ReDiCom: Resilient Communication for First Responders in Disaster Management

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

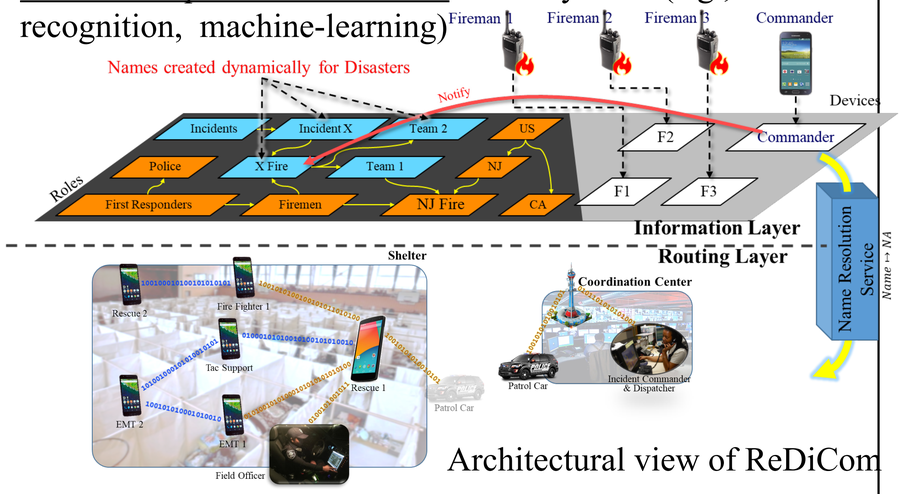
Prof. Murat Yuksel
University of Central Florida

Prof. Hulya Seferoglu
University of Illinois at Chicago

Dr. Jiachen Chen
WINLAB, Rutgers University

Motivation & Building Blocks of ReDiCom

- **Keys to effective response to an incident**
- Effective communication *within and among* first responder team members: *teams formed dynamically*
- Comm. among law enforcement, emergency, EMT, transport *& other services* and *public;* based on nature & scale of incidents
  - Infrastructure may be impacted: provide *resilience*
  - *Limited computation resources* for heavy tasks (e.g., face recognition, machine-learning)



Architectural view of ReDiCom

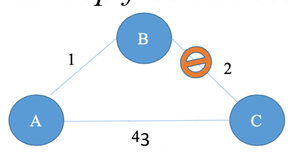
ReDiCom Architecture

- Information Layer
  - Supports dynamically formed teams: Graph-based namespace for flexibility; Pub/sub for timeliness
- Network Layer
  - Supports data dissemination: connected and delay-tolerant links
  - Supports data transmission over flat names (identifiers)
- Link Layer
  - Exploit Device-to-Device communication: Complement infrastructure-based communications
- Coded Device-to-Device Computation
  - Distributed computing for first responders with little or no infrastructure support
  - Higher reliability, smaller delay, lower communication cost

Please see page 2 for Consensus Protocol and ReDiCom Implementation

Binary State Distance Vector Routing

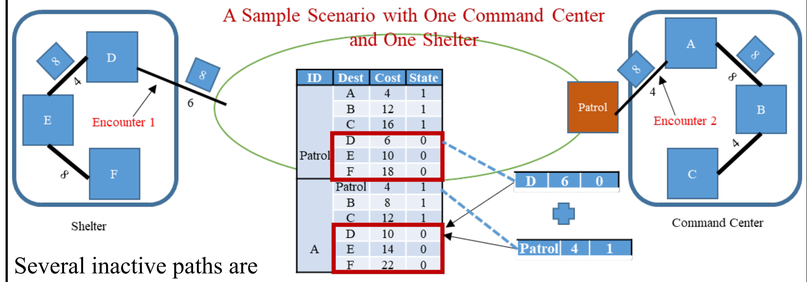
- Conventional *epidemic routing* for DTNs is too costly, causes too many data packet replicas.
- We design Binary State DVRP, a nearly unicast routing that combines:
  - *distance vector* entries with binary state and
  - conventional *epidemic* routing
- A dynamic delay tolerant networking (DTN) routing protocol that:
  - Can work with D2D links (appearing as *TCP-like sockets* to net-layer)
  - Can work with *heterogeneous D2D links* (e.g., Bluetooth or WiFi-Direct)
  - Can handle node *failures* and *resurrections*
  - Is *loop-free* and *scalable*



ID	Dest	Cost	State
A	B	1	1
	C	3	0
B	A	1	1
	C	2	0
C	B	2	0
	A	3	1

State:
0 – Inactive
1 – Active

Two types of **DVT** entries, **active** and **inactive**, for constructing **FT**: active for current paths, inactive for paths that were complete in the past.



ID	Dest	Cost	State
Patrol	A	4	1
	B	12	1
	C	16	1
	D	6	0
	E	10	0
	F	18	0
A	Patrol	4	1
	B	8	1
	C	12	1
	D	10	0
	E	14	0
	F	22	0

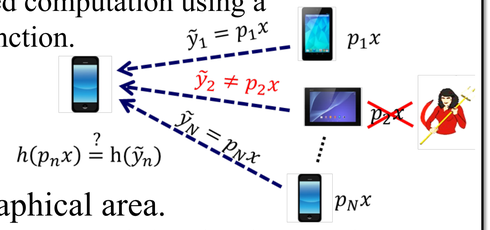
Several inactive paths are installed by nodes at the command center by following binary-state precedence rules. After convergence: (1) Patrol *active-forwards* towards the command center. (2) A, B and C could *inactive-forward* towards the shelter.

Future work

- Enhanced buffering rules on data messages through *priority queues* based on forming a precedence rule
- ID-based group separation for aggregation in tables

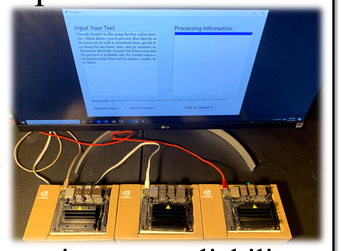
Coded Computation & Communication

- Computation:
  - Distributed computing can be crucial in first responder and PSC systems when there is little or no infrastructure support.
  - Our work:
    - *Divide* a computationally intensive task into small subtasks.
    - Offload each subtask to multiple first responder/civilian devices via *coding* to improve resiliency of the system.
    - Develop secure coded computation using a *light-weight hash* function.
- Communication:
  - A group of first responder devices is in the same geographical area. A common content, e.g., crucial voice instructions, can be broadcast over infrastructure-based links. D2D connections and coding improves reliability.



  - Question: which coded packets over each interface?
  - Approach:
    - Convert the problem to a two-layer connected graph (one for cellular, one for D2D).
    - Maximum independent set gives packets and network codes to be transmitted for cellular and D2D.

Demo: Distributed Text-to-Speech



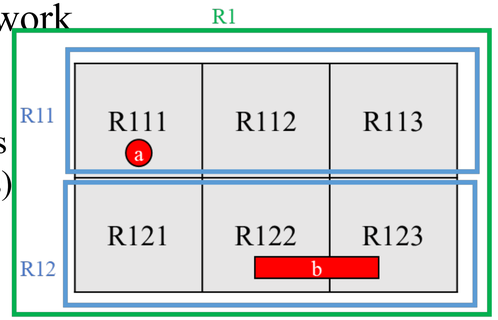
Future work

- Development of coding techniques to improve reliability of our architecture.
- Heterogenous capabilities including varying # of communication interfaces, battery levels & computation power.

Consistent Dissemination using Consensus

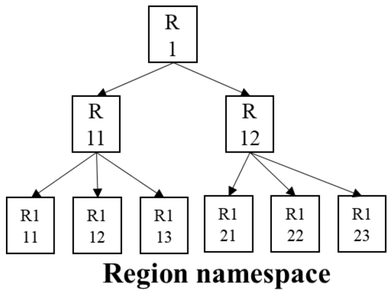
➢ Consistent dissemination in disconnected network

- Delay-tolerant network environment; no network infrastructure or clock synch.
- Application: Map-based geo-tagging for first responders
- Number of mobile users (first responders, commanders) disseminate critical information to each other
  - Updates to a common shared database.
  - Consistency is important and challenging

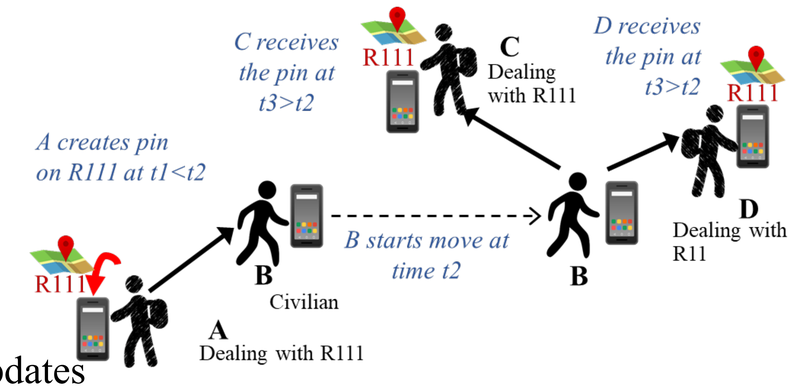


Region-ed map (base + data)

➢ Goal: ensure delivery and consistency of the data across users

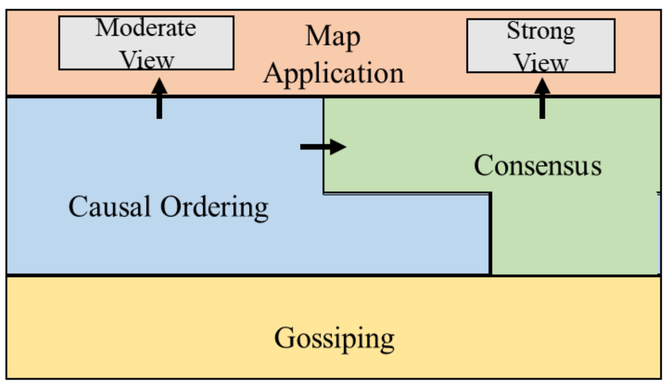


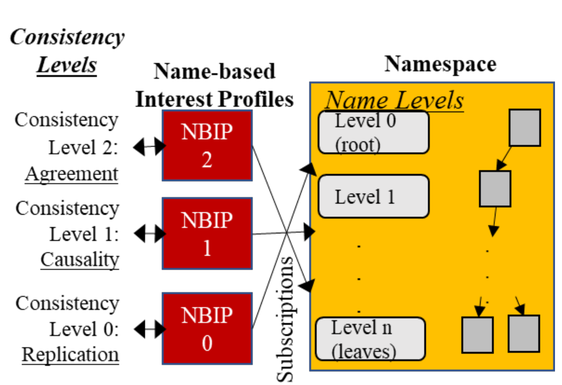
Region namespace



➢ Achieving Consistent Updates

- **Consistency level 0: Replication** – make sure users receive created updates
- **Consistency level 1: Causality** – users get causally ordered view of updates
- **Consistency level 2: Agreement** – users get an agreed upon view of updates Name-based Interest Profile (NBIP) limits each user's participation according to relevant namespace subsets
- Every user has map & namespace offline; Dynamic updates - disseminated consistently
- Consensus –use and enhance One-Third Rule (OTR)-suitable for asynchronous environment



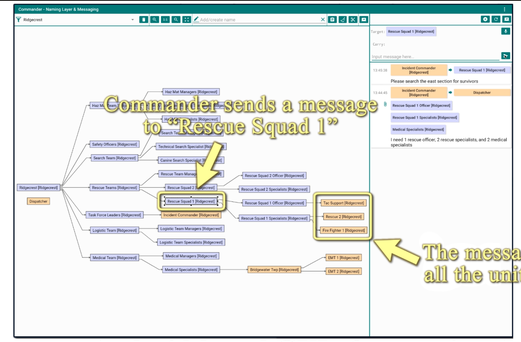


Our method achieves much higher relevant completeness and thus consistency, with much less usage of mobile phone storage

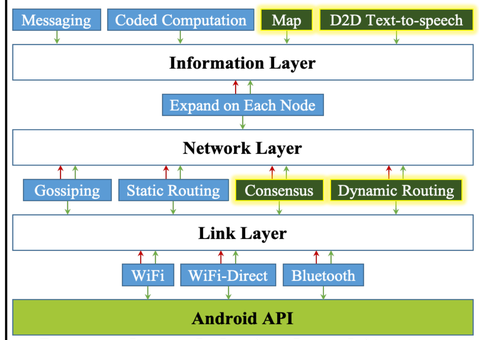
Integration of ReDiCom Capabilities in Implementation

➢ ReDiCom Demo in PSCR'19

- Allow commanders to *dynamically* instantiate templates (preplans)
- Send messages to *different levels* in (and along) the command chain via graph-based namespace
- *D2D* communication *& DTN* for use in infrastructure-less environments



Android-based GUI of messaging App



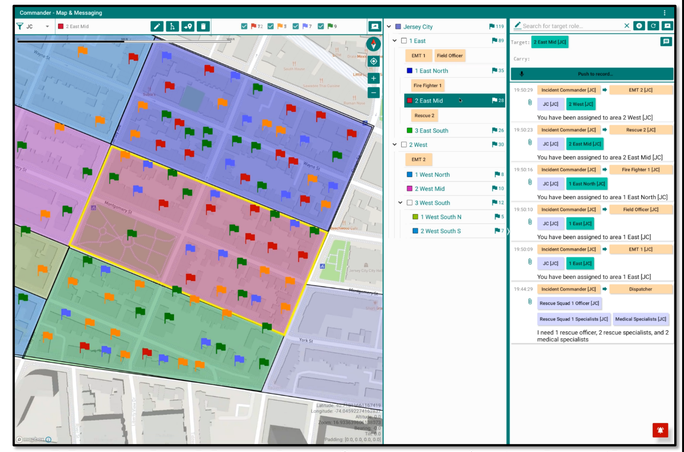
Layered, modularized architecture

➢ Improvements on ReDiCom architecture

- *Modularized* design
  - Enabled different teams develop different components
- Better inter-module communications
  - Remove heavy logic from UI thread, better *responsiveness*
  - *Address dead-lock* issues with RCU & message queue
- New packet library, *avoid copying* between layers
- Android-based GUI, *better stability*

➢ Map tool

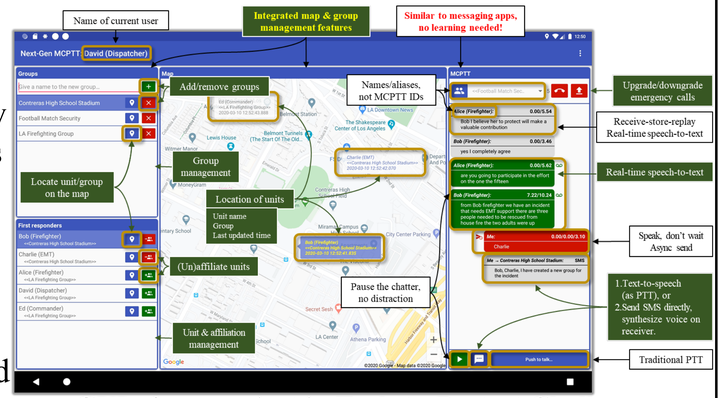
- Create/edit a *hierarchy* of areas
- *Assign units* to areas at different levels
- Add/update markers & send messages *along the location hierarchy*
- Topic hierarchy (markers) & recipient hierarchy (messages) *unified in the graph-based namespace*
- Offline mode (*D2D & DTN* support)



Map tool: add markers & communicate along the location hierarchy

➢ Tech-to-Protect Challenge

- We address 3 issues with MCPTT
  - Messages cannot be stored/replayed
  - Messages cannot be paused, distracting
  - Difficult to synchronize with high latency
- By: (1) storing the messages on the devices for replay & pause, (2) allowing user talk without acquiring a floor, (3) real-time text-to-speech & speech-to-text, (4) map functionality, and (5) other improvements like call prioritization, etc.
- First Prize in NY site; "Superior" award in national contest (best in contest #2)



GUI of our project in Tech-to-Protect Challenge