ReDiCom: Resilient Communication for First Responders in Disaster Management

Project Members

Prof. K. K. Ramakrishnan – University of California, Riverside

Prof. Murat Yuksel – University of Central Florida

Prof. Hulya Seferoglu – University of Illinois at Chicago

Dr. Jiachen Chen – WINLAB, Rutgers University

DISCLAIMER

This presentation was produced by guest speaker(s) and presented at the National Institute of Standards and Technology's 2019 Public Safety Broadband Stakeholder Meeting. The contents of this presentation do not necessarily reflect the views or policies of the National Institute of Standards and Technology or the U.S. Government.

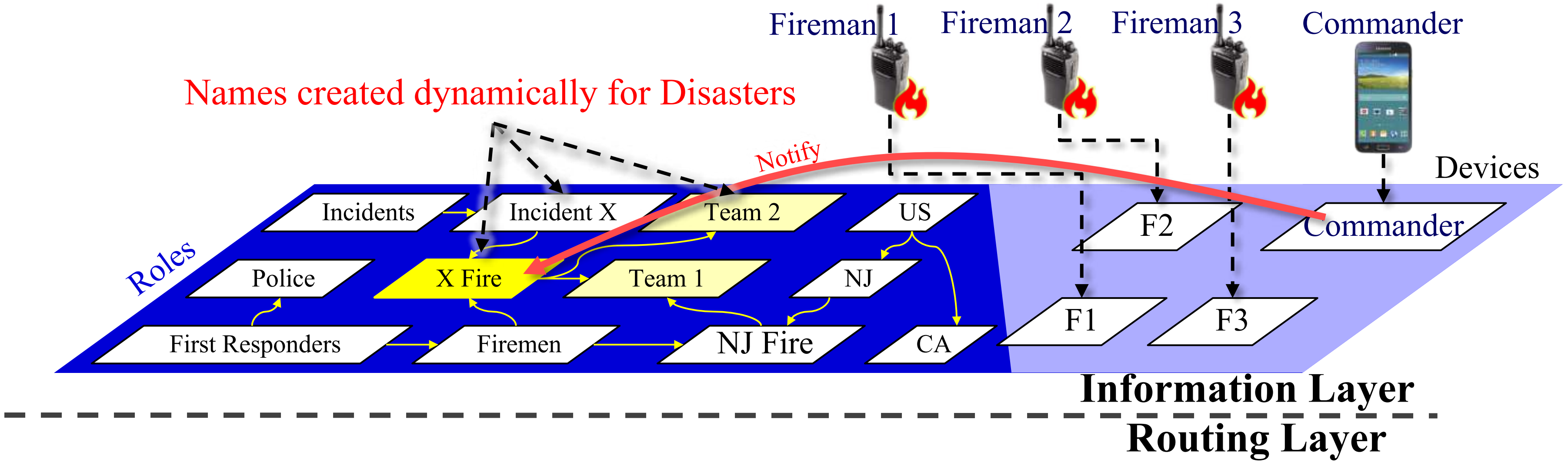
Posted with permission

Importance of Communication for Disaster Management

- Communication is key to improving outcomes in the aftermath of a disaster
- However, it is in the aftermath of a disaster that we are likely to face communication challenges:
  - Infrastructure may be impacted
  - Communication channels may be congested
- Keys to an effective response to a catastrophic incident:
  - Effective communication within and among dynamically formed first responder teams
    - Public safety teams comprising: law enforcement, health, emergency, transport and other special services, depending on the nature and scale of the emergency
  - Communication with stranded individuals and the public at large
- **<u>Project Objective:</u> A network architecture for information and communication resilience in disaster management.**

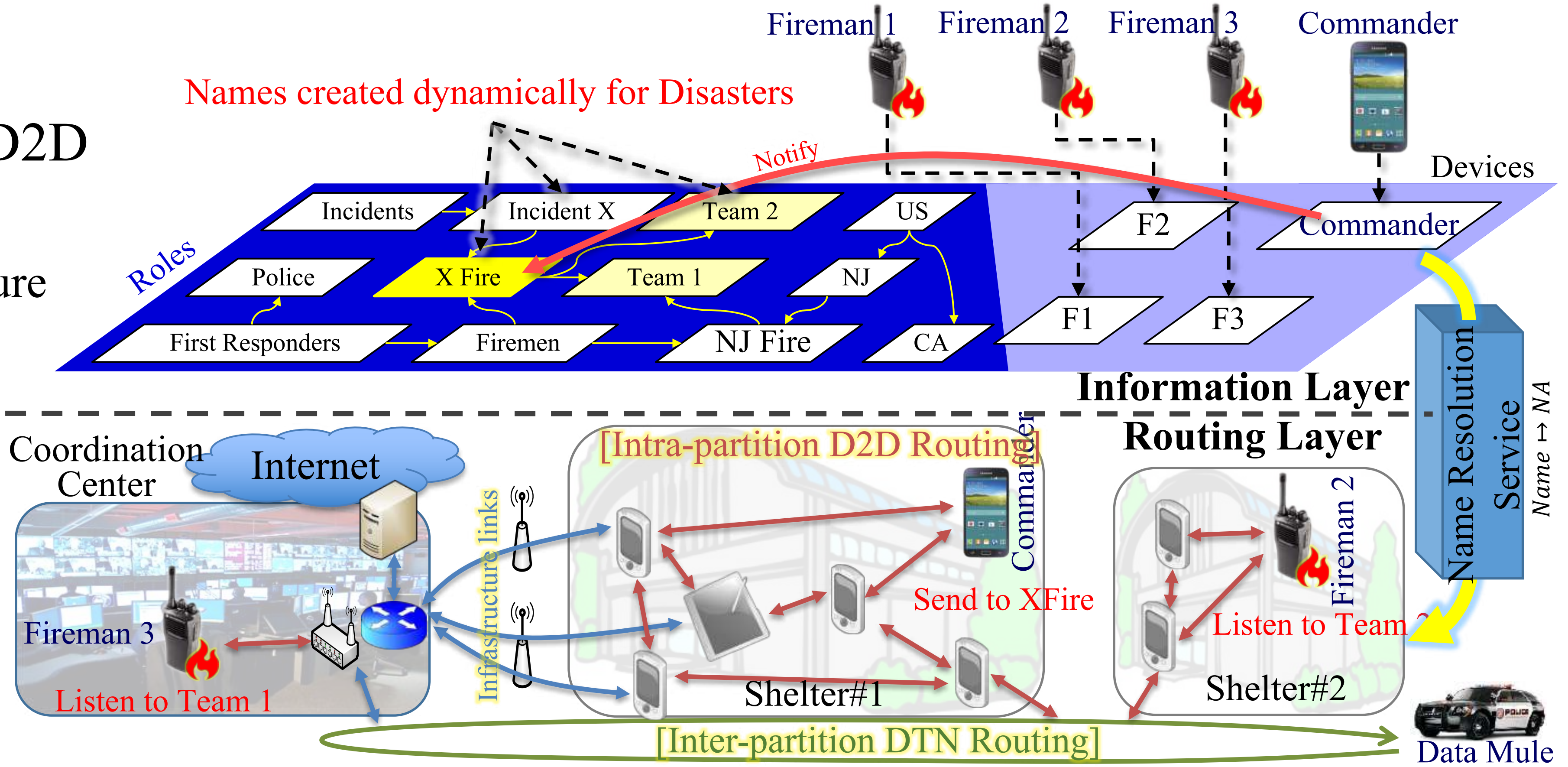
Proposed Architecture

- **Information Layer**
  - Facilitate communication among dynamically formed first-responder teams
  - Information-Centric (Role-Based) Communication
    - Communication based on dynamically created roles, rather than locations
- **Routing Layer**



Proposed Architecture

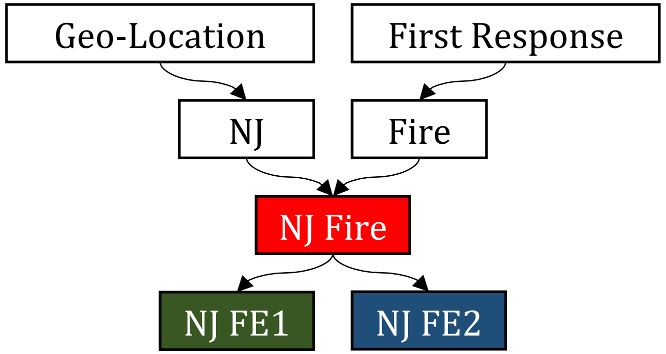
- **Information Layer**
  - Facilitate communication among dynamically formed first-responder teams
  - Information-Centric (Role-Based) Communication
    - Communication based on dynamically created roles, rather than locations
- **Routing Layer**
  - Intra- & inter-partition D2D communication
    - Resilient to the infrastructure disruptions
    - Smart mode-switching
  - Coding for robustness and resiliency
    - D2D computation
  - Content-based security



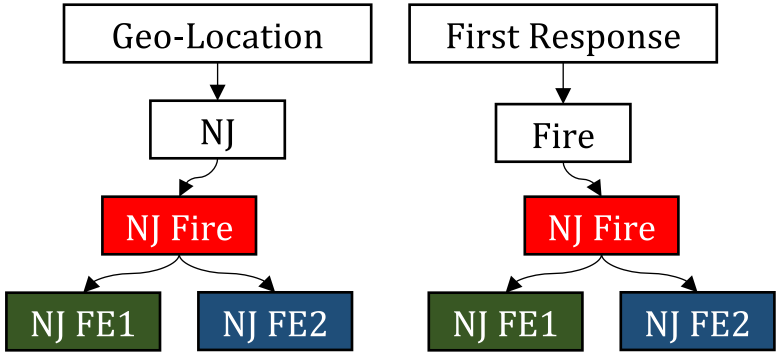
Namespace Design

- **Multi-dimensional**
  - E.g. `FireEngine1` has Time, Location and Department attributes (dimensions)
- **Graph structure**
  - More efficient than NDN-style strict hierarchy
- **Dynamic**
  - Edges (relations) pop in and out of existence
- **Publish/Subscribe service interface**
  - Support a publish/subscribe capability for users to share information
  - Multiple entities can publish to a name
  - Uses a shared multicast structure in network, using rendezvous points (RPs)

Graph Structure



Hierarchical Structure

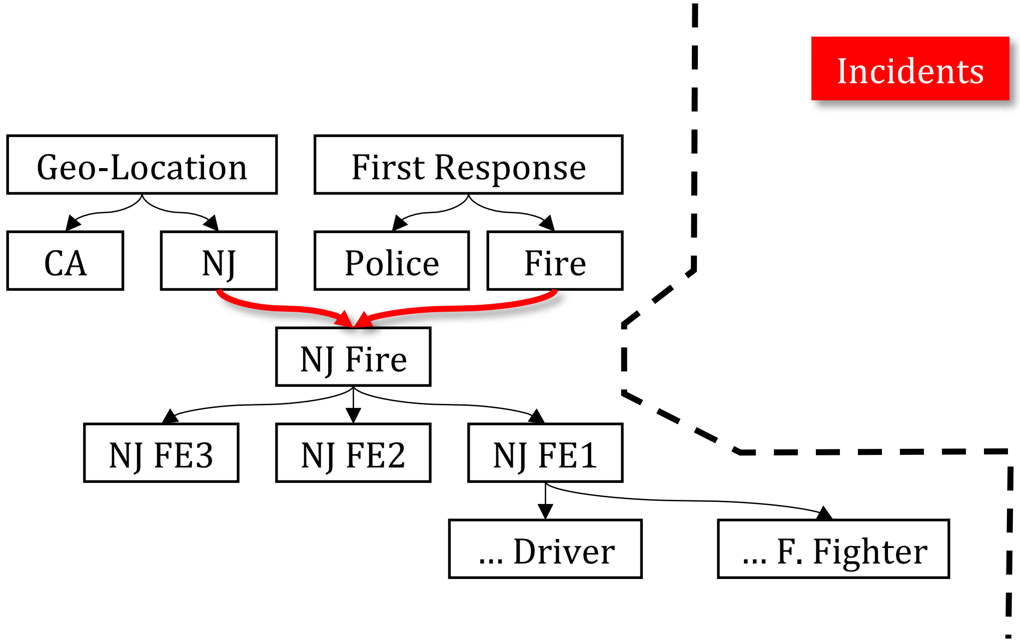


Hierarchical names:

```
/Geo-Location/NJ/NJ Fire
/First Response/Fire/NJ Fire
/Geo-Location/NJ/NJ Fire/NJ FE1
/First Response/Fire/NJ Fire/ NJ FE1
/Geo-Location/NJ/NJ Fire/NJ FE2
/First Response/Fire/NJ Fire/ NJ FE2
```

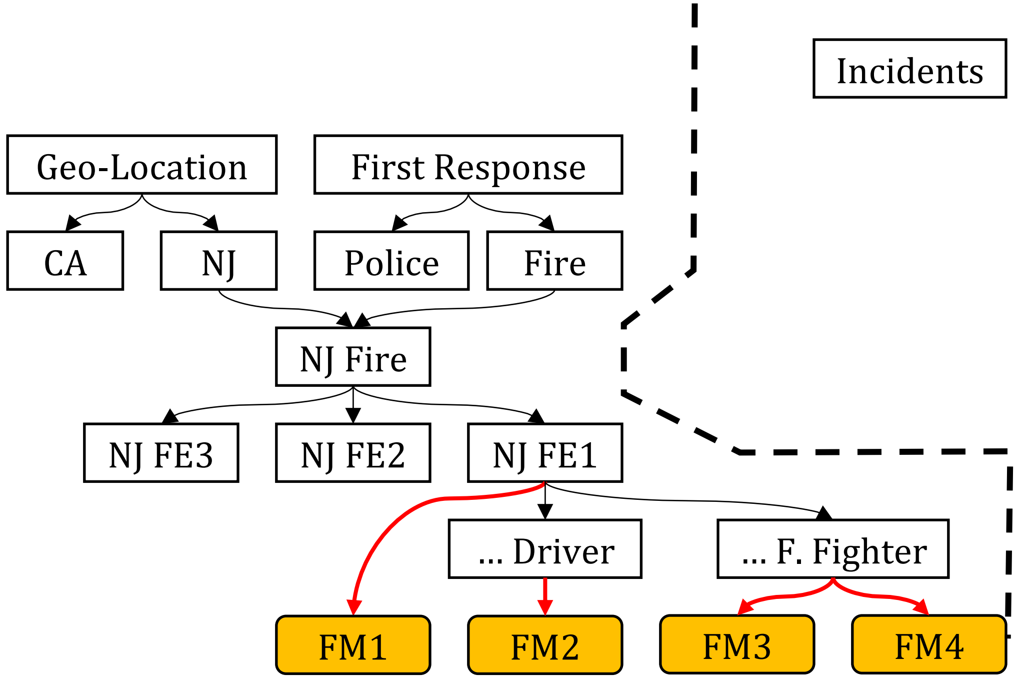
Improve Efficiency of Disaster Management by Graph-based Namespace

- Example namespace
  - Organizational structure: need information flow to members
    - Graph enables multiple dimensions (geo-location & functionality)
  - Incident place holder



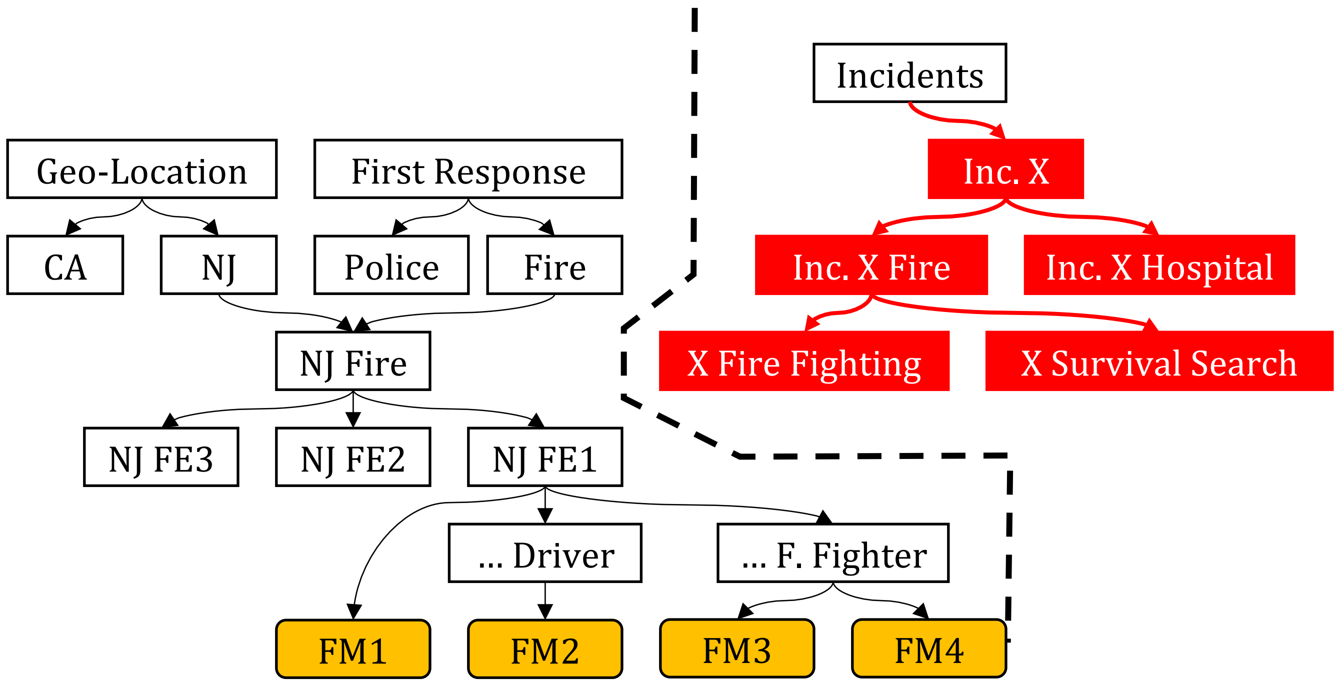
Improve Efficiency of Disaster Management by Graph-based Namespace

- Example namespace
  - Organizational structure: need information flow to members
    - Graph enables multiple dimensions (geo-location & functionality)
  - Incident place holder
- First responders instantiate roles



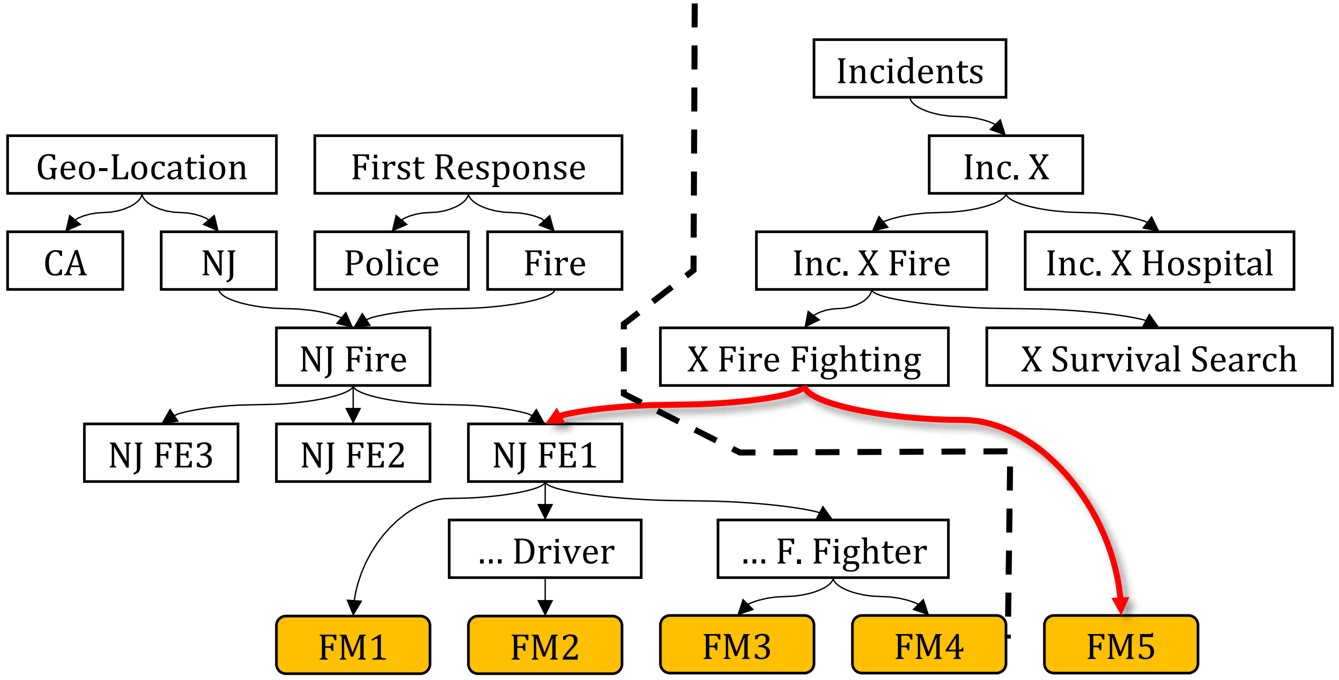
Improve Efficiency of Disaster Management by Graph-based Namespace

- Example namespace
  - Organizational structure: need information flow to members
    - Graph enables multiple dimensions (geo-location & functionality)
  - Incident place holder
- First responders instantiate roles
- Instantiate a disaster management template: preplanned namespaces



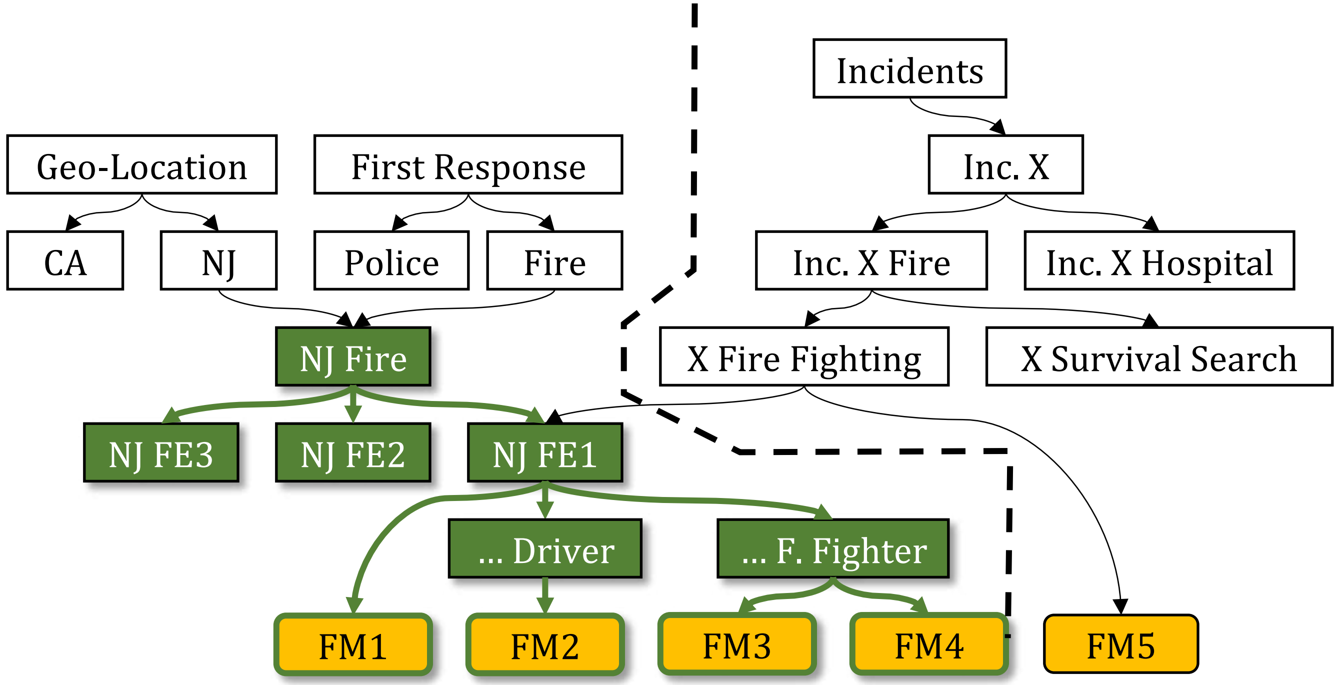
Improve Efficiency of Disaster Management by Graph-based Namespace

- Example namespace
  - Organizational structure: need information flow to members
    - Graph enables multiple dimensions (geo-location & functionality)
  - Incident place holder
- First responders instantiate roles
- Instantiate a disaster management template: preplanned namespaces
- Dispatch units to deal with functions in an incident



Improve Efficiency of Disaster Management by Graph-based Namespace

- Example namespace
  - Organizational structure: need information flow to members
    - Graph enables multiple dimensions (geo-location & functionality)
  - Incident place holder
- First responders instantiate roles
- Instantiate a disaster management template: preplanned namespaces
- Dispatch units to deal with functions in an incident
- Send messages to a role, e.g., “NJ Fire”



Improve Efficiency of Disaster Management by Graph-based Namespace

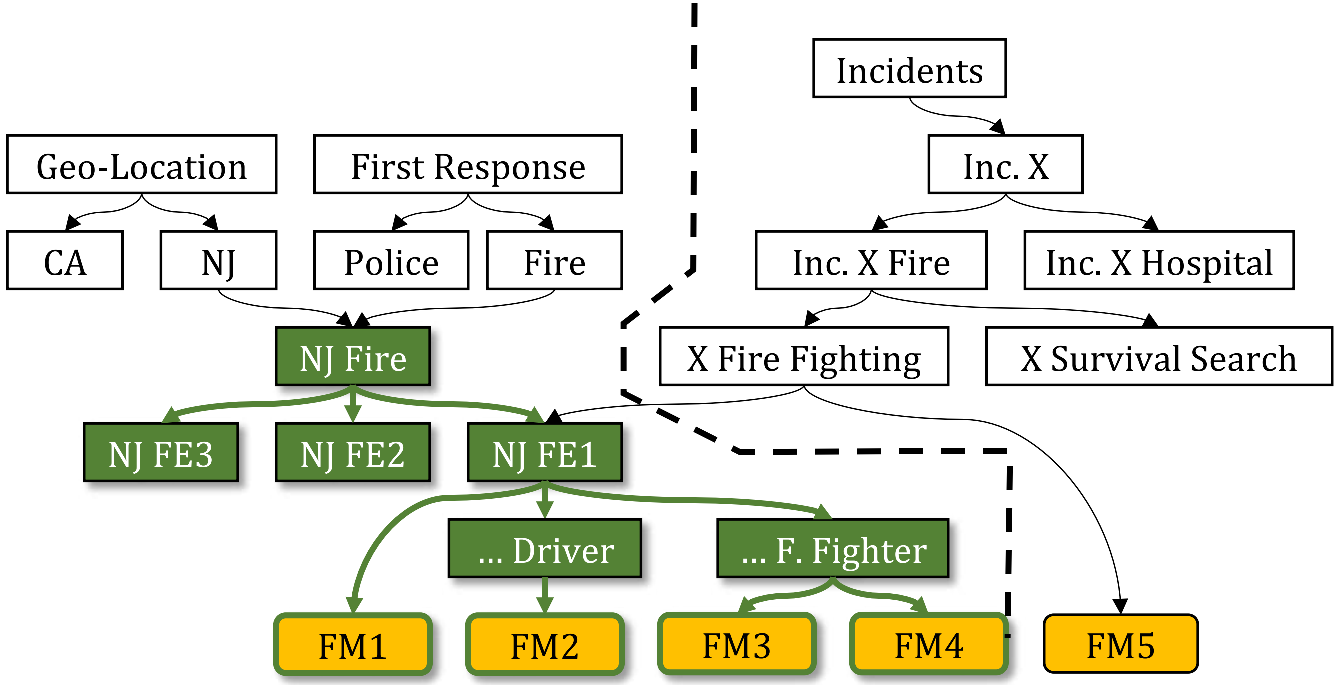
- Example namespace
  - Organizational structure: need information flow to members
    - Graph enables multiple dimensions (geo-location & functionality)
  - Incident place holder
- First responders instantiate roles
- Instantiate a disaster management template: preplanned namespaces
- Dispatch units to deal with functions in an incident
- Send messages to a role, e.g., “NJ Fire”

Need: Support a graph-based namespace in the network

Dynamic Nature of Namespace

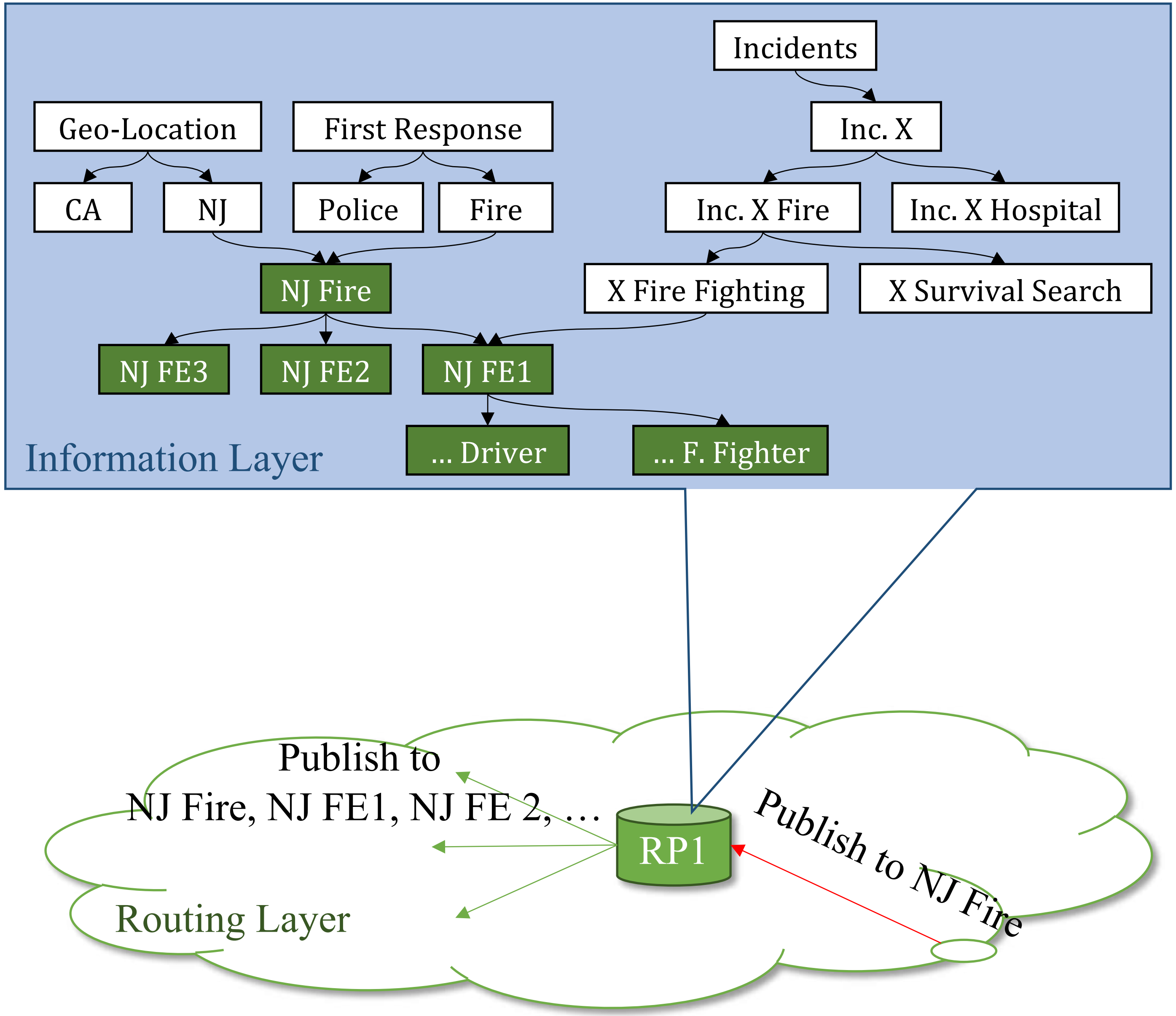
Dynamic installations of disaster namespaces

The namespace can evolve according to the situation



Solution Overview

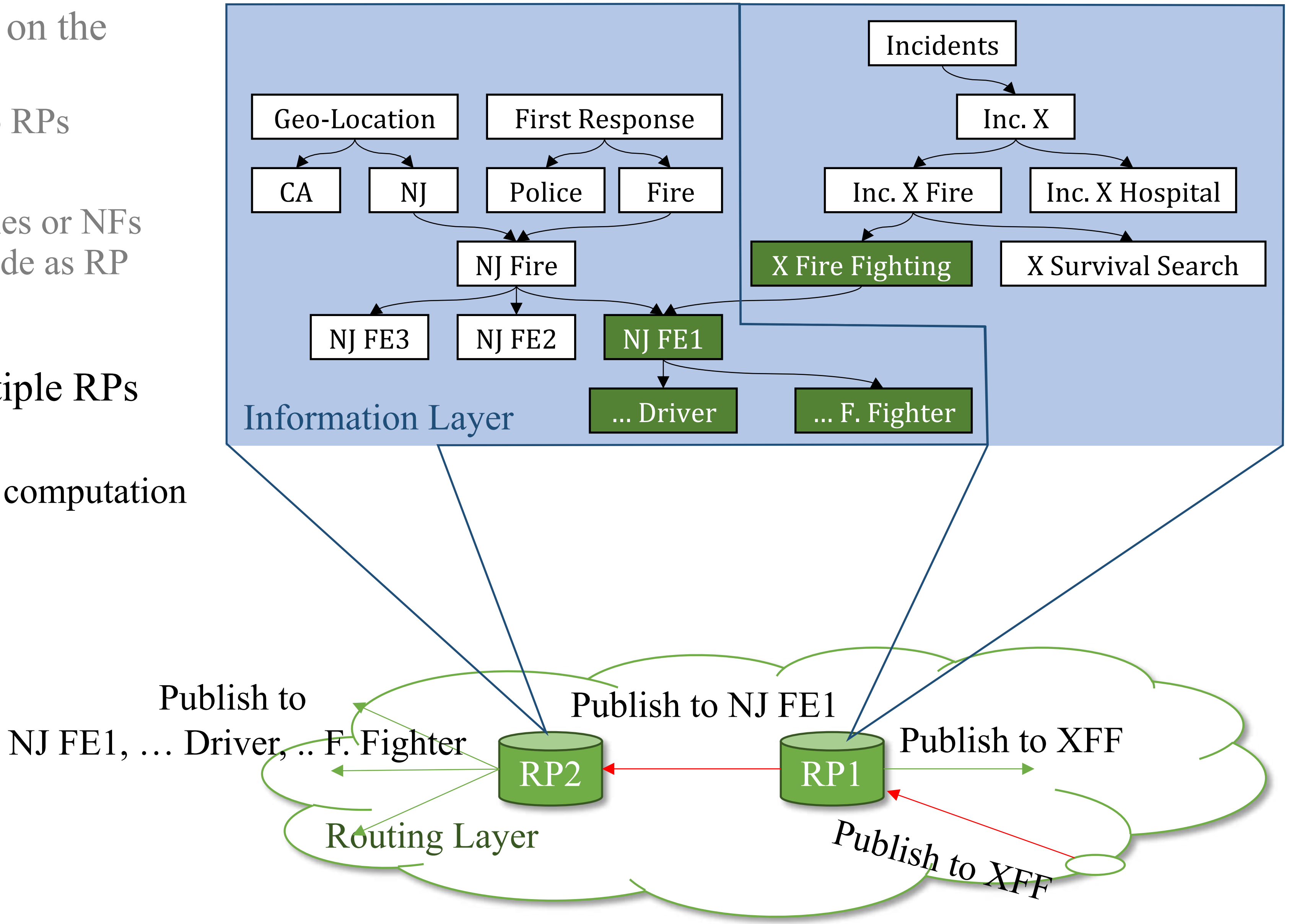
- Place the expansion functionality on the Rendezvous Point(s)
  - Multicast traffic will anyway go to RPs
  - Avoid triangular routing of traffic
  - Can be implemented as middleboxes or NFs physically residing on the same node as RP



Solution Overview

- Place the expansion functionality on the Rendezvous Point(s)
  - Multicast traffic will anyway go to RPs
  - Avoid triangular routing of traffic
  - Can be implemented as middleboxes or NFs physically residing on the same node as RP

- Distribute the namespace on multiple RPs
  - Avoid traffic concentration
  - Reduce & localize the storage and computation
  - Minimize inter-RP traffic

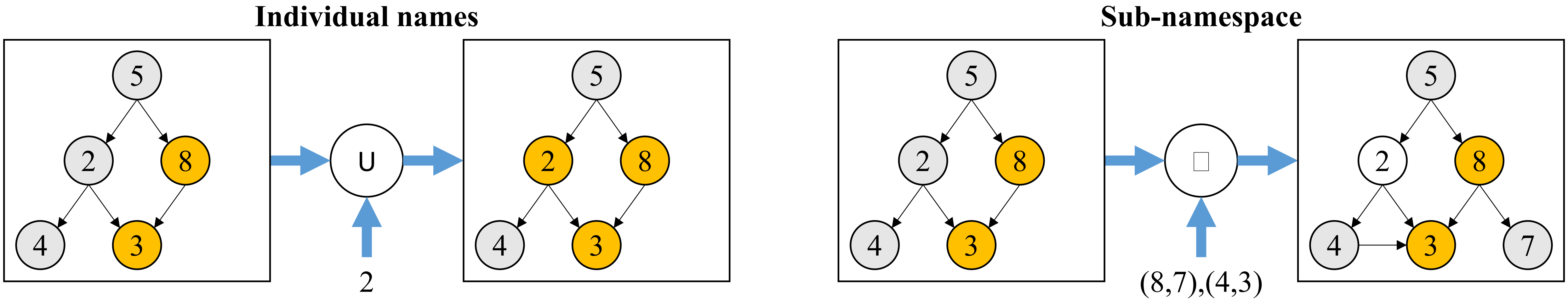


Propagation of Name Space and Messages in Disconnected Context

- In an environment where there are frequent problems with connectivity (because of lack of infrastructure, limitations of device-device connectivity, battery power limits), need to ensure we can still function
- We address a number of problems:
  - Name discovery: propagation of reachability and namespace updates
  - Critical messages being propagated from node to node (e.g., within a shelter)
- We seek to use existing techniques developed for delay-tolerant networks and disruption-tolerant networks
  - Gossip protocol: Epidemic routing for propagation of information in DTNs
- Name discovery procedures
  - Connected environment
    - Spreading new information
      - Propagate new name reachability announcement into network
    - Querying for information
      - Request for individual names or partial namespace
  - Disconnected environments
    - Exchange and share information between newly arrived mule and an encountered node in a new region

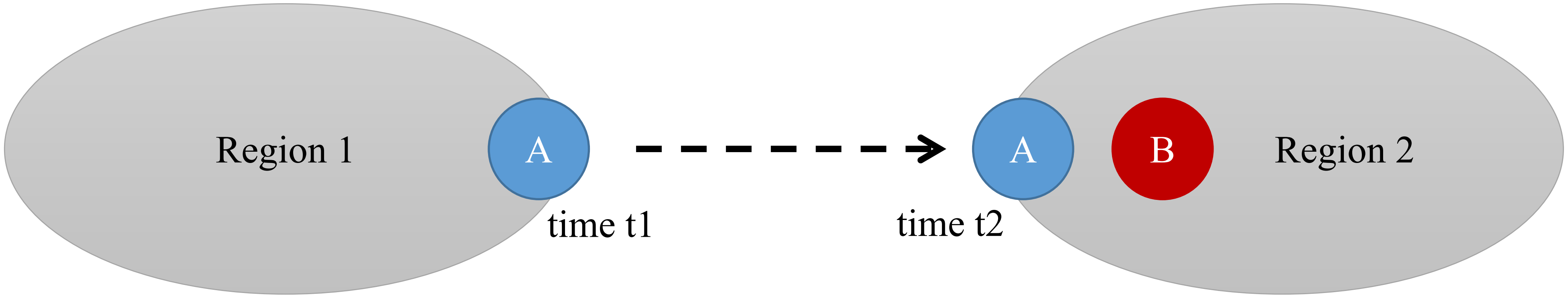
Propagating Name Reachability

- Name discovery
  - Individual names
    - I know the structure of graph (nodes and edges); want to know which nodes are reachable
  - Sub-namespaces
    - My graph is incomplete; want to get nodes and edges I don't have – so as to complete knowledge of my namespace
- Current information gets combined with the new update
  - Reachable names are "colored"



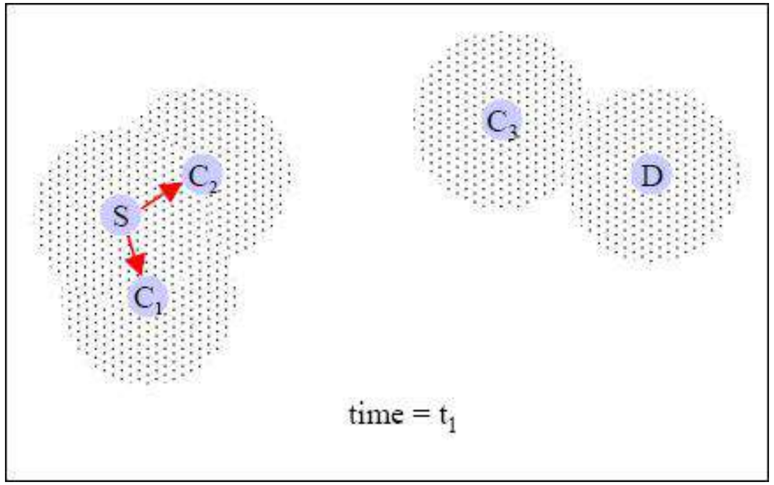
Overview of Information Exchange

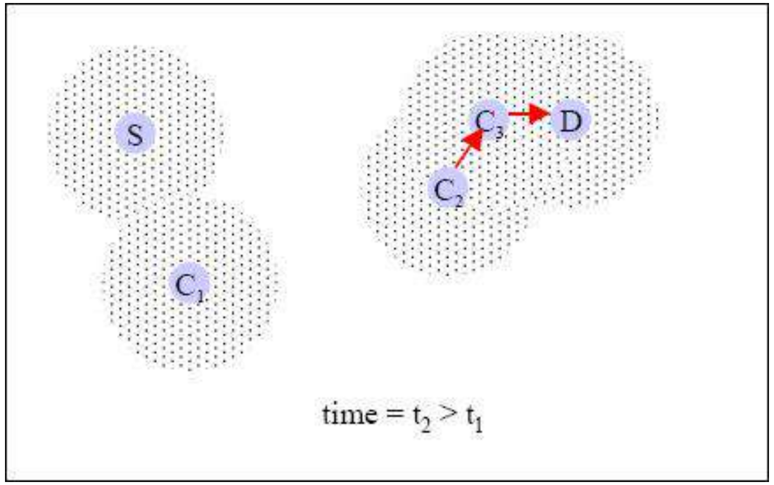
- In disconnected (fragmented) environment, there may not be a path between two nodes at any time
- Mule A is in Region 1 at time t1, moves to Region 2 at time t2, carrying knowledge he has accumulated regarding Region 1 info
- At t2, A and B exchange their info (Region 1 and Region 2 info)
- B has to be selected by A: random, lowest/nearest node ID, highest power, largest info set size, etc.



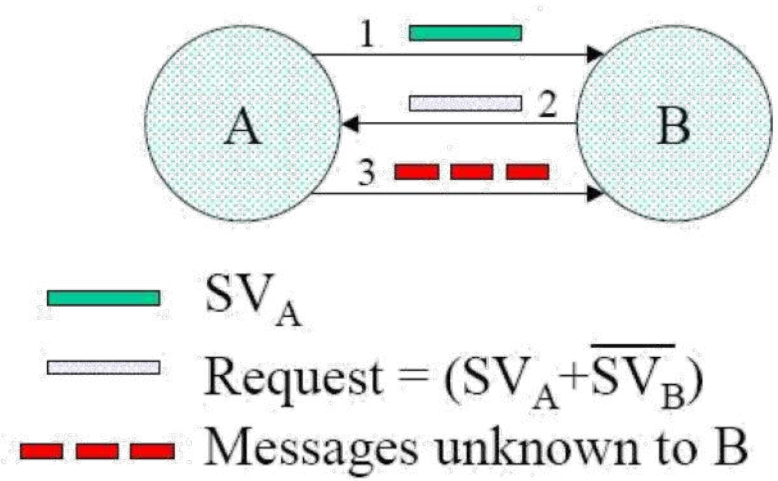
Epidemic Routing

- Amin Vahdat et al, "Epidemic routing for partially connected ad hoc networks", 2000.
- Goal: to deliver messages with high probability even when there is never a fully connected path





- Anti-Entropy sessions: relay messages
  - Summary vector: a list of ids of the messages that A has.
  - Message IDs are globally unique: node id + sequence #



Simulation Studies

- We have been using the ONE simulator, to study the performance of Name space propagation

- Study the effect of parameters:
  - Probability of acceptance (Gossiping probability): whether to accept an arriving message or discard
  - Max Hop: if the max. hop limit is reached, discard the message

- Two experiment settings
  - Setting 1: Random walk mobility in small environment
    - First responders and civilians
    - Investigate the impact of max hop and acceptance probability
  - Setting 2: Map-based mobility in larger environment

    2a) First responders, civilians, and high-speed mules
      - Investigate the impact of number of mules and mule speeds

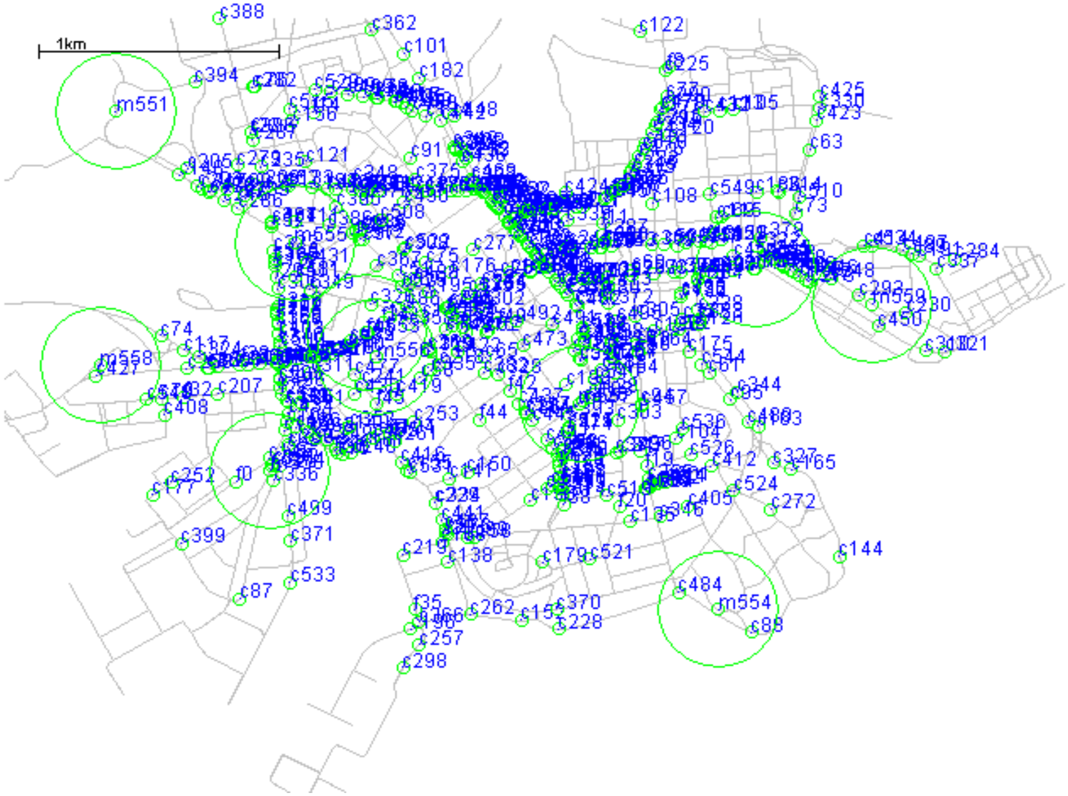
    2b) Incident managers, civilians and high-speed mules

Setting 2: Map-based Mobility

- World size: 4500×3400 meters

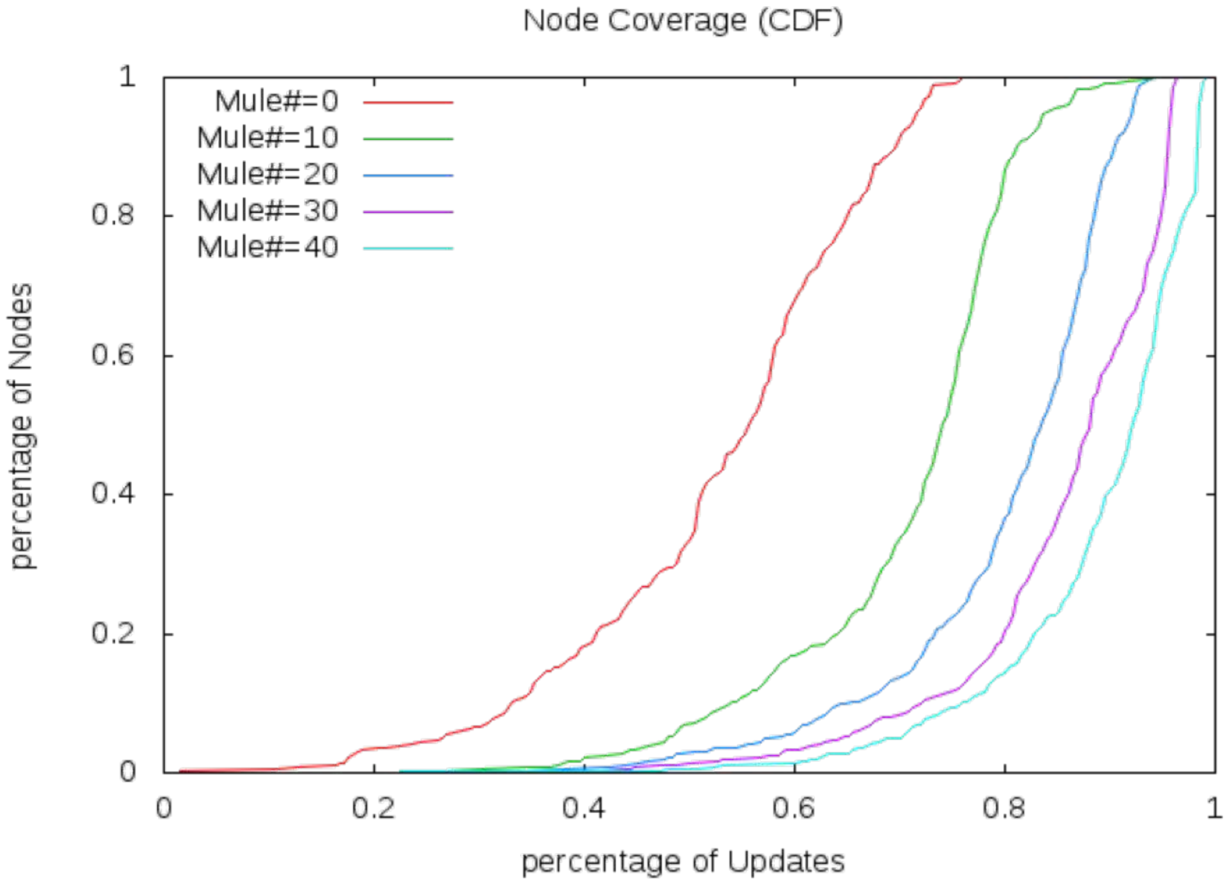
- Mobility model: map-based (Helsinki)
  - Randomly walking over map-based (shortest) paths
    a) 50 first responders and 500 civilian nodes
    b) 2 first responders propagating namespace changes (update different parts of the namespace) and 500 civilians

- High-speed mules
  - Vary number of mules
  - Vehicle type, typically higher movement speed (see each expt.)
  - Additional wireless interface: 1 Gbps, 250 meters range
  - More buffer capacity: 1000 MB
  - They collect namespace upon contact, store, carry and relay.



Impact of No. of High-speed Mules

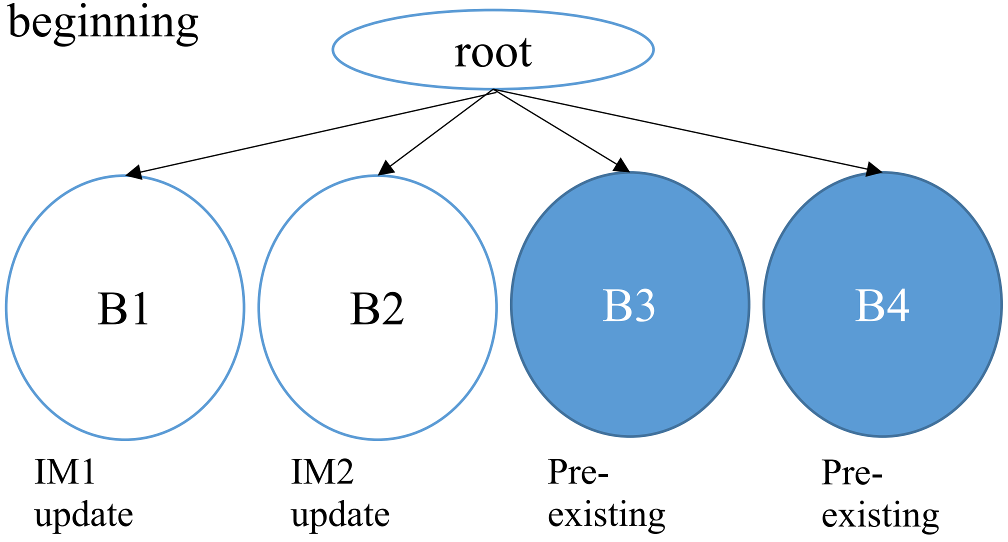
- Mules speed: 20 – 30 m/s (45 – 65 mph)
- Update interval: 600s;
- Acceptance probability: 0.75;
- Max hop: 100



- More mules helps the network get more connected and provide more coverage

Experiment with Incident Managers

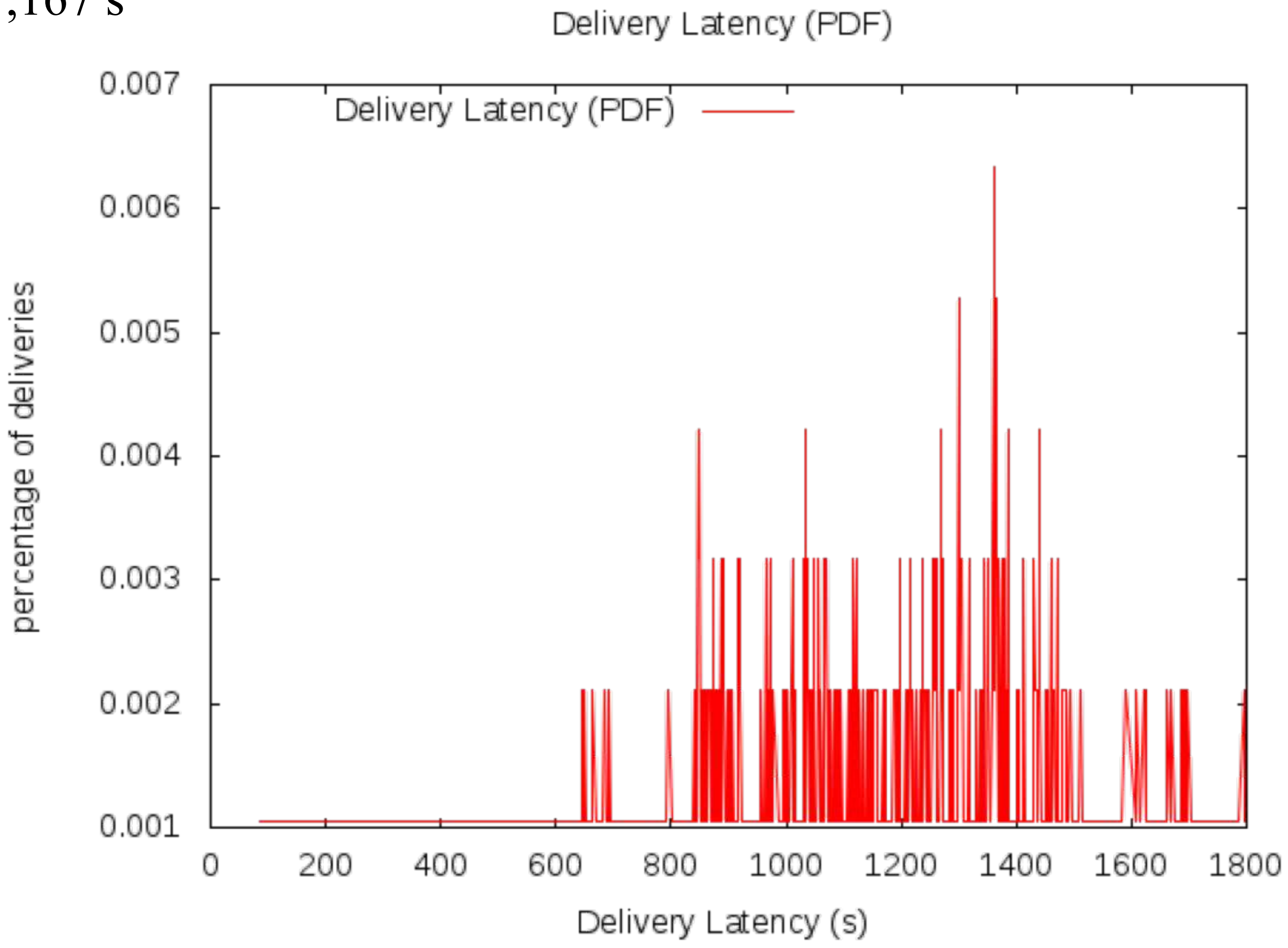
- 2 incident managers, 500 civilians, 5 high-speed mules
  - Incident managers send namespace update (one update each)
  - Namespace example: 4 branches at root; everyone has B3 and B4 in the beginning
  - IM1 sends B1, IM2 sends B2
- Each branch of namespace: 1MB size
- There are 2 updates (complete message or none)
- Map-based mobility over Helsinki map



- Results
  - Relays: 948
  - Average Coverage: 0.91 (each node gets 91% of updates)
  - Average Namespace completeness: 0.955 (95.5% of namespace is distributed to everyone)
  - Contacts: 17,429
  - Average relay per contact: 0.05 (only 2 messages)

Experiment with Incident Managers

- Propagation of name space:
  - Average latency: 1,167 s

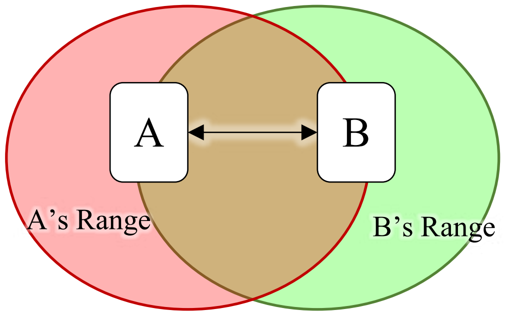


D2DMesh: D2D Mesh Network

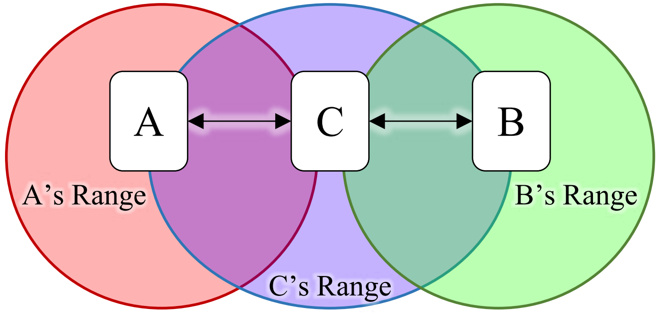
- A user-space D2D app that
  - Facilitates direct communication between mobile devices without using the infrastructure.
  - Utilize multi-hop heterogeneous wireless interfaces.
  - Enables sharing of services such as SMS, Internet, and camera.
- This application could
  - Help to maintain connectivity in infrastructure-less situations.
  - Be used by the first responders during disasters and for reaching to victims.
  - Help to crowdsource resources of individual devices.
  - Reduce traffic over the core network by offloading local traffic to D2D links.

Challenges:

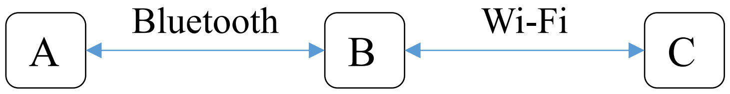
- Peer discovery
- Handling heterogeneous links from user space
- Quantifying D2D link quality
- D2D topology control
- Effective routing
- Vendor-specific issues (We use Android)



(a) Direct communication between A and B



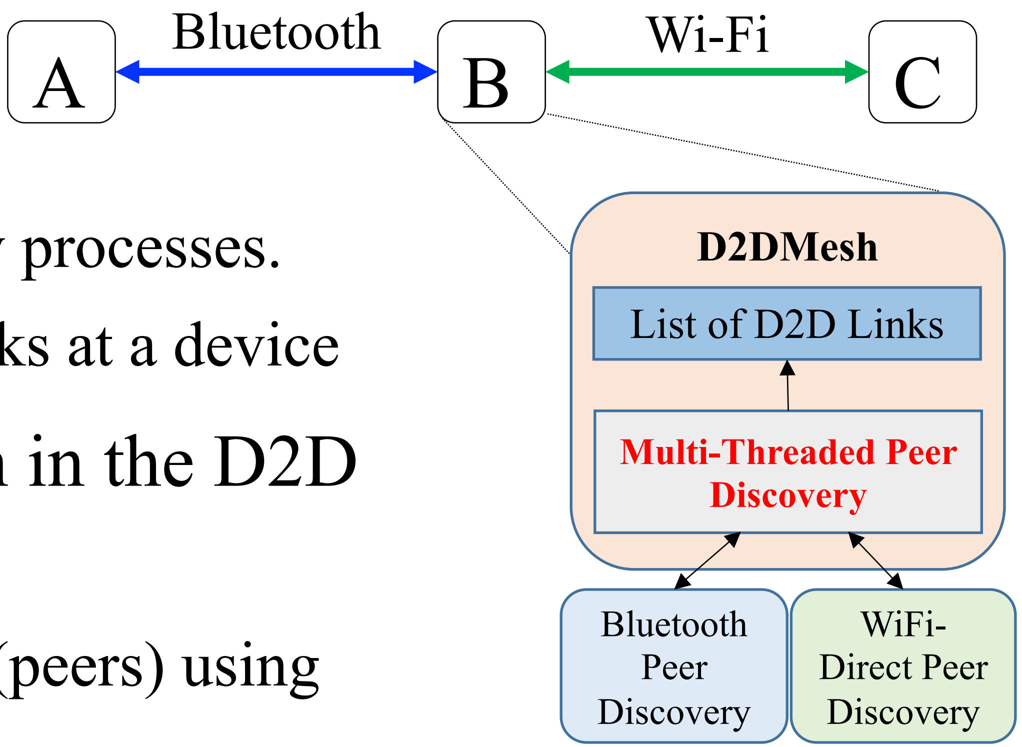
(b) Communication between A and B via C



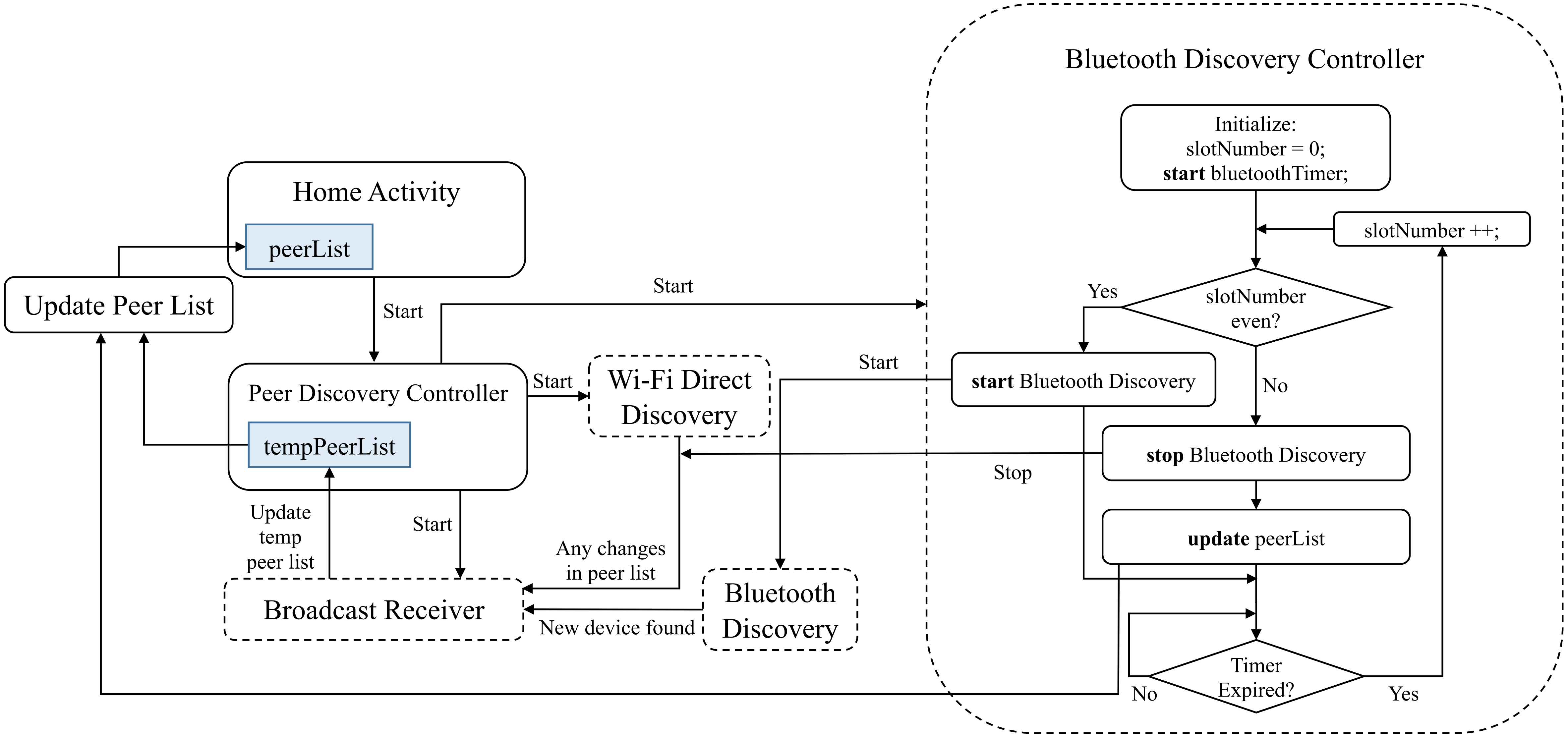
(c) Communication via Bluetooth and Wi-Fi

Simultaneous D2D Peer Discovery

- How to detect D2D links at the user space?
  - Both WiFi-Direct and Bluetooth use internal peer discovery
  - User apps can only start/stop these peer discovery processes.
  - Have to keep track of available heterogeneous links at a device
- Periodic peer discovery to handle dynamism in the D2D relationships
  - D2DMesh periodically discovers nearby devices (peers) using simultaneously both Bluetooth and WiFi-Direct
  - Discovery process runs on alternative time slots

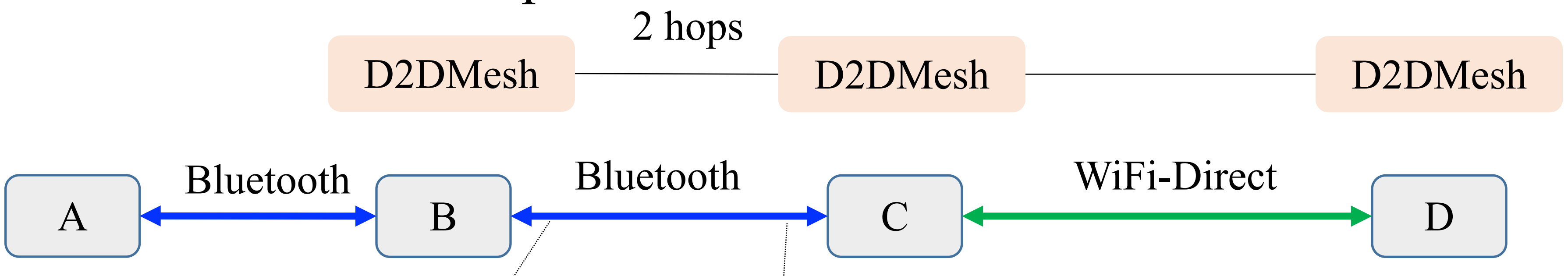


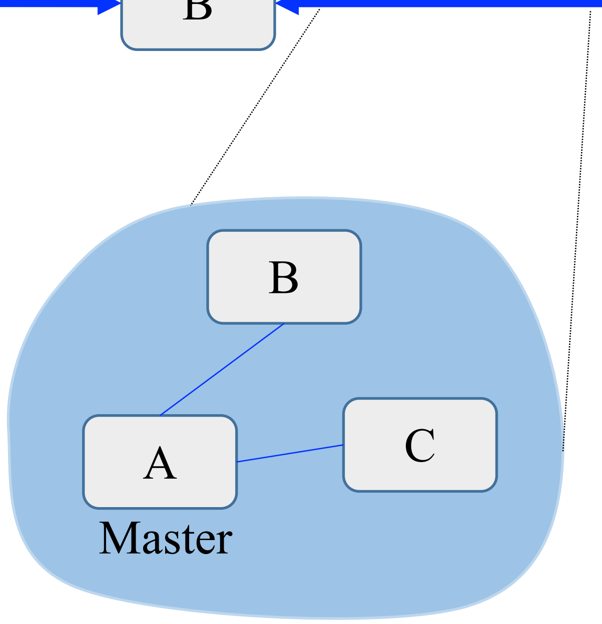
Simultaneous D2D Peer Discovery



D2D Link Quality Measurement

- How to quantify a D2D link's quality – for later use in topology control and routing?
  - WiFi-Direct and Bluetooth maintain a star topology within their groups
  - Links can involve two hops





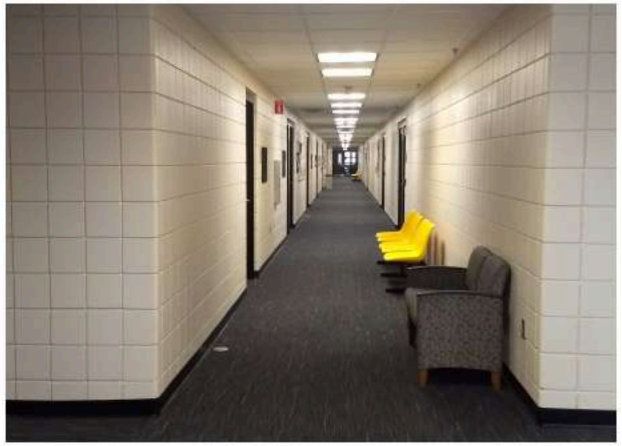
Careful measurement of link quality metrics is needed:

- RSSI
- RTT
- Loss
- Throughput

D2D Link Quality Measurement

- 4 devices from different vendors
- Outdoor: up to 80 m in a field
- Indoor: up to 40 m
  - 60 m long, 10 m wide corridor
  - LoS or NLoS

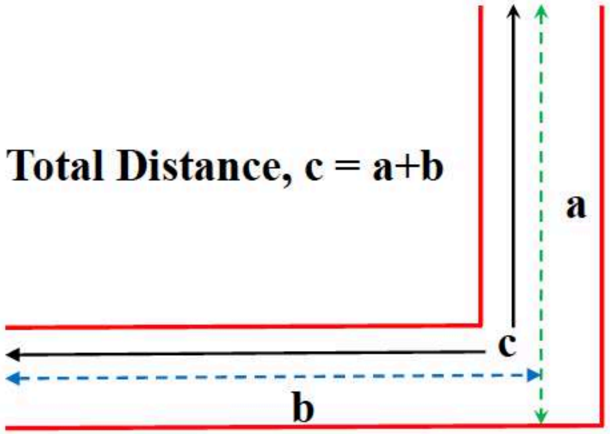
Vendors	Model	Processor	RAM
LG	Nexus 5	2.26 GHz quad core	2GB
Motorola	Moto E4	1.4 GHz quad core	2GB
Sony	Xperia L1	1.45 GHz quad core	2GB
Nokia	Nokia 2	1.3 GHz quad core	1GB



(a) Indoor LoS

(b) Outdoor LoS

(c) Indoor NLoS

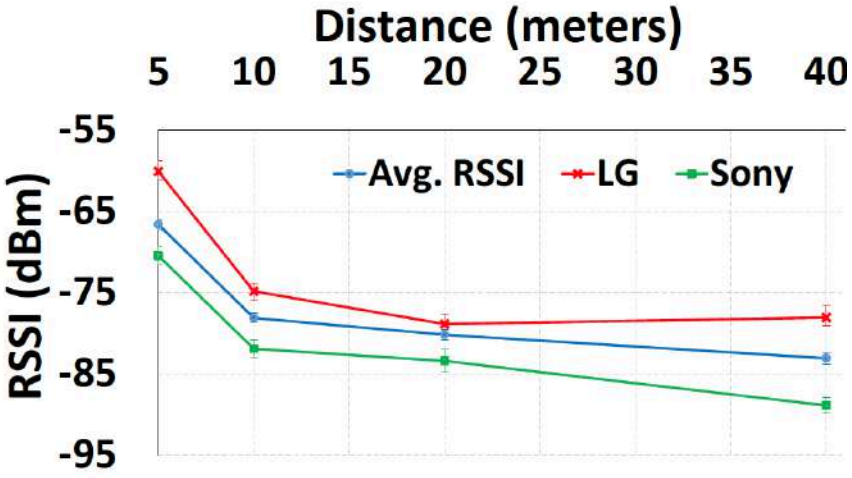


(d) NLoS Setup

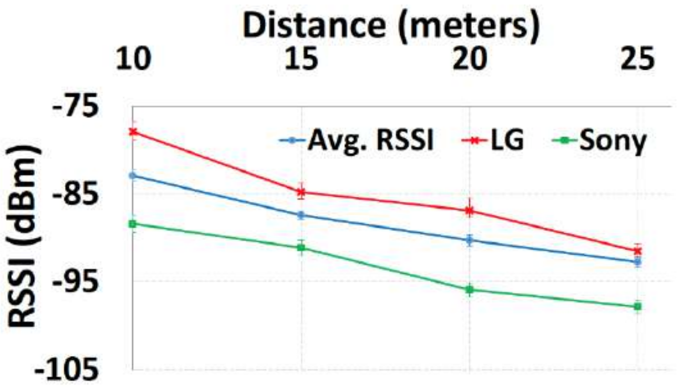
D2D Link Quality Measurements – Bluetooth

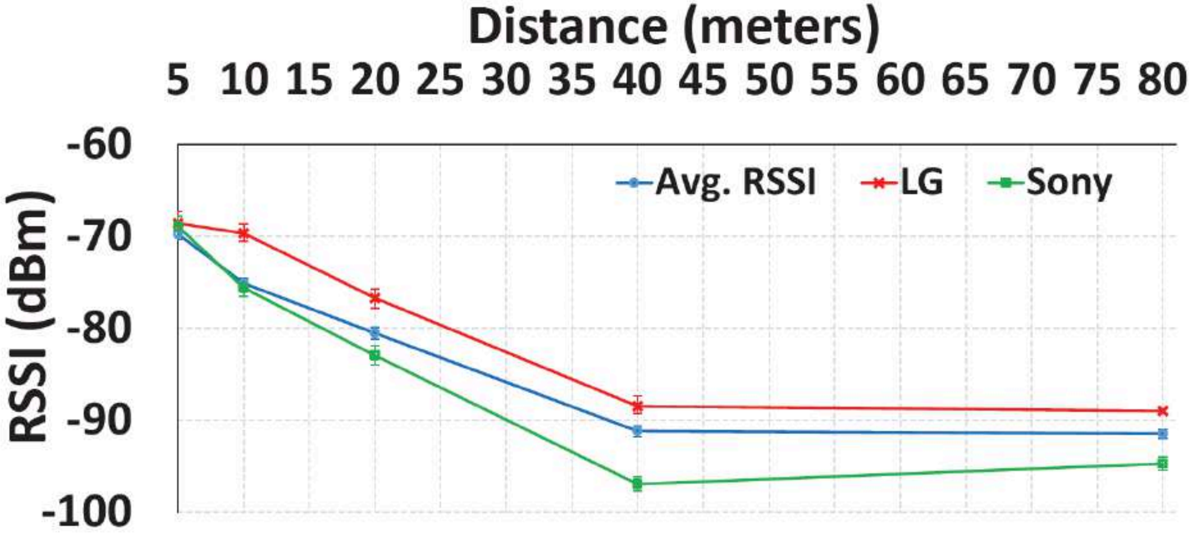
- RSSI

Indoor LoS



Indoor NLoS





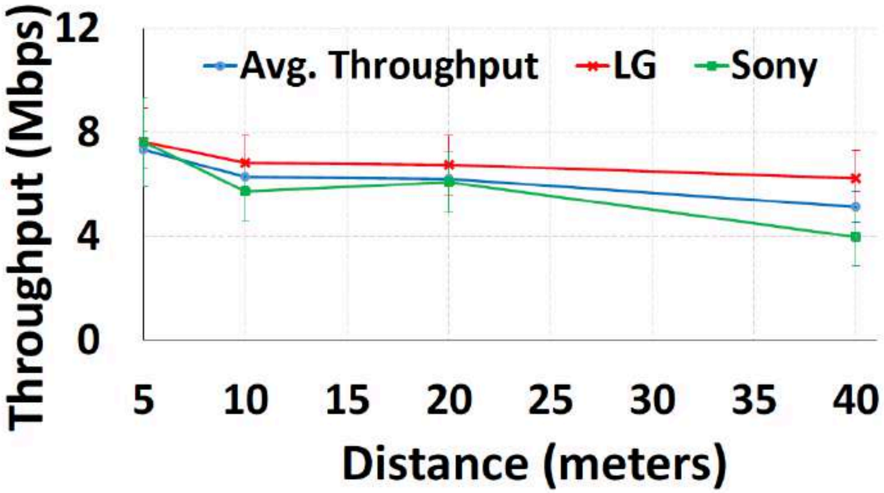
Outdoor

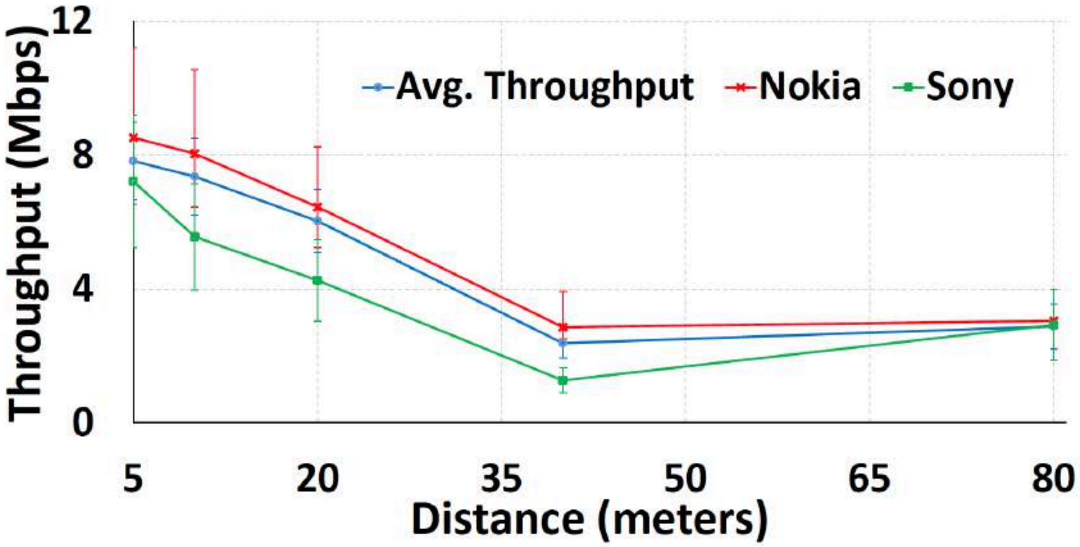
- For both indoor and outdoor, RSSI gets weaker with increasing distance.
- LG did the best while Sony did the worst performance.

D2D Link Quality Measurements – Wi-Fi Direct

- TCP throughput

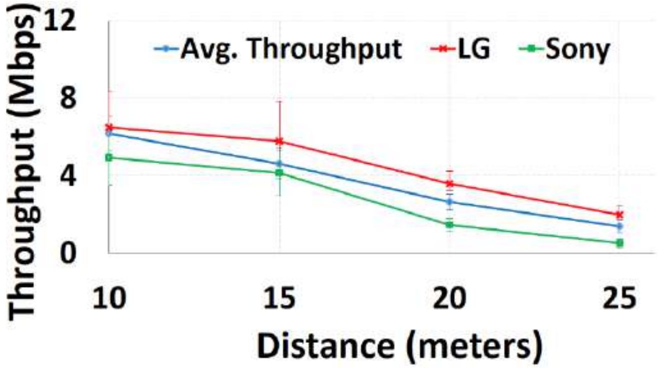
Indoor LoS





Outdoor

Indoor NLoS



- 10 MB file transferred for measurements
- Takes several seconds

- TCP throughput reduces per distance. Indoor LoS is the most reliable and agnostic to distance.
- Sony is the worst performer while LG or Nokia did the best performance.

D2D Link Quality Measurements: Correlations

- Can Bluetooth RSSI indicate the Wi-Fi Direct link quality?

Environment	Parameter	Correlation
Indoor LoS	Bluetooth RTT	0.77
	WiFi-Direct RTT	-0.46
	WiFi-Direct TCP Throughput	0.93
	WiFi-Direct UDP Throughput	-0.46
Indoor NLoS	Bluetooth RTT	-0.87
	WiFi-Direct RTT	-0.68
	WiFi-Direct TCP Throughput	0.99
	WiFi-Direct UDP Throughput	0.96
Outdoor LoS	Bluetooth RTT	-0.88
	WiFi-Direct RTT	-0.96
	WiFi-Direct TCP Throughput	0.98
	WiFi-Direct UDP Throughput	0.59

- Bluetooth RSSI is enough for quantifying WiFi Direct quality, but only at short distances.
  - WiFi Direct link quality at long distances is to be addressed.
- RTT is clearly affected by system issues, not by the link’s channel quality.

Prediction of Cell Tower Locations from Crowdsourced Data: Motivation

- Contacting 911 is challenging during disasters due to damaged or overloaded cell towers.
  - How reliable is the cellular system?
  - How many D2D hops to reach a cell tower?
- Knowing cell tower locations would help:
  - to predict resilience of the cellular system
  - to guide D2D communication
  - to forecast victims who might get affected during disasters

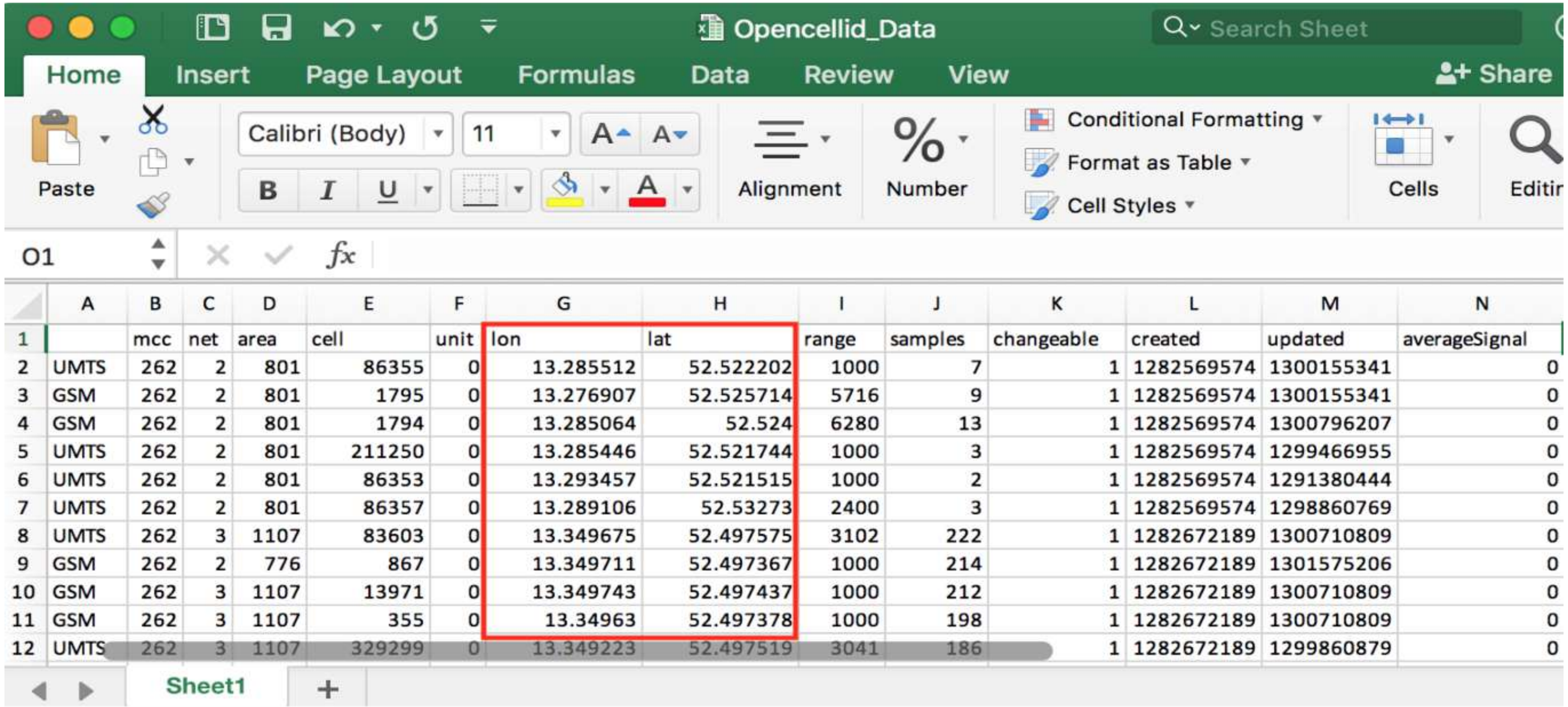
(D: down, U: up w/o ALI, R: reroute w/o ALI, A: reroute w/ ALI, Abnormal %: % of answer positions down or w/o ALI)

Date	County	PSAPs (Answer Positions)					Abnormal(%)	Cell sites down (%)
		Total	D	U	R	A		
9/10	Monroe	3 (11)	2 (7)				63.64	87 (80.56)
	Collier	2 (39)	2 (39)				100.00	160 (75.47)
	Hendry	4 (8)	2 (3)		1 (2)		62.50	31 (67.39)
	Lee	5 (41)	2 (15)	1 (14)	1 (2)		75.61	186 (54.23)
	Miami-Dade	7 (212)				1 (19)	0.00	739 (51.50)
	Broward	6 (126)					0.00	443 (47.94)
	Palm Beach	19 (142)				2 (13)	0.00	311 (42.84)
9/11	Monroe	3 (11)	2 (7)				63.64	89 (82.41)
	Collier	2 (39)		1 (33)	1 (6)		100.00	154 (72.64)
	Hendry	4 (8)		3 (5)			62.50	36 (78.26)
	Lee	5 (41)		4 (39)		1 (2)	95.12	170 (49.56)
	Miami-Dade	7 (212)				1 (19)	0.00	602 (41.95)
	Broward	6 (126)				1 (18)	0.00	353 (38.20)
	Palm Beach	19 (142)				2 (13)	0.00	244 (33.61)

- **Why crowdsourced data?**
  - Cellular providers are not obliged to provide cell locations.
  - FCC reports only provide aggregate data (cell tower count in a county) without any location information.

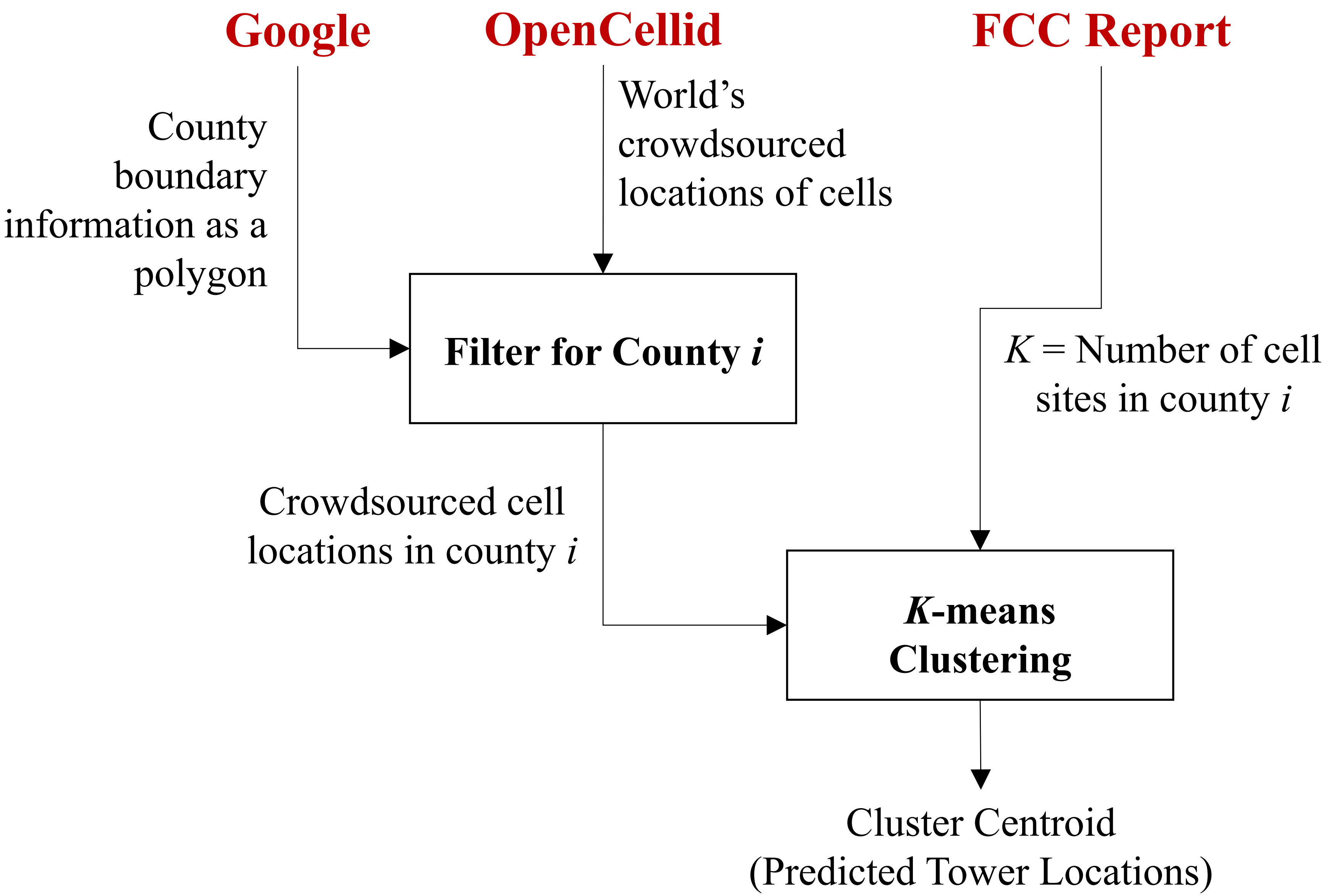
Prediction of Cell Tower Locations from Crowdsourced Data

- **Our approach:** Use crowdsourced dataset and try to predict the tower locations.
- **OpenCellid**
  - The World's largest collaborative community project for collecting GPS locations of cellular network antennas.

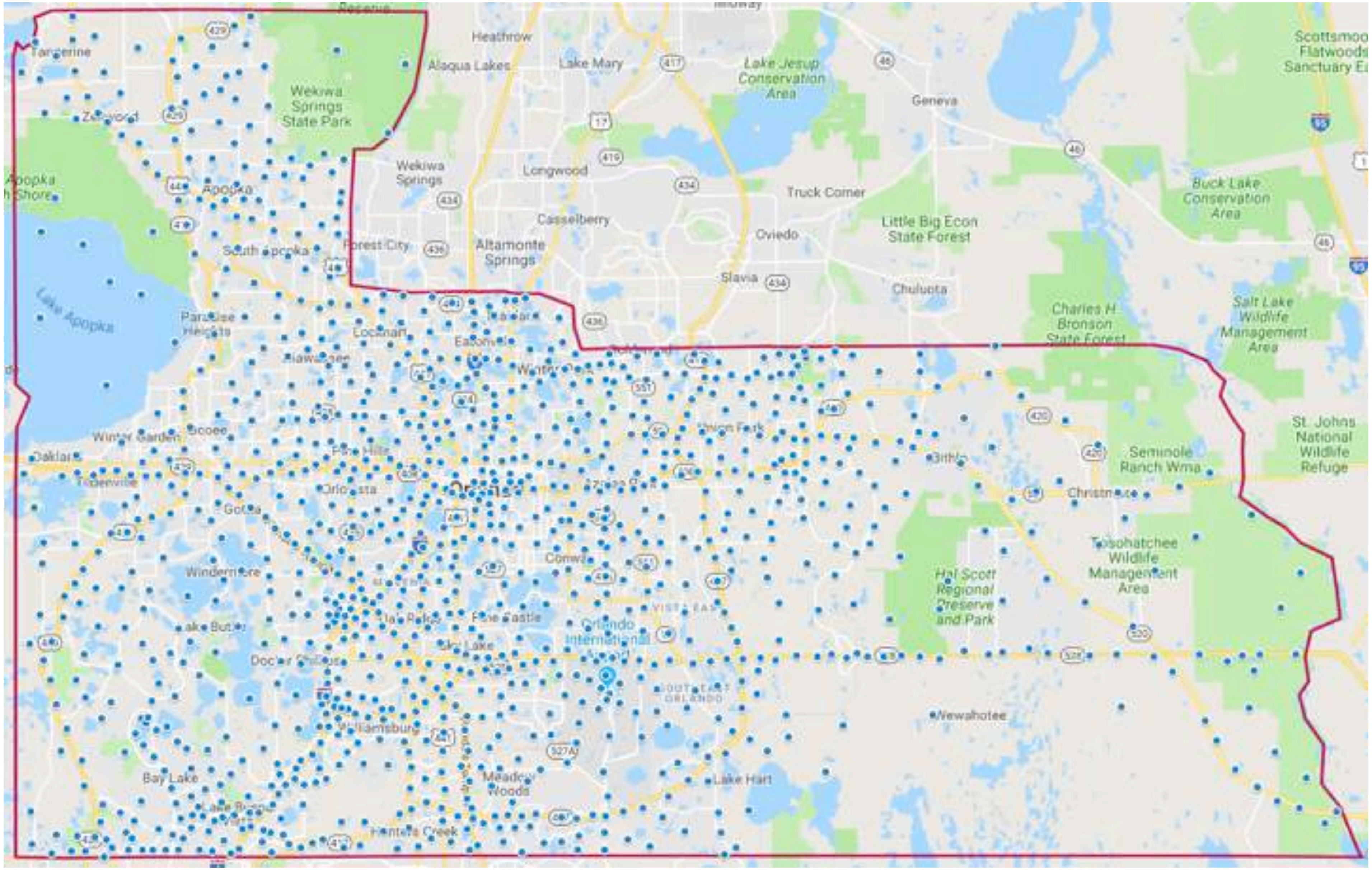


	mcc	net	area	cell	unit	lon	lat	range	samples	changeable	created	updated	averageSignal
UMTS	262	2	801	86355	0	13.285512	52.522202	1000	7	1	1282569574	1300155341	0
GSM	262	2	801	1795	0	13.276907	52.525714	5716	9	1	1282569574	1300155341	0
GSM	262	2	801	1794	0	13.285064	52.524	6280	13	1	1282569574	1300796207	0
UMTS	262	2	801	211250	0	13.285446	52.521744	1000	3	1	1282569574	1299466955	0
UMTS	262	2	801	86353	0	13.293457	52.521515	1000	2	1	1282569574	1291380444	0
UMTS	262	2	801	86357	0	13.289106	52.53273	2400	3	1	1282569574	1298860769	0
UMTS	262	3	1107	83603	0	13.349675	52.497575	3102	222	1	1282672189	1300710809	0
GSM	262	2	776	867	0	13.349711	52.497367	1000	214	1	1282672189	1301575206	0
GSM	262	3	1107	13971	0	13.349743	52.497437	1000	212	1	1282672189	1300710809	0
GSM	262	3	1107	355	0	13.34963	52.497378	1000	198	1	1282672189	1300710809	0
UMTS	262	3	1107	329299	0	13.349223	52.497519	3041	186	1	1282672189	1299860879	0

Overall Methodology



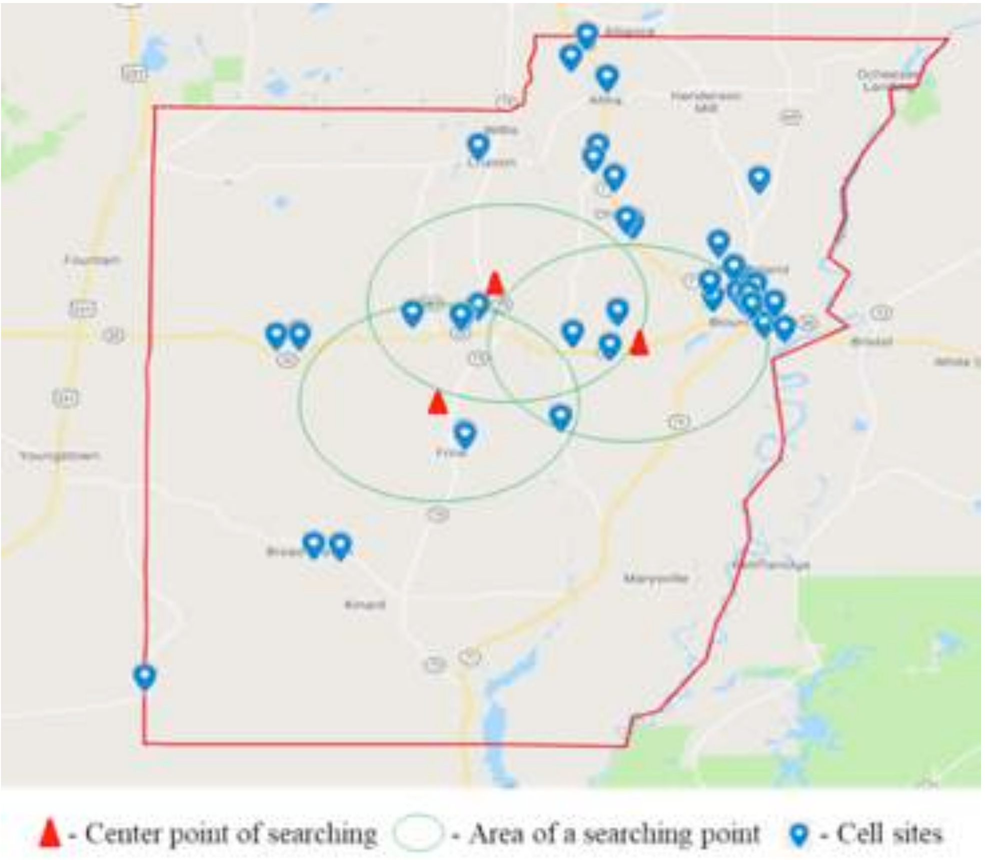
Predicted Towers for Orange County



How to Validate the Predicted Locations?

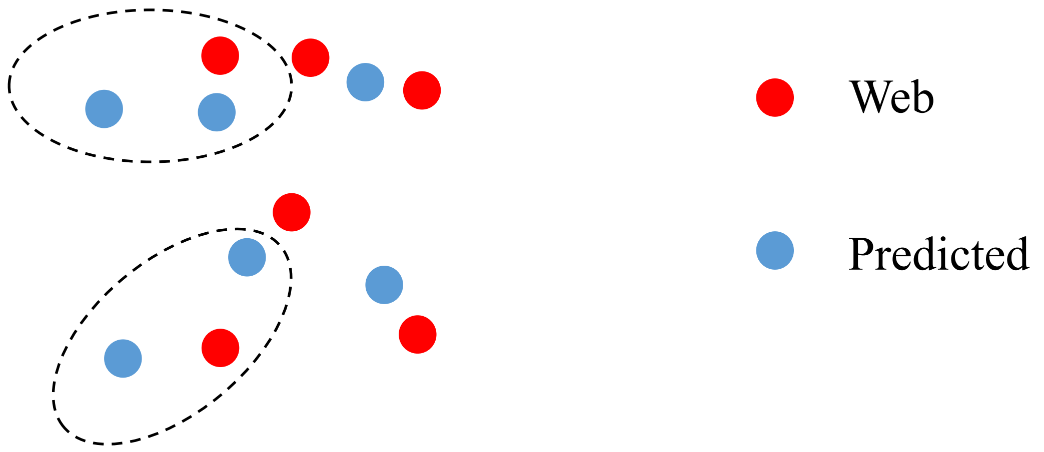
- How do we know our predictions are correct?
  - No ground-truth about the cell tower locations.
  - No provider would give out the cell locations.
- The only option is to use public websites that give an estimated location of the cell antennas.
- We get cell antenna locations from AntennaSearch.com
- The numbers of cell sites in FCC report and the website are different

County	FCC Report	AntennaSearch.com
Orange	1,152	1,605
Calhoun	15	52
Union	13	22



How to Validate the Predicted Locations?

- How close are the predicted tower locations to the web locations?
  - Need one-to-one mapping of the predicted and web tower locations (based on distance)
- We find the nearest "web tower" for a particular "predicted tower"
- When designing the mapping algorithm, we faced two issues:
  - Case I: A single web tower can be the nearest for multiple predicted towers.
  - Case II: Multiple predicted towers can have equal distance for a web tower.



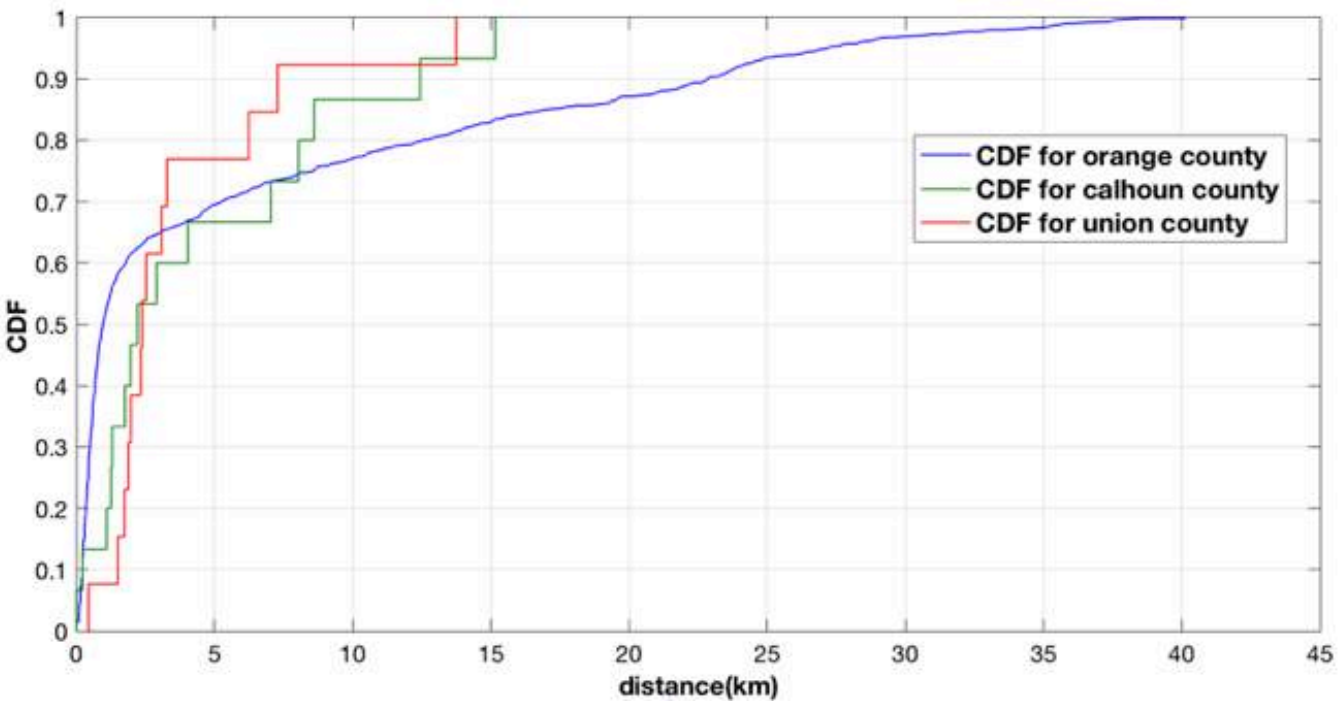
Predicted (P) to Web (W) Towers Mapping Algorithm

- Given two sets of locations: P for the predicted and W for the web towers:
- Find distances from a location of P to every location in W and store in a matrix D_{MN}
- Sort every row in D_{MN} to get the minimum distance and we get another matrix which is sorted, $D_{MN(sorted)}$
  - Pick a row m (i.e., a predicted tower)
  - Assign the nearest web tower n (i.e., the first element of row m) to the predicted tower m
  - Remove the entire row m and all other elements corresponding to the web tower n
  - Continue the above steps until all the rows are removed

- Case I: Multiple rows m_1 and m_2 have the same web tower n as their first element. In this case, we compare the distances $d_{m1,n}$ and $d_{m2,n}$, and pick the minimum one to map.

- Case II: It is still possible to have equal distance ($d_{m1,n} = d_{m2,n}$) for multiple nearest locations. In that case, we randomly choose a predicted point to map.

Predicted (P) to Web (W) Towers Mapping Algorithm



Calhoun County

- FCC shows cell sites = 15
- AntennaSearch.com shows cell sites = 52
- Precise cell tower location prediction = 2
- 50% cell towers (7) are located within **2 km**

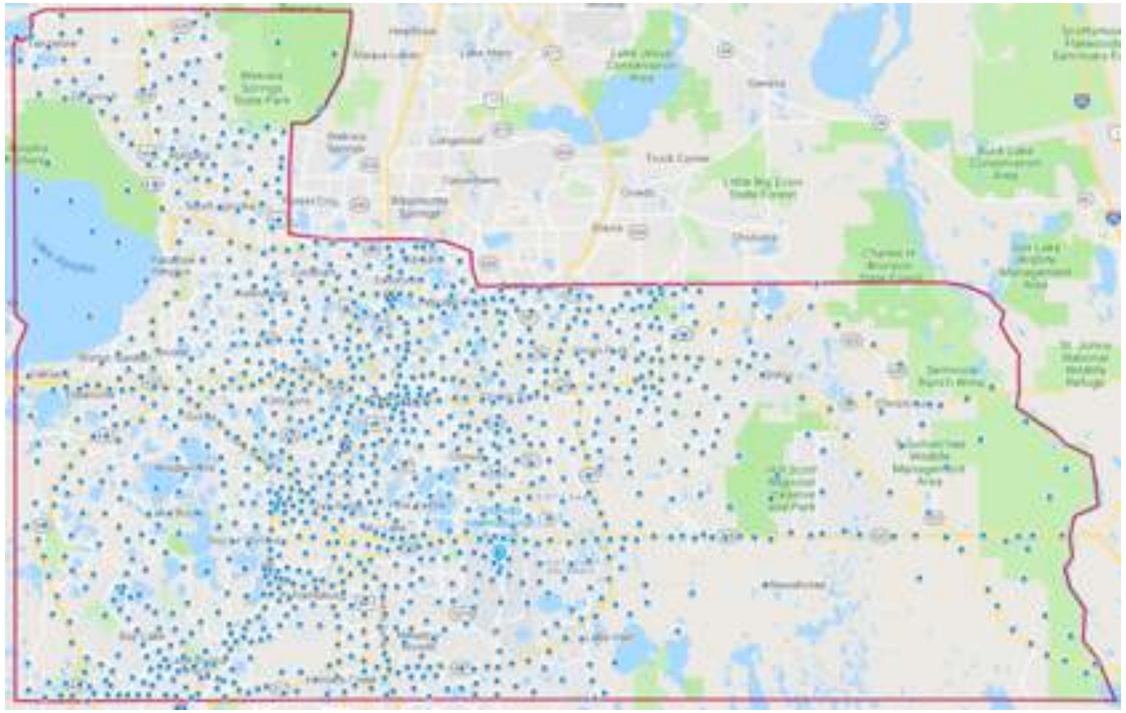
Union County

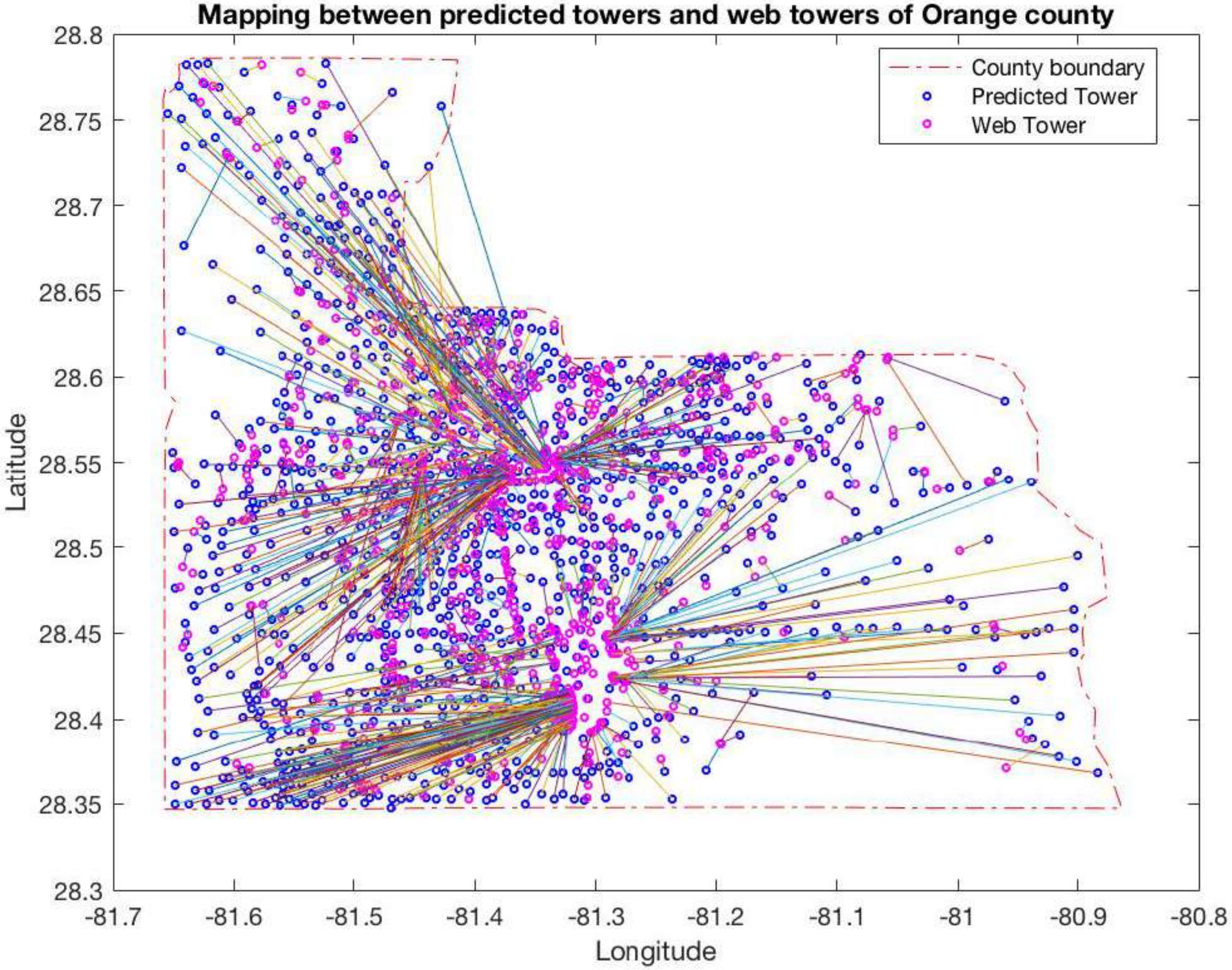
- FCC shows cell sites = 13
- AntennaSearch.com shows cell sites = 22
- 50% cell towers (7) are located within **2.4 km**

Orange County

- FCC shows cell sites = 1,152
- AntennaSearch.com shows cell sites = 1,605
- 17 exactly matched predicted cell tower locations.
- More than 50% of cell towers (584) are located within **1 km**

Predicted Towers for Orange County





Future Extension

- The crowdsourced dataset has more fields to utilize, e.g., range, number of samples, cell type.

- Try weighted clustering based on
  - the number of measurement samples for an entry in the crowdsourced dataset
  - the population map of the counties

- More accuracy in predicting the cell towers, e.g., randomize which predicted tower to pick.

Coding for Reliable D2D Computation

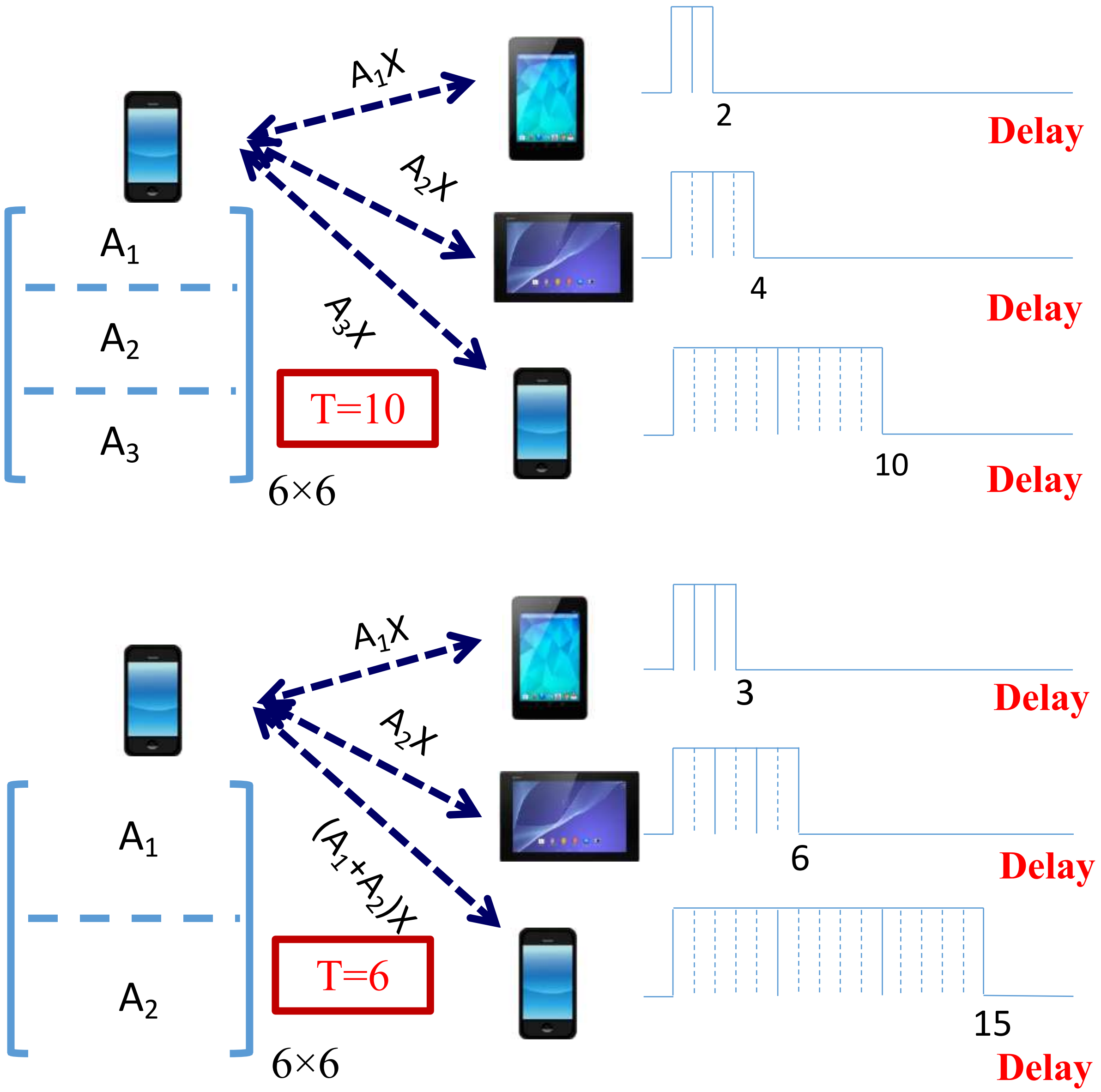
- Distributed computing can be crucial in first responder and PSC systems when there is little or no infrastructure support.
  - *E.g.*, creating a map showing the disaster area.

- Our approach:
  - Divide a computationally intensive task into small subtasks
  - Offload each subtask to multiple first responder/civilian devices after coding to improve resiliency of the system.

- Challenge: Heterogeneous nature of the first responder/civilian devices.
  - Different and time-varying computing power and energy resources
  - Mobility

How Does Coding Help for Computation?

- Calculation of matrix multiplication $y = Ax$
- Trivial Approach:
  - A is divided into 3 submatrices with equal size.
  - 3 tasks each consisting of one of the submatrices
- Coded Computation:
  - A is divided into 2 submatrices with equal size.
  - 3 coded tasks are generated from the 2 submatrices
- Advantage of coded computation:
  - Smaller delay
  - Higher reliability
  - Lower communication cost



Coding for Reliable D2D Computation

- Distributed computing can be crucial in first responder and PSC systems when there is little or no infrastructure support.
  - *E.g.*, creating a map showing the disaster area.

- Our approach:
  - Divide a computationally intensive task into small subtasks
  - Offload each subtask to multiple first responder/civilian devices after coding to improve resiliency of the system.

- Challenge: Heterogeneous nature of the first responder/civilian devices.
  - Different and time-varying computing power and energy resources
  - Mobility

System Model

- Setup: Master / worker setup. The master wishes to compute $y = Ax$, where A is an R by R matrix, x is an R by 1 vector.

- Coding: Packetize each row of A into a packet and create R packets; $\rho_1, \rho_2, \ldots \rho_R$. These packets are coded using Fountain codes to $v_1, v_2, \ldots v_{R+K}$, where K is the coding overhead.
  - Fountain codes suits well with the dynamic property of our work thanks to their rateless property, low encoding and decoding complexity, and low overhead.
  - Fountain codes perform better than
    - Repetition codes thanks to randomization of sub-tasks by mixing them,
    - MDS codes as they require a priori task allocation, thus not suitable for dynamic and adaptive framework, and
    - Network coding as the decoding complexity of network coding is high.

System Model

- Delay Model:

$$D_n = RTT_n^{data} + \sum_{i=1}^{r_n} \beta_{n,i}$$

Delay of worker n for offloading r_n tasks

Average RTT of sending and receiving a packet

The runtime of packet/task $p_{n,i}$ at worker n.

- RTT_n^{data} in this formulation is due to transmitting the first packet $p_{n,1}$, and receiving the last computed packet $p_{n,r_n}x$. The other packets can be transmitted while the worker is busy with processing previously transmitted packets.

Problem Formulation

- Problem: Determine the task offloading set $\mathcal{R} = \{r_1, r_2, \dots, r_N\}$ that minimizes the total task completion delay, *i.e.*, determine $\mathcal{R}$ that solves the following optimization problem:

$$\min_{\mathcal{R}} \max_{n \in \mathcal{N}} \Big(\overbrace{RTT_n^{data} + \textstyle\sum_{i=1}^{r_n} \beta_{n,i}}^{D_n} \Big)$$

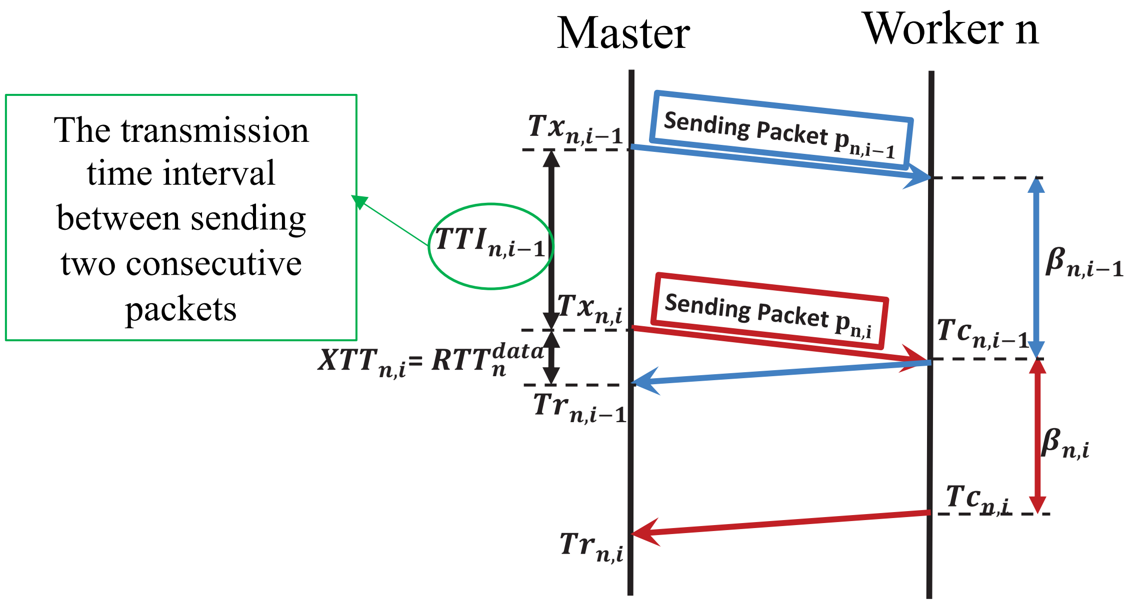
$$\text{subject to } \textstyle\sum_{n \in \mathcal{N}} r_n = R, r_n \in \mathbb{N}, n \in \mathcal{N}$$

- The solution to the above problem is challenging as
  - $RTT_n^{data} + \sum_{i=1}^{r_n} \beta_{n,i}$ is a random variable, not known a priori
  - Integer programming problem
  - Need a dynamic and online solution

C3P

- Approach: Inspired by the ARQ mechanism, the master transmits packets to workers gradually, estimates the runtime of each worker n based on the frequency of the received ACKs, and decides to send more/less coded packets to that worker.

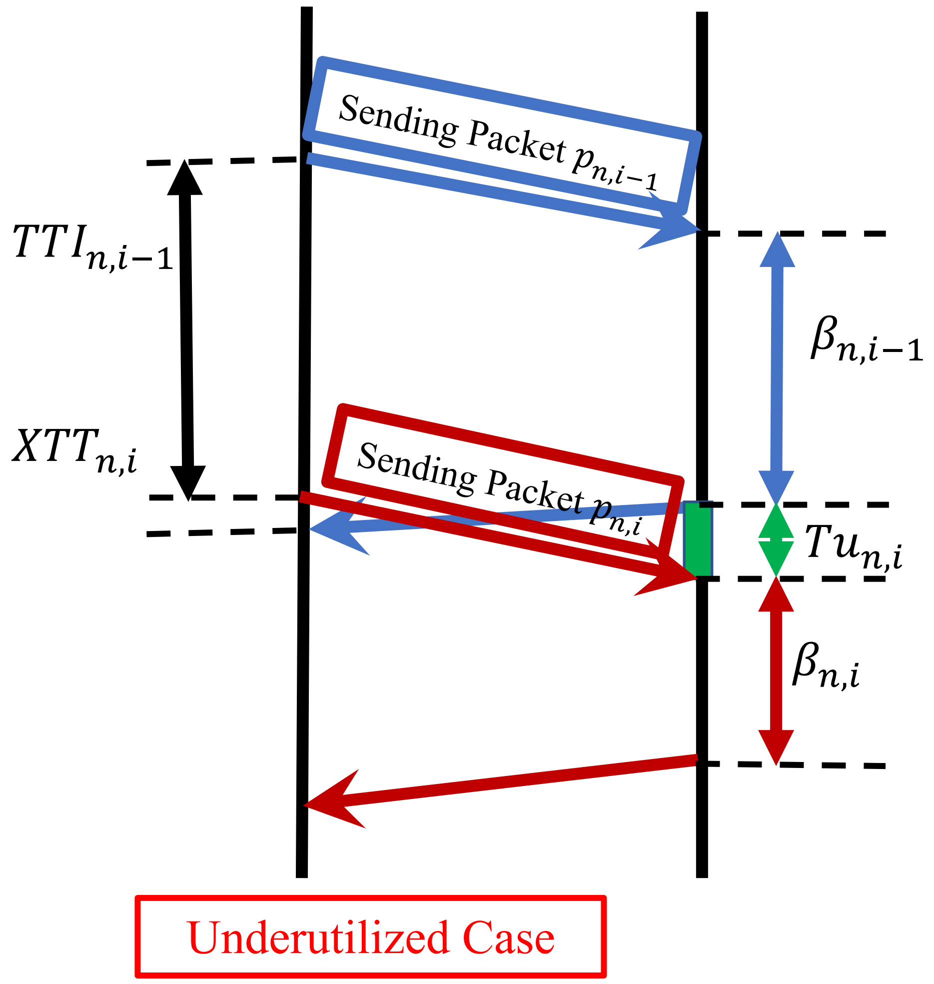
- Ideal Scenario:

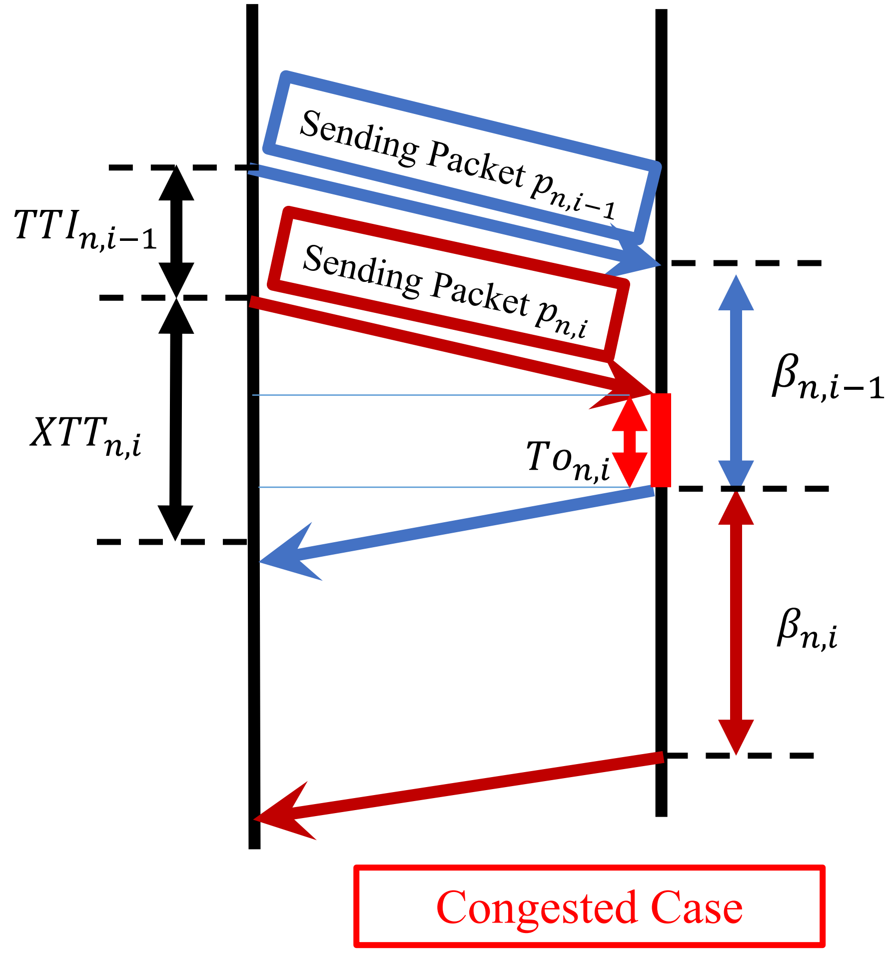


For the best system efficiency and smallest task completion delay, $TTI_{n,i} = \beta_{n,i}$

C3P

- In practice:





C3P

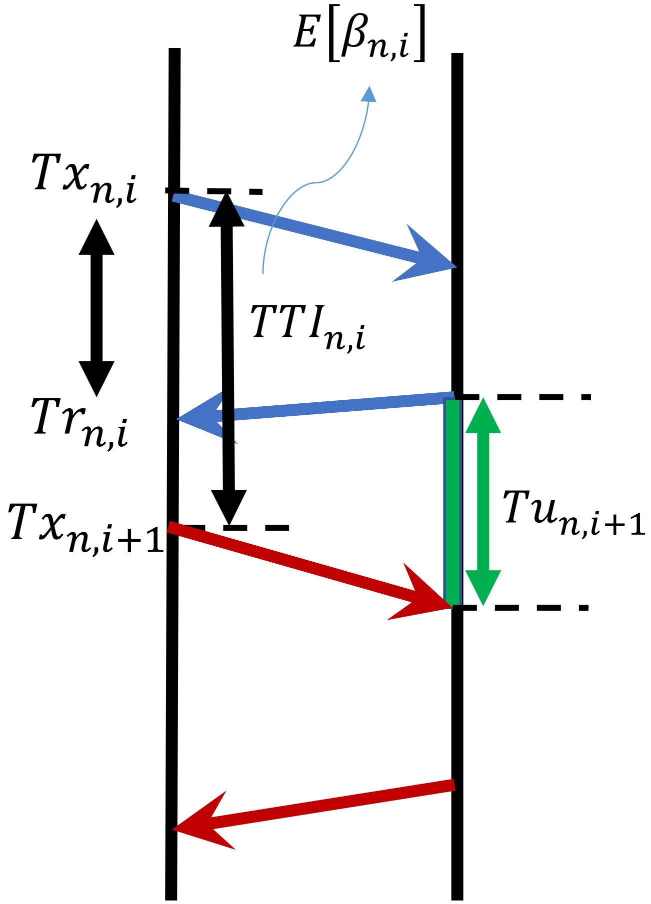
- Our solution:

  Set $TTI_{n,i} = \min(E[\beta_{n,i}], Tr_{n,i} - Tx_{n,i})$

- In C3P, $E[\beta_{n,i}]$ is estimated using runtimes of previously received packets:

$$E[\beta_{n,i}] \approx \frac{\sum_{n=1}^{m_n} \beta_{n,i}}{m_n}$$

- $\beta_{n,i}, i = 1,2,\ldots,m_n$ at the master device:
  - Put timestamps on sub-tasks to directly access the runtimes at the master.
  - The master device can estimate the runtimes by taking into account transmission and ACK times of sub-tasks. More efficient in terms of communication overhead.



Performance Analysis of C3P – Task Completion Delay & Efficiency

- Problem: $\min_{\mathcal{R}} \max_{n\in\mathcal{N}} \left(RTT_n^{data} + \sum_{i=1}^{r_n} \beta_{n,i}\right)$

  subject to $\sum_{n\in\mathcal{N}} r_n = R, r_n \in \mathbb{N}, n \in \mathcal{N}$

- Non-causal solution: Assuming perfect knowledge of $\beta_{n,i}$

$r_n^{best} = \operatorname{argmin}_{r_n} \max_{n\in\mathcal{N}} \left(RTT_n^{data} + \sum_{i=1}^{r_n} \beta_{n,i}\right), T^{best} = \max_{n\in\mathcal{N}} \left(RTT_n^{data} + \sum_{i=1}^{r_n^{best}} \beta_{n,i}\right)$

$T^{C3P} - T^{best}$ gap becomes finite when $R \to \infty$.

- Static solution: Assuming $D_n \approx \sum_{i=1}^{r_n} \beta_{n,i}$, the average solution

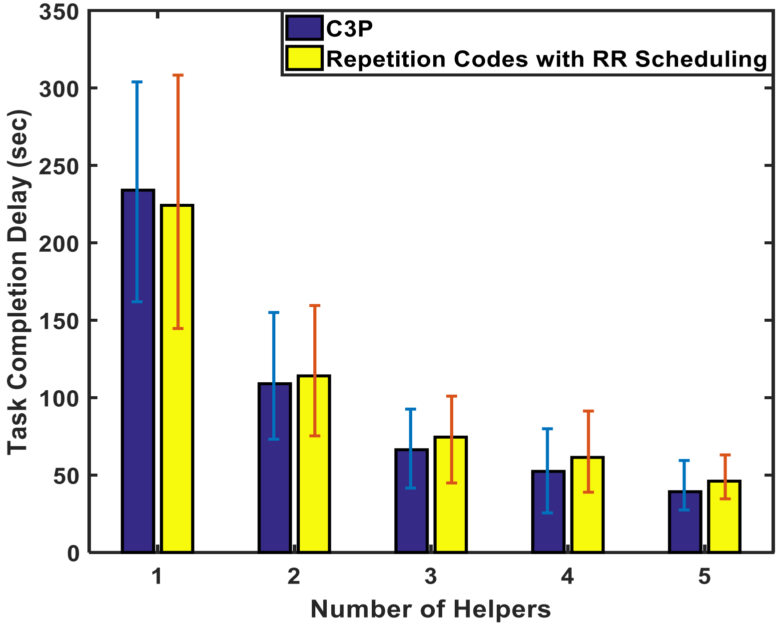
$$r_n^{static} = \frac{R}{E[\beta_{n,i}]\sum_{n=1}^{N} 1/E[\beta_{n,i}]} \quad T^{static} = \frac{R}{\sum_{n=1}^{N} 1/E[\beta_{n,i}]}$$

$T^{C3P} - T^{static} \to 0$ when $R \to \infty$.

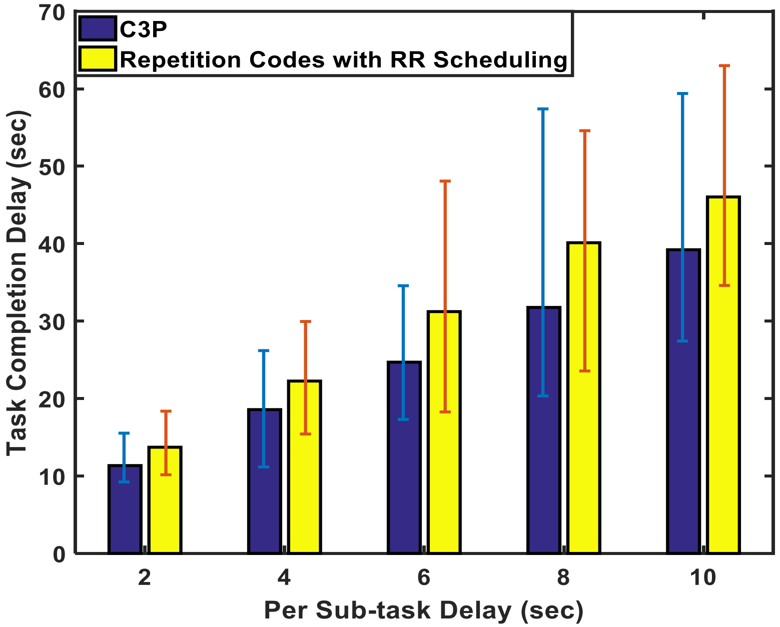
Efficiency of C3P is more than 99%.

Performance Analysis of C3P – Implementation

- Android 6.0.1 based Nexus 6P and Nexus 5 smartphones
- Workers are connected to the master device using Wi-Fi Direct connections
- A is a $1K \times 10K$ matrix and x is a $10K$ column vector
- Matrix A is divided into 20 sub-matrices, each of which is a 50×10K matrix.

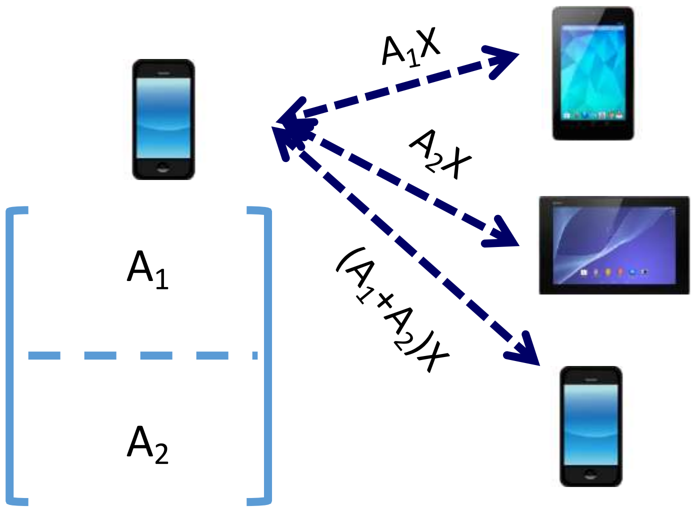


Per sub-task delay 10 sec



5 workers

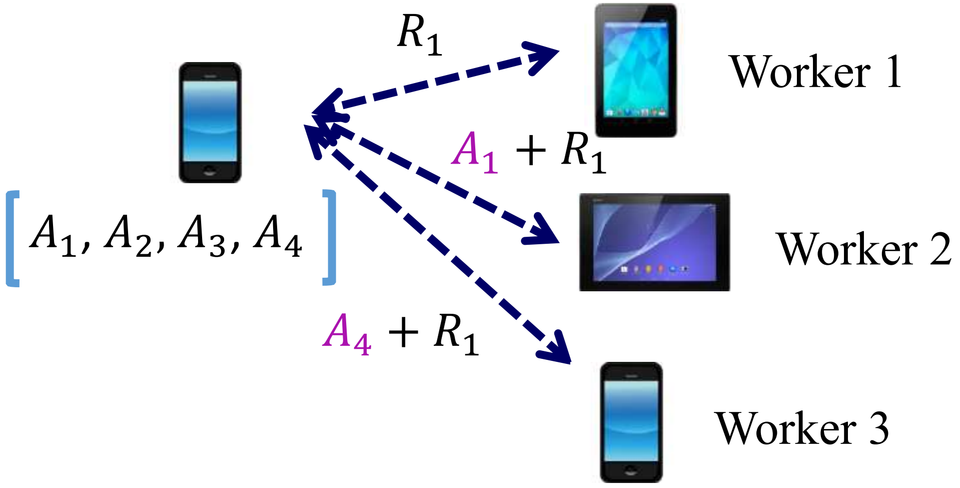
How about Privacy?



A_1, A_2, and x are revealed to workers.

PRAC

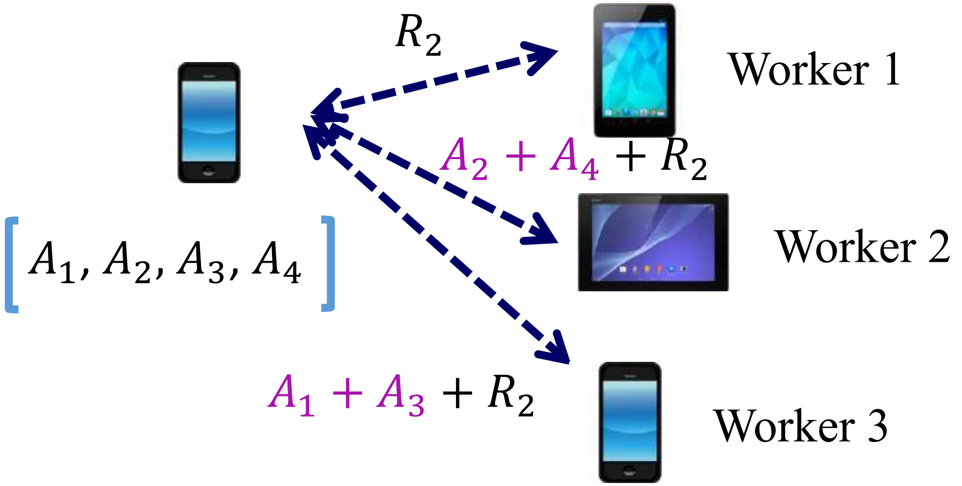
- Master/worker setup. Eavesdropping attack.
- Use random keys to mask data.
- Data packets are coded using Fountain codes, keys are coded using MDS codes.
- <u>Example:</u> One malicious worker in homogenous setup



Time	Worker 1	Worker 2	Worker 3
$t = 1$	$R_1 x$	$(A_1 + R_1)x$	$(A_4 + R_1)x$

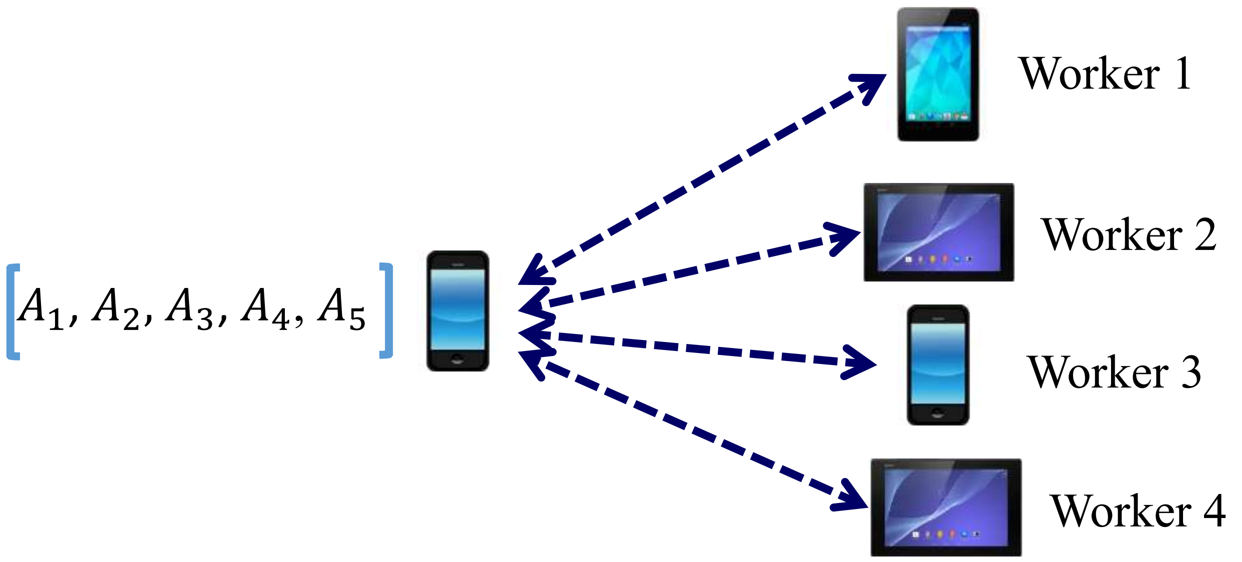
PRAC

- Master/worker setup. Eavesdropping attack.
- Use random keys to mask data.
- Data packets are coded using Fountain codes, keys are coded using MDS codes.
- Example: One malicious worker in homogenous setup



Time	Worker 1	Worker 2	Worker 3
$t = 1$	R_1x	$(A_1 + R_1)x$	$(A_4+R_1)x$
$t = 2$	R_2x	$(A_2 + A_4 + R_2)x$	$(A_1 + A_3 + R_2)x$

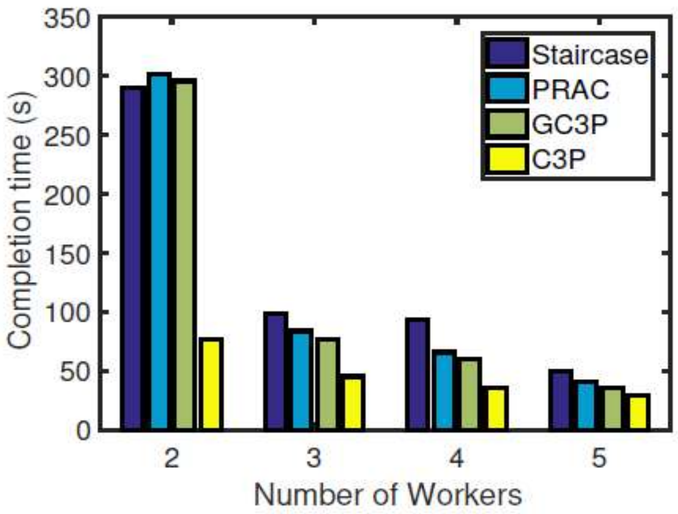
PRAC in Heterogeneous Setup – Two Colluding Workers



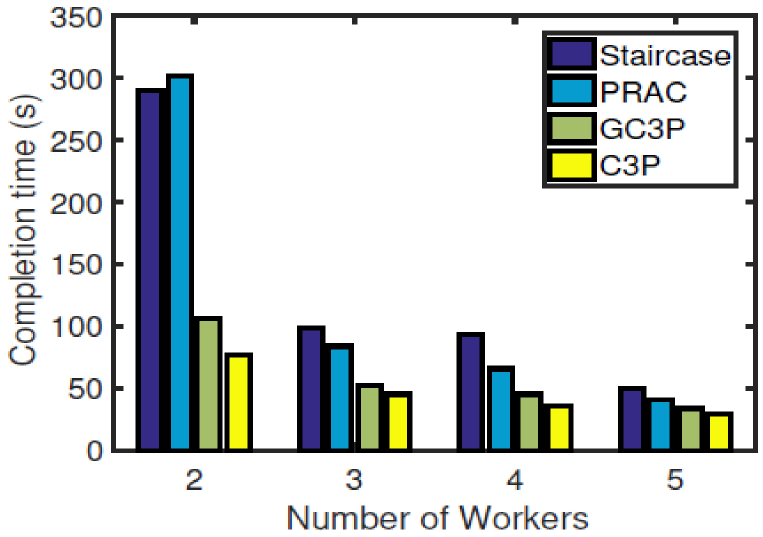
Time	Worker 1	Worker 2	Worker 3	Worker 4
$t = 1$	$R_{1,1}$	$R_{1,2}$	$A_4 + R_{1,1} + R_{1,2}$	$A_3 + A_4 + A_6 + R_{1,1} + R_{1,2}$
$t = 2$				$R_{2,1}$
$t = 3$	$R_{2,2}$			
$t = 4$		$A_3 + R_{2,1} + R_{2,2}$	$A_4 + A_5 + R_{2,1} + 2R_{2,2}$	
$t = 5$		$R_{3,1}$		
$t = 6$	$A_2 + R_{3,1} + R_{3,2}$			$R_{3,2}$
$t = 7$		$R_{4,1}$	$A_1 + R_{3,1} + 2R_{3,2}$	
$t = 8$	$R_{4,2}$			$A_2 + A_3 + R_{4,1} + R_{4,2}$

Performance Evaluation of PRAC - Implementation

- Android 6.0.1 based Nexus 6P and Nexus 5 smartphones
- Workers are connected to the master device using Wi-Fi Direct connections
- A is a $10K$ row vector and and x is a $10K$ column vector
- The master device needs to complete 60 $y = Ax$ calculations
- Two groups of workers: fast workers exponential delay with mean 2 seconds, slow workers with mean 5 seconds.



Fast worker is adversarial for GC3P

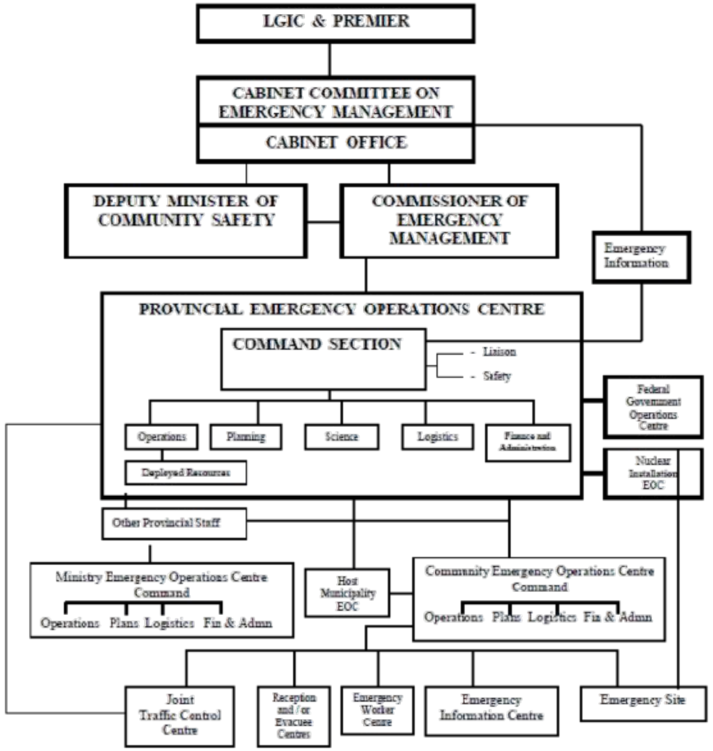


Slow worker is adversarial for GC3P

Demo: Use of ReDiCom in A Disaster

- Managing a disaster requires a lot of man power
  - Dynamic command chains
    - Dynamically formed teams – role-based communication, preplans (templates)
    - Different dimensions & granularities:
      - Geo-location, function, incident, …
      - Unit, team, everyone, …
    - Dynamic role changes
    - Usually done manually, on a white board, keeping track of every personnel
  - Heavy computation workload
    - Face recognition, video processing, …
  - Fragmented network
    - Need runners (mules) to carry messages around

ReDiCom tries to automate many of these actions so that the officers can focus on saving lives

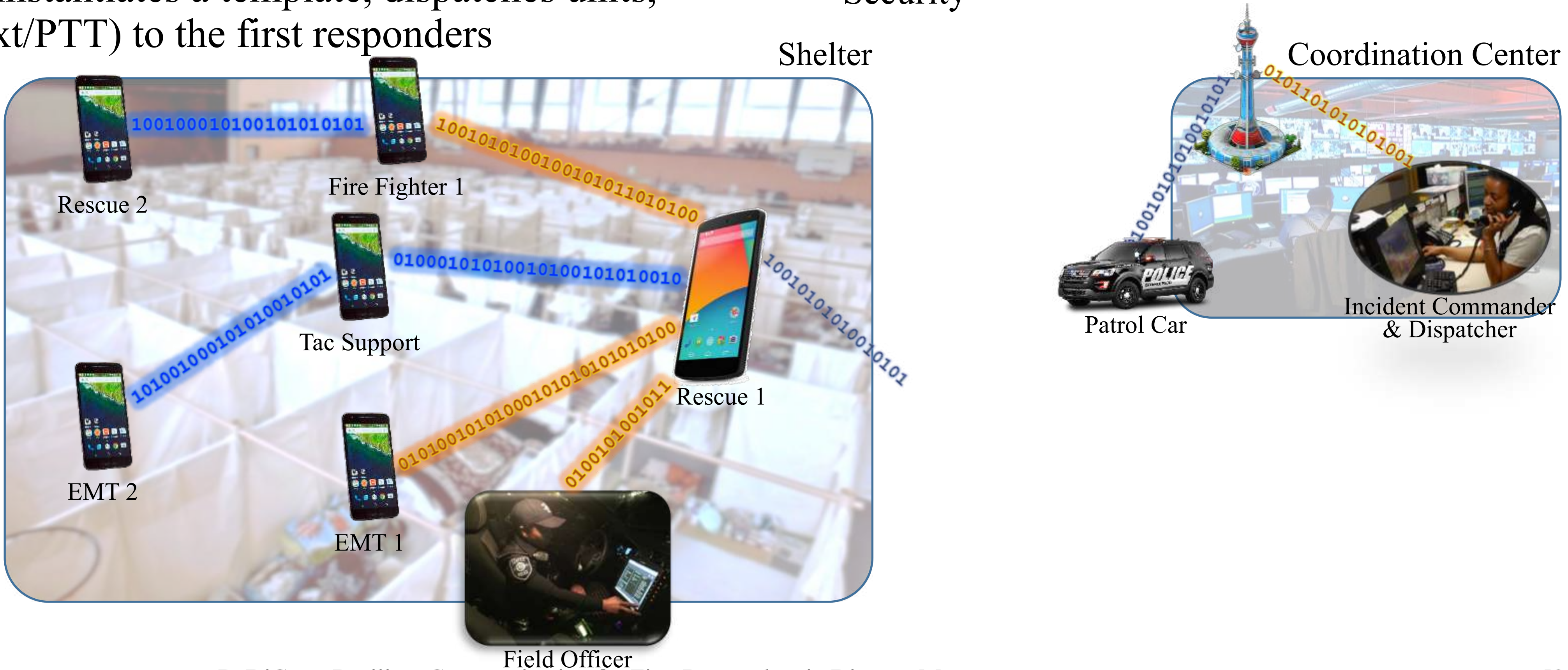


https://www.emergencymanagementontario.ca/english/beprepared/ontariohazards/nuclear/provincial_nuclear_emergency_response_plan.html

Use of ReDiCom in A Disaster

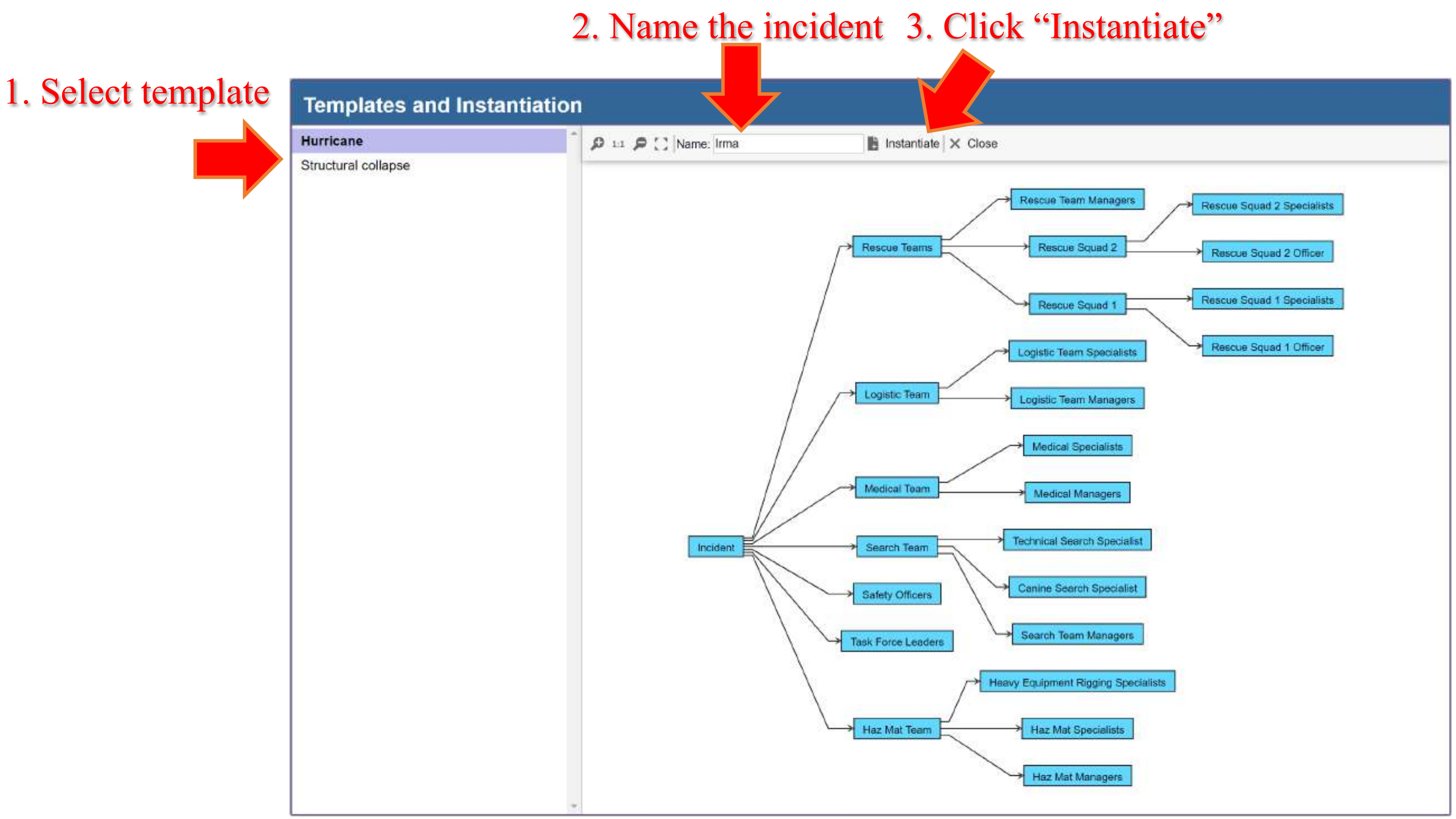
- Coordination Center & Shelter
- Leverage different communication technologies
  - WiFi Direct, Bluetooth, and (Legacy) WiFi
- Use graph-based namespace to manage command chains
- Coordination center instantiates a template, dispatches units, & sends messages (txt/PTT) to the first responders
- Patrol car carries the updates & messages to the shelter and disseminate among the first responders

- Use coded computation for work-offloading inside the shelter
  - Face recognition
  - Coded computation
  - Weighted work distribution
  - Security



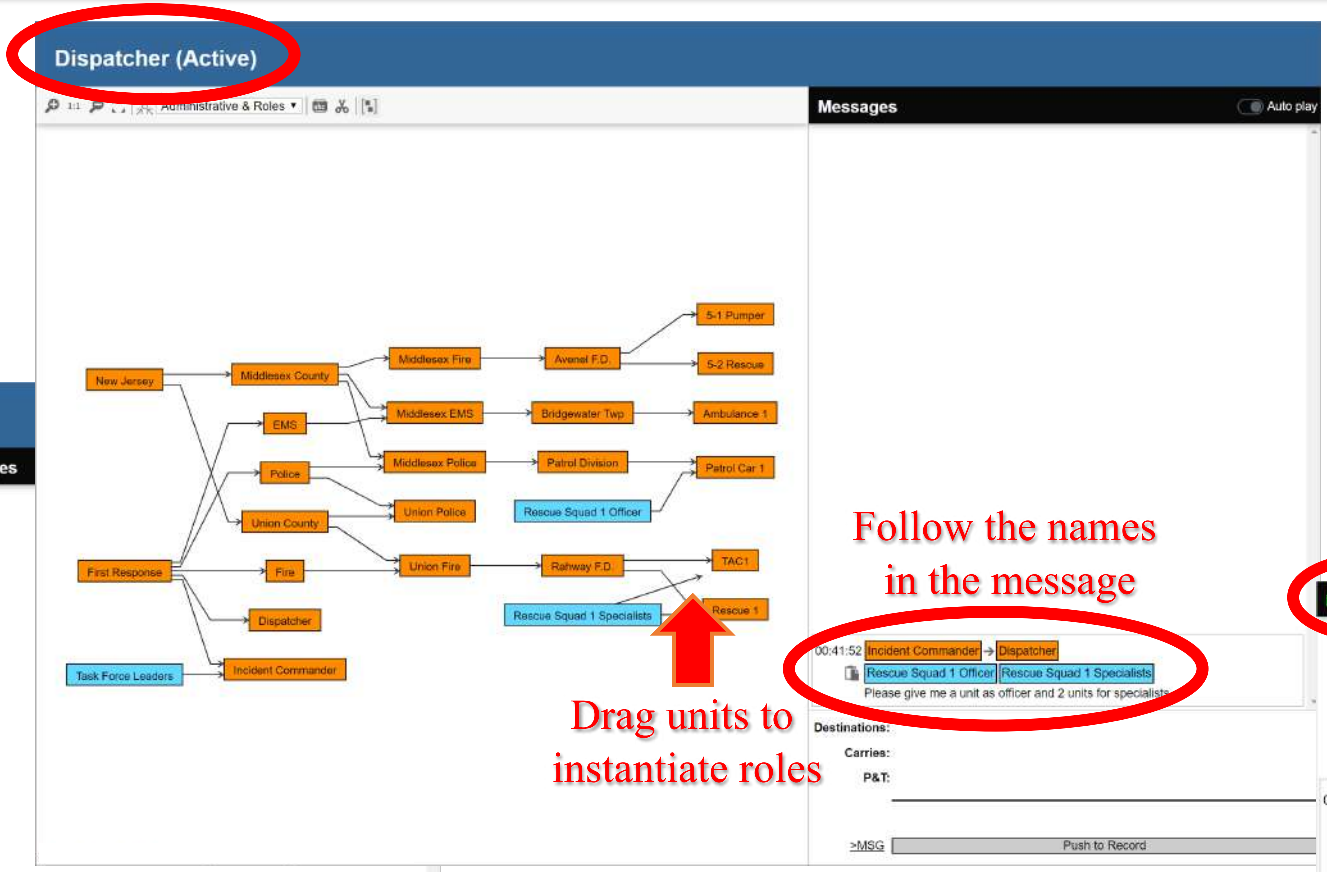
Application 1: Messaging

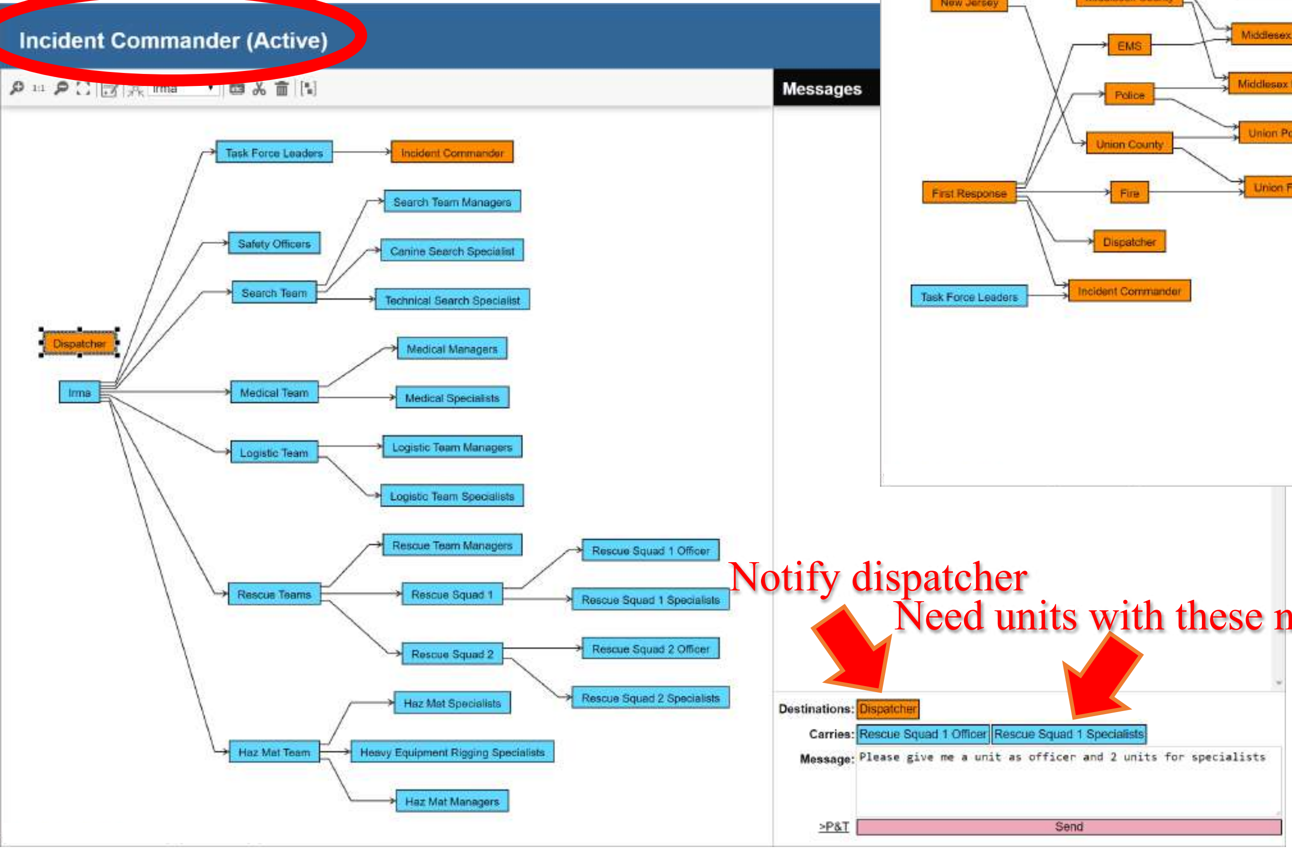
- Dynamically instantiate a template
  - Templates: roles defined by preplans
  - Instantiate: create new names in the namespace based on templates

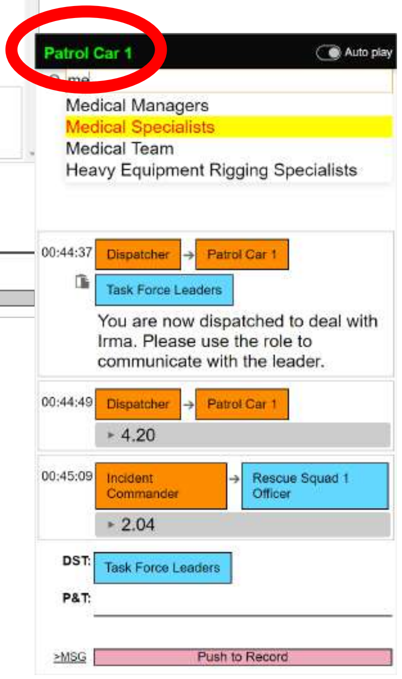


Application 1: Messaging

- Dynamically dispatch units
  - Filtered views for different roles
- Dynamically change roles or relationships

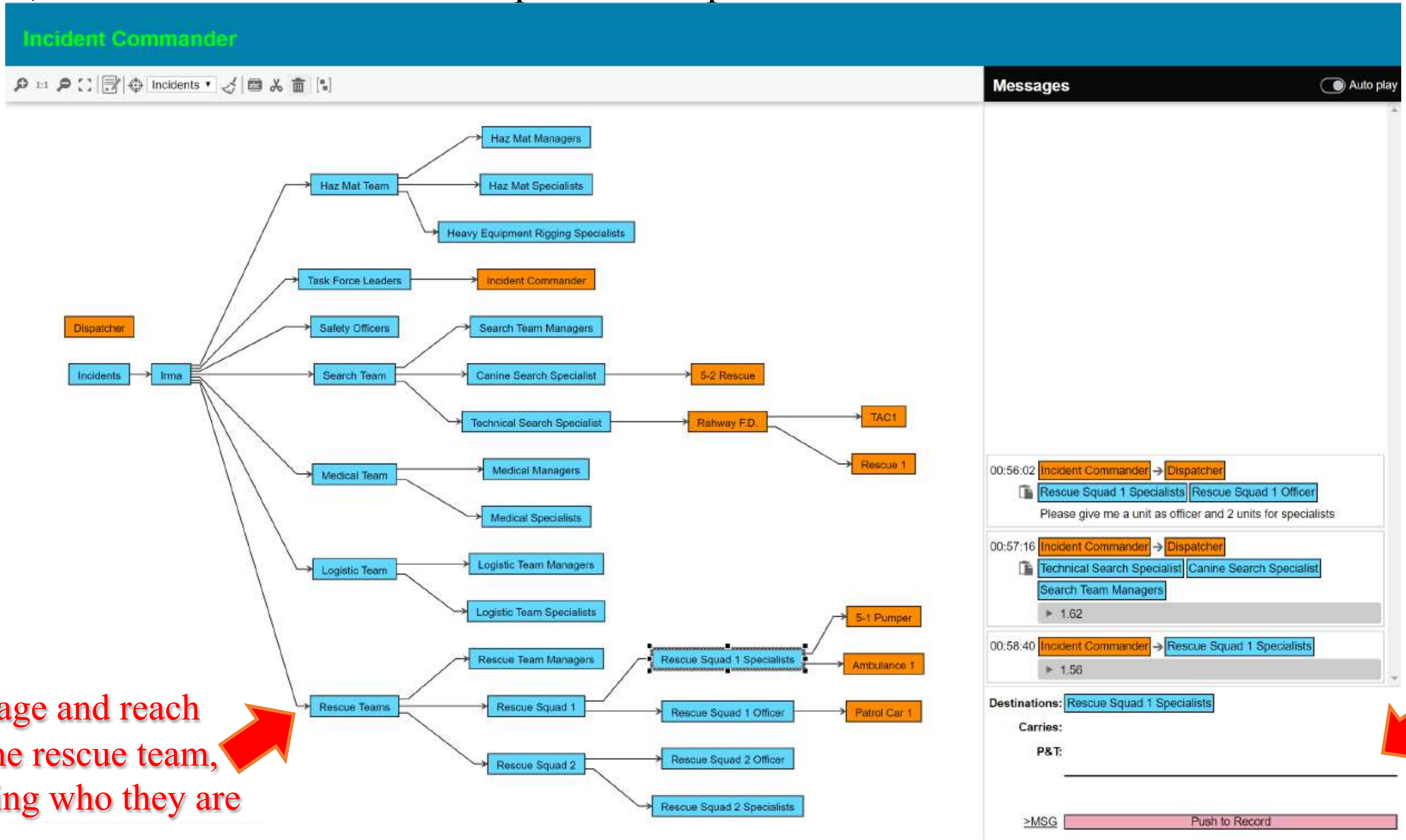






Application 1: Messaging

- Role-based communication
  - Exchange data (voice and text) based on roles
    - Send commands at different granularity: send a command and reach all the units below
  - Serialized voice playout, avoid issues of concurrent speech with push-to-talk



Send a message and reach everyone in the rescue team, without worrying who they are

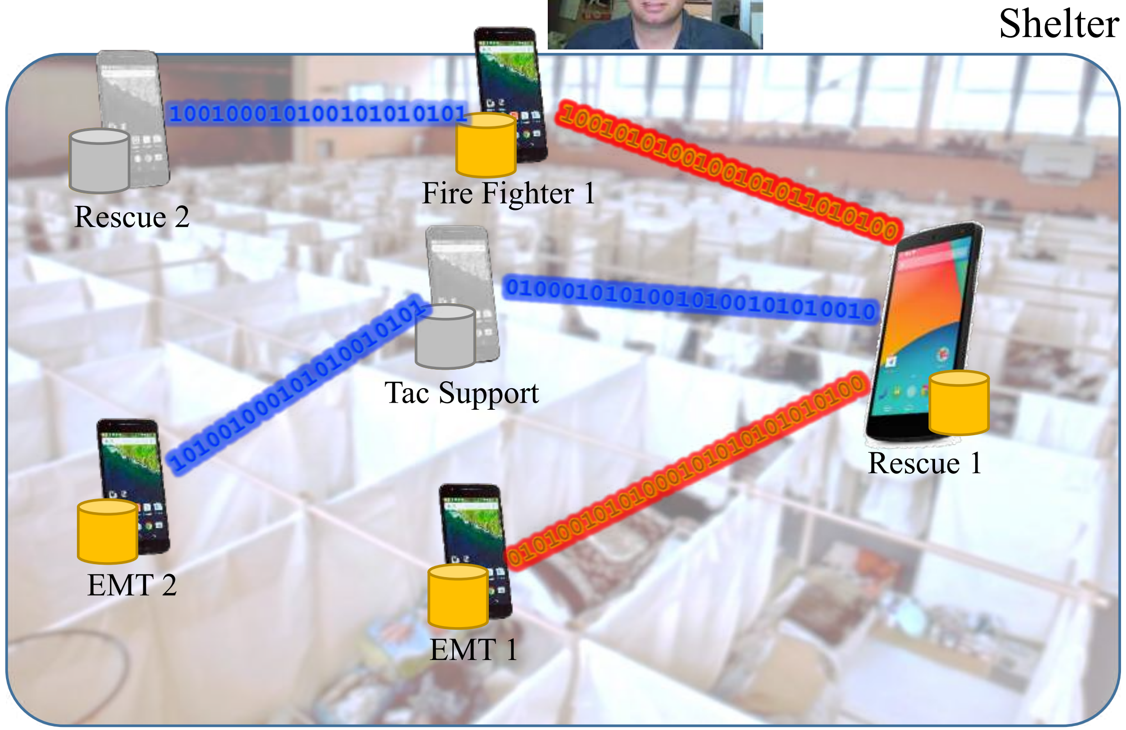
Support text & voice

Application 2: Coded Computation

- Face recognition
  - Step 0: each device has a pool of candidate people
    - Dataset: Caltech 1999 front face dataset
    - http://www.vision.caltech.edu/html-files/archive.html

  - Step 1:
    - Master (Rescue 1) queries for the available workers (send message to a predefined name)
    - Available workers respond and identify their own names
  - Step 2: Master sends the target and a subset test images to the workers according to workers' capability

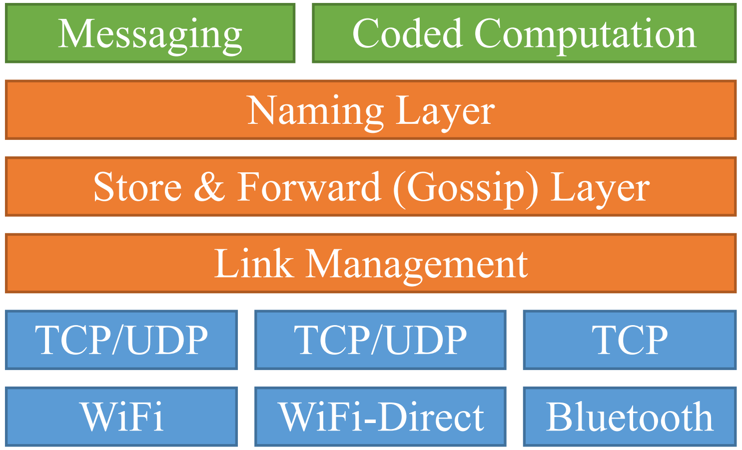
  - Step 3: Worker that identifies the target person sends the image ID back to the master

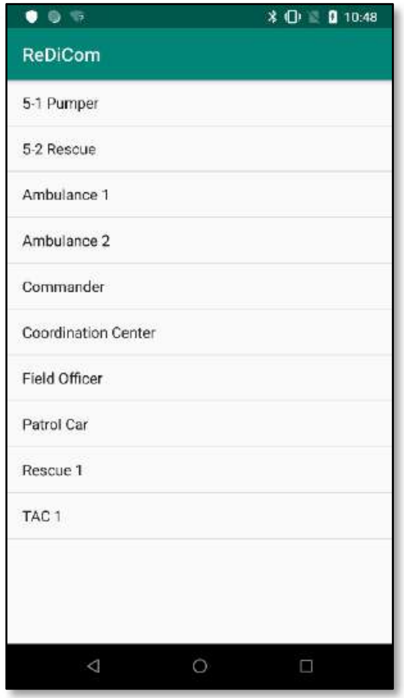


- Coded Cooperative Computation Protocol (C3P)
  - Matrix multiplication is the cornerstone of machine learning apps
  - Master does not want to wait for all the results from the workers
  - Step 2: Master divides and encodes the matrices using Fountain Codes
  - Step 3:
    - Master sends the coded sub matrices to available workers
    - Workers respond with the result of the multiplication
  - Step 4: Master decodes received matrices to get the result

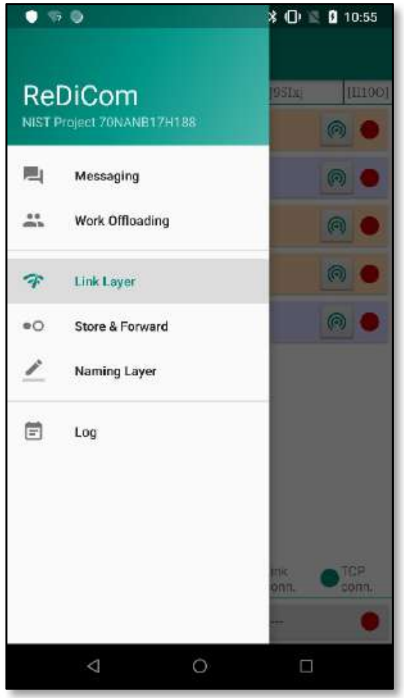
Architecture of ReDiCom

- Layered view
  - <u>Application layer:</u> messaging & coded computation
  - <u>Naming layer:</u> provides namespace management and message delivery over names
  - <u>Store & forward layer:</u> provides robust content delivery via DTN (epidemic routing)
  - <u>Link layer:</u> provides connectivity and reachability of physical nodes via Bluetooth, (legacy) WiFi and WiFi Direct, including fragmentation and aggregation, keepalive…





Role Selection



Apps and Layers

Store & Forward Layer

Link Layer

Log

Wrap Up & Plans for Next Year

- "Communication saves lives": provide a much improved framework for developing a communication system for first responders: deliver relevant information in a timely manner, even with infrastructure failures
  - Information layer for organizing teams
  - Integrated dissemination service model: publish/subscribe as a first-class capability
  - Namespace discovery and message propagation in disconnected/disruption prone environments
  - Exploited Device-Device communication: included Bluetooth and WiFi Direct
  - Exploited coding to improve communication over impaired channels
  - Used peer devices to develop D2D computation and also have secure computation capabilities, especially when infrastructure is down
- Integration of all the diverse components
- Introduce authorization to access and update namespaces; send and/or receive to a name/role
- Routing in disaster scenarios
- Coding for reliable and secure communication and computing in disaster scenarios
- Evaluating the overall performance and effectiveness of architecture