BUPT Talk

Deep Learning and Control for Ultra Reliable Low Latency Communications

Walid Saad

Electrical and Computer Engineering Department,
Network sciEnce, Wireless, and Security (NEWS) Group

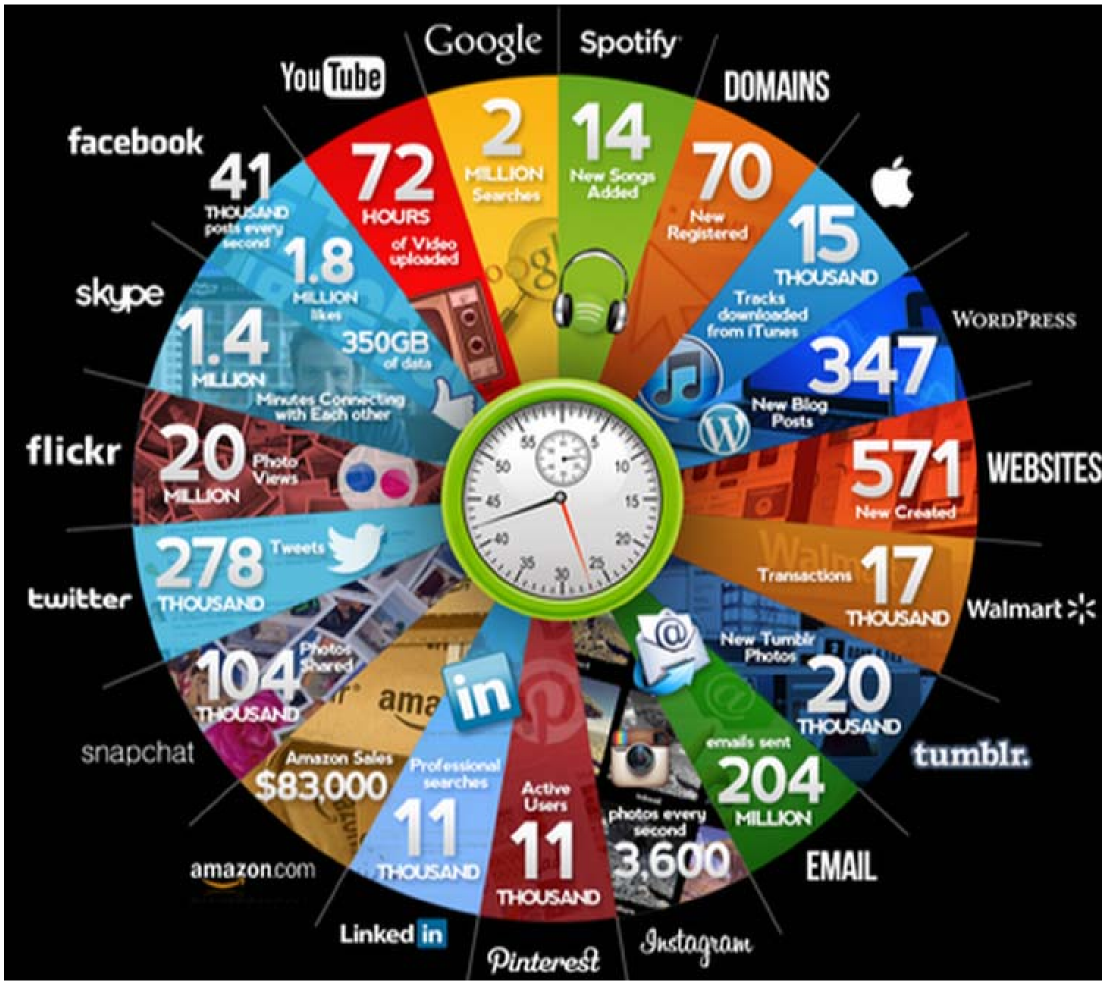
Virginia Tech

Email: walids@vt.edu
Group: http://www.netsciwis.com
Personal: http://resume.walid-saad.com

Outline

- Why URLLC?
  - Towards supporting autonomous systems
  - Service requirements
  - Whats new in this area?
- Experienced Deep Learning for Model-Free URLLC
  - System model and problem formualtion
  - Key results
- Integrated Control and Communications
  - System model and problem formualtion
  - Key results
- Overview on other research

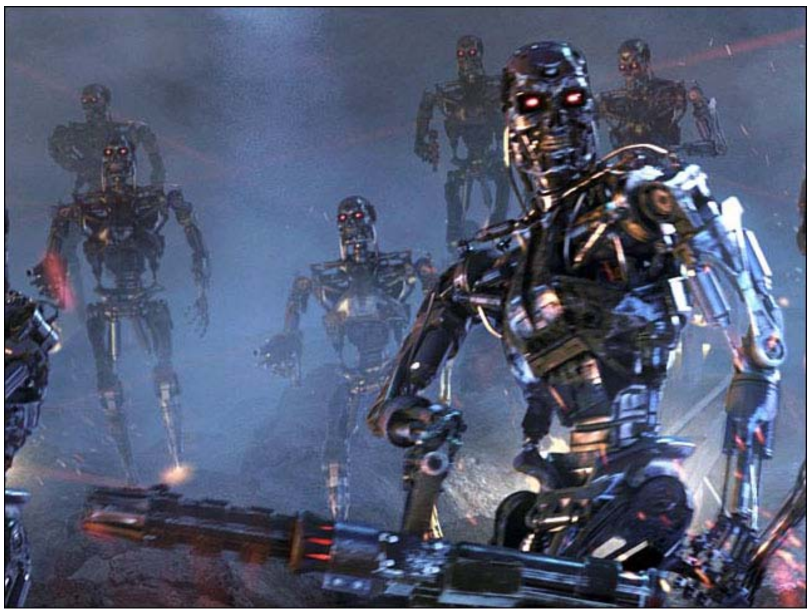
What happens online in 60 seconds?



- www.rethink-wireless.com (January 2012)
  - "[…] iPhones caused disastrous performance problems on AT&T networks"
  - "Verizon Wireless blamed similar factors for several outages on its LTE network last year […]"
  - "DoCoMo demands Google's help with IP Signalling Storm"

Autonomous Machines

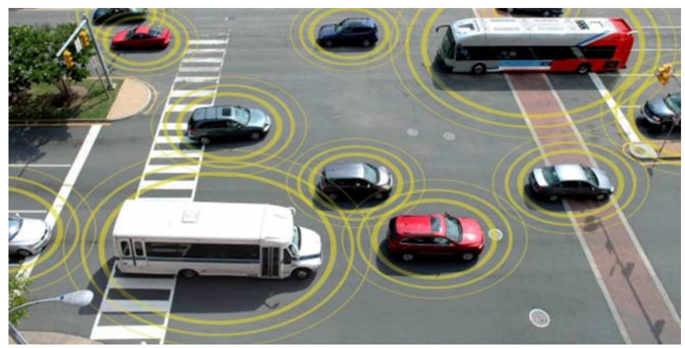
- All these services sought to connect ***people***
- Next decade….
  - We need to connect autonomous ***machines….***



  - ***…..for the well-being of people***

What happens in an autonomous system/smart city?

- This was just the iPhone's impact, what about a full smart city?



- Massive number of devices with services that are more **critical** than *Facebook/Instagram posts!*

Communication Requirements

- How to effectively connect all those infrastructure and components of a city?
  - Communication should be *high data rate* – **terrabytes of data to be delivered!**
  - Communication should be *high-speed* – **low delay and response time (few millisecond target!)**
  - Communication should be *pervasive* – **anytime, anywhere, anyone!**
  - Communication should be *reliable* – **five 9's!**
- Ultra-reliable low latency communications (URLLC)!

Conventional Services vs. Smart City/Autonomy Services

	Conventional services	URLLC services
Key goal	• High **data rates**	• Low **latency** • Rate **constraints**
Links	• **Few** simultaneous transmissions ($<$ 100)	• **Massive** (1000+) transmissions
Direction	• **To** users	• **From** sensors or **to autonomous** vehicles
Traffic	• **Large** packets	• **Small or mixed** packets
Devices	• **Human-centric:** Smartphones, tablets, etc.	• **Machine-centric:** sensors, drones, etc.

URLLC: What's new?

- URLLC has been around for a while but prior art…
  - Focused on *IoT sensors* (uplink) – **autonomous vehicles/drones are different (downlink?)!**
  - Assumes *known models* for traffic (M/M/1 etc.)– **latency has many components, hard to model!**
  - Considers slow deep reinforcement learning (DRL) – **learning in URLLC must handle extreme, rare conditions!**
  - Assumes *rate* can be *ignored* – **autonomous systems may need some form of rate guarantees!**
  - Comes up with *arbitrary latency numbers* – **latency is driven by the autonomous vehicles control!**
- ***Problem 1: Experienced DRL for Model-free URLLC***
- ***Problem 2: Control meets Communication for URLLC***

System Model

- Consider the downlink of a single-cell wireless network whose base station is sending latency-sensitive control message to autonomous vehicles
- We consider a downlink OFDMA system with resource blocks that must be allocated
- The downlink rate from the BS to a user *i* will be

$$r_i(t) = \sum_{j=1}^{K} \rho_{ij}(t) B \log_2 \left(1 + \frac{p_{ij}(t) h_{ij}(t)}{\sigma^2} \right)$$

RB allocation indicator

Bandwidth

Power allocated over RB *j*

Channel gain

Problem Formulation

- Reliability is defined as the probability of end-to-end packet delay exceeding a threshold
- We can map this to the following constraint:

$$r_i(t) > \underbrace{\phi}_{\text{Unknown relationship}}(\lambda_i(t), \beta(t), \gamma_i, D_i^{\max}) > \underbrace{\lambda_i(t)}_{\text{Arrival rate}}\underbrace{\beta_i(t)}_{\text{Packet size}}$$

- We do not make any assumptions for a delay model
  - Delay is intrinsically hard to model, most models are often unrealistic and have some hidden drawbacks
  - Delay has many components, hard to model their combination precisely

Problem Formulation

- Our goal is to solve the following problem

$$\min_{p_{ij},\rho_{ij}} \quad \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{i=1}^{N} \sum_{j=1}^{K} p_{ij}(\tau),$$

Reliability constraint

$$\text{s.t} \quad \Pr\{D_i > D_i^{\max}\} < 1 - \gamma_i^*, \quad \forall i \in \mathcal{N},$$

$$r_i(t) > \lambda_i(t)\beta_i(t), \quad \forall i \in \mathcal{N}, \quad \forall t$$

Rate constraint

Feasibility constraints

$$p_{ij}(t) \geq 0, \quad \rho_{ij}(t) \in \{0,1\},$$

$$\forall i \in \mathcal{N}, \quad \forall j \in \mathcal{K}, \quad \forall t,$$

$$\sum_i \rho_{ij}(t) = 1, \quad \forall j \in \mathcal{K}, \quad \forall t.$$

- Explicit rate guarantees imposed
- Challenging to solve because of our model-free assumption

Handling Model-Free

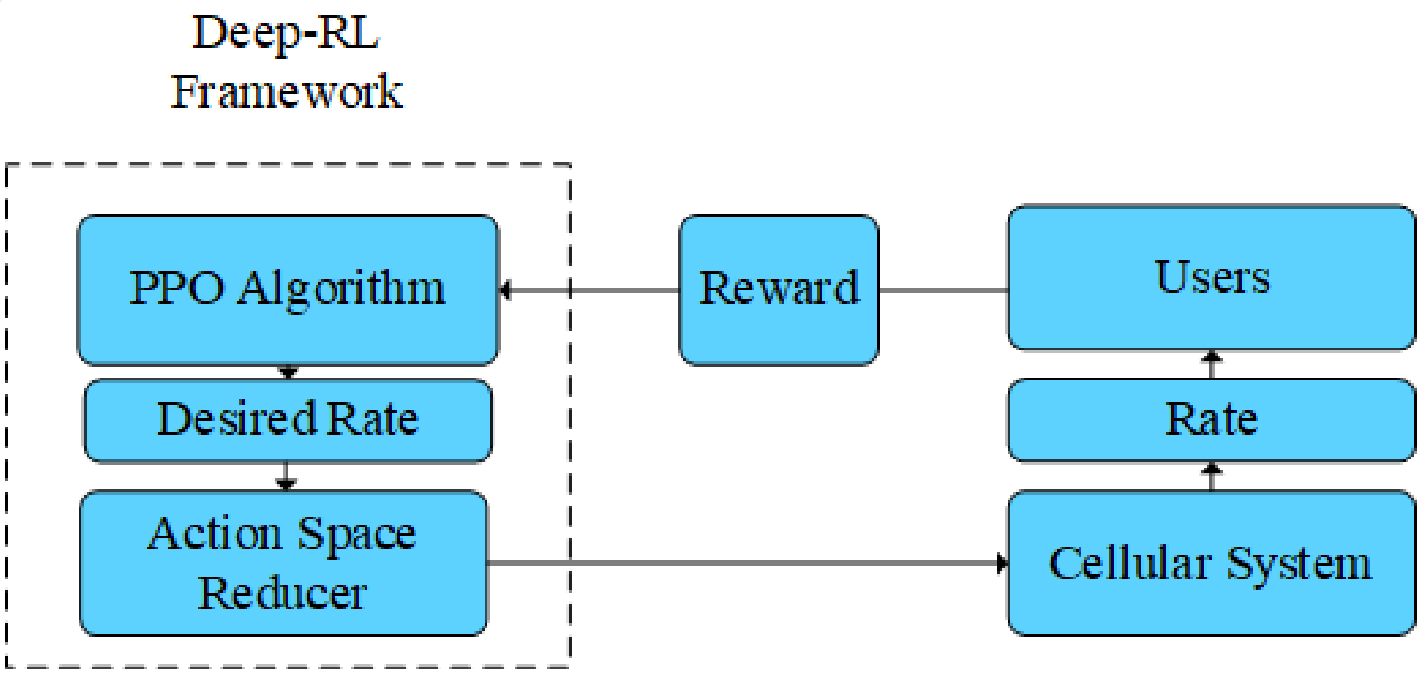
- In reality, a wireless network can empirically measure the delay

$$\gamma_i(t) = 1 - \Pr\left\{D_i > D_i^{\max}\right\} \approx 1 - \frac{\mu_i'(t)}{\mu_i(t)}$$

Ratio of number of packets with delay excess and total number of packets

- Network can "learn" the delay once it connects with a user
- How to learn? Reinforcement learning is natural but…
  - ….classical solutions cannot handle the large state space
- Solution: Deep reinforcement learning
  - Deep RL used because it is appropriate to handle our large state space not because it is "fashionable"

Deep-RL for Model-Free URLLC



- **State space:** number of packets transmitted, packet size, and channel gains
- **PPO:** Proximal policy optimization determines target rates
- **Action space reducer:** Deep-RL made tractable

Deep-RL for Model-Free URLLC

- The reward function used by deep-RL:

$$R(\boldsymbol{a}_t, \boldsymbol{s}_t) = -\sum_{i \in \mathcal{N}} w_i(t)(1 - \gamma_i(\boldsymbol{a}_t, \boldsymbol{s}_t)) - \alpha P(\boldsymbol{a}_t)$$

$$w_i(t+1) = \max\{w_i(t) + \gamma_i^* - \gamma_i(t), 0\}$$

Time-varying weight that control the reliability

- **Theorem 1:** By maximizing this reward, after convergence of the deep-RL algorithm, the reliability of each user is guaranteed, such that:

$$\gamma_i(t) \geq \gamma^* \, \forall i \in \mathcal{N}$$

  - Implicitly ensures rate requirements as well

Action Space Reduction

- The action space for the deep-RL is too large
  - Non-wireless prior work: Small action space (e.g. Atari)
  - Wireless prior work on deep-RL does not handle the large action space, but maintains complexity
- Two-step solution
  - Use the PPO algorithm optimize rate, rather than RB/power
  - Map PPO outcomes to original actions (action space reducer)
  - Action space reducer: a re-formulated optimization problem
- But, is deep-RL reliable and suitable for URLLC?
  - No! Can be **slow** to converge and **unreliable** extreme cases
  - Solution? Use generative adversarial networks (GANs)!

What is GAN?

- A **generative** model seeks to create data that is not seen before, but fits some input data distribution

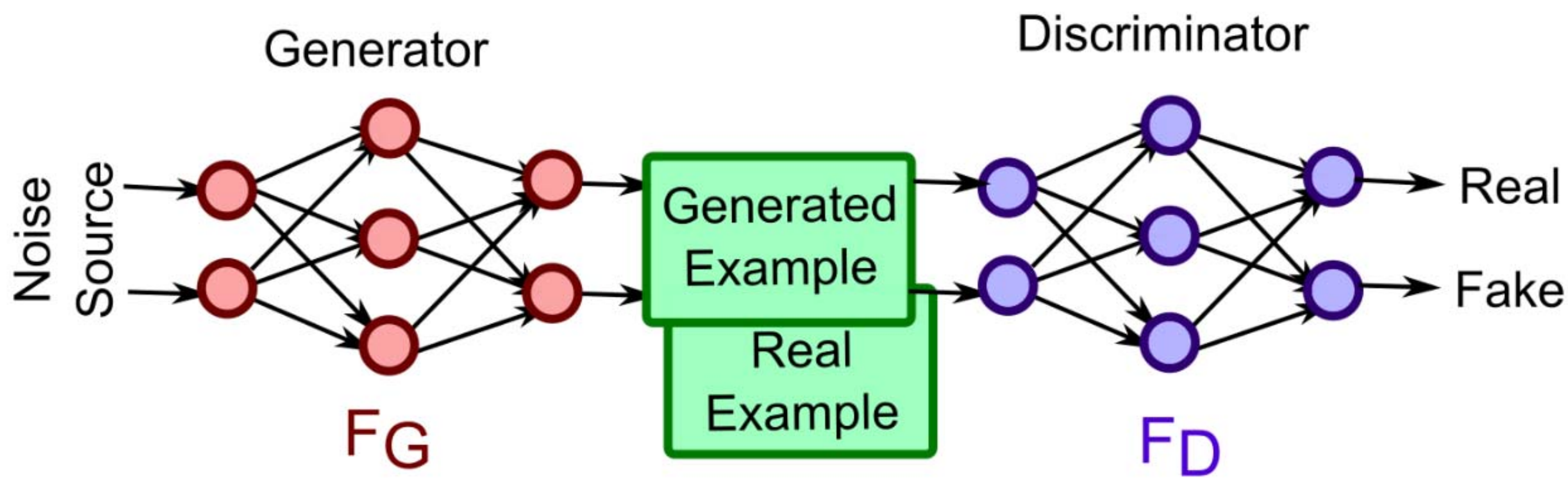
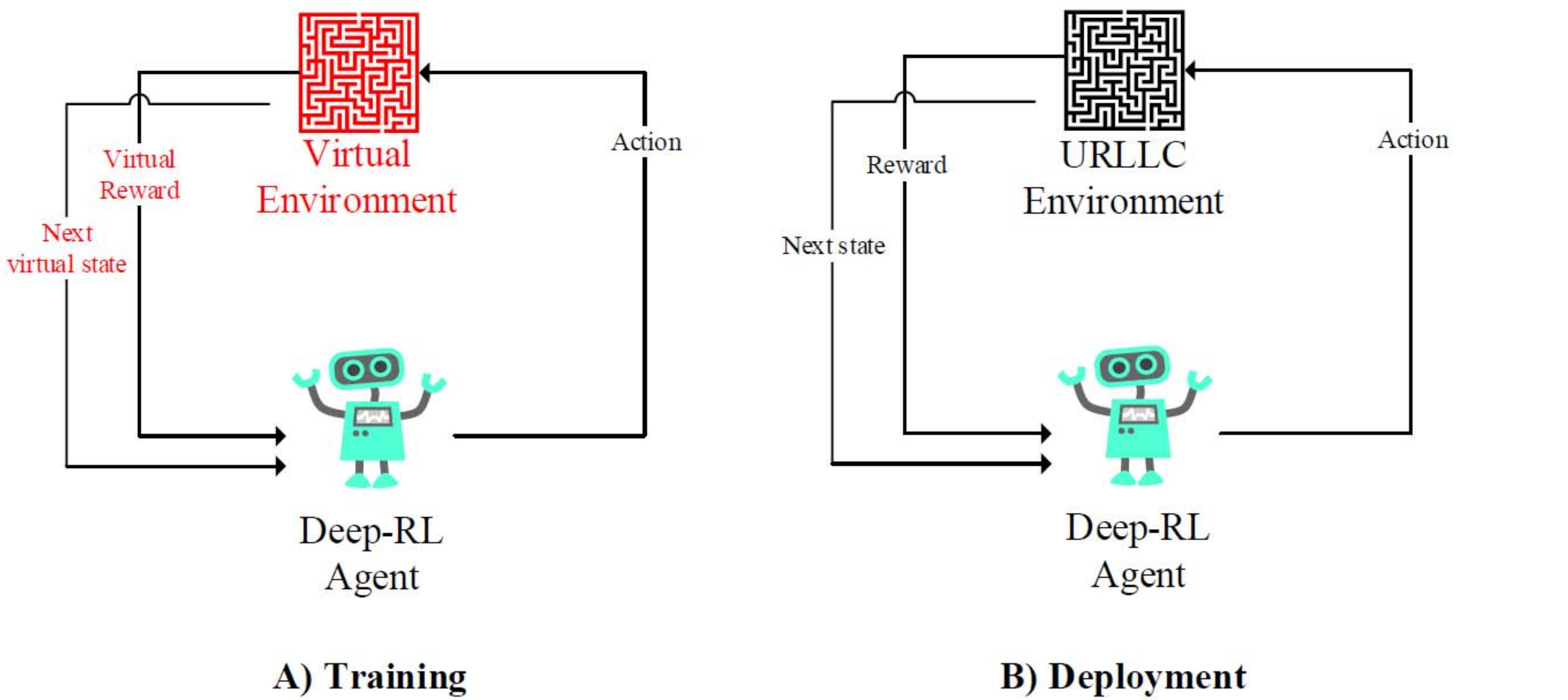


Figure source: http://hunterheidenreich.com/blog/what-is-a-gan/

- Generator: Tries to generate fake data
- Discriminator: Figure out whether data is fake or real
  - Adversarial interactions between the two (game theory)

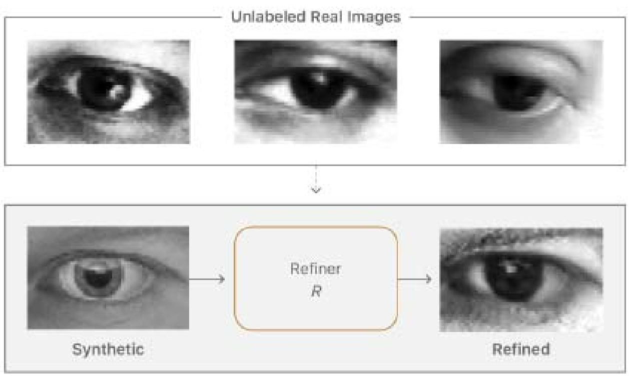
Experienced Deep RL

- Use GAN to create a "virtual environment" for training

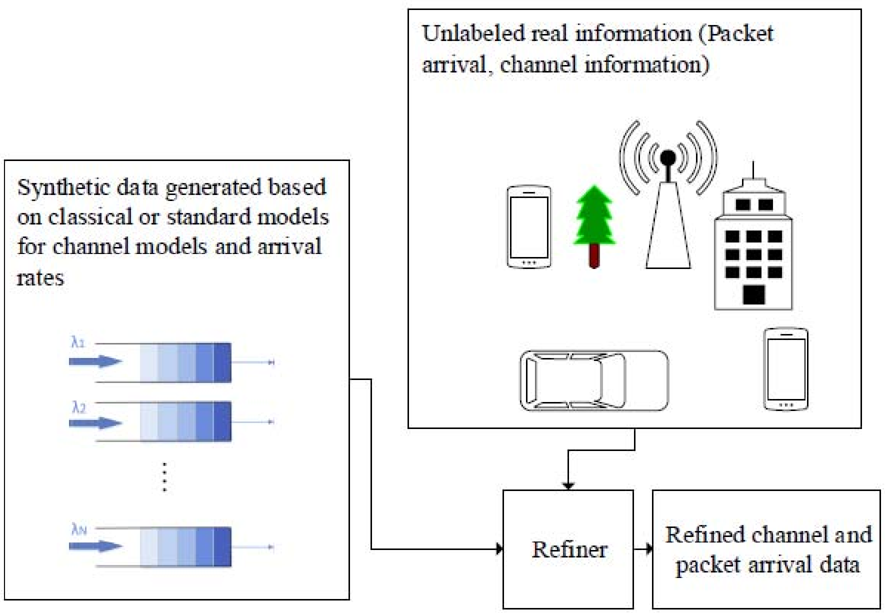


- Virtual environment is created by GAN using a mix of (limited) real data and synthetic (simulated) data

Experienced Deep RL



a)



b)

- GAN-based refiner
  - Proposed by Apple for computer vision
- Inputs
  - Unlabeled real data
  - Synthetic model data
- Output
  - Refined (and larger) dataset that includes new network conditions (extreme events) that can train your deep RL

Experienced Deep RL

- We train our deep RL using the GAN-refined data
  - We now have an experienced agent that has been exposed to extreme (rare) network conditions/events
  - The experienced agent will be able to better cope with extreme events as well as to converge faster in a URLLC system by eliminating transient period
- The refiner (which is a neural network) is trained as follows:

$$\theta_R^* = \arg\min_{\theta_R}\max_{\theta_D} f(\theta_R, \theta_D) = \arg\min_{\theta_R} f(\theta_R, \theta_D^*(\theta_R)),$$

$$\text{s.t.} \quad \mathbb{E}_{z\sim \mathrm{g_{sim}}}[\|F(z;\theta_R) - z\|] < \epsilon_r,$$

Refiner weights

Discrimnator weights

Refined, real-like data

Control Real vs. synthetic

Experienced Deep RL

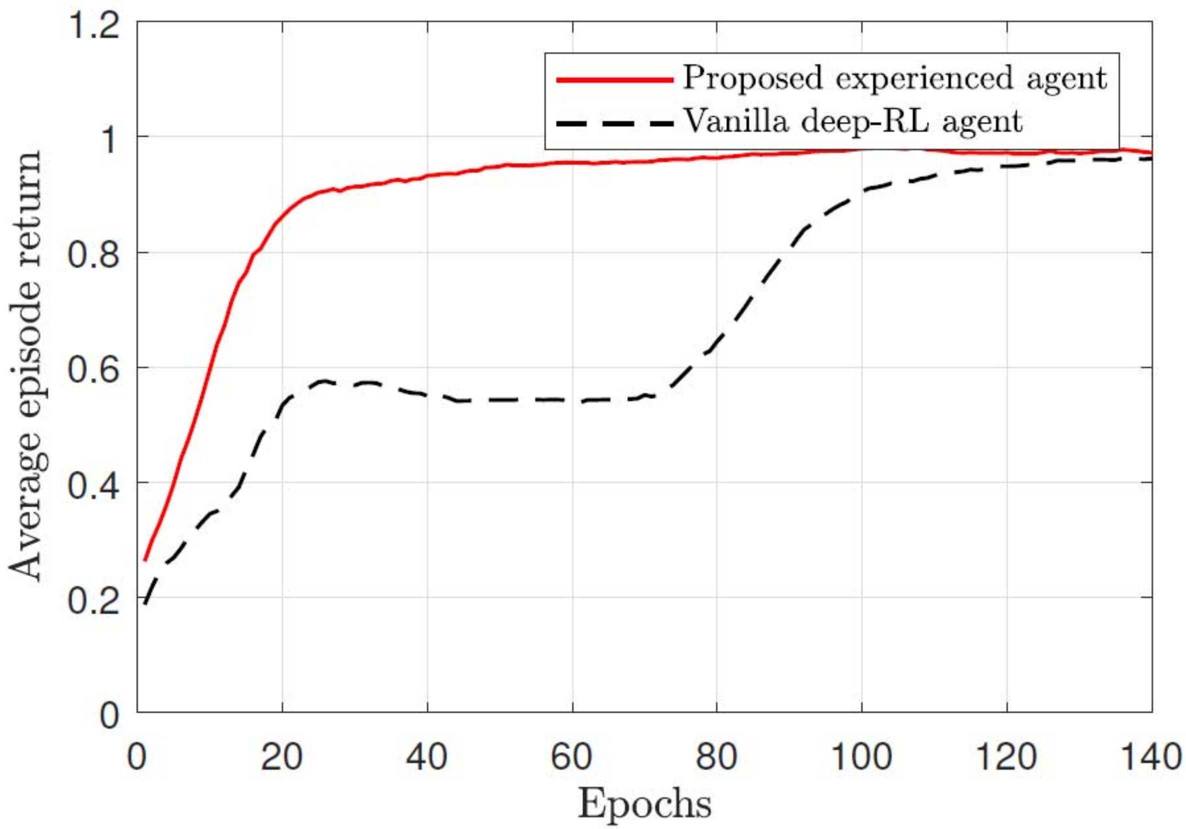
- Theorem 2: The refiner cannot be trained (i.e., problem is infeasible) if:

$$\epsilon_r < \epsilon_r^t \qquad \epsilon_r^t = \sqrt{||\mu_R||^2 + ||\mu_z||^2 - 2\mu_R^T\mu_z}.$$

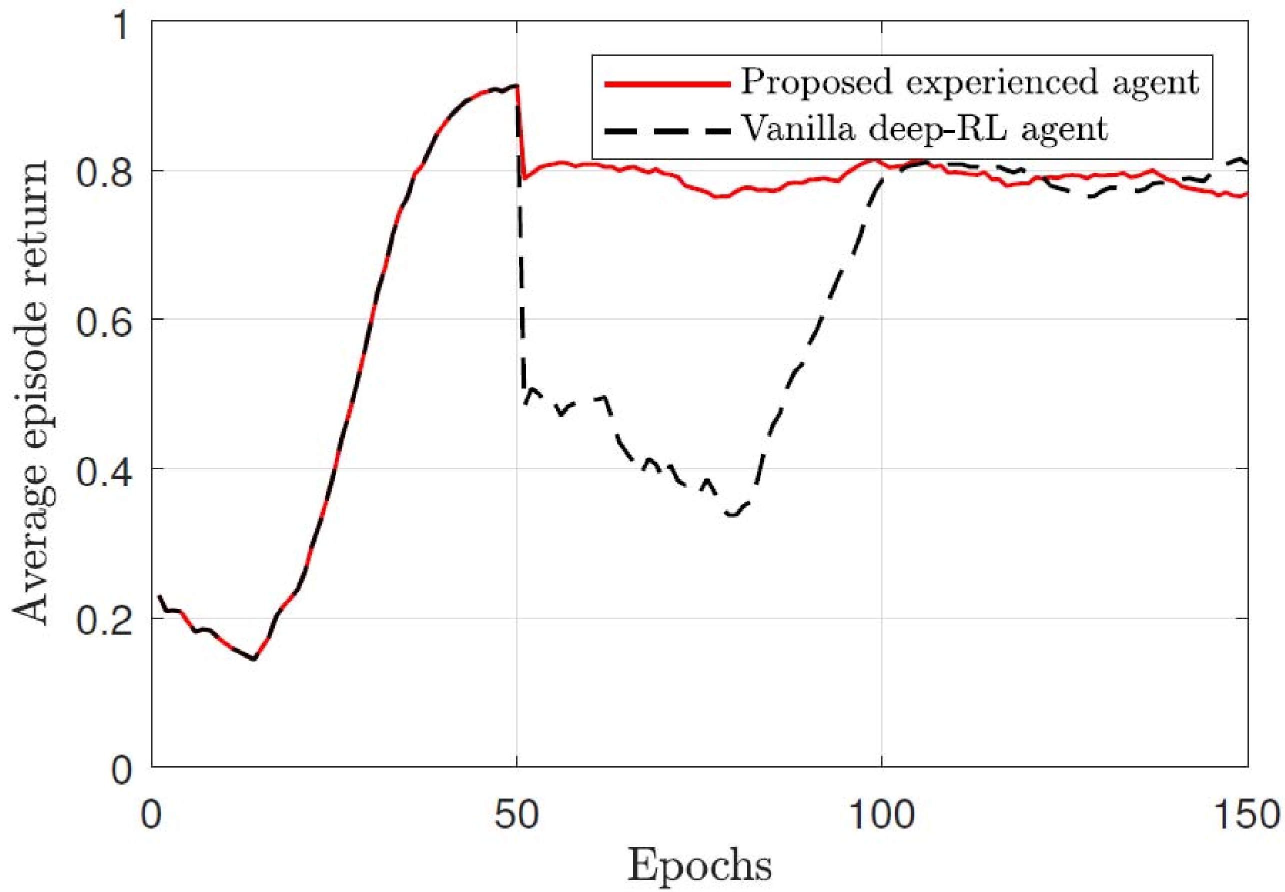
  - Threshold is function of the expected values of synthetic data and refiner output
  - We can control how our data is being generated
  - There is also an upper bound but hard to characterize mathematically
- Using our GAN and these results, we can create a training environment for ANY deep RL agent

Simulation Results

- We use a real dataset with specific packet sizes and interarrival times (with some modification)

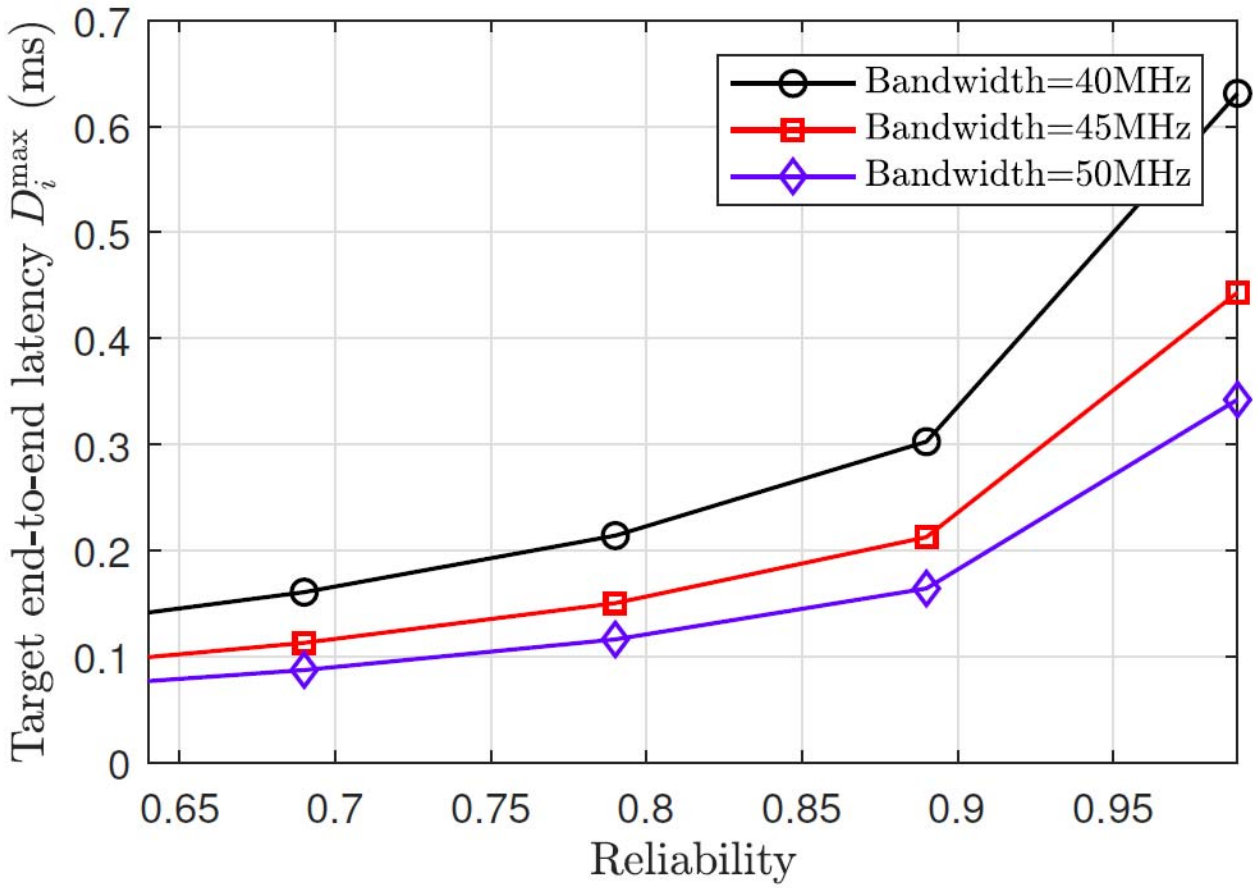


Simulation Results



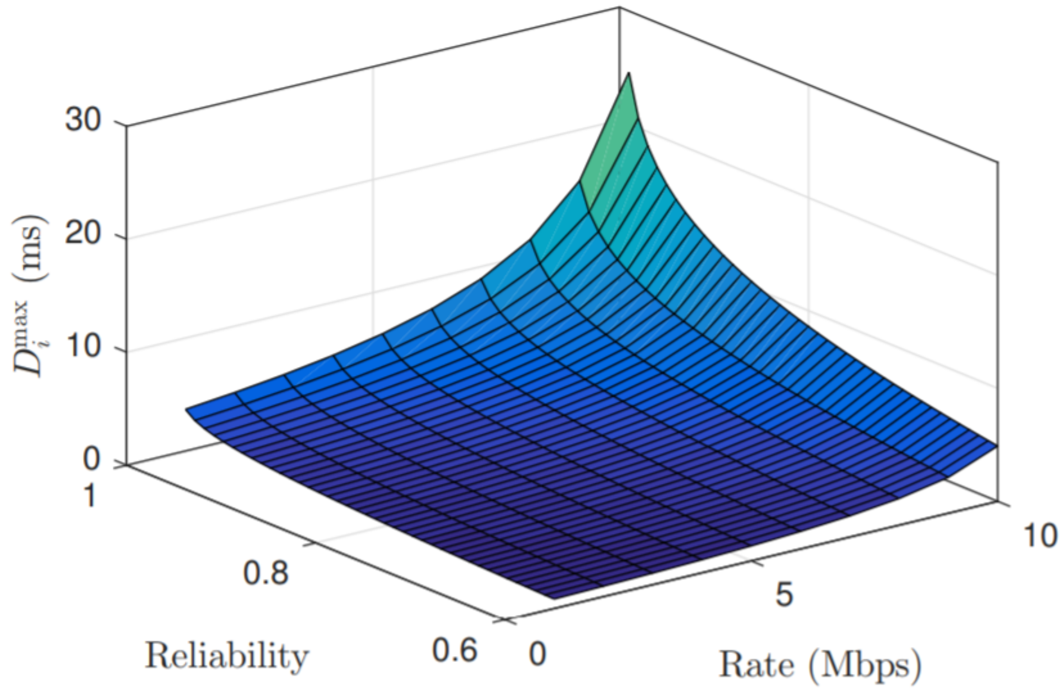
- Experience allows a very smooth handling of extreme events compared to vanilla deep-RL

Simulation Results



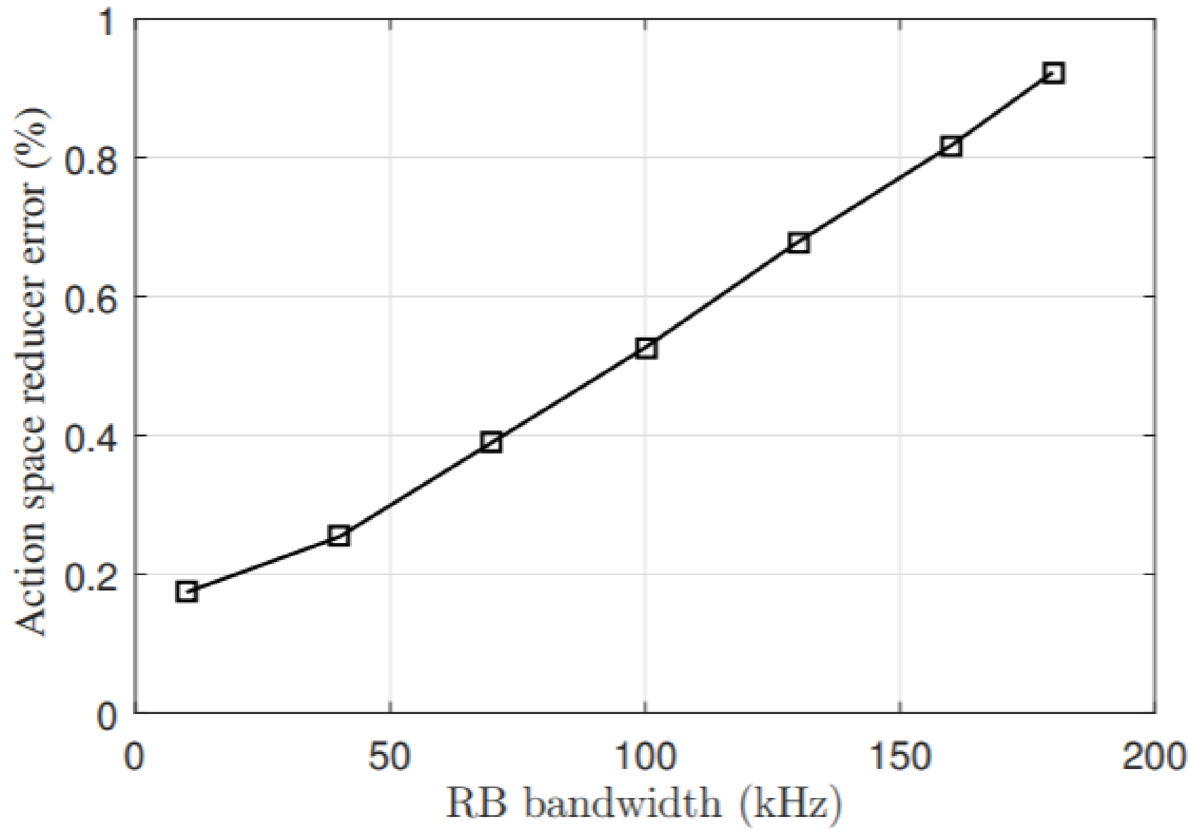
- Reliability-achievable latency have a clear tradeoff that can be improved with more bandwidth

Simulation Results



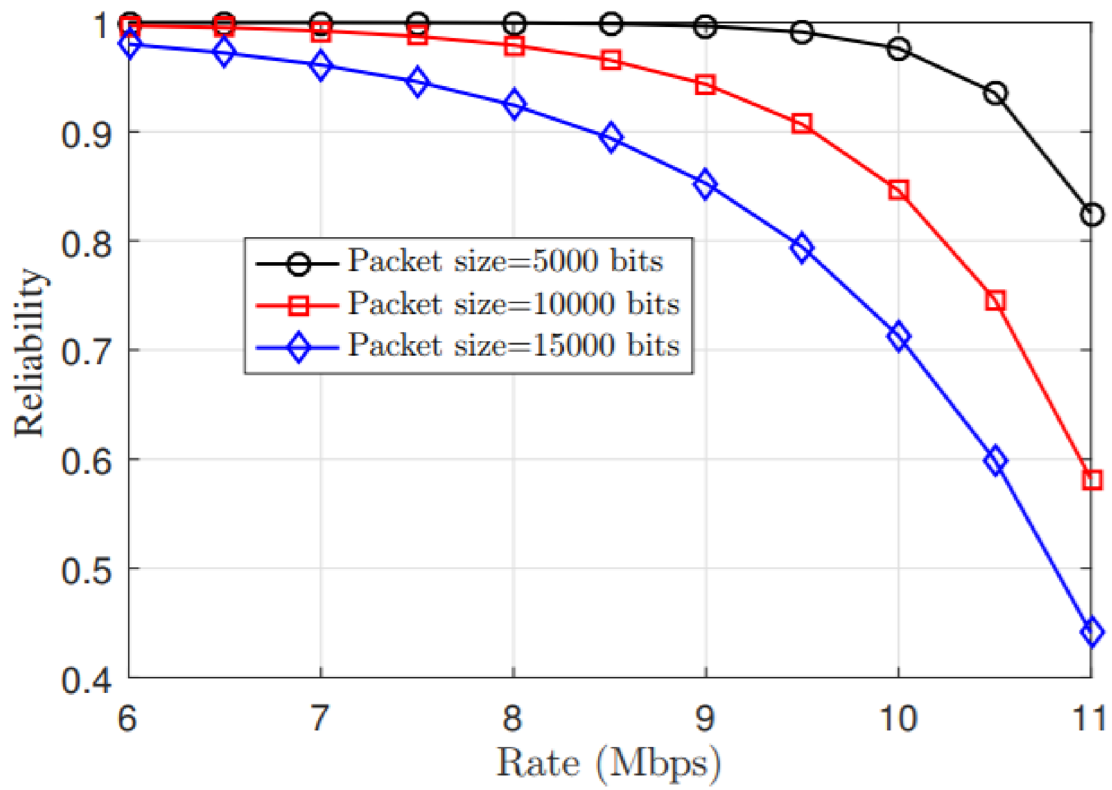
- Rate-reliability-latency tradeoff
  - 99% reliability, 4.2 ms latency, but rate of 1 Mbps
  - To gain 1% reliability, 47% lower delay but 7-times lower rate
  - Higher rate, higher power needed to have higher rates

Simulation Results



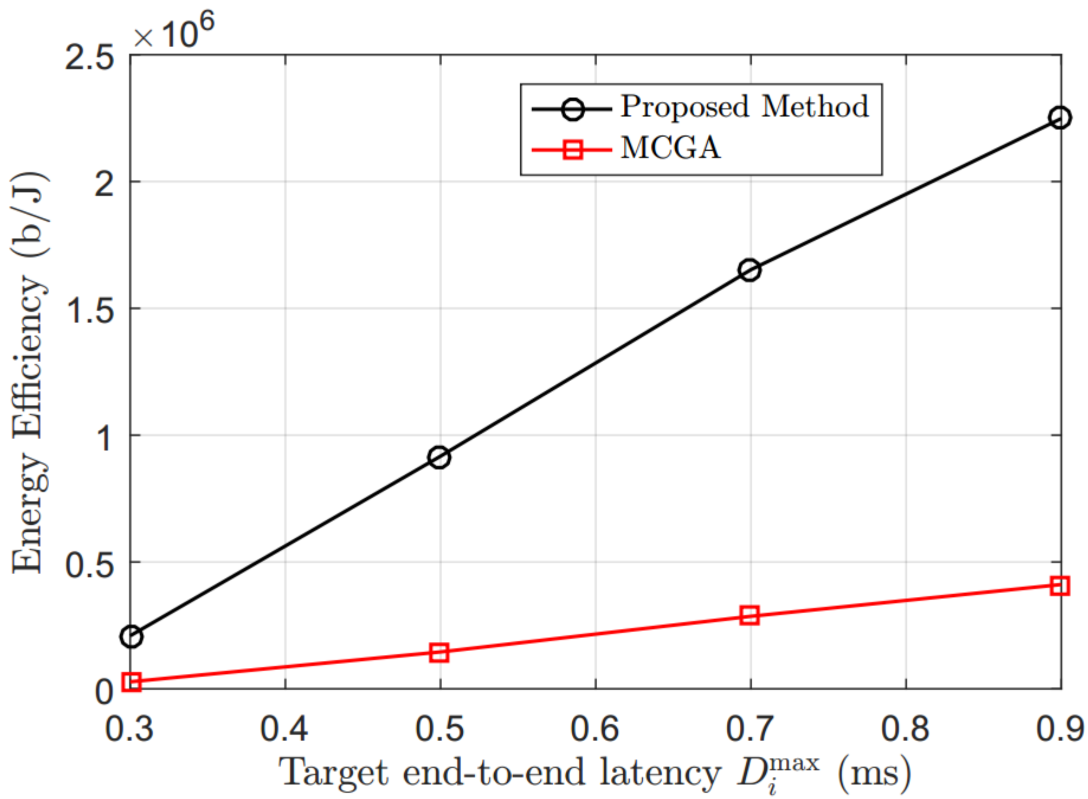
- For smaller bandwidth, the error due to approximation is smaller
- But error is anyway less than 1%!

Simulation Results



- Higher rate are more affected by packet size
- Higher reliability less rate corroborated
- 90% reliability at about 9 Mbps rate (10 ms delay)

Simulation Results



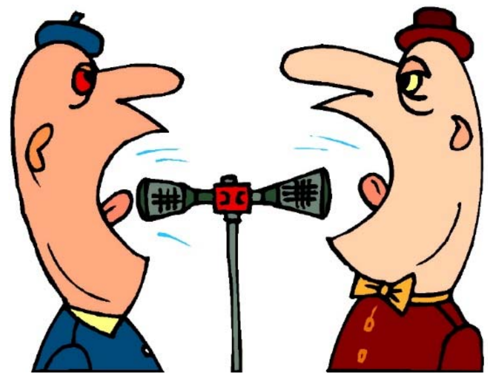
- MCGA: round-robin assignment of RBs to highest channel gain, we use a fixed power allocation
- Significantly more energy efficient than the baseline

Remarks

- The proposed approach can be easily extended to the uplink as well
  - Change in the problem formulation but the framework can work better
  - Issues such as interference and multiple access will have to be accounted for
- Use of a neural network instead of PPO to enhance/reduce possible overhead
- Extensions to multi-agent scenario
- The use of experienced deep RL in other problems

Control meets Communications

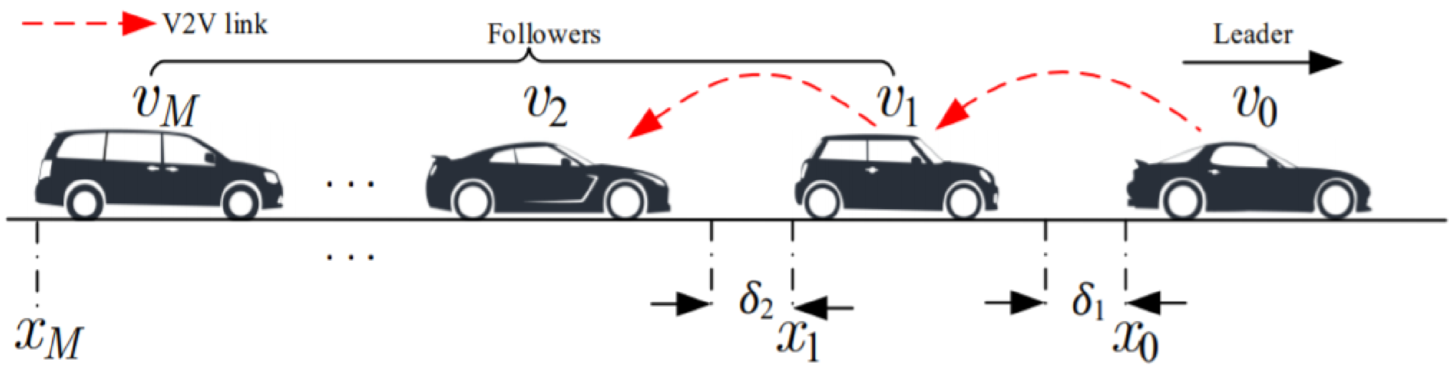
- What does low latency mean?
- We want 1 ms….no, we want 10 ms…. no, 5 ms is reasonable…
  - Futile and endless debate….



- Let's think realistically of the actual application
- Autonomous vehicles
  - Prime URLLC candidate
  - Their URLLC needs come directly from **their control system!**

System Model

- Consider an autonomous vehicular platoon (leader-follower)



- Operation governed by control dynamics (acceleration):

$$u_i(t) = a_i(t)[V(d_{i-1,i}(t)) - v_i(t)] + b_i(t)[v_{i-1}(t - \tau_{i-1,i}(t)) - v_i(t)]$$

Control gain

Headway-dependent velocity

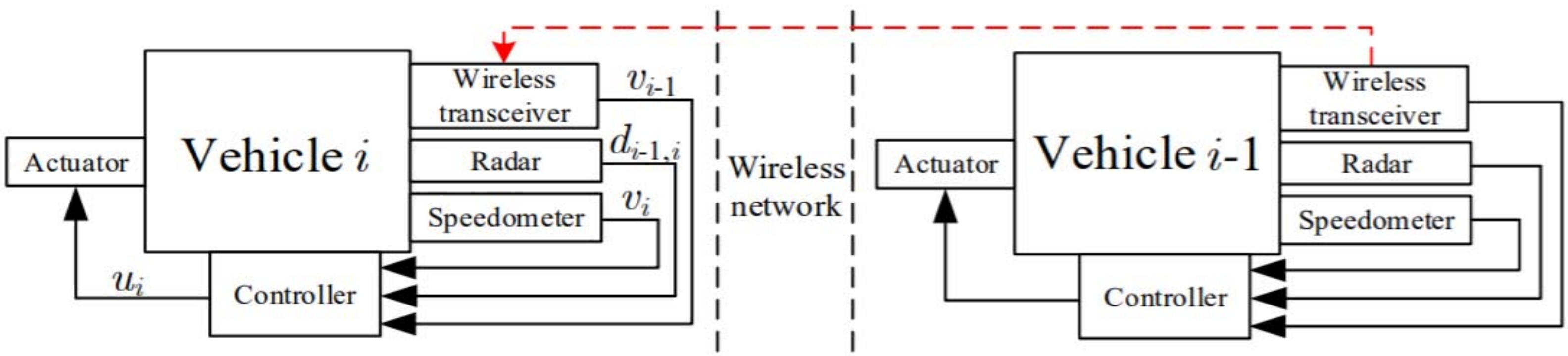
Speed

Control gain

Delay to receive speed measurement

System Model

- Basic platoon structure



- The control system can become unstable if information is heavily delayed (or erroneous, but we focus on delay)
- Joint communication and control design is necessary
  - Reliability and latency of communicated V2X information will have a direct control system impact!

Control System Stability

- Plant stability (convergence of errors) can be guaranteed if:

$$\tau_{i-1,i}(t) \leq \tau_1 = \frac{\lambda_{\min}(\boldsymbol{M}_3)}{\lambda_{\max}(\boldsymbol{M}_4)}$$

  - Matrices are easy to compute function of constants (e.g., sparse/dense headway distance, max speed, etc.)

- String stability (resilience to disturbance) guaranteed if:

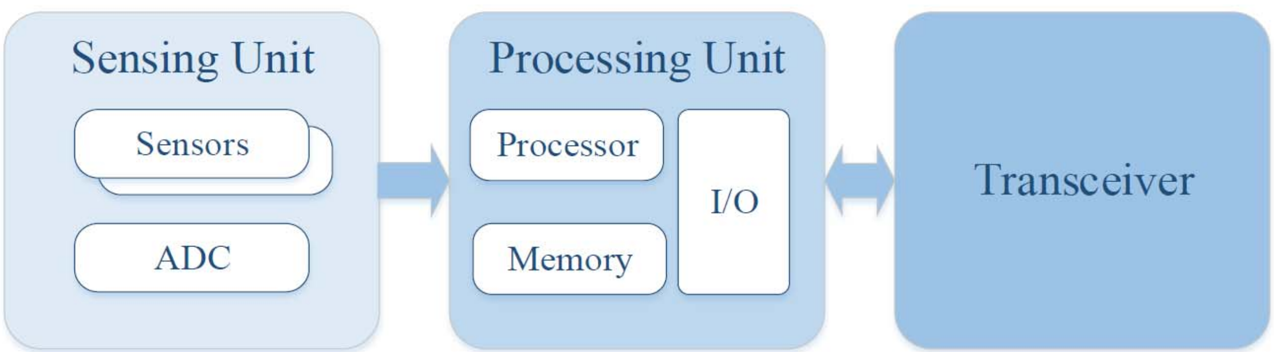
$$\tau_{i-1,i}(t) \leq \tau_2 = \frac{C^2 - 2A - B^2}{2AC}$$

  - Also easy to compute constants

- Concrete target latency: $\tau_{i-1,i}(t) \leq \min(\tau_1, \tau_2)$

Some Remarks

- We have established a concrete link between the wireless network delay and the control system delay
  - This differs from network control systems by the fact that we **model the V2X system as a wireless network**, control works assume it to be a black box
  - Prior wireless URLLC works ignore the control system
- We assume the delay stems from two queues



Reliability Analysis

- Using this queuing model and a standard wireless network modeling (V2V channels, interference, etc.), we can derive a lower bound for the reliability

$$\mathbb{P}(T_1+T_2 \leq \min(\tau_1,\tau_2)) \geq \max\left(1-\frac{\bar{T}_1+\bar{T}_2}{\min(\tau_1,\tau_2)}, 1-\exp\left(\bar{T}_1+\bar{T}_2-\min(\tau_1,\tau_2)\ln\left(\frac{\min(\tau_1,\tau_2)}{\bar{T}_1+\bar{T}_2}\right)\right)\right)$$

  - Function of delay expressions and wireless parameters
  - Explicitly links wireless and control parameters
  - Exact approximation in closed-form if no processing delay
- Perform any joint design for both wireless and control
  - Given wireless parameters, what are optimal control parameters?
  - Given control parameters, how to manage wireless resources?

Example Problem

- Optimal control system gains

$$\max_{a,b} \max\left(1 - \frac{\bar{T}_1 + \bar{T}_2}{\min(\tau_1, \tau_2)}, 1 - \exp\left(\bar{T}_1 + \bar{T}_2 - \min(\tau_1, \tau_2) \ln\left(\frac{\min(\tau_1, \tau_2)}{\bar{T}_1 + \bar{T}_2}\right)\right)\right)$$

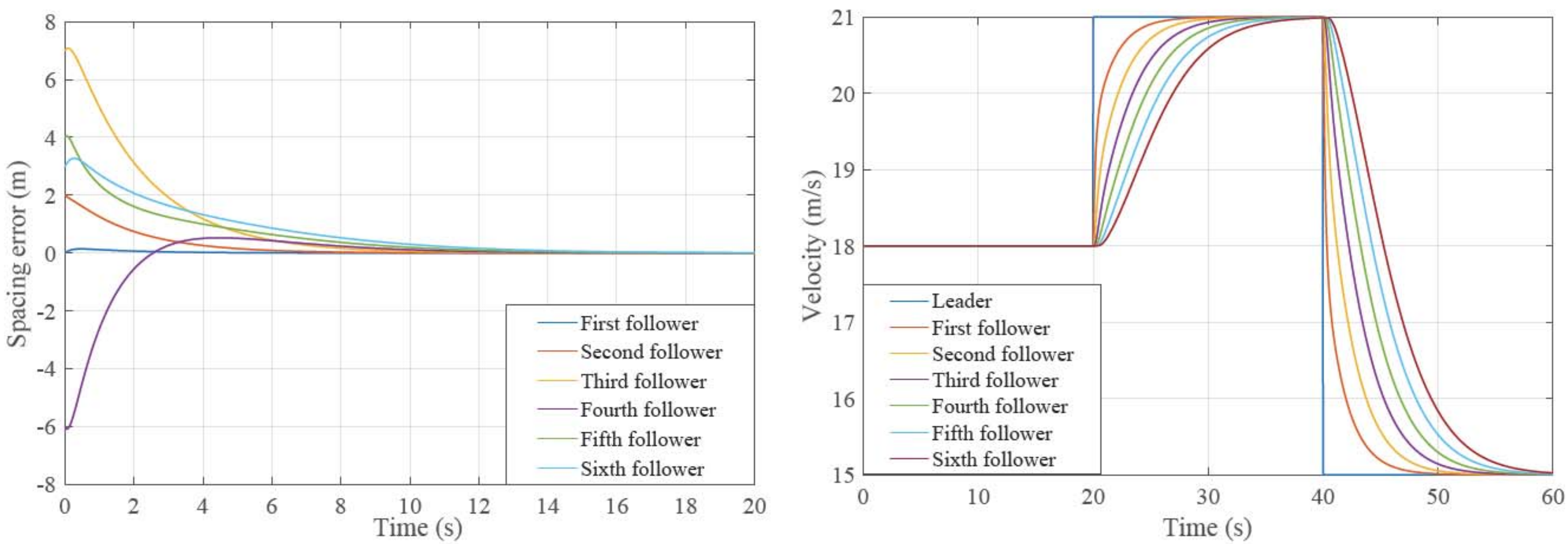
$$\text{s.t.}\quad \bar{T}_1 + \bar{T}_2 \le \min(\tau_1, \tau_2),$$

$$a_{\min} \le a \le a_{\max}, b_{\min} \le b \le b_{\max},$$

$$a^2 + b^2 + 2ab - 4a \ge 0, a + 2b - 2 \ge 0$$

- Can be handled analytically by showing that it is equivalent to a convex optimization problem
- One can address several similar optimizations (joint comm and control) that can follow from the work
- Latency is not arbitrary, it comes from the system!

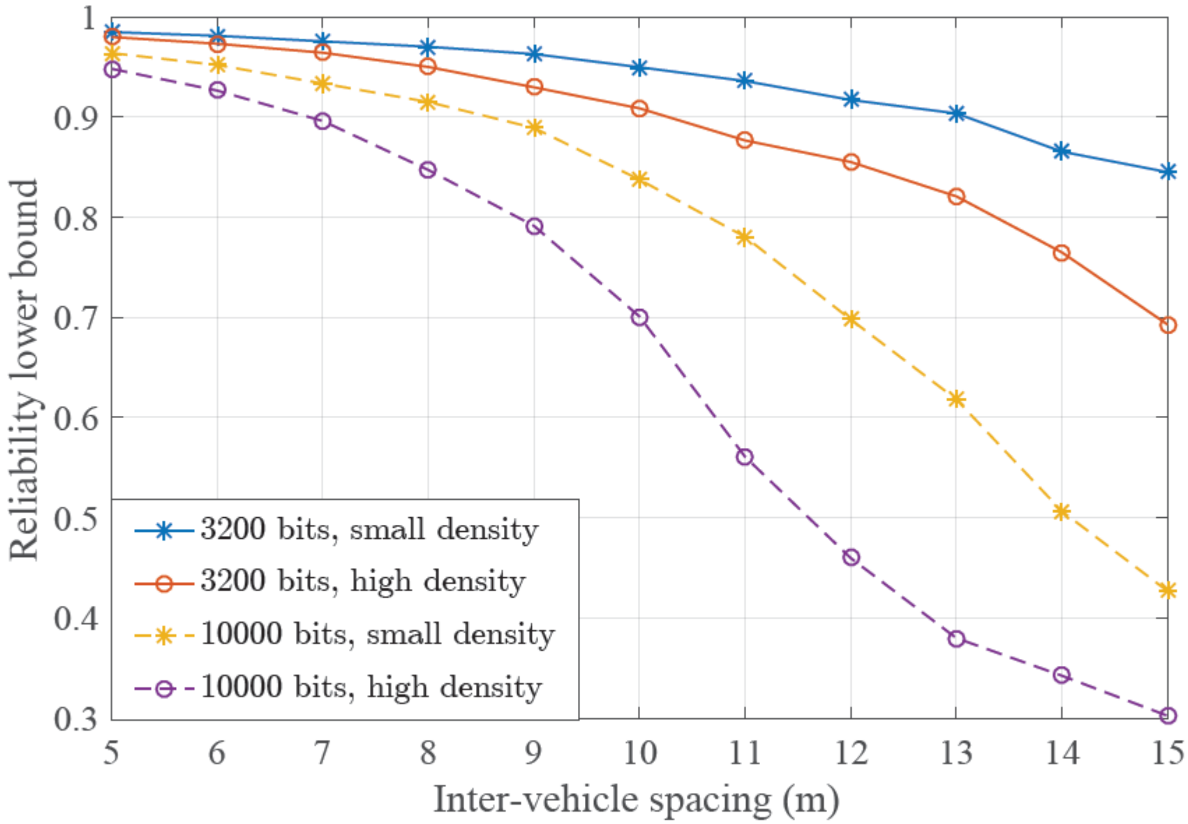
Simulation Results



(a) Plant stability. (b) String stability.

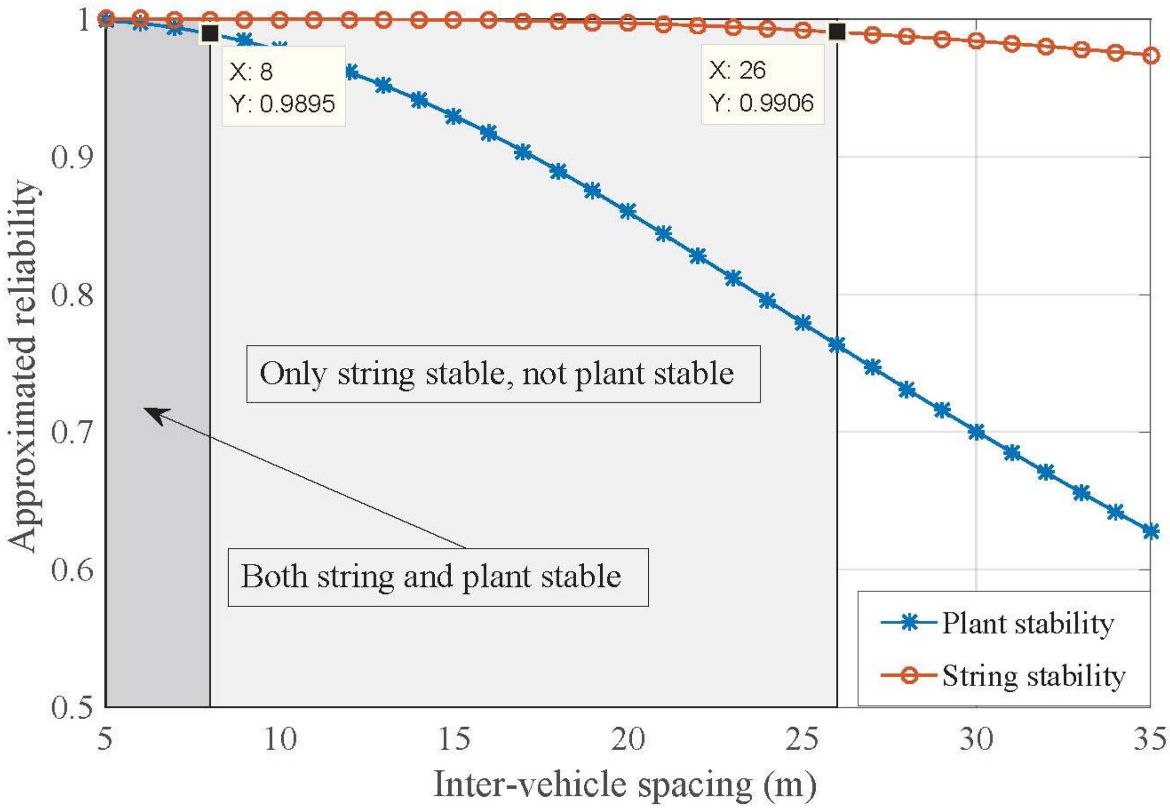
- Sanity check result (13.9 ms delay for plant stability)
  - Errors converge to zero (plant stability)
  - Disturbances do not propagate (string stability)

Simulation Results



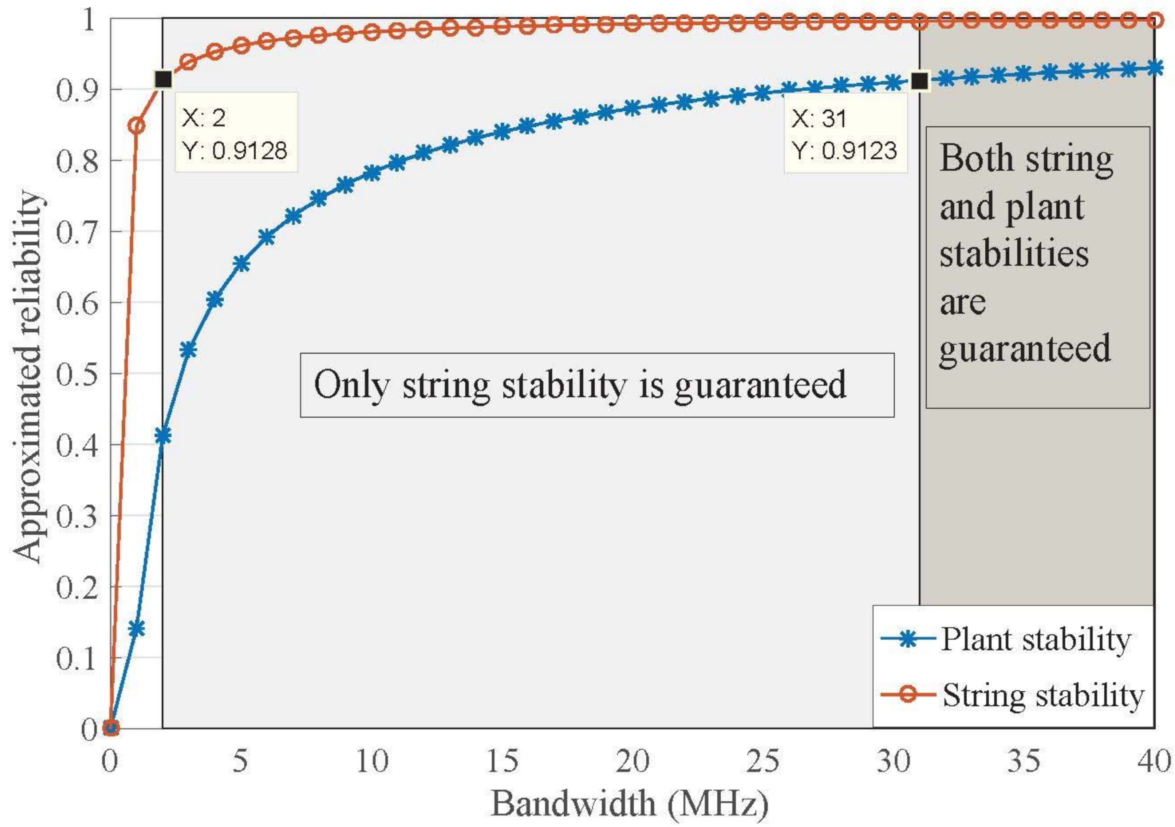
- Reliability decreases with inter-distance/density/packet size
- Target distance can now be designed based on the wireless network reliability!

Simulation Results



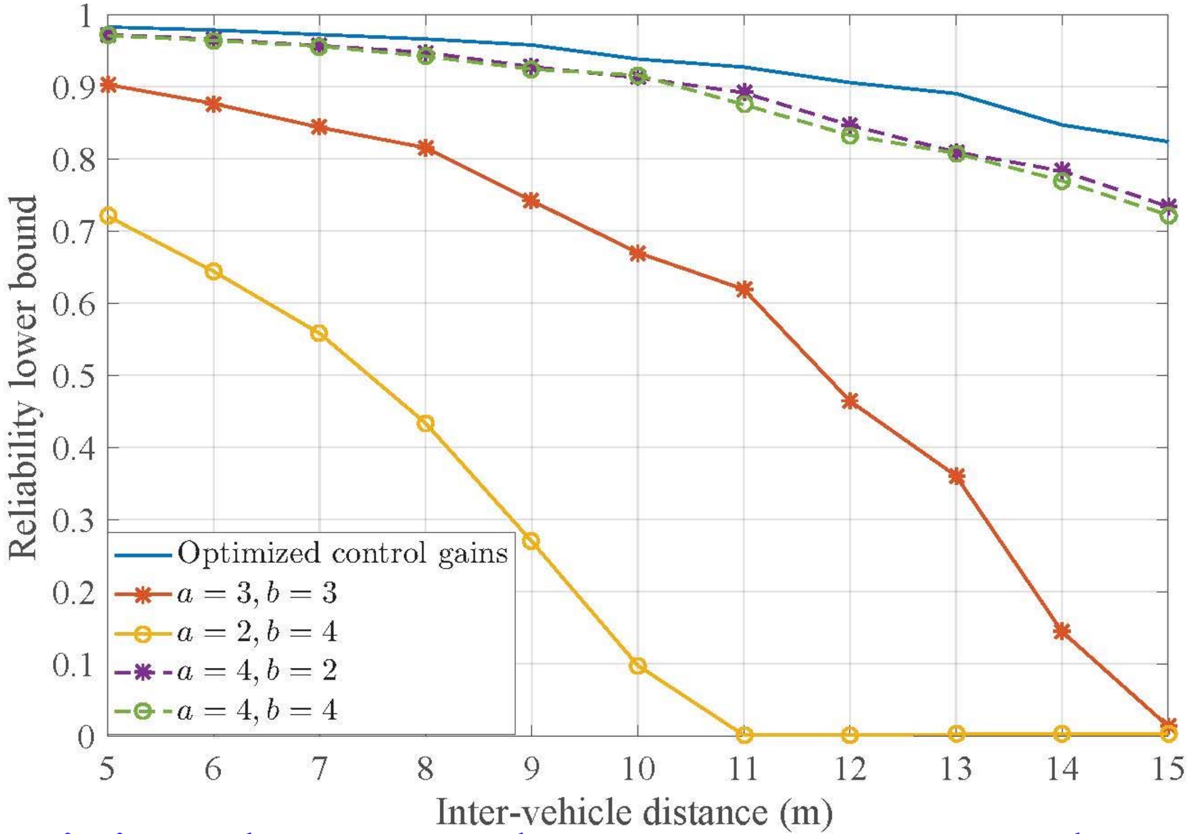
- Approximated reliability for platoons with different spacing
- Design guideline of choosing spacing to achieve a target reliability to support a stable control system

Simulation Results



- Approximated reliability with different total bandwidth
- We can see how wireless metrics directly impact the control system's stability

Simulation Results



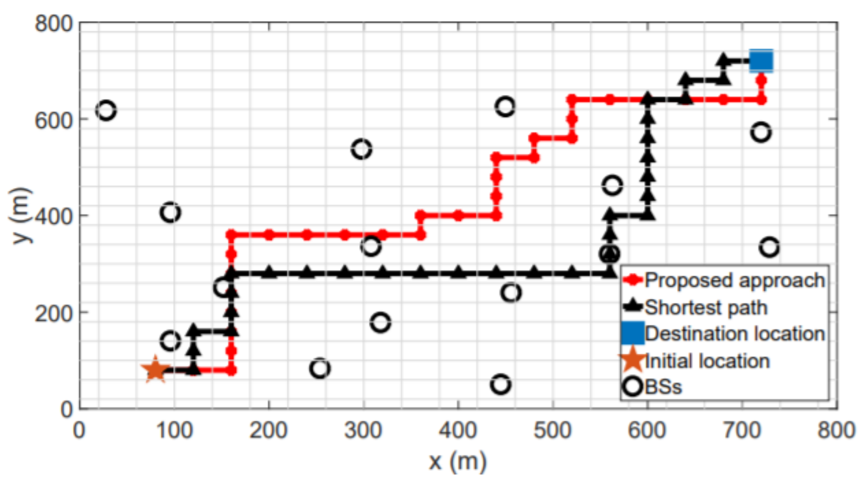
- Optimizing the control parameters can enhance reliability significantly
- Control and communications cannot be oblivious

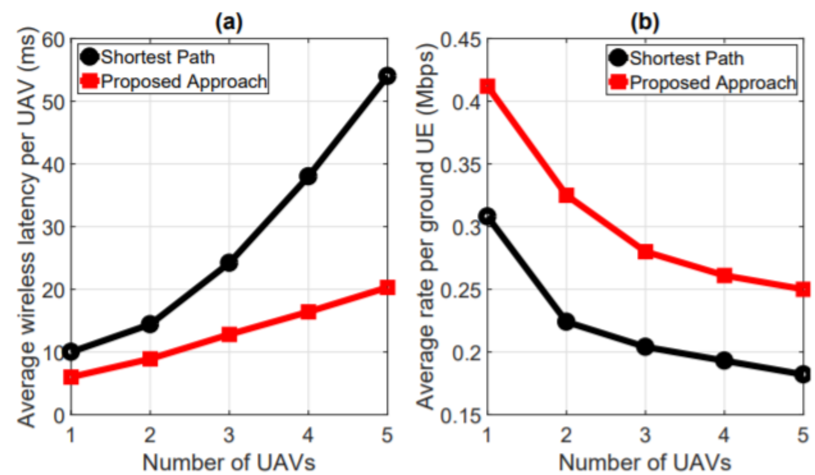
What else?

- Joint communications and control
  - What about independent (non-platoon) autonomous vehicles? (with T. Zeng, ICC'18)
  - Presence of dependencies across wireless and control systems (with T. Zeng, GC'19 and ongoing)
  - Robust learning and control (with. A. Ferdowsi et al., TCOM'19)
  - For UAVs (T. Zeng et al. Asilomar'19, M. Mozaffari et al., TCOM'19)
- Reliability and general URLLC
  - Reliance on non-average analysis (using tails and risk notions from economics), (works with S. Samarakoon et al., M. Khairy et al., C. Chaccour et al., B Zhou et al., N. Tran et al., and several others)
  - URLLC to support federated learning, i.e., communication for learning rather than learning for communications (with M. Chen et al. and M. Bennis et al.)
  - Terahertz frequencies (with C. Chaccour et al.)

Other Applications: Deep RL for UAVs

- How can a cellular-connected drone-UE navigate while minimizing LoS interference on ground BSs?
- Tradeoffs: energy efficiency, wireless latency, interference on ground users

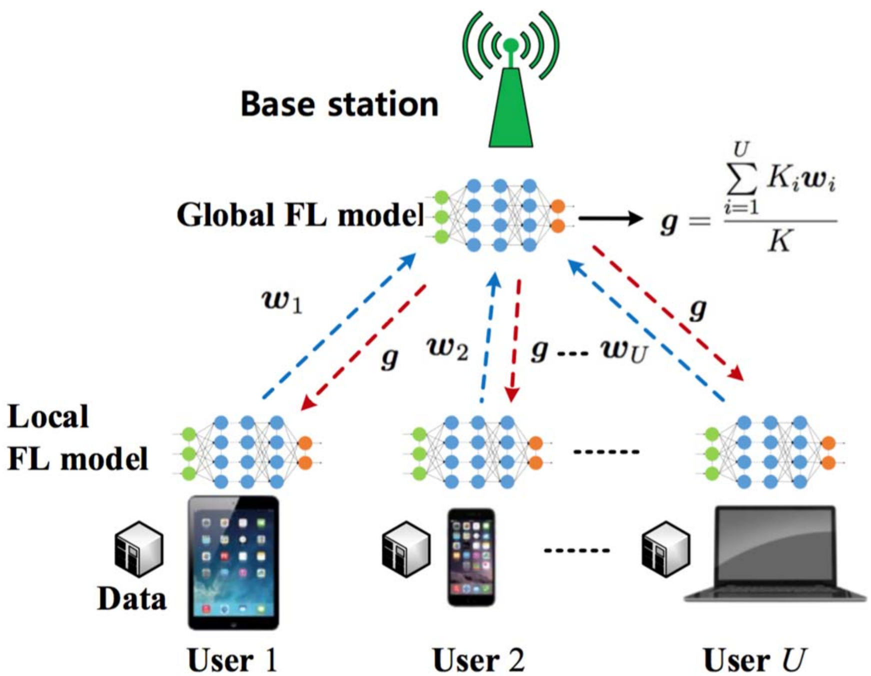


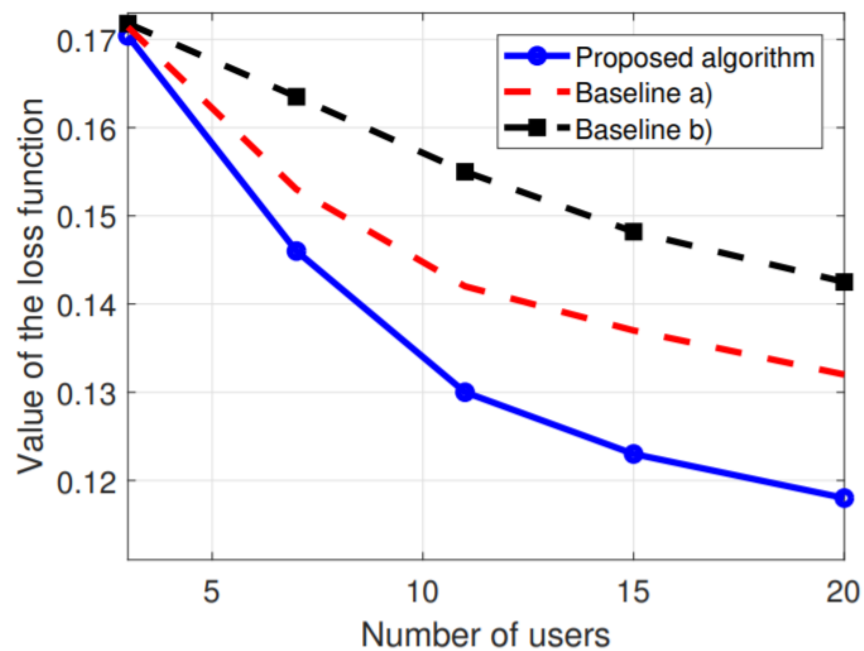


- U. Challita, W. Saad, and C. Bettstetter "Interference Management for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach", *IEEE Trans. on Wireless Communications*, 2019.

Other Applications: Reliable Federated Learning

- How can federated learning be deployed reliably over wireless networks?

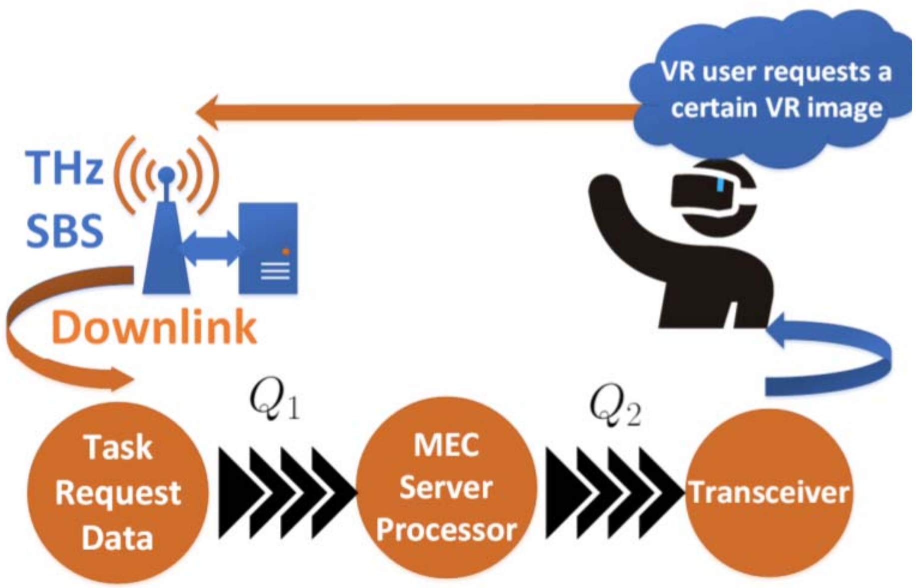


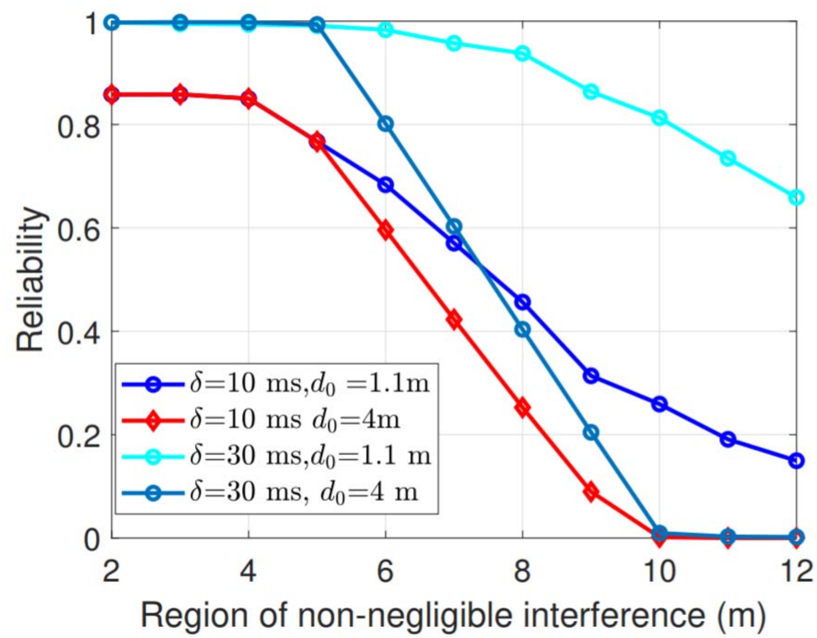


- M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A Joint Learning and Communications Framework for Federated Learning over Wireless Networks", arXiv:1909.07972, 2019.

Other Applications: THz Reliability

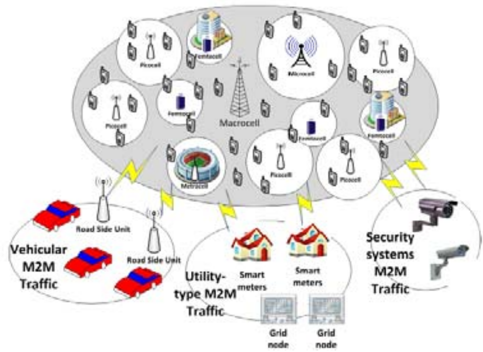
- Can THz frequency bands provide reliable communications for virtual reality applications?

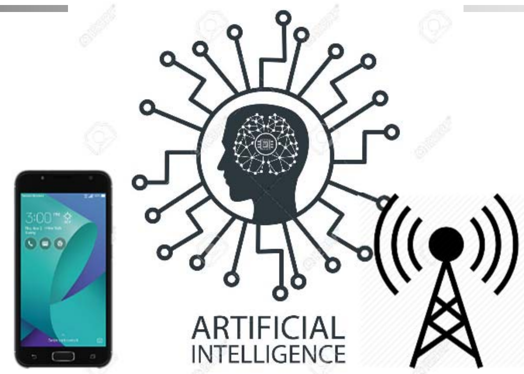




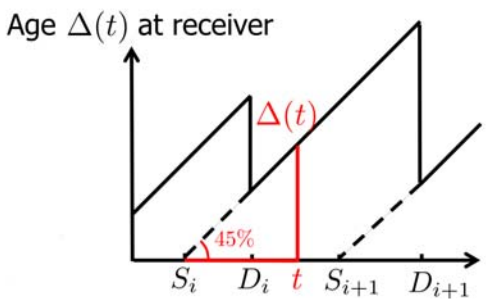
- C. Chaccour, R. Amer, B. Zhou, and W. Saad, "On the Reliability of Wireless Virtual Reality at Terahertz (THz) Frequencies", in Proc. of 10th IFIP NTMS, Mobility & Wireless Networks Track, 2019.

Other research areas

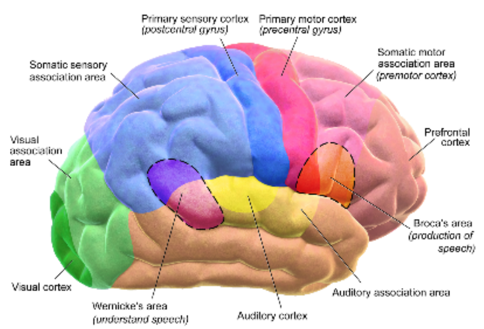




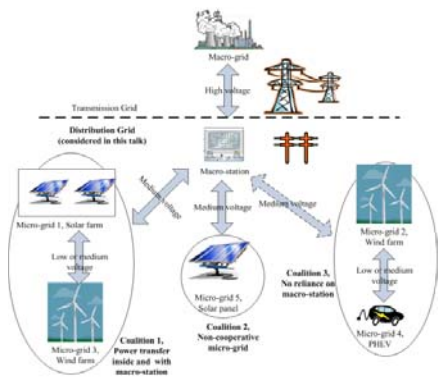
- Internet of things
  - **Ultra reliable, low latency** comm. with ML/GAN
  - Edge computing
  - Learning with finite memory
  - Security
- AI-enabled wireless
  - AI-based self-organization
  - **Federated/distributed on device edge learning**
  - Meta-learning
  - Learning meets comm.
- V2X and autonomous vehicles
  - Wireless and control
  - Edge analytics
  - Resource management
  - Security
- Age of information
  - Performance analysis of Internet of Things systems with age of information considerations
  - Game theory and resource management



Other research areas



- Human-aware wireless
  - Learning human behavior
  - Brain-aware wireless comm.
  - Brain-computer over wireless



- Smart grids
  - Demand side management
  - Game theory
  - Renewables/storage/EV

- Smart cities
  - Big data for smart city optimization
  - Resilience
  - Security

- Game theory
  - Foundations
  - Applications to CPS, security, wireless

- Capsule ML networks
- Blockchains
- Food-energy-water
- High-frequency comm.

NEWS@VT Group

- Acknowledgement to students/collaborators: Ali Taleb Zadeh Kasgari, Mingzhe Chen, Tengchan Zeng, Anibal Sanjab, Christina Chaccour, Omid Semiari, Mohammad Mozaffari, Gilsoo Lee, Mehdi Bennis, and Merouane Debbah
- NEWS@VT Group Current Members
  - 10 PhD students, 2 postdocs
  - Many visitors from China, Korea, UK, etc.
  - Open positions (PhD) for 2020
- Many potential collaborations with BUPT students and faculty across

Conclusions

- URLLC is an exciting area, particularly when dealing with autonomy
  - Restricting URLLC to "uplink, short packet, sensors" doesn't cut it anymore
- Reliable and extreme-trained deep learning
- Rate-reliability-latency tradeoff
  - An interesting tradeoff worthy of investigation
- Latency and reliability requirement come from the physical system
  - When dealing with autonomous vehicles, this physical system is the control system

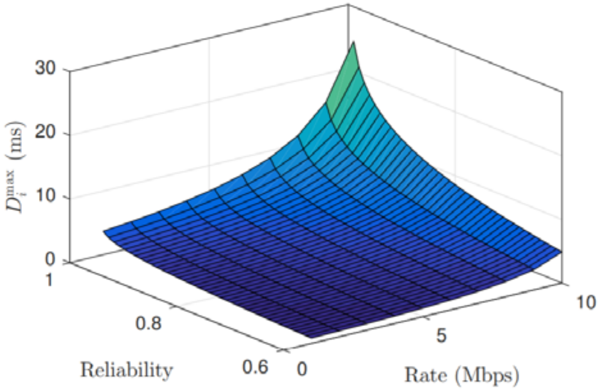
Finally….

Communications

Connect

Thank You

谢谢



Autonomy

***Act* (intelligently)**