

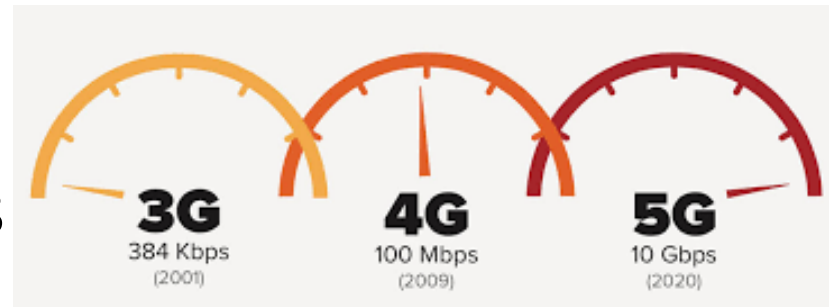
Machine Learning Enables Mobile Networks: Anticipatory Mobility Management for Ultra- Low Latency Mobile Networking

Professor Kwang-Cheng Chen, IEEE Fellow
Department of Electrical Engineering
University of South Florida
email: kwangcheng@usf.edu

Thanks to earlier research support from IBM, INTEL,
MediaTek, and Huawei

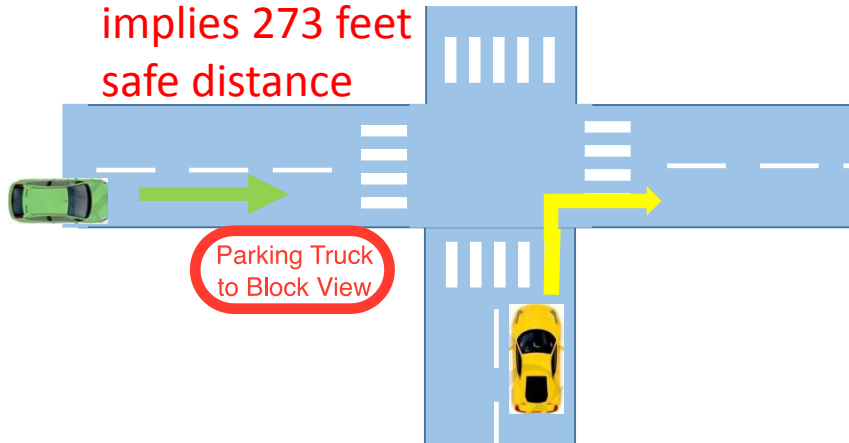
5G Technology

- The well known pillar technologies for 5G
 - Enhanced Mobile Broadband (eMBB)
 - Ultra Reliable and Low Latency Communication (uRLLC)
 - Mission critical services
 - Massive M2M/IoT Communication (mMTC)
- Field trials are coming
- Machine learning emerges

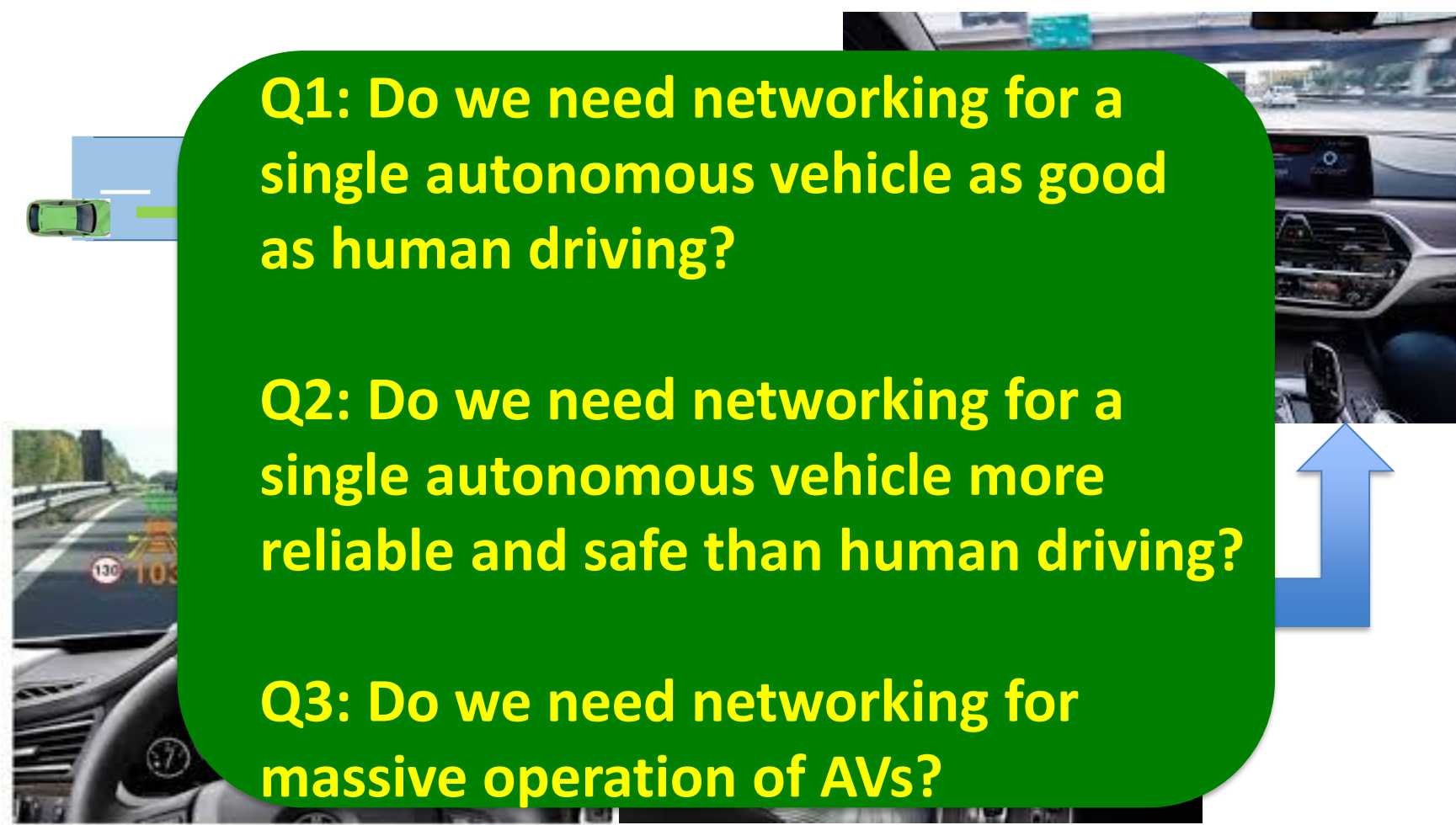


From Driving Assistance to Autonomous Driving

Green car at 50 MPH
implies 273 feet
safe distance



From Driving Assistance to Autonomous Driving



Q1: Do we need networking for a single autonomous vehicle as good as human driving?

Q2: Do we need networking for a single autonomous vehicle more reliable and safe than human driving?

Q3: Do we need networking for massive operation of AVs?

A new technology paradigm emerges

ULTRA-LOW END-TO-END LATENCY MOBILE NETWORKING AND MASSIVE MTC DEVICES

AI Without Wireless Networking

The New York Times | <https://nyti.ms/2u3QDYx>

TECHNOLOGY

Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam

By DAISUKE WAKABAYASHI MARCH 19, 2018

SAN FRANCISCO — Arizona officials saw opportunity when Uber and other companies began testing driverless cars a few years ago. Promising to keep oversight light, they invited the companies to test their robotic vehicles on the state's roads.

Then on Sunday night, an autonomous car operated by Uber — and with an emergency backup driver behind the wheel — struck and killed a woman on a street in Tempe, Ariz. It was believed to be the first pedestrian death associated with self-driving technology. The company quickly suspended testing in Tempe as well as in Pittsburgh, San Francisco and Toronto.

The accident was a reminder that self-driving technology is still in the experimental stage, and governments are still trying to figure out how to regulate it.

Uber, Waymo and a long list of tech companies and automakers have begun to expand testing of their self-driving vehicles in cities around the country. The



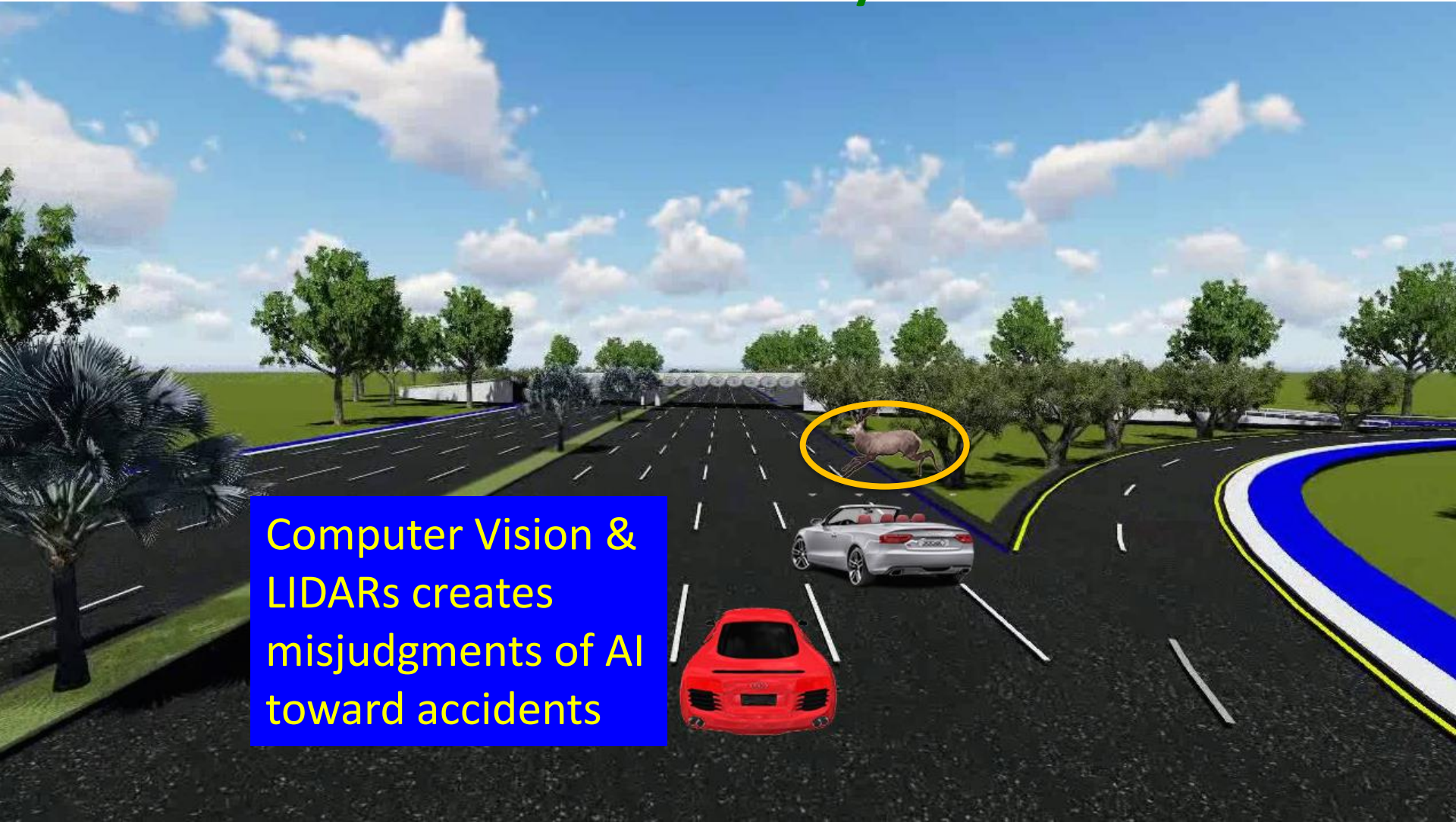
Single-agent AI of only on-board sensors does not satisfactorily work!



UNIVERSITY OF
SOUTH FLORIDA
COLLEGE OF ENGINEERING
Department of Electrical Engineering

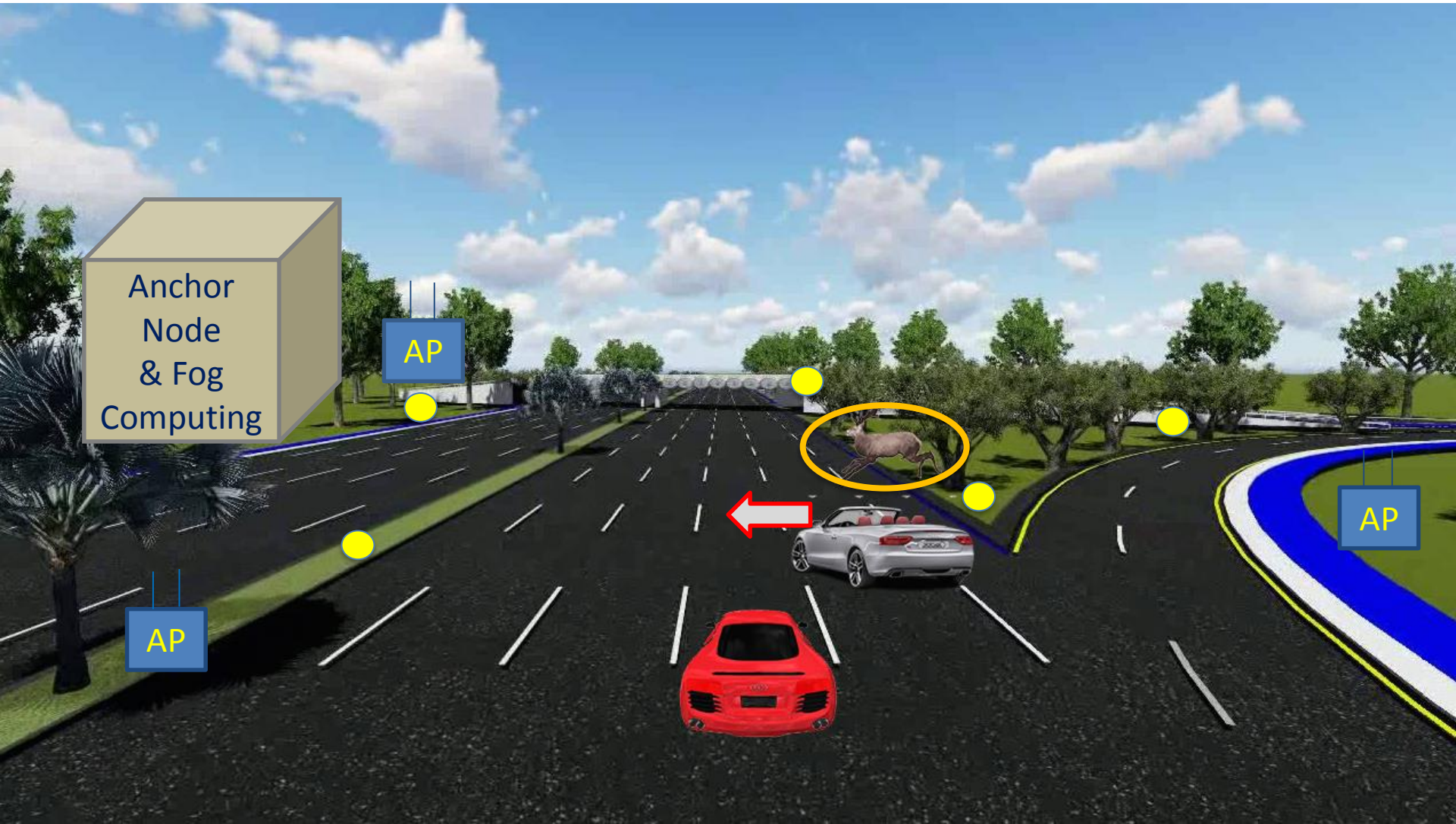
Vision is Not Safe Enough

To see is not necessarily to believe!

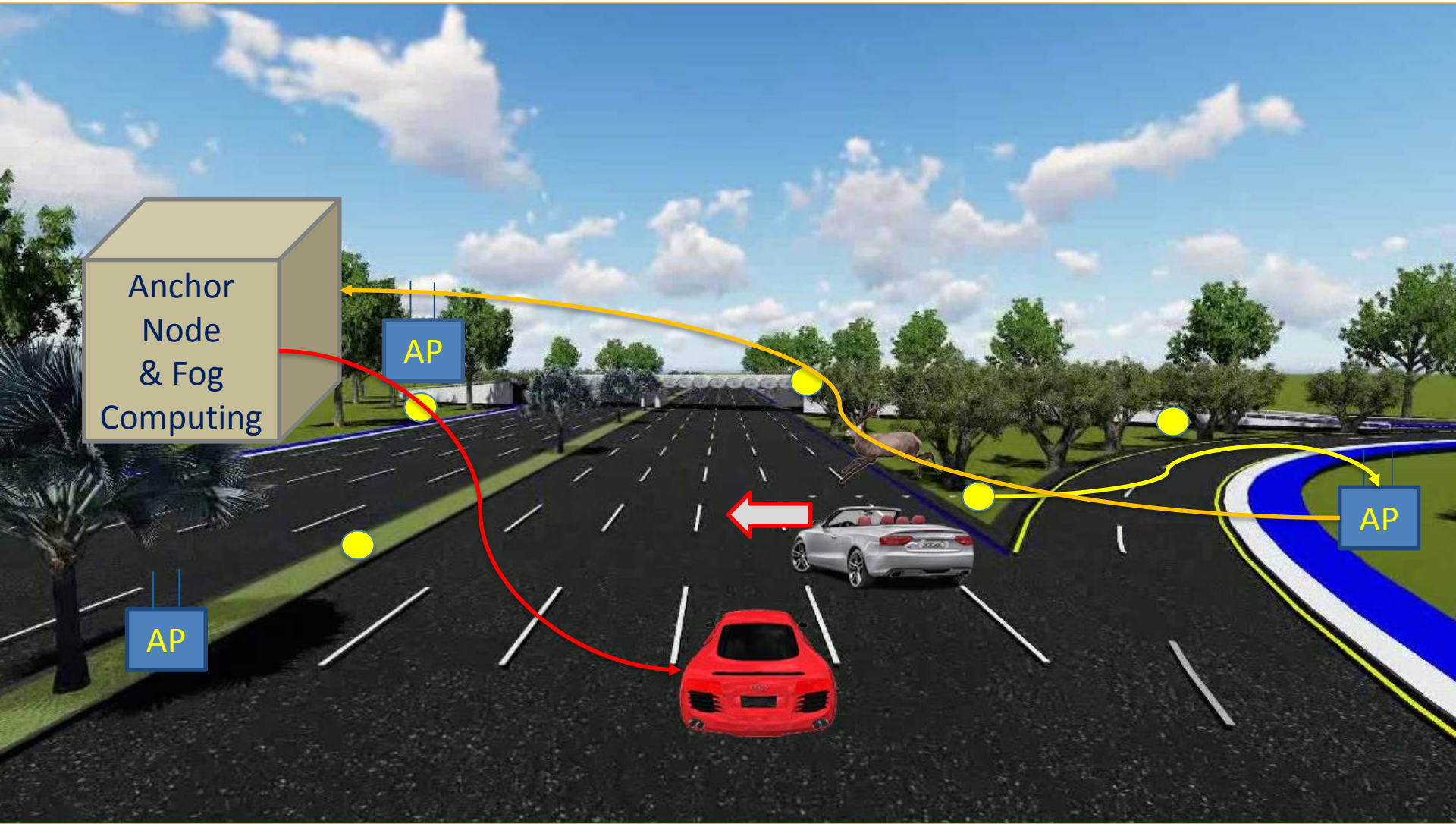


Computer Vision &
LIDARs creates
misjudgments of AI
toward accidents

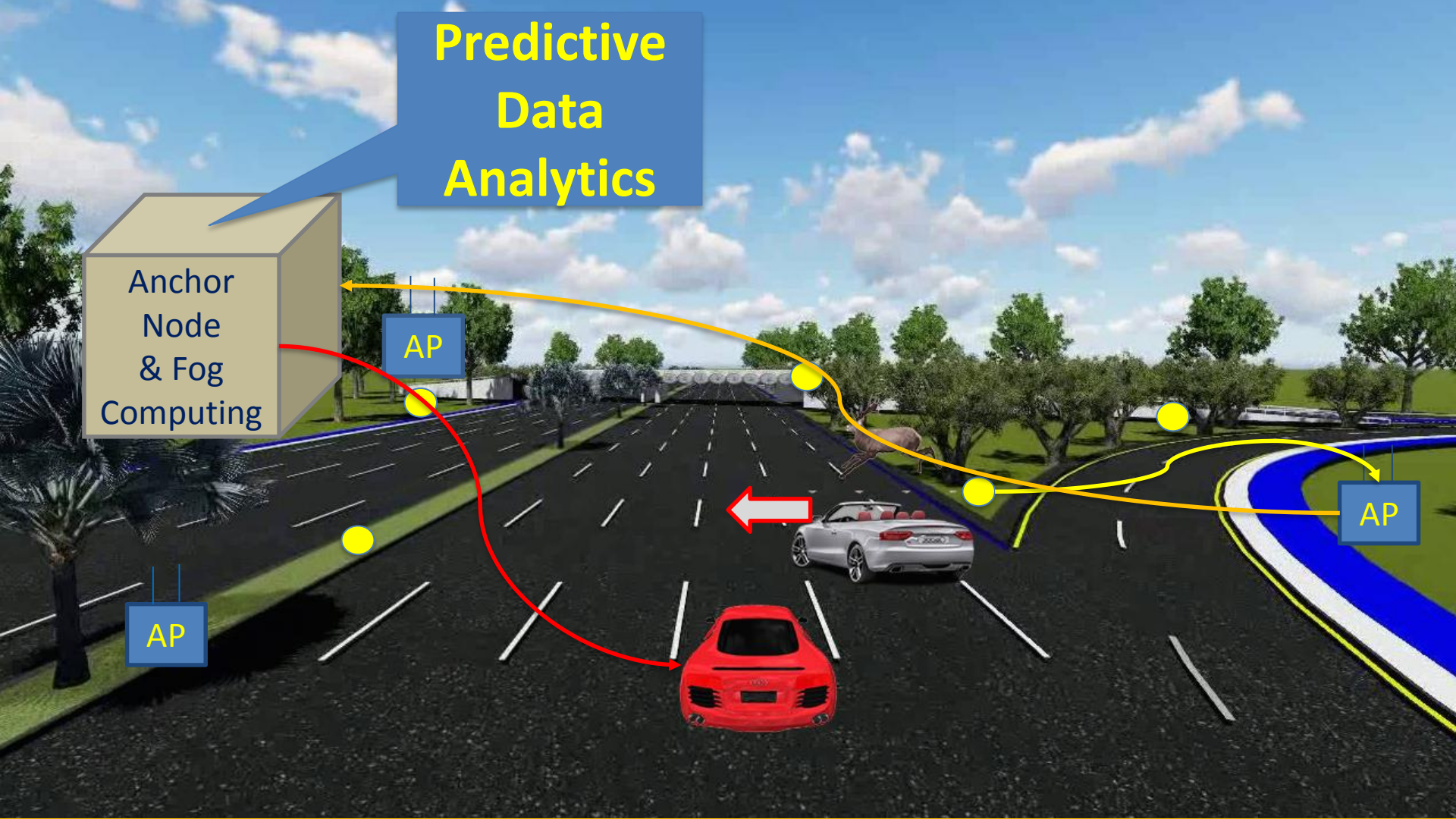
Fog Computing and Roadside Sensors



New Technology Paradigm



Ultra-Low Latency Wireless Networking



Predictive
Data
Analytics

Anchor
Node
& Fog
Computing

AP

AP

AP

Ultra-Low Latency Wireless Networking

Predictive
Data
Analytic

Anchor
Network
& Fog
Computing

AI and Mobile
Computing Meets
Networking and
Communication

AP

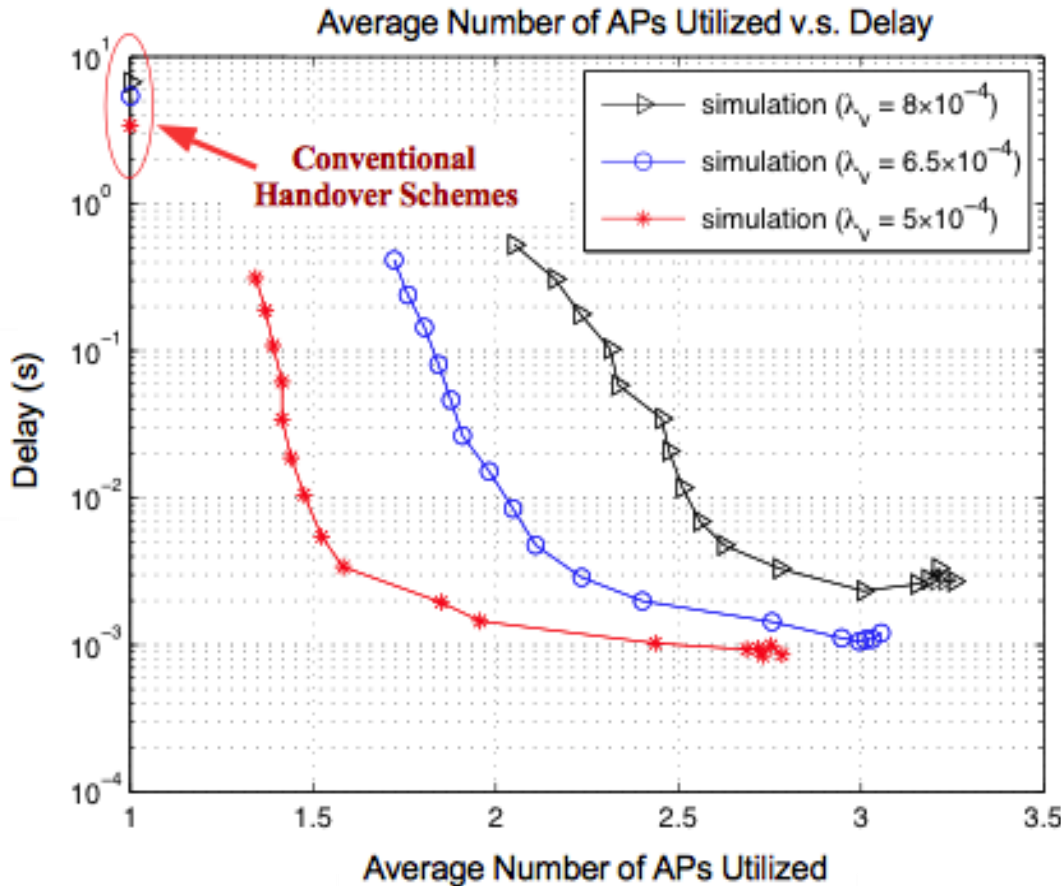
AP



K.-C. Chen, T. Zhang, R.D. Gitlin, G. Fettweis, “Ultra-Low Latency Mobile Networking”, to appear in the *IEEE Network Magazine*.

NEW ROAD TOWARD URLLC IN MOBILE NETWORKING

Ultra-Low Networking Latency is Possible by Open-Loop Wireless Communication and Proactive network association

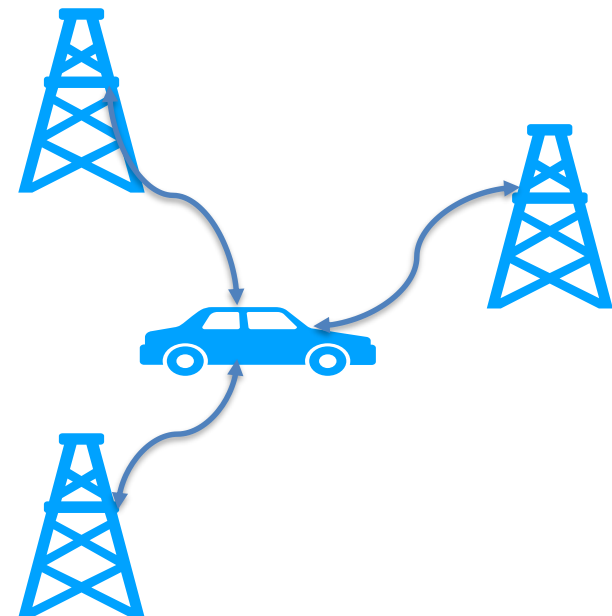
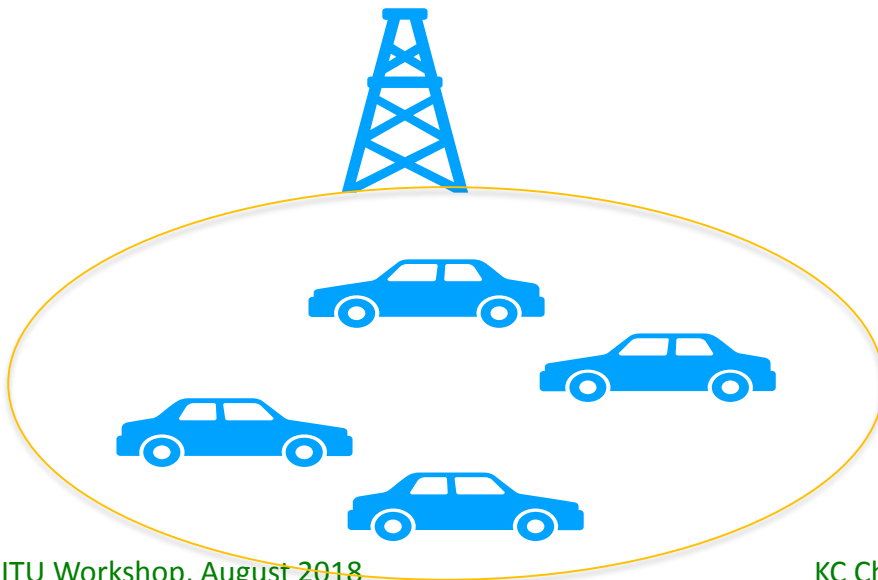


Delay performance of the conventional handover schemes, that each IMM is served by only one AP, and the proposed stochastic network optimization procedure are shown for different densities of randomly distributed IMM, represented by λ_v .

[S.-C. Hung, K.-C. Chen, IEEE T-MC, early access]

Concept of Virtual Cell

- Traditionally, one BS serves multiple mobile stations
 - Edge of cell suffers low SINR and complex handover mechanism of closed-loop and centralized control
- Virtual Cell: one mobile station is served by multiple BSs via cooperative communication
 - Suitable for virtual networks
 - No clearly defined operation

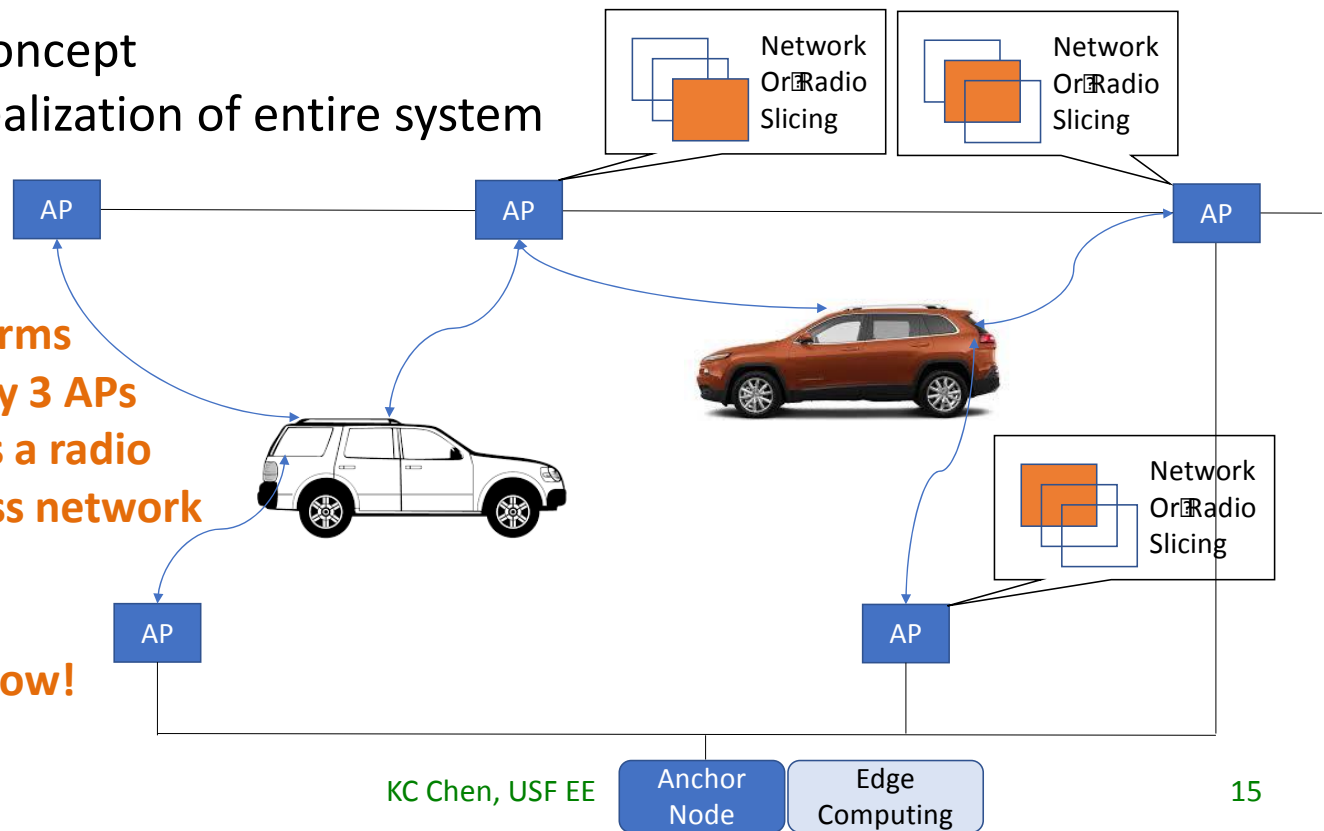


Vehicle-Centric Networking

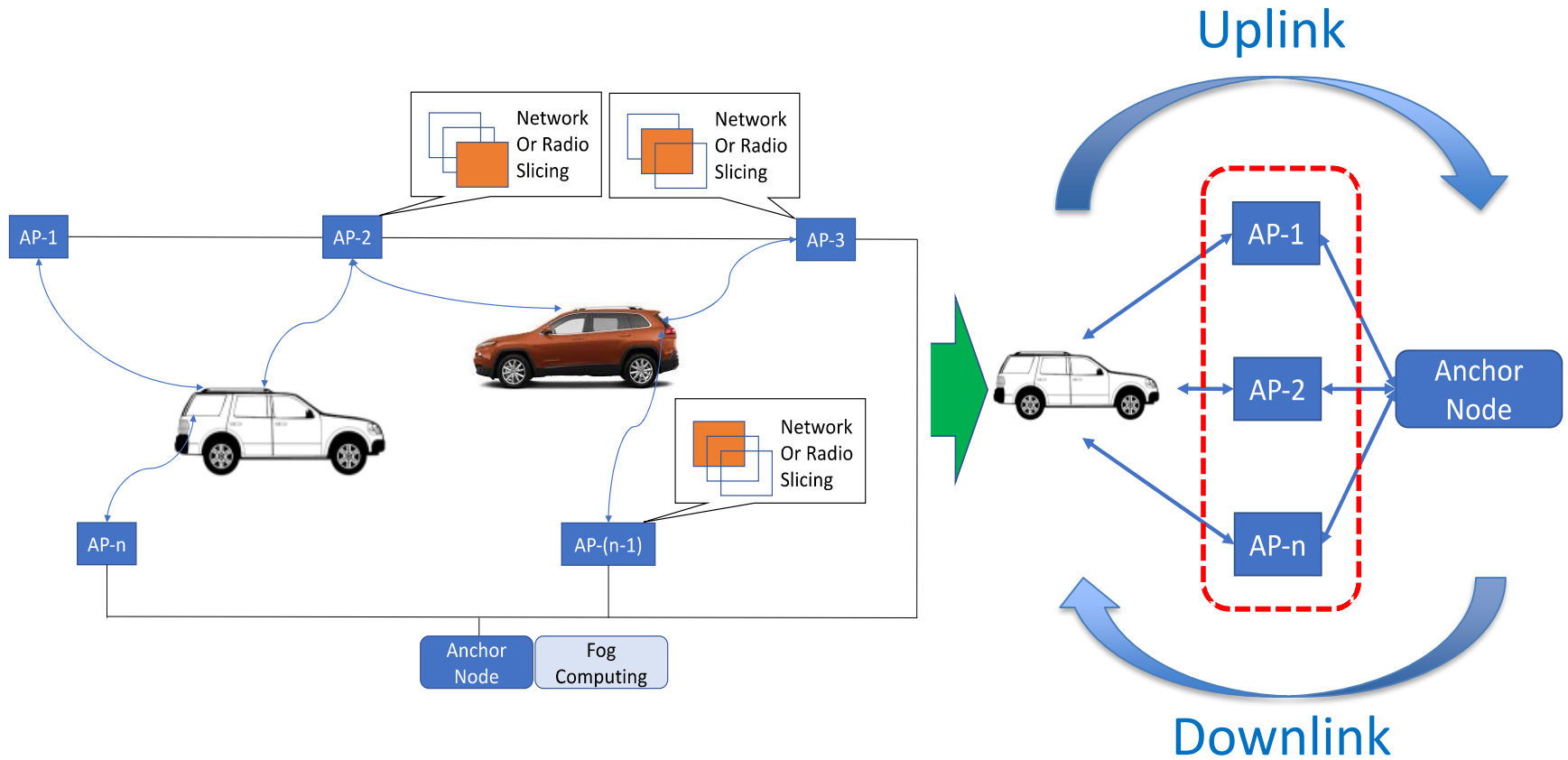
- Each vehicle is the center (i.e. the only mobile station) of a virtual cell. Multiple APs serve this virtual cell.
- Network/Radio slicing of virtual networking at each AP to serve the virtual cell.
 - Not a new concept
 - Lacking of realization of entire system

The orange vehicle forms a virtual cell served by 3 APs and each AP supports a radio slice of virtual wireless network to this virtual cell.

SDN/NFV is needed now!

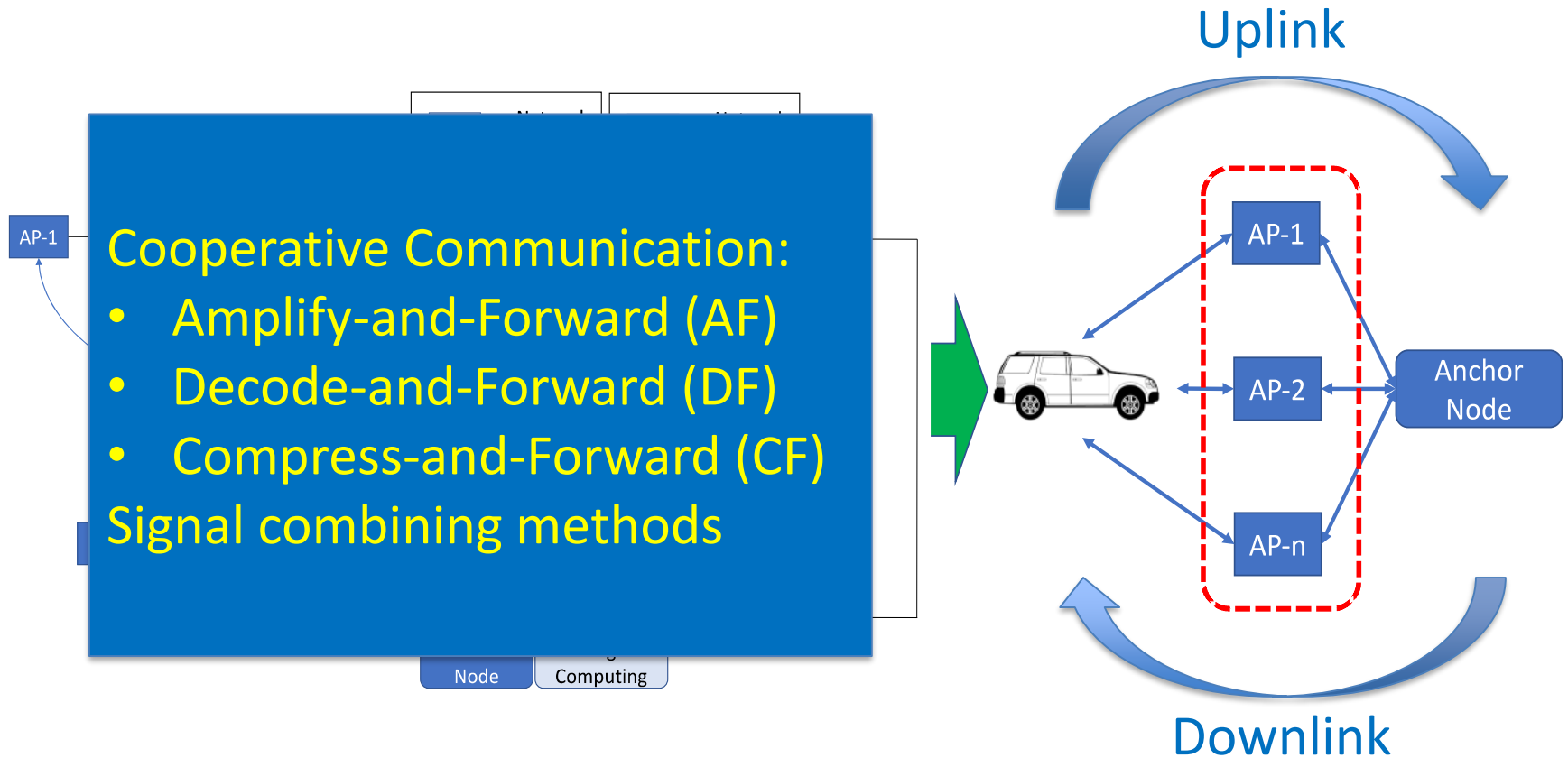


Multi-hop Multi-path Networking Between AV and AN



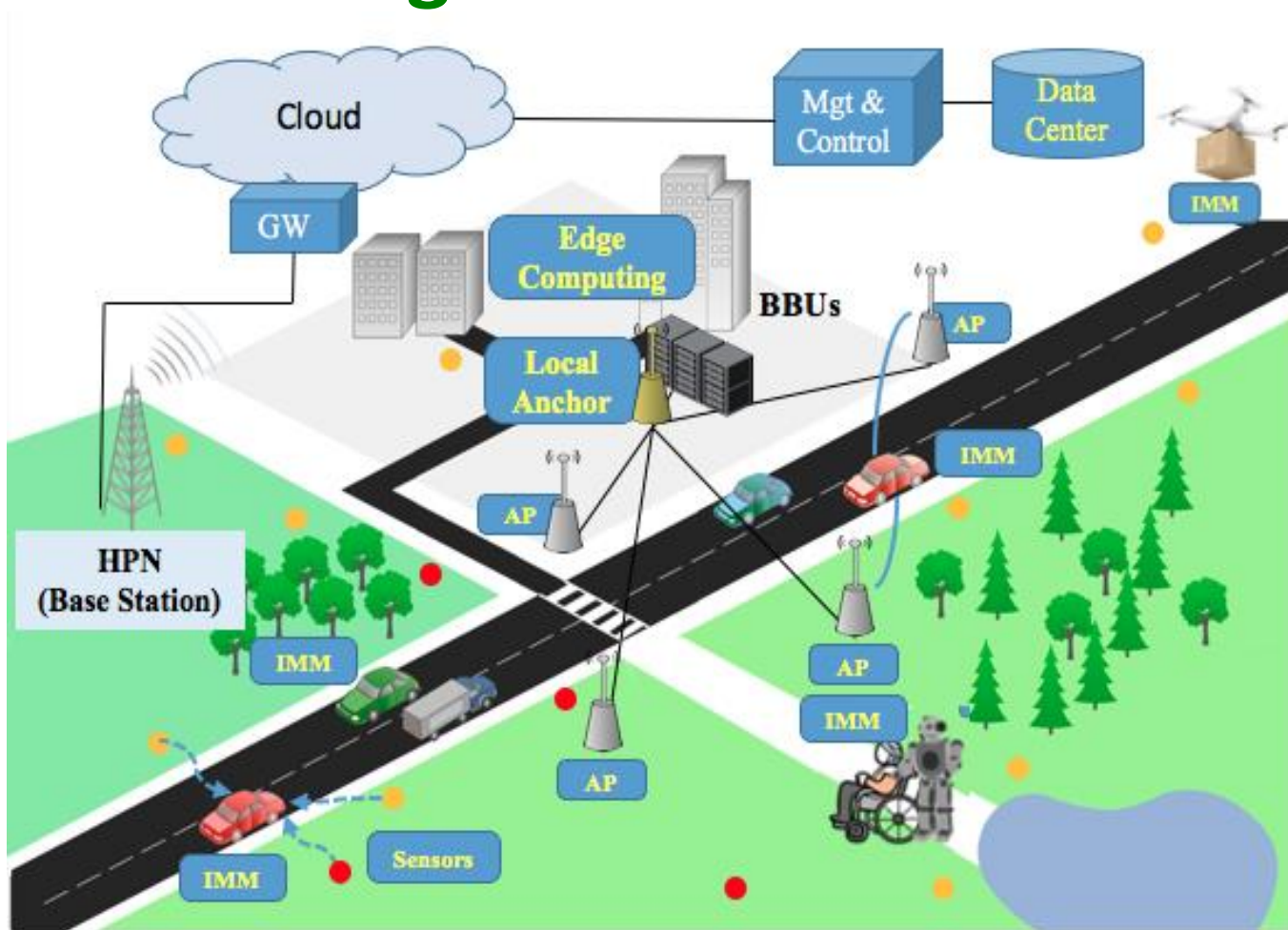
Please recall path-time codes can be exactly applied in multi-path two-hop networking!

Multi-hop Multi-path Networking Between AV and AN

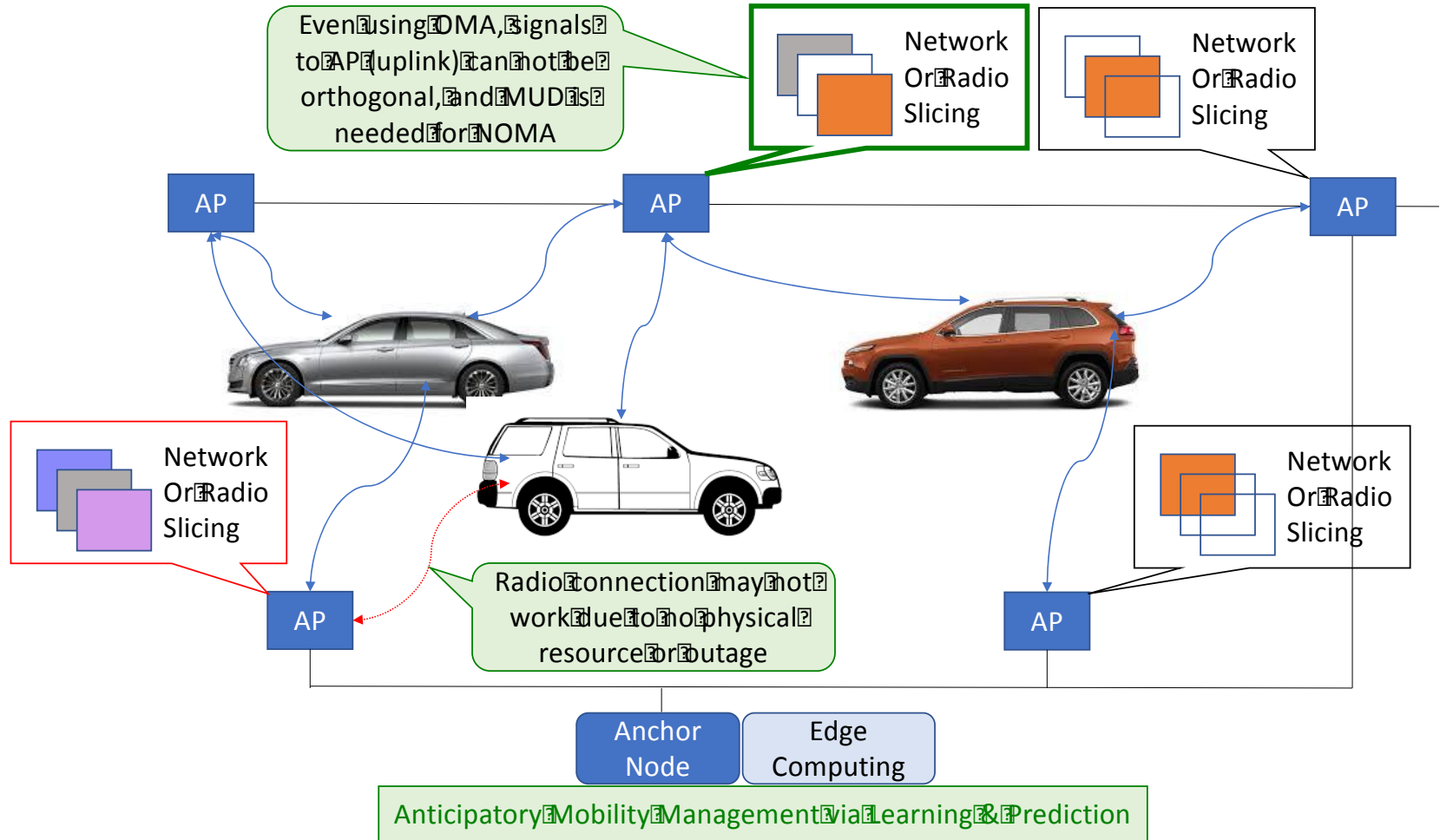


Please recall path-time codes can be exactly applied in multi-path two-hop networking!

Heterogeneous Computing & Networking for Intelligent Mobile Machines



Non-Orthogonal Multiple Access & Anticipatory Mobility Management & Open-Loop Error Control

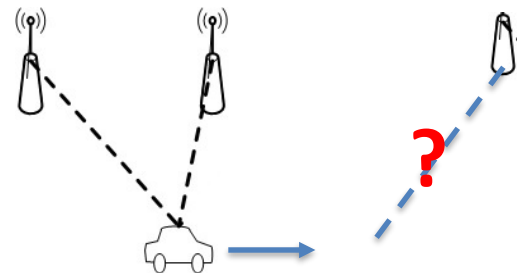


Anticipatory Mobility Management

- Technology Challenge:
 - Due to proactive network association, anchor node must determine candidate APs for ultra-low latency message delivery in the downlink
- Machine learning and artificial intelligence is not only preferred in such networking, and is a must. We name as **anticipatory mobility management**.
 - However, it is actually a new problem in machine learning to develop the optimal methodology, though many methods might work.

MACHINE LEARNING PARADIGMS FOR NEXT-GENERATION WIRELESS NETWORKS

CHUNXIAO JIANG, HAIJUN ZHANG, YONG REN, ZHU HAN,
KWANG-CHENG CHEN, AND LAJOS HANZO



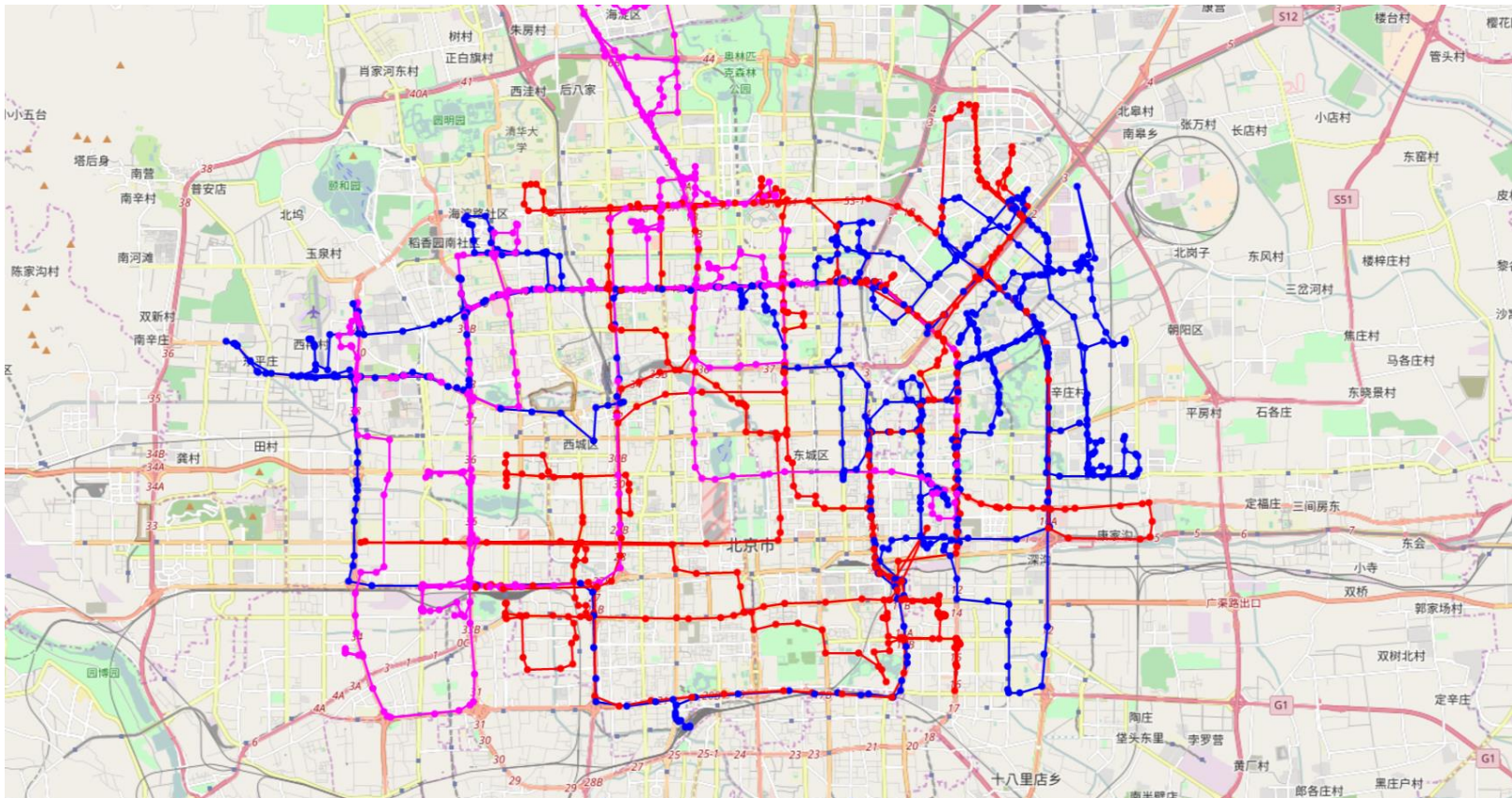
IEEE Wireless Communications. April, 2017

Machine Learning Applied to Networking

- Machine learning has been applied to wired and wireless networks for decades
 - Supervised Learning: EM optimization etc. to MIMO communications
 - Unsupervised Learning: Clustering in distributed networks and ad hoc networks
 - Reinforcement Learning: Markov decision process and thus POMDP and Q-learning, to optimize resource utilization in fixed and wireless networks
- Amount and diversity of data gives a new frontier of technology opportunities.
- Common ground between statistical communication theory and statistical learning theory

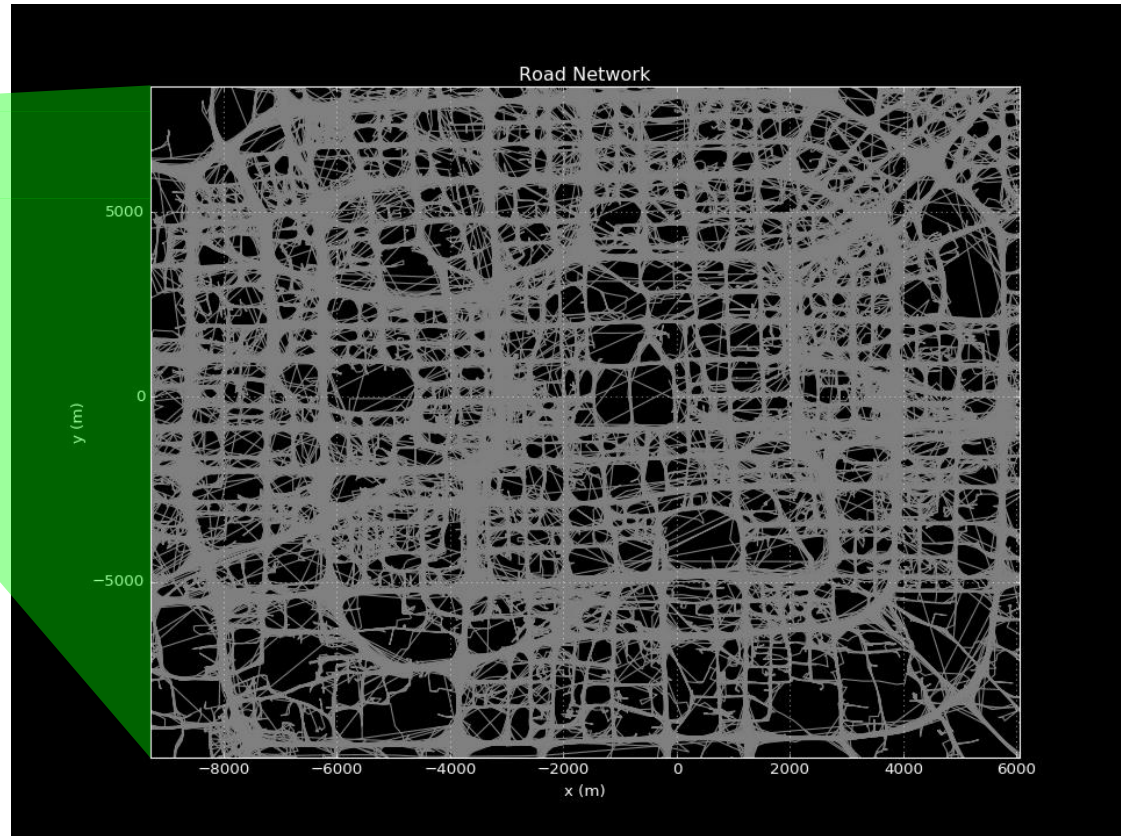
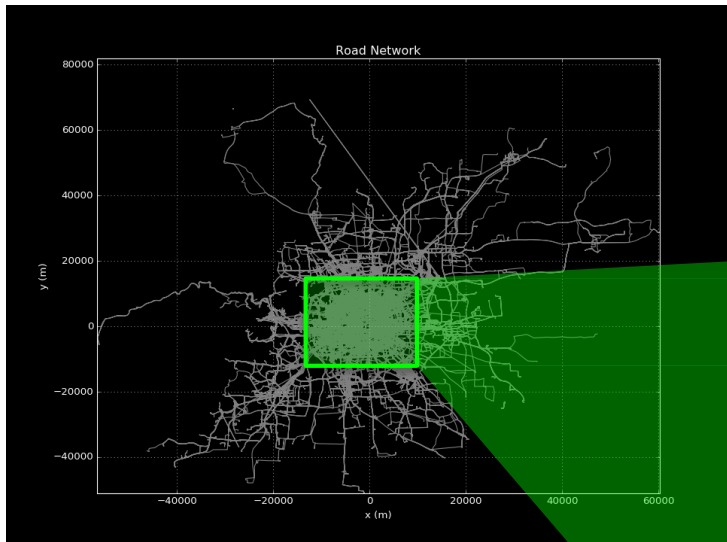
Big Data Analytics

Taxi data might be most similar to mission-oriented behavior of autonomous vehicles. We consider a dataset of more than 12,000 taxis in Beijing for a month.

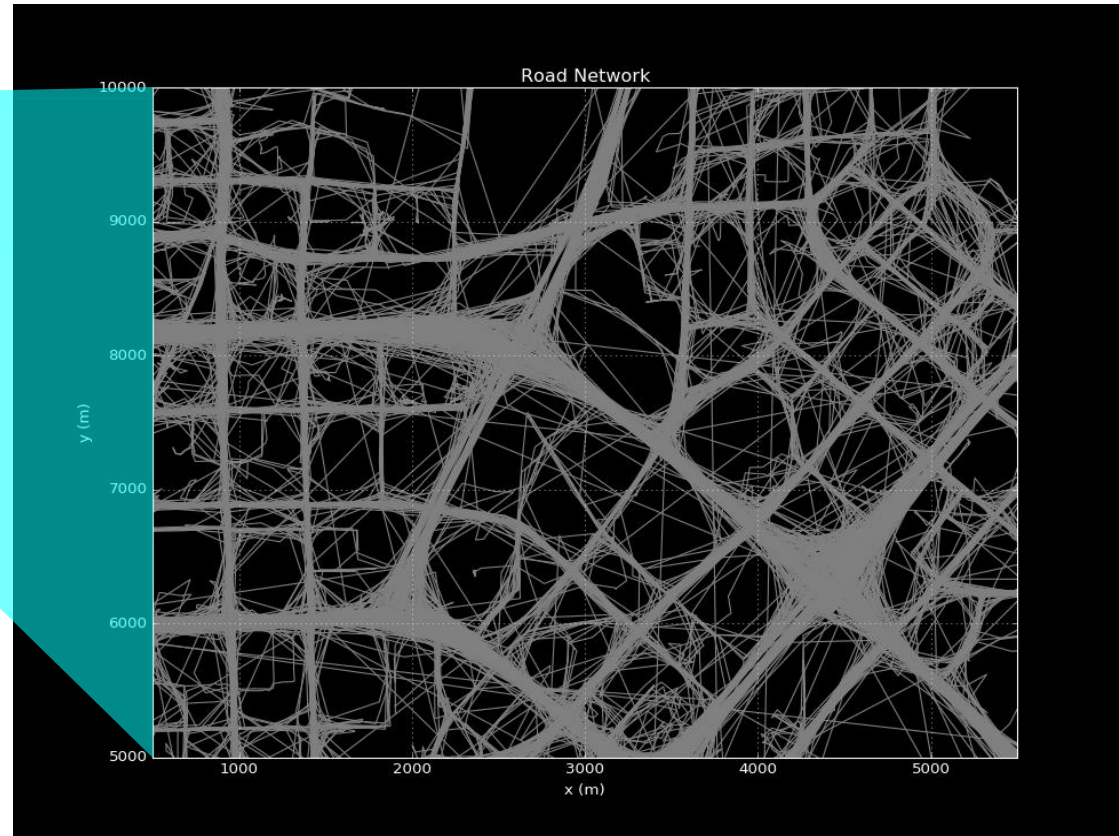
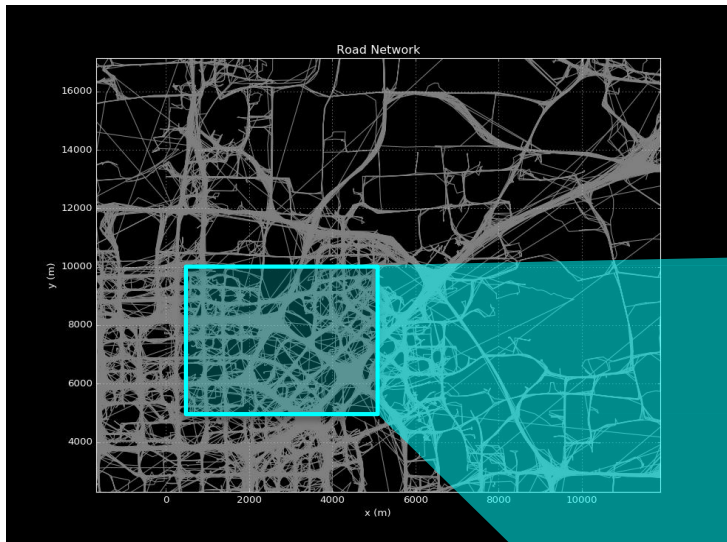


Trajectories for 3 different vehicles over a 24h period

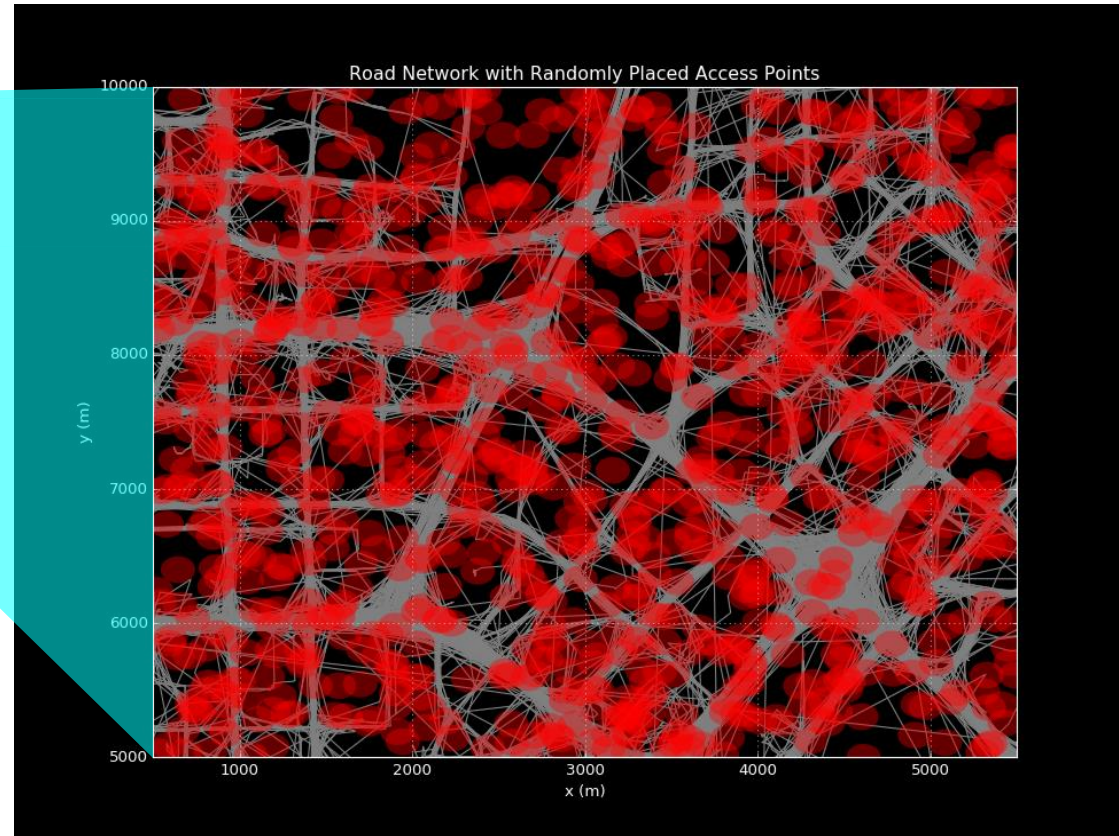
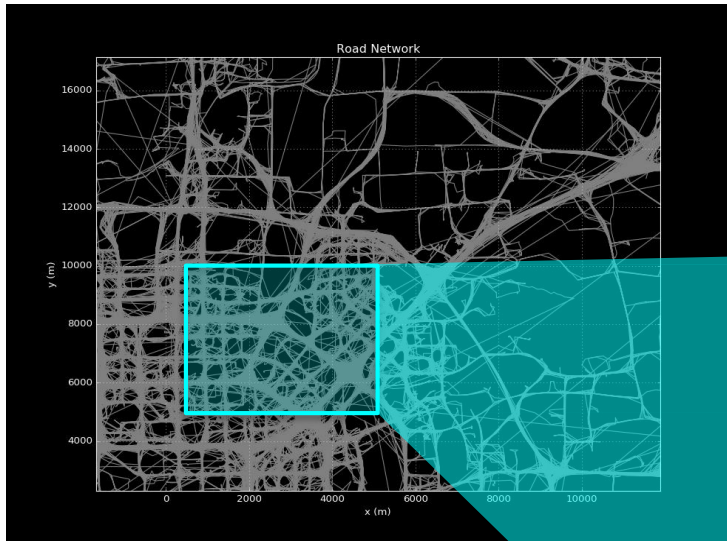
Zooming onto the City with All Trajectories



One Representative Region of Interest

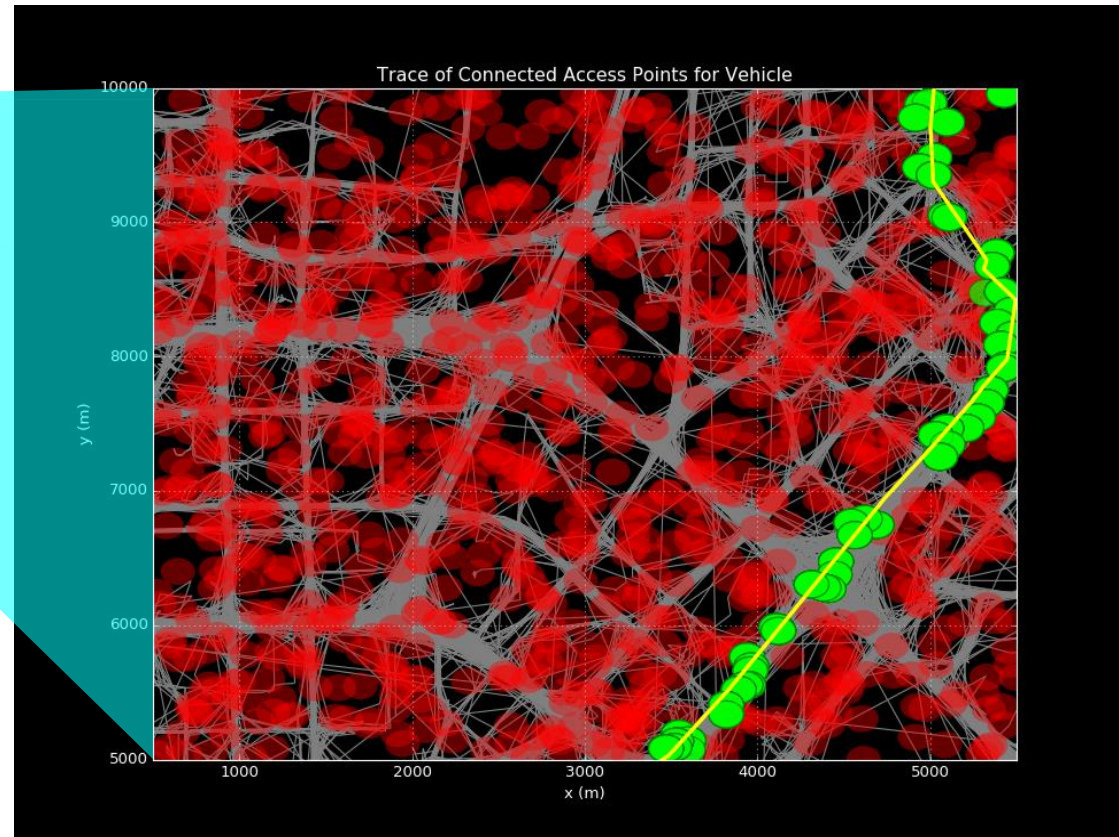
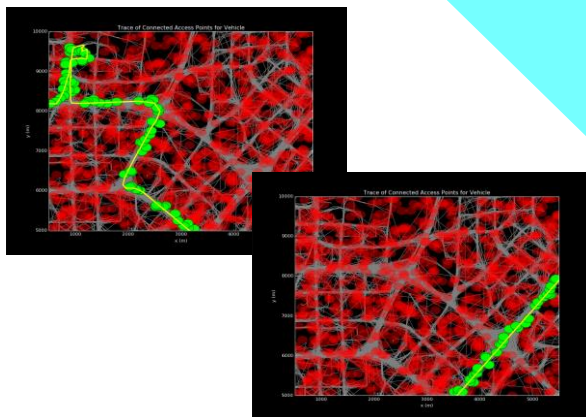
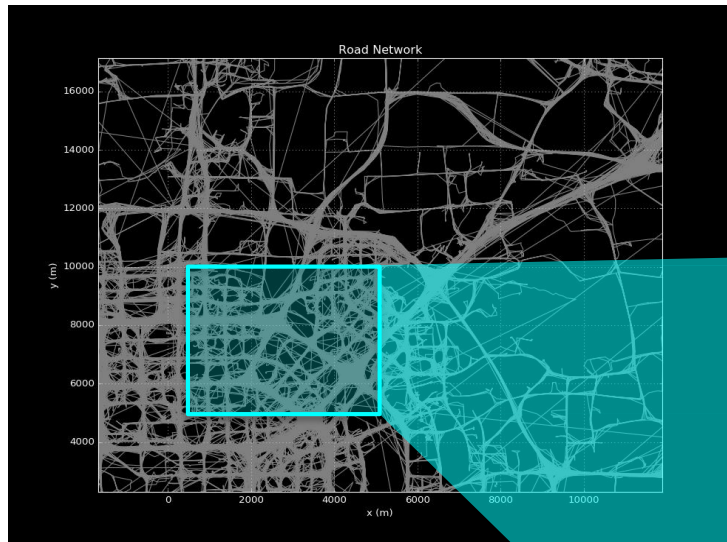


Randomly Deployed APs

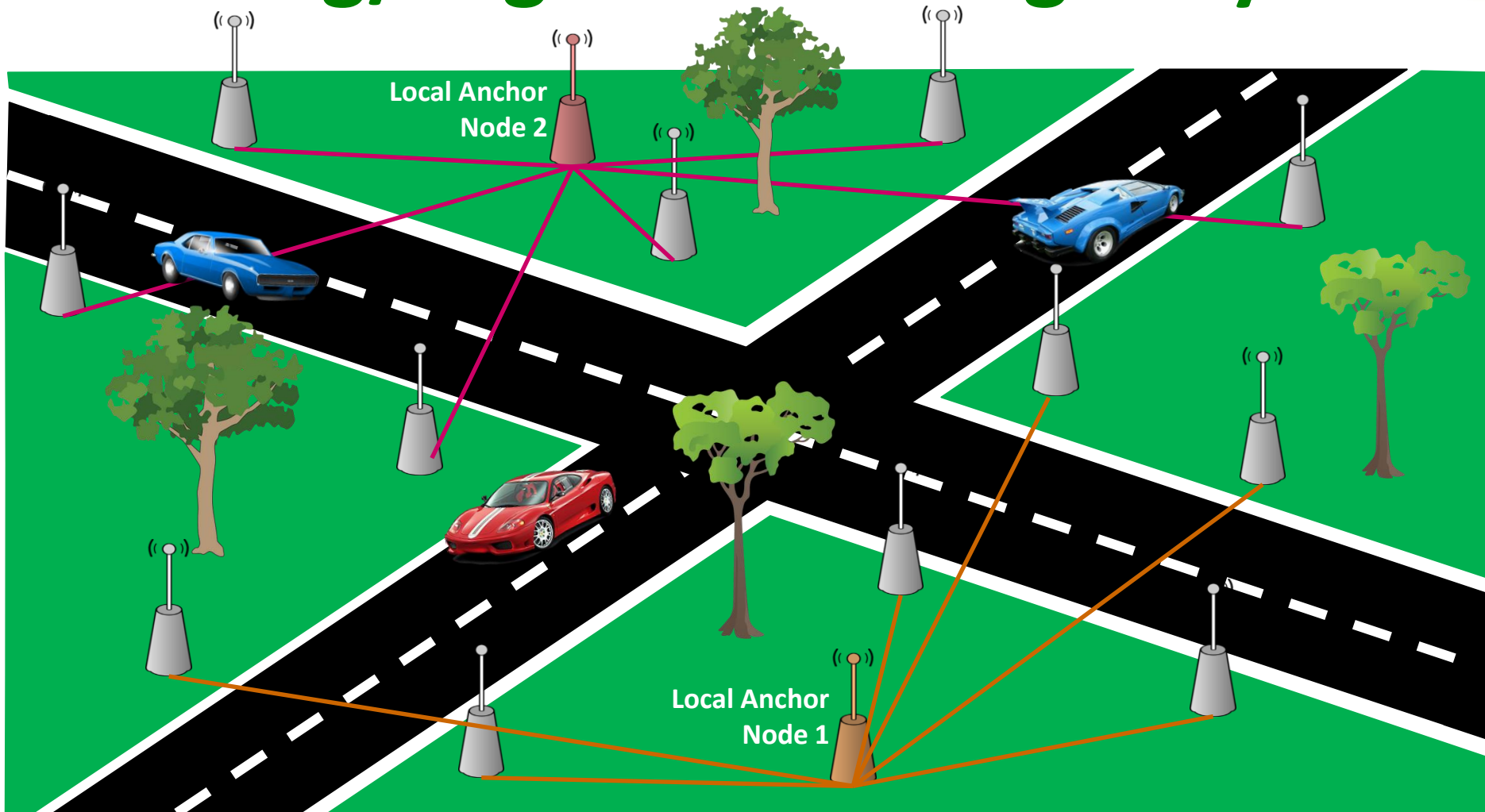


Randomly deployed APs stand for the worst case scenario

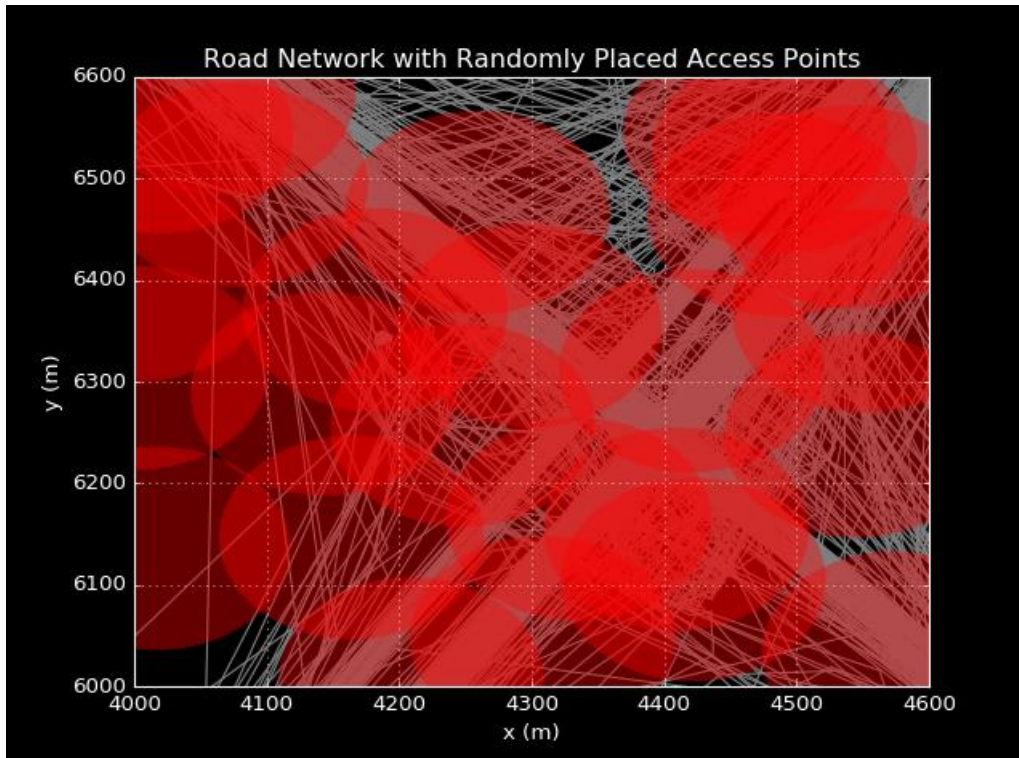
Illustration of Connections to a Vehicle



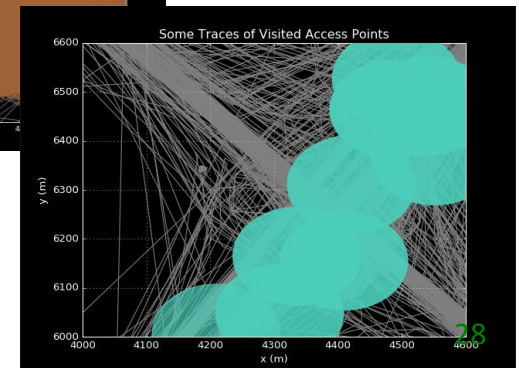
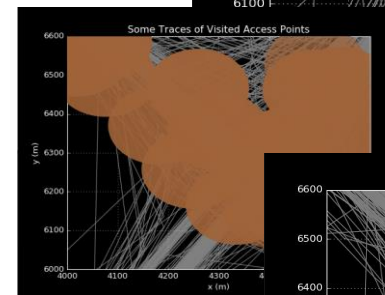
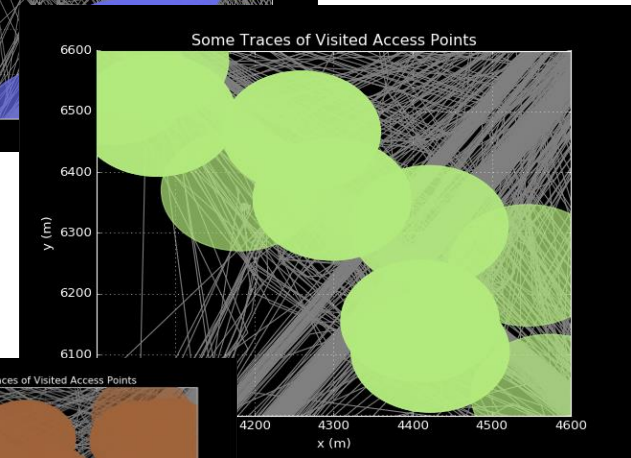
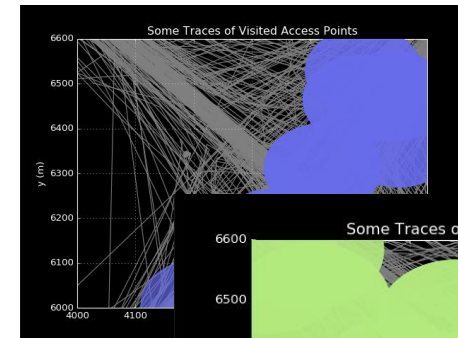
Fog/Edge Networking Only



Microscopic View



25 Access Points within 600m x 600m square area
in average



Prediction on Connected APs

- n Access Points (APs) associated with a particular Anchor Node (AN)
 - This is NOT a traditional vehicular mobility prediction
- Connections for a particular vehicle at a particular time instance given by **connection vector**
 - e.g. [0 1 0 0 0 1 0 1] denotes connections to APs 2, 6 and 8
- Predict the next connection vector given m previous connection vectors
 - For $m = 3, n = 8$

$$\mathbf{X} \left\{ \begin{array}{l} 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0 \\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1 \\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0 \end{array} \right.$$

$$\mathbf{Y} = 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1$$

Problem Formation: Prediction of APs via Big Vehicular Data

Each virtual cell can proactively associate to K APs.

- It associates with K_{max} APs of the strongest SINR (or other signaling to indicate suitability) if more than K_{max} APs are in radio range.
- It associates with K APs, if $K \leq K_{max}$ APs are in radio range.

Suppose each AP has an ID. Given the rule of association, we could obtain time series representation by defining AP association vector as

$$X(t) = [X^{(1)}(t)X^{(2)}(t)\dots X^{(d)}(t)] \in \{0, 1\}^d$$
$$X^{(i)}(t) = \begin{cases} 1, & \text{if the } i^{\text{th}} \text{ AP is connected} \\ 0, & \text{if the } i^{\text{th}} \text{ AP is not connected} \end{cases} \quad (1)$$

Problem. *Considering one single vehicle driving on roads, given a series of time t_1, \dots, t_n and the corresponding AP association vector $X(t_1), \dots, X(t_n)$ of the vehicle, predict $X(t_{n+1})$ for $t_{n+1} > t_n$.*

Edge Networking

Representation of
Knowledge

Prediction

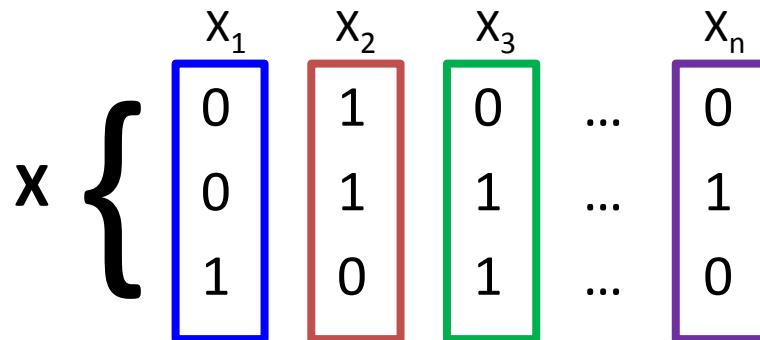
(trying NOT to use GPS data
due to accuracy and privacy)

Naïve Bayesian Approach: Benchmark

- Each AP treated independently $\mathbf{Y} = [Y_1 Y_2 \dots Y_n]$

$$\frac{P(Y_i = 1|\mathbf{X})}{P(Y_i = 0|\mathbf{X})} = \frac{P(\mathbf{X}|Y_i = 1)P(Y_i = 1)}{P(\mathbf{X}|Y_i = 0)P(Y_i = 0)}$$

- Assume independence over the APs in \mathbf{X} as well

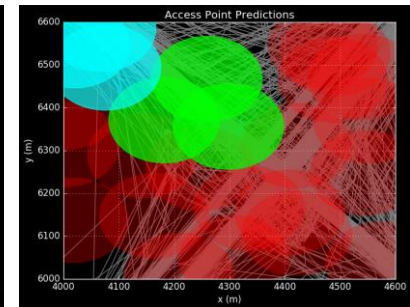
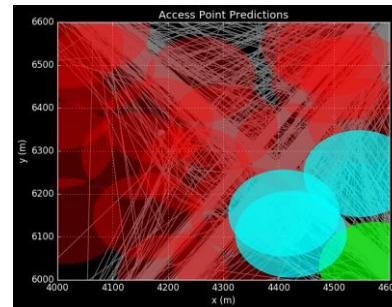
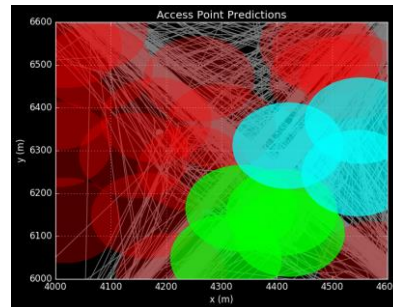
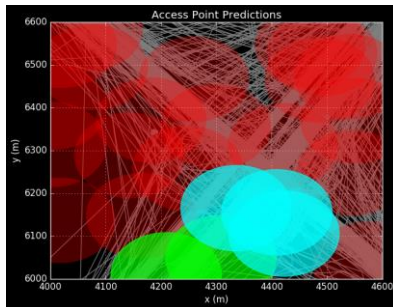


- $P(\mathbf{X}|Y_i) = P(X_1 = "001"|Y_i)P(X_1 = "110"|Y_i) \dots P(X_1 = "010"|Y_i)$

Benchmark Performance

- Simulation with a maximum of 3 simultaneous connections

Actually Connected # APs	Correctly Predicted # of APs			
	0	1	2	3
1	81.81%	18.18%	-	-
2	3.47%	29.14%	67.5%	-
3	0.53%	4.73%	17.78%	76.94%



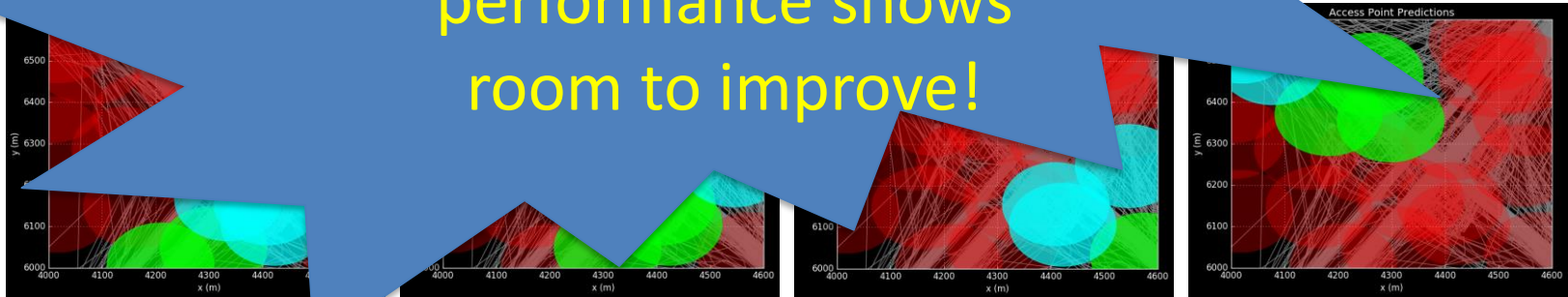
Light Green – Past Connections, Light Blue – Predicted Connections

Benchmark Performance

- Simulation with a maximum of 3 simultaneous connections

Actually Connected # APs	Correctly Predicted # of APs		
	0	2	3
1			
2			-
3			76.94%

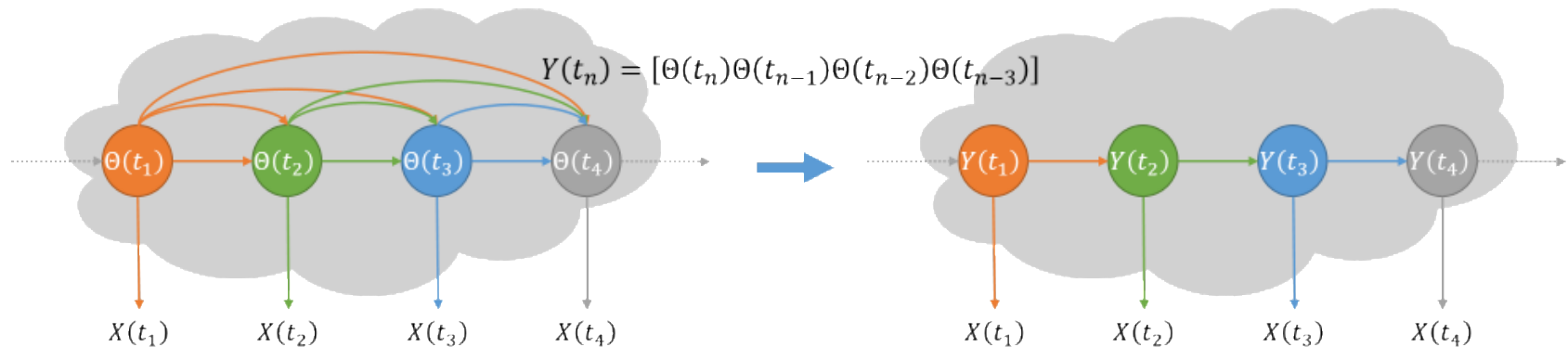
The idea of Anticipatory Mobility Management works and benchmark performance shows room to improve!



Light Green – Past connections, Light Blue – Predicted Connections

Learning Process

- The recursive process at t_n involves three stages:
 - with observing a new association vector, derive the corresponding location and update the transition probability between locations
 - based on the derived locations and the transition probability, predict the location at time t_{n+1} .
 - according to the predicted location and vectors of APs, obtain the prediction for the association vector at t_{n+1} .



Recursive Bayesian Estimation

- We intend to estimate $P_{X_{n+1}|X_{1:n}}(x_{n+1}|x_{1:n})$
- Taking location Θ into consideration,

$$P_{X_{n+1}|\rightarrow_{n+1}}(x_{n+1}|\sqrt{n+1})P_{\rightarrow_{n+1}|X_{1:n}}(\sqrt{n+1}|x_{1:n})$$

- To form the basis of optimal Bayesian estimation

Based on the Markovian property, there is a recurrence relation between the posterior belief, $P_{\rightarrow_n|X_{1:n}}(\sqrt{n}|x_{1:n})$, and the prior belief, $P_{\rightarrow_n|X_{1:n-1}}(\sqrt{n}|x_{1:n-1})$, satisfying

$$\begin{aligned}
 & P_{\rightarrow_n|X_{1:n-1}}(\sqrt{n}|x_{1:n-1}) \\
 = & \int P_{\rightarrow_n|\rightarrow_{n-1}}(\sqrt{n}|\sqrt{n-1})P_{\rightarrow_{n-1}|X_{1:n-1}}(\sqrt{n-1}|x_{1:n-1})d\sqrt{n-1}
 \end{aligned} \tag{6}$$

and

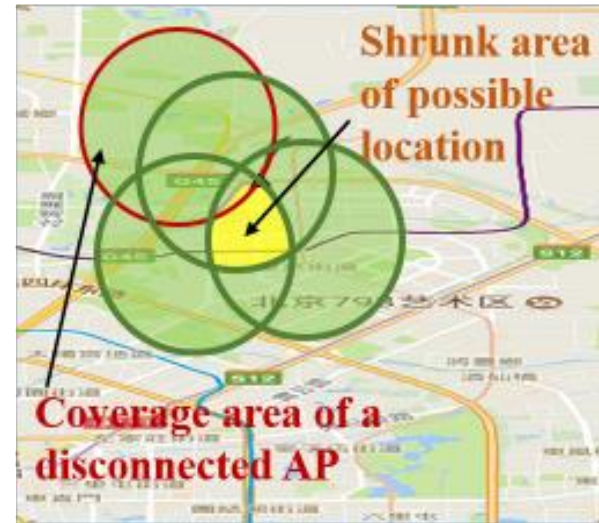
$$P_{\rightarrow_n|X_{1:n}}(\sqrt{n}|x_{1:n}) = \mathbb{R} \frac{P_{X_n|\rightarrow_n}(x_n|\sqrt{n})P_{\rightarrow_n|X_{1:n-1}}(\sqrt{n}|x_{1:n-1})}{\int P_{X_n|\rightarrow_n}(x_n|\sqrt{n})P_{\rightarrow_n|X_{1:n-1}}(\sqrt{n}|x_{1:n-1})d\sqrt{n}} \tag{7}$$

Posterior Belief of the Location

Inferring location from association vector



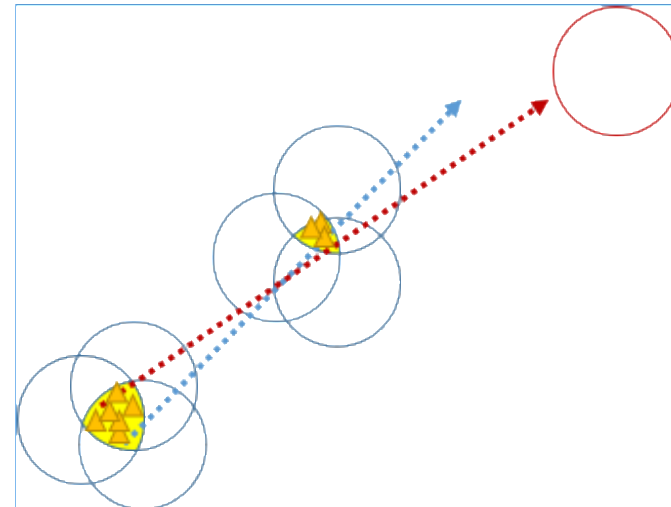
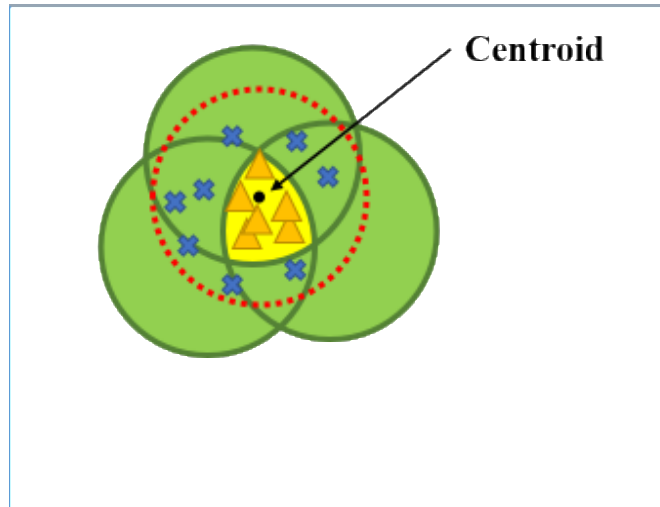
(a) Possible positioning basing on information from connected APs



(b) Possible positioning with additional information from disconnected APs

In case (a), with considering connected APs, the enclosed area that the vehicle possibly locates is the intersection of the coverage area of the connected APs (yellow colored). In case (b), in addition to the connected APs, some of the disconnected APs also provide extra information about the possible location.

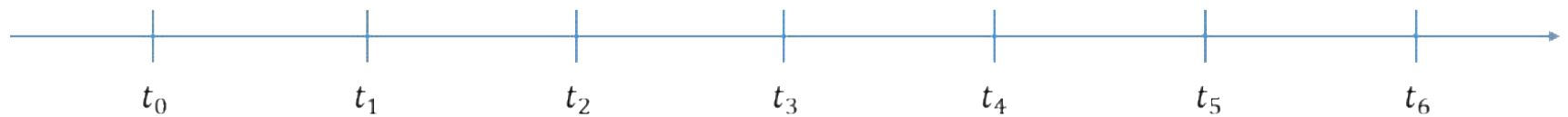
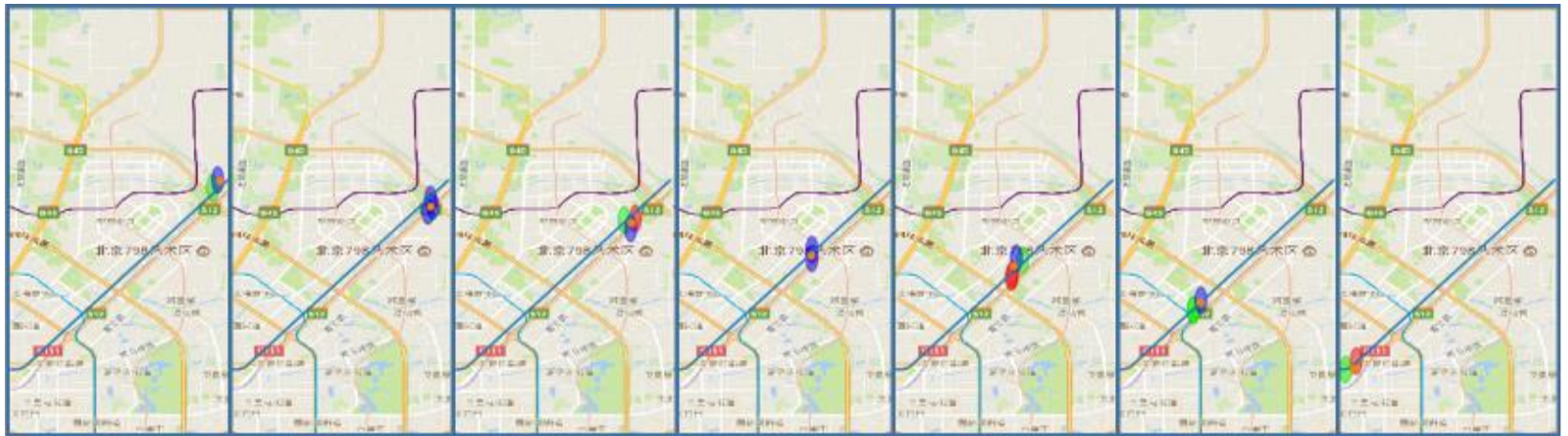
Further Location Prediction



Utilizing Monte Carlo method for obtaining a set of points for representing the area of possible location and making prediction accordingly. For randomly spread points, we retain the ones in the yellow colored area only for representing the possible location. Based on these points, we may estimate the moving velocity accordingly, and make use of it to predict the future location

$$\hat{v}(t_{n+1}) \hat{=} \hat{v}(t_n) + v \cdot (t_{n+1} - t_n)$$

Illustration of Prediction



An example of AP prediction with the ordering of time corresponding to the frames. The ellipses in the figure indicate the coverage area of APs. The blue colored ones means that we predict the AP to be connected and it is indeed connected, the green ones means that we predict the AP to be disconnected but it is actually connected, and the red ones means that we predict the AP to be connected but it is actually disconnected. The orange marker in the figure represents the predicted location of the car.

Satisfactory Prediction Is Achievable

Method of Calculating Optimal Velocity				
# of connected APs	# of correctly predicted APs			
	0	1	2	3
1	26.68%	73.32%		
2	8.49%	13.68%	77.83%	
3	1.25%	2.12%	2.74%	93.89%

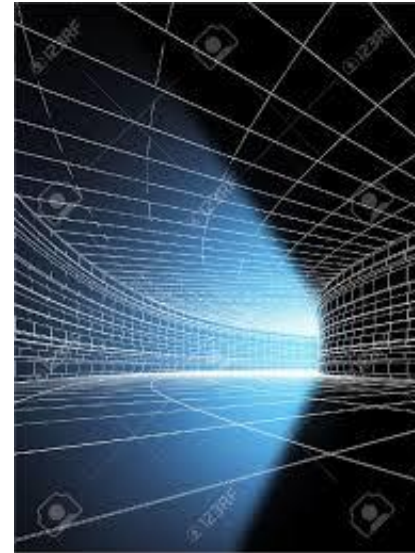
Please note the APs are randomly deployed, and the case of one correctly predicted AP can work.

Optimal deployment of APs based on map can improve a lot.

[IEEE ICC 2018]

Machine Learning Enables Mobile Networking

- AMM is possible by the aid of data analytics, using real-time learning, location and velocity estimation, and previous connected APs.
 - Downlink ultra-low latency networking and proactive network association is realizable by AMM, and therefore ultra-low latency mobile networking can be realized
 - For many cases, simple facilitation of machine learning works well, particularly for real-time operation
 - The key is to identify a proper machine learning technique(s) to facilitate the goal, based on the property of available data
- Future enhancements of AMM
 - AP deployment based on the map, particularly avoid no coverage and poor coverage by only one AP.
 - More methods like random forest
 - Deep learning assists AN and transfer learning to the edge.



In the 2018 IEEE Globecom, we further demonstrate that wireless networking can renovate machine learning and thus enhance AI systems.

**IT IS A LONG JOURNEY ... BUT WE CAN
SEE THE LIGHT FROM END OF TUNNEL**

Thank you for your attention & Questions?

Statistical Learning

Statistical learning theory was introduced in the late 1960's and remained a mathematical statistical analysis as the problem of function estimation from a given collection of data. Particularly since the invention of widely applied support vector machines (SVMs) in the mid-1990's, statistical learning theory has been shown to be useful to develop new learning algorithms.



Vladimir N. Vapnik is the main contributor of Vapnik-Chervonenkis (VC) theory of statistical learning theory and a co-inventor of support vector machines. He was born in the Soviet Union in 1936 and received the PhD in statistics at the Institute of Control Sciences in 1964. In 1990, Dr. Vapnik moved to the US and joined the Adaptive Systems Department, AT&T Bell Labs., where he and his colleagues developed support vector machines. He was inducted into the National Academy of Engineering in 2006.

Machine Learning

- The canonical model of the learning conducted in a general statistical framework by minimizing the expected loss using observed data, which consists of three components
 - A generator of random vectors x , obtained independently from a fixed but generally unknown distribution $P(x)$
 - A supervisor who returns an output vector y for each input vector x , according to a conditional distribution $P(y|x)$ that is fixed but again generally unknown
 - A learning machine capable of selecting the one from a set of functions $f(x, \alpha)$, $\alpha \in \mathfrak{A}$, to predict the supervisor's response in the "best" possible way
- **supervised learning**
 - A teacher tells the results of learning to be good or not
- **unsupervised learning**

Supervised Learning

- To identify the criterion of selecting best possible approximation to the supervisor's response, we intend to measure the discrepancy $D(y, f(x, \alpha))$ between the response $f(x, \alpha)$ by the learning machine and the response of the supervisor to a given input x , where such a measure is also known as *loss* or *distortion*.

$$R(\alpha) = \int D(y, f(x, \alpha)) dP(x, y)$$

Types of Learning Problems

- Pattern Recognition:** This class of learning problems is also known as *classification* in literature. Let the supervisor's output y take on the discrete values, say binary valued as $y \in \{0,1\}$. Then, $f(x, \alpha), \alpha \in \mathfrak{A}$, become a set of *indicator* functions. One example of loss function is defined as

$$D(y, f(x, \alpha)) = \begin{cases} 0, & \text{if } y = f(x, \alpha) \\ 1, & \text{if } y \neq f(x, \alpha) \end{cases}$$

- Regression and Estimation:** Suppose the supervisor answers real-valued y and $f(x, \alpha), \alpha \in \mathfrak{A}$ is a set of real functions which contains the optimal *regression function*

$$f(x, \alpha_{opt}) = \int y dP(y|x)$$

In case $f(x, \alpha)$ belongs to L_2 functional, the optimal regression function is to minimize the risk functional of $D(y, f(x, \alpha)) = [y - f(x, \alpha)]^2$

- Density Estimation:** For a set of densities $p(x, \alpha), \alpha \in \mathfrak{A}$, density estimation considers to minimize the risk functional of

$$D[p(x, \alpha)] = -\log p(x, \alpha)$$

Linear Prediction

Given input data $\mathbb{X} = (X_1, \dots, X_r)$, the predictor Y is obtained through the linear model

$$\hat{Y} = \hat{b}_0 + \sum_{j=1}^r X_j \hat{b}_j$$

\hat{b}_0 is known as the *bias*, which usually can be included in \mathbb{X} . Then, if $\hat{\mathbb{b}} = (\hat{b}_1, \dots, \hat{b}_r)$,

$$\hat{Y} = \mathbb{X}^T \hat{\mathbb{b}}$$

As described in earlier chapters, there are many ways to define measure of performance. The most common measure is *least square-error*. In machine learning, such an approach is to identify coefficients \mathbb{b} to minimize *residual sum of squares*

$$\text{RSS}(\mathbb{b}) = \sum_{i=1}^N (y_i - \mathbf{x}_i^T \mathbb{b})^2$$

$\text{RSS}(\mathbb{b})$ is a quadratic function of parameters and hence minimum always exists but may not be unique.

$$\text{RSS}(\mathbb{b}) = (\mathbf{y} - \mathbb{X}\mathbb{b})^T (\mathbf{y} - \mathbb{X}\mathbb{b})$$

\mathbf{y} is an N -vector of outputs from training set. Differentiating with respect to \mathbb{b} , we get the normal equations

$$\hat{\mathbb{b}} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbf{y}$$

Advanced Regression

Remark (Ridge Regression): Ridge regression shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized RSS.

$$\hat{b}^{ridge} = \arg \min_b \left\{ \sum_{i=1}^N \left(y_i - b_0 - \sum_{j=1}^r x_{ij} b_j \right)^2 + \lambda \sum_{j=1}^r b_j^2 \right\}$$

where $\lambda > 0$ controls the scale of shrinkage, or the convergence speed. The method can be also used in neural networks as *weight decay*.

Remark (LASSO): The *least absolute shrinkage and selection operator* (LASSO) method is evolved from Ridge regression using L_p norm.

$$\hat{b}^{lasso} = \arg \min_b \left\{ \sum_{i=1}^N \left(y_i - b_0 - \sum_{j=1}^r x_{ij} b_j \right)^2 + \lambda \sum_{j=1}^r \|b_j\|_p \right\}$$

which has been widely applied in modern statistical data analysis.

K-Nearest-Neighbor

- Being widely used in pattern classification, the nearest-neighbor method utilizes those observations in the training set \mathcal{T} closest in the input space to form \hat{Y} . The k -nearest neighbor is defined as

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample set. We need a metric to define “closest” and we usually assume Euclidean distance.

- We seek a function $f(X)$ to predict Y given input values from X . The expected prediction error (based on Euclidean distance) is

$$\text{EPE}(f) = \mathbb{E}\{[Y - f(X)]^2\}$$

- The predictor is $f(x) = \mathbb{E}\{Y|X = x\}$ In other words, the predictor is just the conditional mean (or conditional expectation), which is also known as the *regression function*.
- On the other hand, the nearest-neighbor method actually attempts to directly implement this concept by training data. At each point x , we might ask for the average of all such y_i 's with input $x_i = x$. As there are typically one observation at any point x ,

$$\hat{f}(x) = \text{AVG}[y_i | x_i \in N_k(x)]$$

$\text{AVG}(\cdot)$ denotes the operation of average, and $N_k(x)$ is the neighborhood containing the k points in \mathcal{T} closest to x .