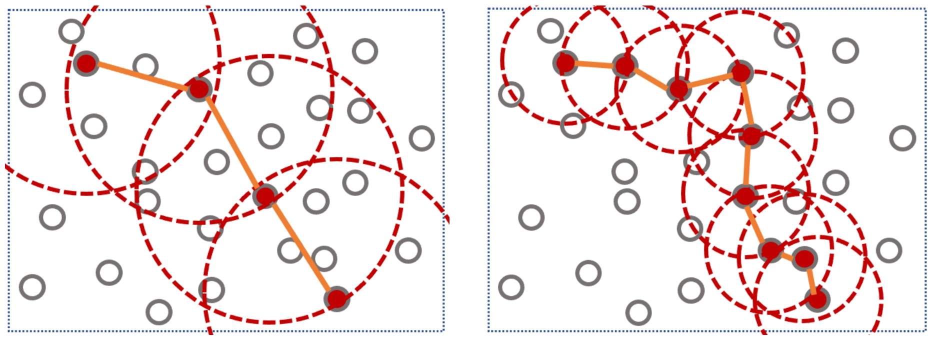


Scheduling Overhead v.s. Network Density

Scheduling overhead increases by size of interfering neighborhood!

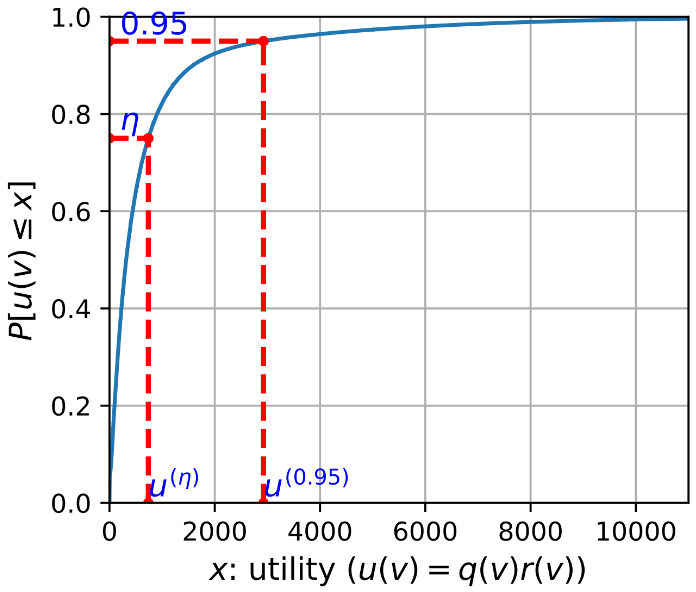
- Massive access 🚫
  - 10 million connections per km^2
- Network Capacity ↓
- Battery life ↓
- Radio footprint ↑ *(Interference, security vulnerability due to topology leakage)*





- Existing solutions
  - Topology control [Santi 2005, Ramanathan 2004, Ray 2016]
  - Sleep scheduling [Ye 2004, Ray 2016, Guha 2011, Long 2020]
  - Cross-layer optimization [Lin 2010, Xiang 2014, Wu 2020]
  - Traffic-based link sparsification

Traffic-based Link Sparsification



- Only the most demanding links contend for scheduling
  - Demand → per-link utility
- Prior knowledge
  - eCDF of Per-link utility
  - Cut-off quantile η
- Statistical thresholding
  - Longer backlog (Queue)
  - Network capacity loss

Conflict density → 5% $\eta = 0.95$

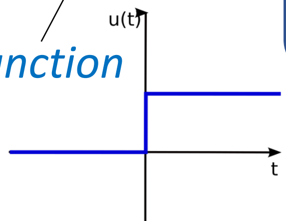
Can we do better than statistical thresholding?

Global cut off threshold

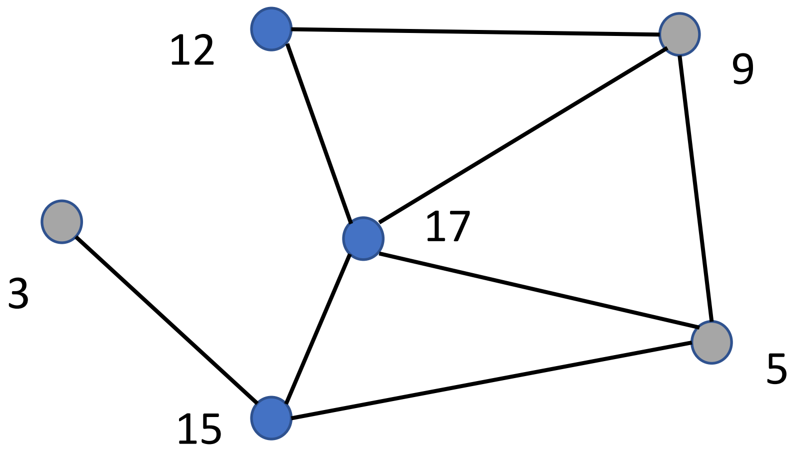
$$h_v(u(v)) = u(v)\, H\left(u(v) - u^{(\eta)}\right)$$

Per-link utility

Heaviside step function



If $u(v) \leq u^{(\eta)}$, then $h_v(u(v)) = 0$
Otherwise, $h_v(u(v)) = u(v).$



$$u^{(\eta)} = 10$$

Topology-aware Link Sparsification

1. Localized functions

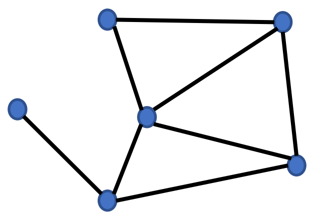
Local threshold

$$h_v(u(v);\mathbf{Z}) = \underline{z_0(v)}u(v)\, H\left(\underline{z_0(v)}u(v) - \underline{z_1(v)}u^{(\eta)}\right)$$

link parameters

$$\mathbf{Z} = [\mathbf{z}_0, \mathbf{z}_1] \in \mathbb{R}^{|\mathcal{V}|\times 2}$$

2. Generalize to different topologies $\mathbf{Z} = \Psi_{\mathcal{G}}(\mathbf{1};\omega)$

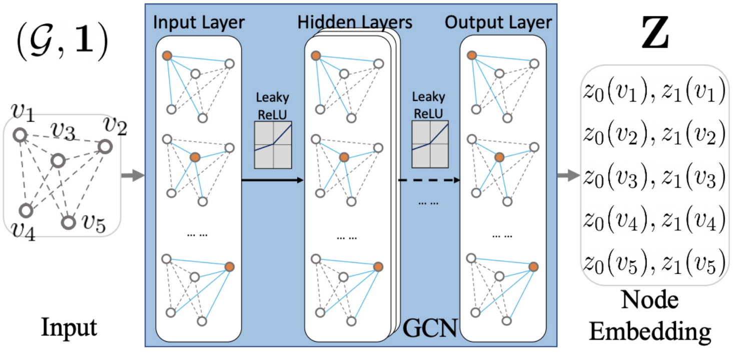


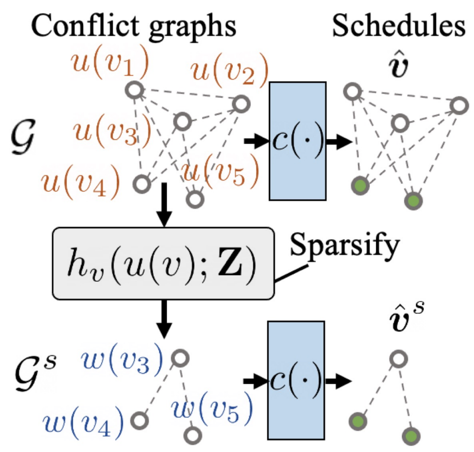
Statistical baseline
(global threshold)

$$h_v(u(v)) = u(v)\, H\left(u(v) - u^{(\eta)}\right)$$

If $u(v) \leq u^{(\eta)}$, then $h_v(u(v)) = 0$

Otherwise, $h_v(u(v)) = u(v)$.





Dense scheduler

Sparse scheduler

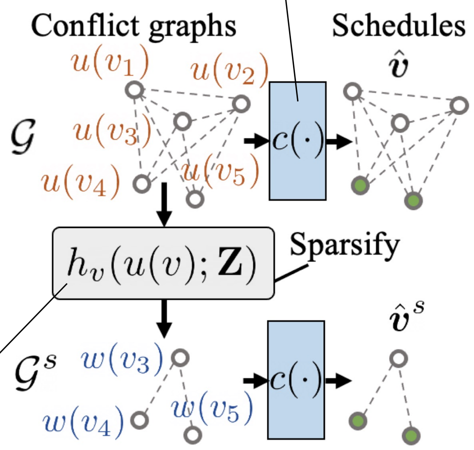
1. GCN can be implemented in a **distributed** manner
2. GCN only runs once a while (until topology changes)

Non-differentiable downstream pipeline

Scheduling contention is a **non-differentiable** discrete function (set operations)

Objective function defined on the output of non-differentiable functions

$$\mathbf{Z} = \Psi_{\mathcal{G}}(\mathbf{1}; \omega)$$



Dense scheduler

Sparse scheduler

Problem 1. *Given a distribution $\mathcal{N}$ over network states $(\mathcal{G}, \mathbf{u})$, we want to obtain the optimal link sparsification functions $\{h_v^*\}$ for all $v \in \mathcal{V}$ as*

$$\{h_v^*\} = \underset{\{h_v\}}{\operatorname{argmax}} \; \mathbb{E}_{\mathcal{N}}\left(u(\hat{\boldsymbol{v}}^s) - \alpha|\mathcal{E}^s|\right) \tag{1a}$$

$$\text{s.t.} \quad \mathcal{G}^s = \mathcal{G} \setminus \{v | v \in \mathcal{V}, h_v(u(v)) \leq 0\}, \tag{1b}$$

$$\mathbf{w}^s = [h_v(u(v))], \text{ for all } v \in \mathcal{V}^s, \tag{1c}$$

$$\hat{\boldsymbol{v}}^s = c(\mathcal{G}^s, \mathbf{w}^s). \tag{1d}$$

Link sparsification is a **non-differentiable** discrete function (step function)

Two-stage customized reinforcement learning

- Stage 1: train topological selectivity $\uparrow \mathbb{E}_{\mathcal{N}}\left(u(\hat{\boldsymbol{v}}^{s,gcn})\right)$
  - Get higher utility with less nodes
- Stage 2: normalize link parameters $\downarrow \mathbb{E}_{\mathcal{N}}\left(|\mathcal{E}^{s}|\right)$

$$\mathbb{E}_{\mathcal{G}}\left(u(\hat{\boldsymbol{v}}^{s,gcn})\right) \to \mathbb{E}_{\mathcal{G}}\left(u(\hat{\boldsymbol{v}}^{s,stat})\right) \quad \mathbb{E}_{\mathcal{G}}\left(z_1(v)u^{(\eta)}\right) \to u^{(\eta)}$$

$$h_v(u(v);\mathbf{Z}) = z_0(v)u(v)\, H\left(z_0(v)u(v) - z_1(v)u^{(\eta)}\right)$$

Experience tuples $(\mathcal{G},\mathbf{u}) \in \mathcal{N}$

$$(\mathcal{G}(i),\mathbf{u}(i),\hat{\boldsymbol{v}}^{s}(i),\boldsymbol{v}^{r}(i),\boldsymbol{\rho}_0(i),\boldsymbol{\rho}_1(i)),\ \text{for } i \in \{0,\dots,N\}.$$

Target vector for $\mathbf{z}_0$

Reward scheduled links with relative total utility

$$\boldsymbol{\rho}_0 = \varepsilon\mathbf{v}^{s} + \mathbf{z}_0 \odot (\mathbf{1}-\mathbf{v}^{s}),\ \varepsilon = u(\hat{\boldsymbol{v}}^{s})/u(\hat{\boldsymbol{v}}^{b})$$

Target vector for $\mathbf{z}_1$

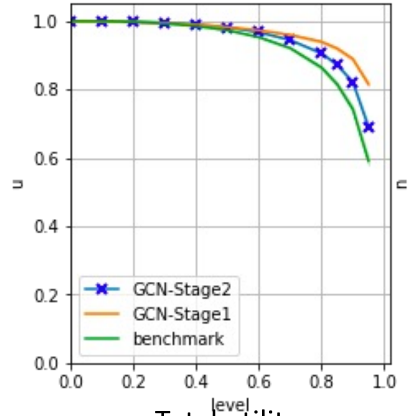
Increase threshold on removed links if utility is larger than baseline, vise versa.

$$\boldsymbol{\rho}_1 = \boldsymbol{\rho}_2,\ \boldsymbol{\rho}_2 = (b/u^{(\eta)})\mathbf{z}_0 \odot \mathbf{u} \odot \mathbf{v}^{r} + \mathbf{z}_1 \odot (\mathbf{1}-\mathbf{v}^{r}), \tag{5a}$$

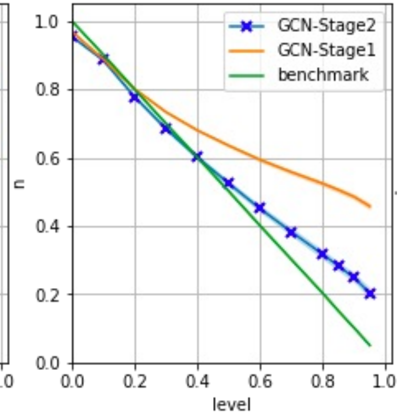
$$\boldsymbol{\rho}_1 = \boldsymbol{\rho}_3/\overline{\boldsymbol{\rho}_3},\ \boldsymbol{\rho}_3 = \boldsymbol{\rho}_2 - 0.2\mathbf{z}_1 \odot \mathbf{v}^{s}, \tag{5b}$$

$$\delta = 0.97 \qquad b = 0.9\mathbb{1}(\varepsilon < \delta) + 1.1\mathbb{1}(\varepsilon \geq \delta).$$

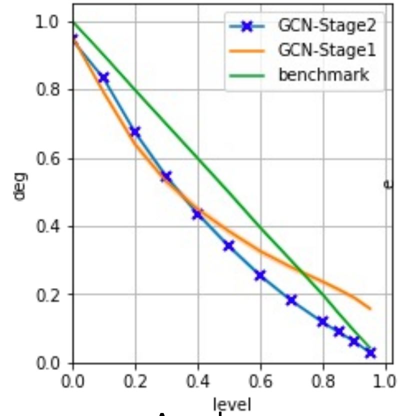
RMS loss function $\ell(\boldsymbol{\omega};\mathcal{G}(i),\mathbf{u}(i)) = |\mathcal{V}|^{-\frac{1}{2}}\|\mathbf{Z}(i) - [\boldsymbol{\rho}_0(i),\boldsymbol{\rho}_1(i)]\|_2.$



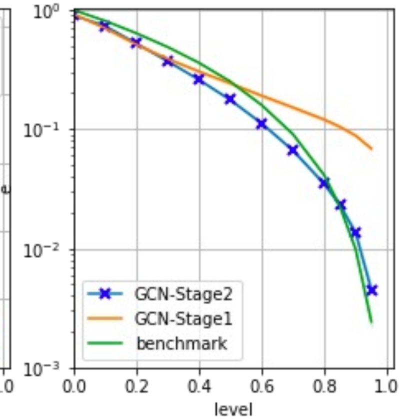
Total utility



Graph size



Avg degree (collision rate in weighted CSMA)

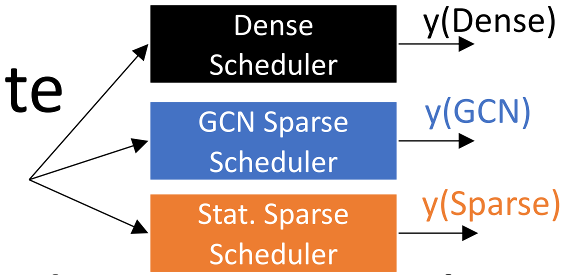


P2P messages in scheduling contention

Performance on identical input network states

- Input network state

  $(\mathcal{G}, \mathbf{u}) \in \mathcal{N}$



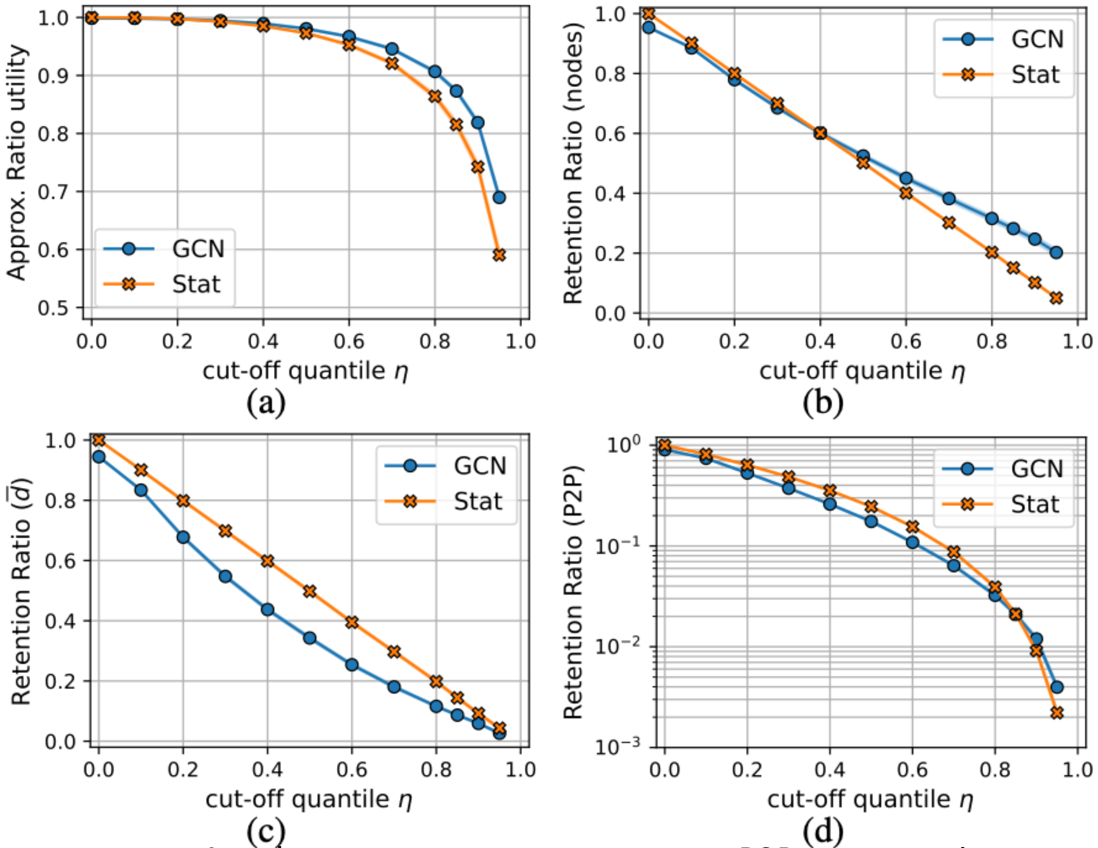
- Results relative to dense scheduler
  - Approx. ratio
  - Retention ratio

$$\frac{\text{y(Sparse)}}{\text{y(Dense)}}$$

- GCN-based link Sparsification
  - Higher total utility
  - Lower average node degree
  - Lower P2P message complexity

Total utility

Graph size



Avg degree (collision rate in weighted CSMA)

P2P messages in Local greedy scheduler

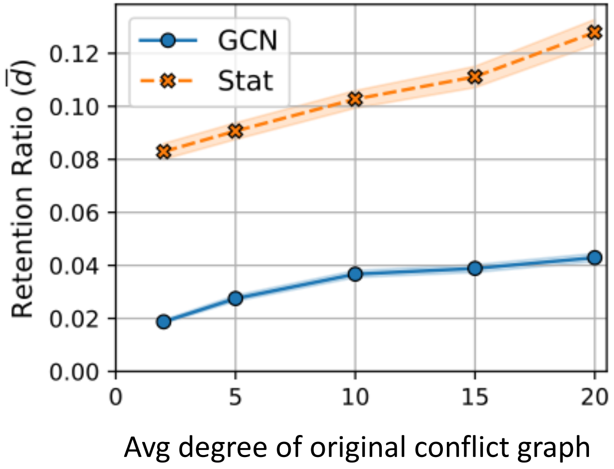
Performance on scheduling

$$(\mathcal{G}(t), \mathbf{u}(t))$$

$$u_v(t) = q_v(t) r_v(t)$$

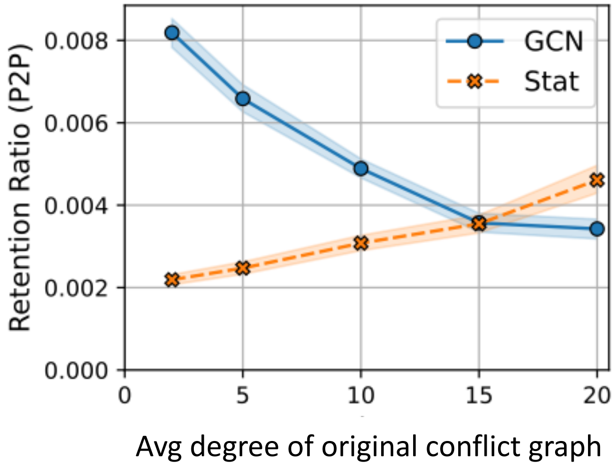
$$q_v(t+1) = \begin{cases} q_v(t) + a_v(t) & \text{if } v \notin \hat{\boldsymbol{v}}(t), \\ q_v(t) + a_v(t) - \min(r_v(t), q_v(t)) & \text{if } v \in \hat{\boldsymbol{v}}(t), \end{cases}$$

- ER conflict graphs
  - Random Point process
  - Uniform transmit power
  - 100~300 links per graph
  - 500 conflict graphs
- Moderate traffic load
  - Packet arrivals: Poisson
  - link rates: normal distribution
  - 300 time slots
  - Same realization of **arrivals** and **link rates** fed to different schedulers
- Cut-off quantile: 0.95



Average degree of sparsified conflict graph

1/3 ~1/5 of statistical baseline



Point-2-point message complexity with local greedy scheduler

Decrease by network density

Conclusion & future work

- Optimize link parameters with GCN
  - Parameterized link sparsification functions reduces scheduling overhead
  - Low computational and communication overhead
- Magic of reinforcement learning
  - No labels or ground truth
  - A good baseline → a better algorithm
- Principled learning method for non-differentiable pipeline [work in progress]
- GNN to optimize local parameters in a network

References

- [Ni 2010] J. Ni, B. Tan, and R. Srikant, “Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks,” in IEEE Intl. Conf. on Computer Comms. (INFOCOM), pp. 1–5, 2010.
- [Joo 2012] C. Joo and N. B. Shroff, “Local greedy approximation for scheduling in multihop wireless networks,” IEEE Trans. on Mobile Computing, vol. 11, no. 3, pp. 414–426, 2012.
- [Zhao 2021] Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, “Distributed scheduling using graph neural networks,” in IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP), pp. 4720–4724, 2021.
- [Santi 2005] P. Santi, “Topology control in wireless ad hoc and sensor networks,” ACM Comput. Surv., vol. 37, p. 164–194, June 2005.
- [Jiang 2010] L. Jiang and J. Walrand, “A distributed CSMA algorithm for throughput and utility maximization in wireless networks,” IEEE/ACM Trans. Netw., vol. 18, no. 3, pp. 960–972, 2010.
- [Ramanathan 2004] R. Ramanathan, “Antenna beamforming and power control for ad hoc networks,” in Mobile Ad Hoc Networking (S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, eds.), pp. 139–173, Hoboken, NJ, USA:John Wiley & Sons, Ltd, 2004.
- [Ray 2021] N. K. Ray and A. K. Turuk, “A hybrid energy efficient protocol for mobile ad hoc networks,” Journal of computer netw. and commun., vol. 2016, 2016.[Lee 2021] M. Lee, G. Yu, and G. Y. Li, “Graph embedding-based wireless link scheduling with few training samples,” IEEE Trans. Wireless Commun., vol. 20, no. 4, pp. 2282–2294, 2021.
- [Ye 2004] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” IEEE/ACM Trans. Netw., vol. 12, no. 3, pp. 493–506, 2004.
- [Guha 2011] S. Guha, P. B. Basu, C.-K. C. Chau, and R. Gibbens, “Green wave sleep scheduling: Optimizing latency and throughput in duty cycling wireless networks,” IEEE J. Sel. Areas Commun., vol. 29, no. 8, pp. 1595–1604, 2011.
- [Jong 2020] J. Long and O. Büyüköztürk, “Collaborative duty cycling strategies in energy harvesting sensor networks,” Comput.-Aided Civ. Infrastruct. Eng., vol. 35, p. 534–548, May 2020.
- [Lin 2010] L. Lin, X. Lin, and N. B. Shroff, “Low-complexity and distributed energy minimization in multihop wireless networks,” IEEE/ACM Trans. Netw., vol. 18, no. 2, pp. 501–514, 2010.
- [Xiang 2014] X. Xiang, C. Lin, and X. Chen, “Energy-efficient link selection and transmission scheduling in mobile cloud computing,” IEEE Commun. Lett., vol. 3, no. 2, pp. 153–156, 2014.
- [Wu 2020] W. Wu, W. Xu, Z. Chen, and M. Yang, “Energy-efficient link scheduling in time-variant dual-hop 60GHz wireless networks,” Concurrency and Computation: Practice and Experience, vol. 32, no. 23, p. e5903, 2020.


Q&A