DistressNet-NG: A Resilient Broadband Communication and Edge Computing Framework for FirstNet

Radu Stoleru,
stoleru@cse.tamu.edu

Wei Zhang, Ala Altaweel, Mengyuan Chao,
Suman Bhunia, Mohammad Sagor

ATM | TEXAS A&M UNIVERSITY

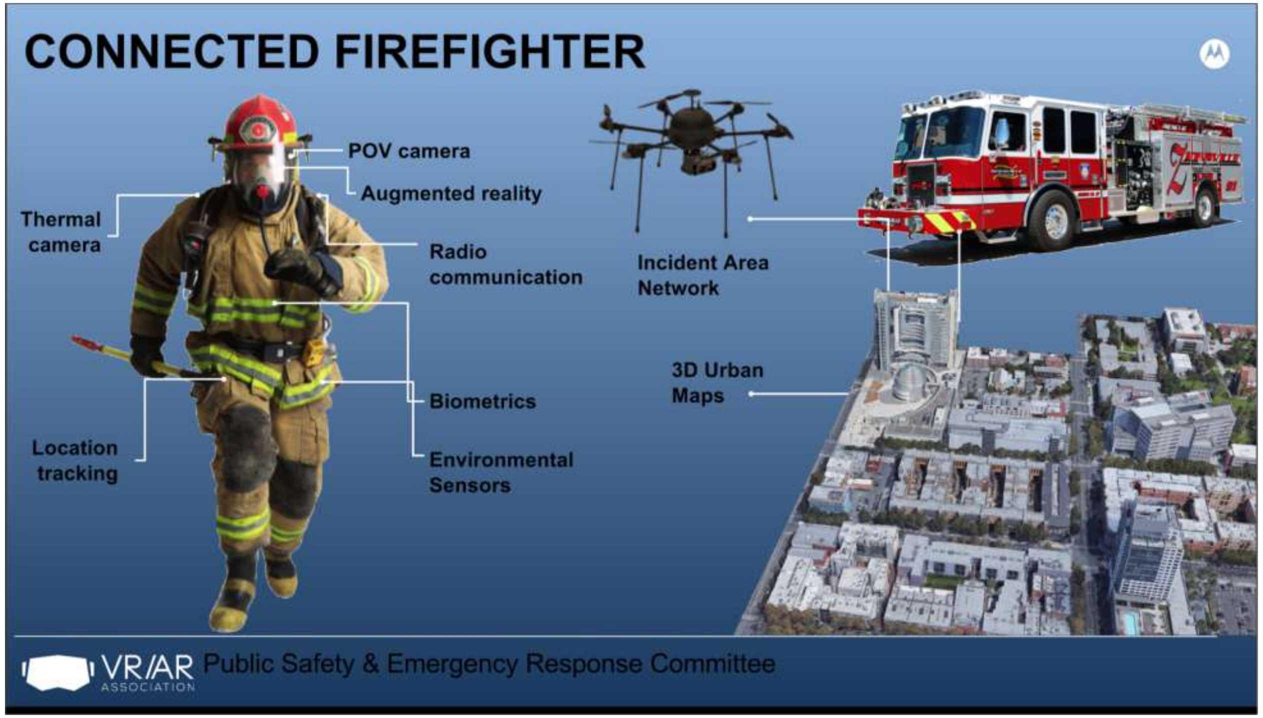
DISCLAIMER

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

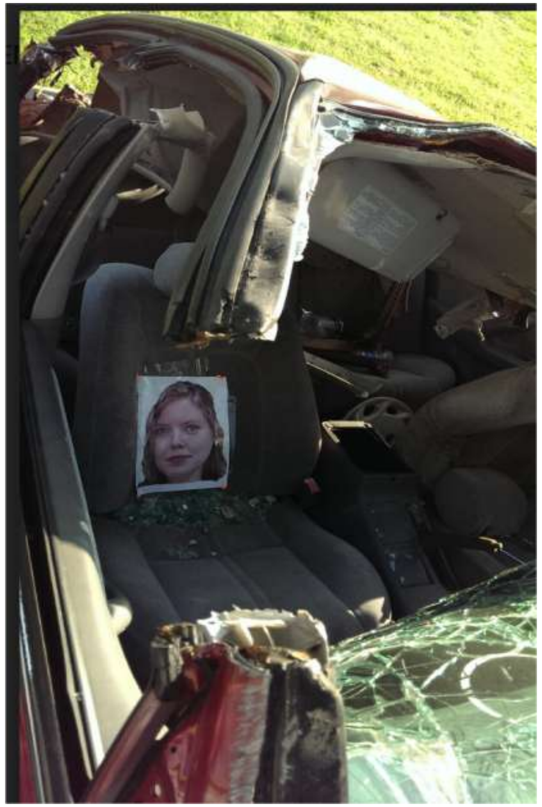
Targeted Applications

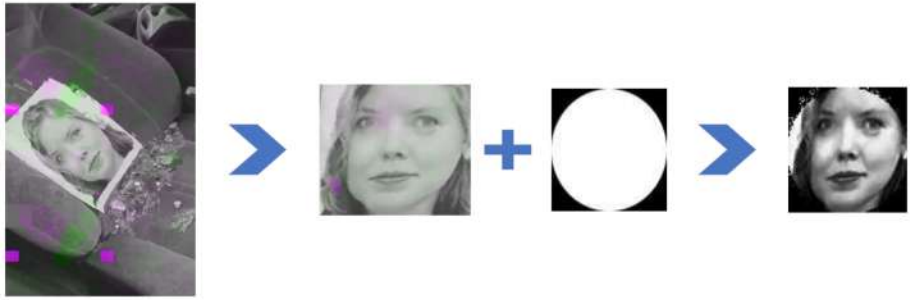
- VR/AR: Video processing in real-time (body-worn cameras) or batch (stored videos on mobile devices or storage devices)
  - Face recognition
  - AR
  - AI



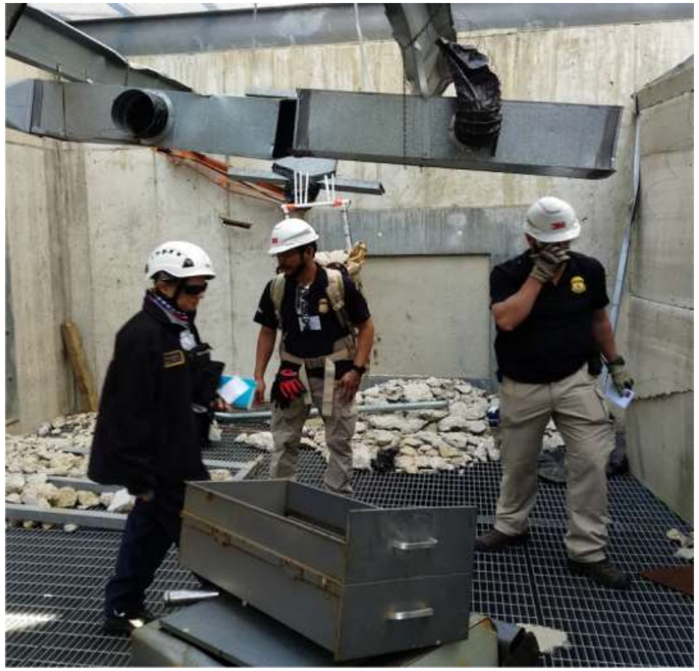
Initial Deployment

- Disaster City (Texas A&M Univ.): March 23, 2018





Second Deployment



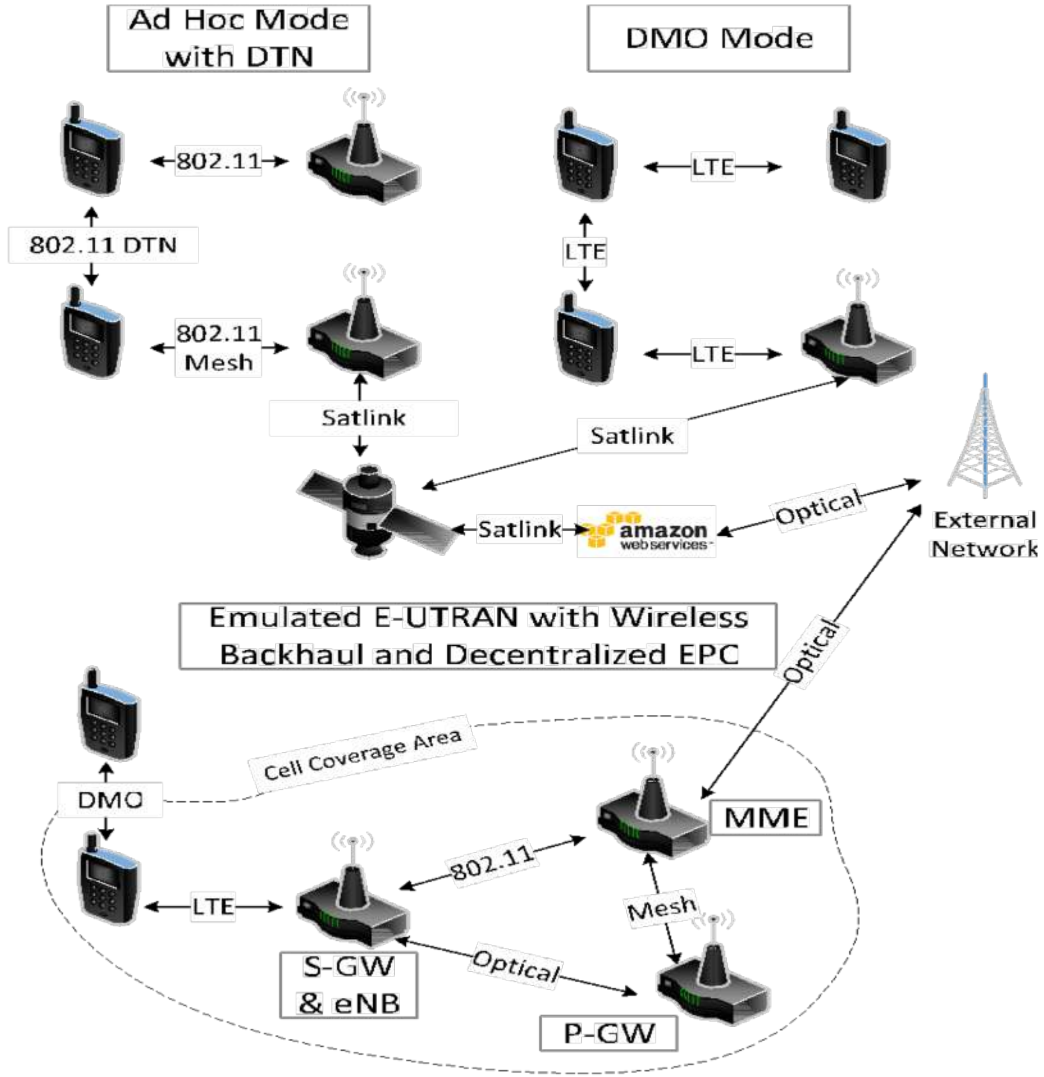
- Disaster City (Texas A&M University): March 26-28, 2019

Outline

- DistressNet-NG Architecture
- Resilient Sockets
- Naming and service discovery
- Mobile Distributed File System (MDFS)
- Edge Computing
  - Real Time Stream Processing (MStorm)
  - Batch Processing (MMR)

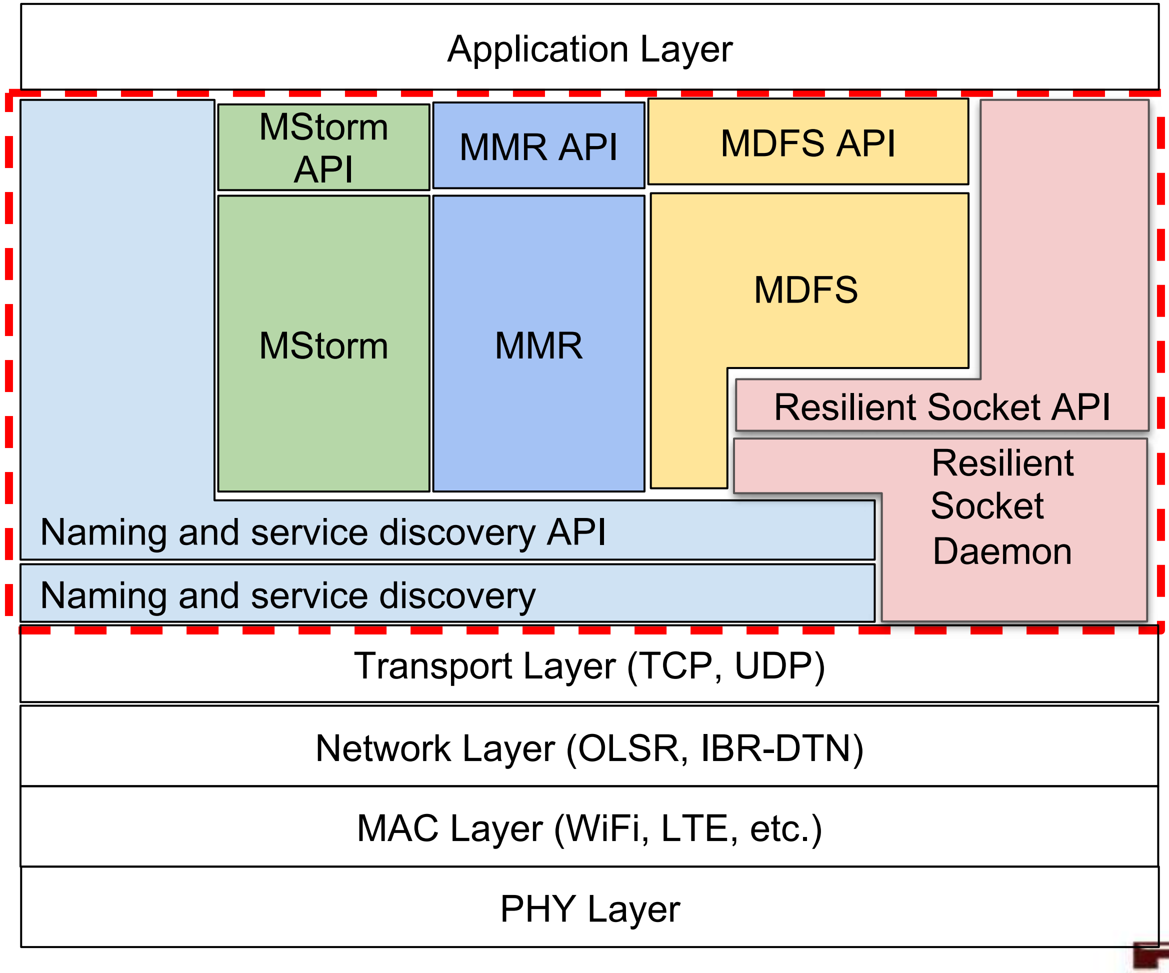
DistressNet-NG Architecture

- Hardware architecture:
  - – UE’s
  - – HPC nodes
  - – LTE
  - – LTE DMO
  - – WiFi mesh
  - – WiFi DTN



DistressNet-NG Architecture

- Software architecture:
  - Resilient Socket (RSock)
  - Naming and Service discovery
  - Mobile Distributed File System (MDFS)
  - Real-Time Stream Processing (MStorm)
  - Batch Processing (MMR)

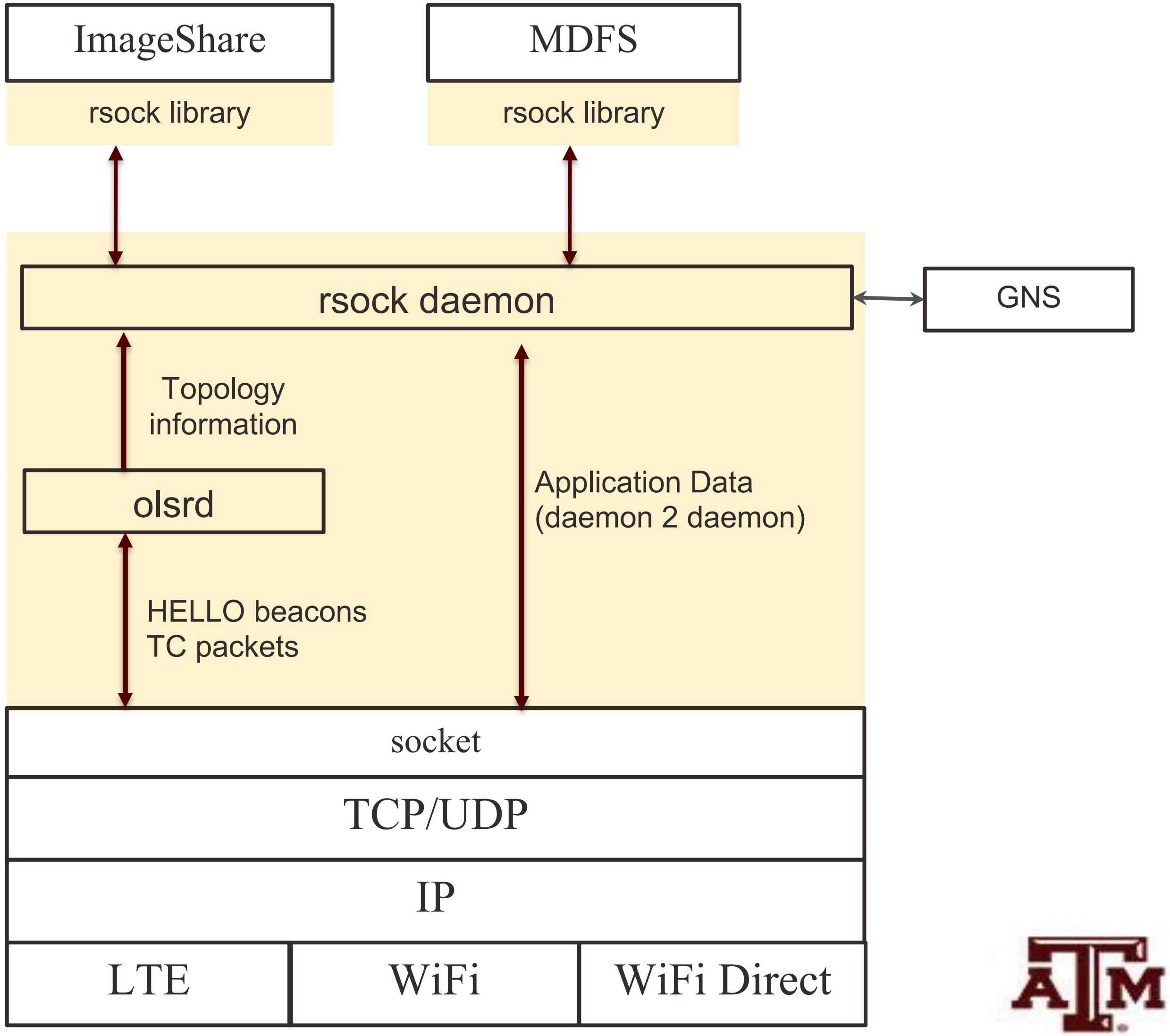

Application Layer
MStorm API
MMR API
MDFS API
MStorm
MMR
MDFS
Resilient Socket API
Resilient Socket Daemon
Naming and service discovery API
Naming and service discovery
Transport Layer (TCP, UDP)
Network Layer (OLSR, IBR-DTN)
MAC Layer (WiFi, LTE, etc.)
PHY Layer

Outline

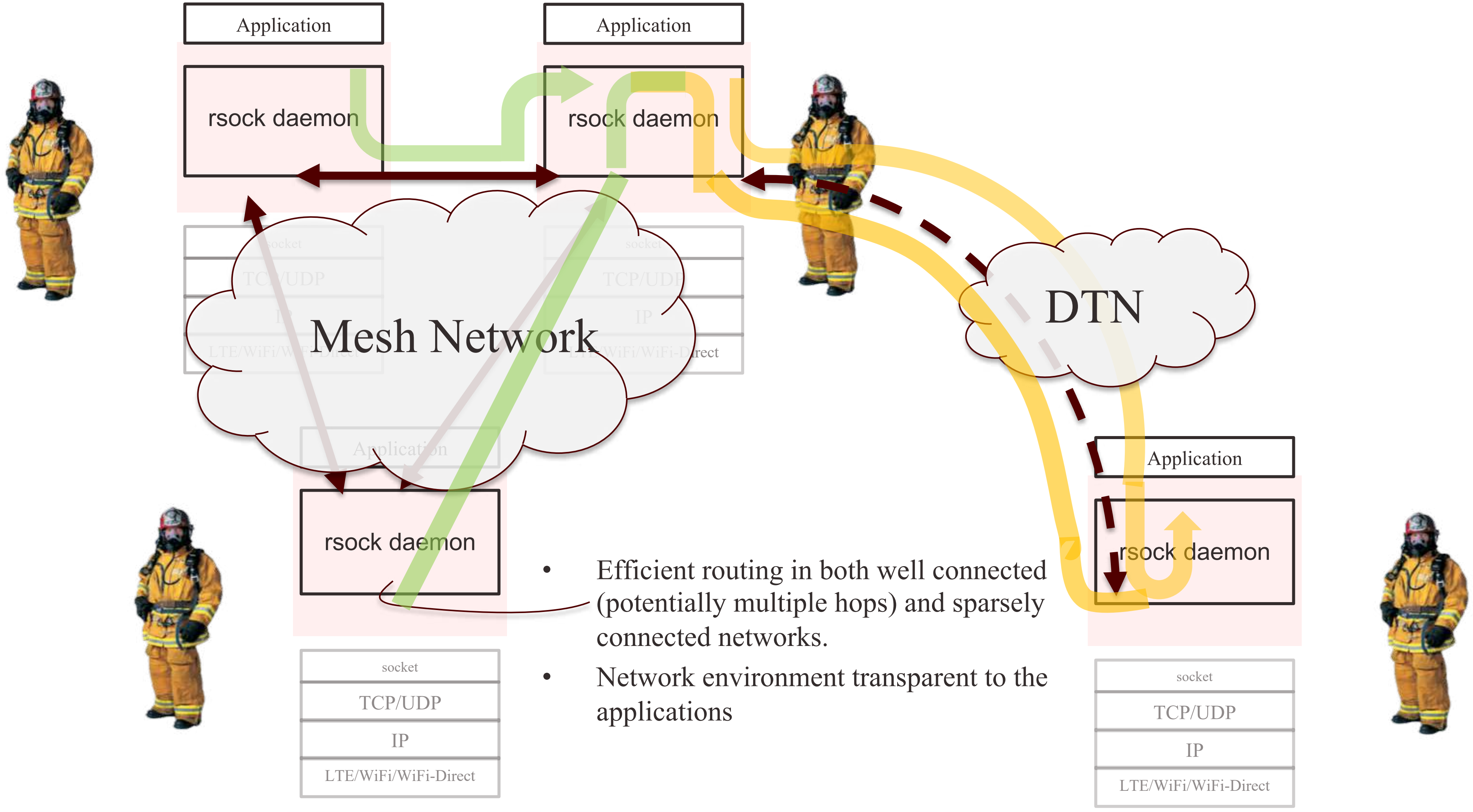
- DistressNet-NG Architecture
- Resilient Sockets
- Naming and service discovery
- Mobile Distributed File System (MDFS)
- Edge Computing
  - Real Time Stream Processing (MStorm)
  - Batch Processing (MMR)

Resilient Socket Architecture

- Overall architecture:
  - rsock library
  - rsock daemon (mesh and DTN)
  - GUID based addressing (no IPs)
  - GNS service
- Switching based architecture
  - Maintains network states
  - Intelligently decides # of replications
- Run as a system service (or a daemon process)



Resilient Socket Architecture



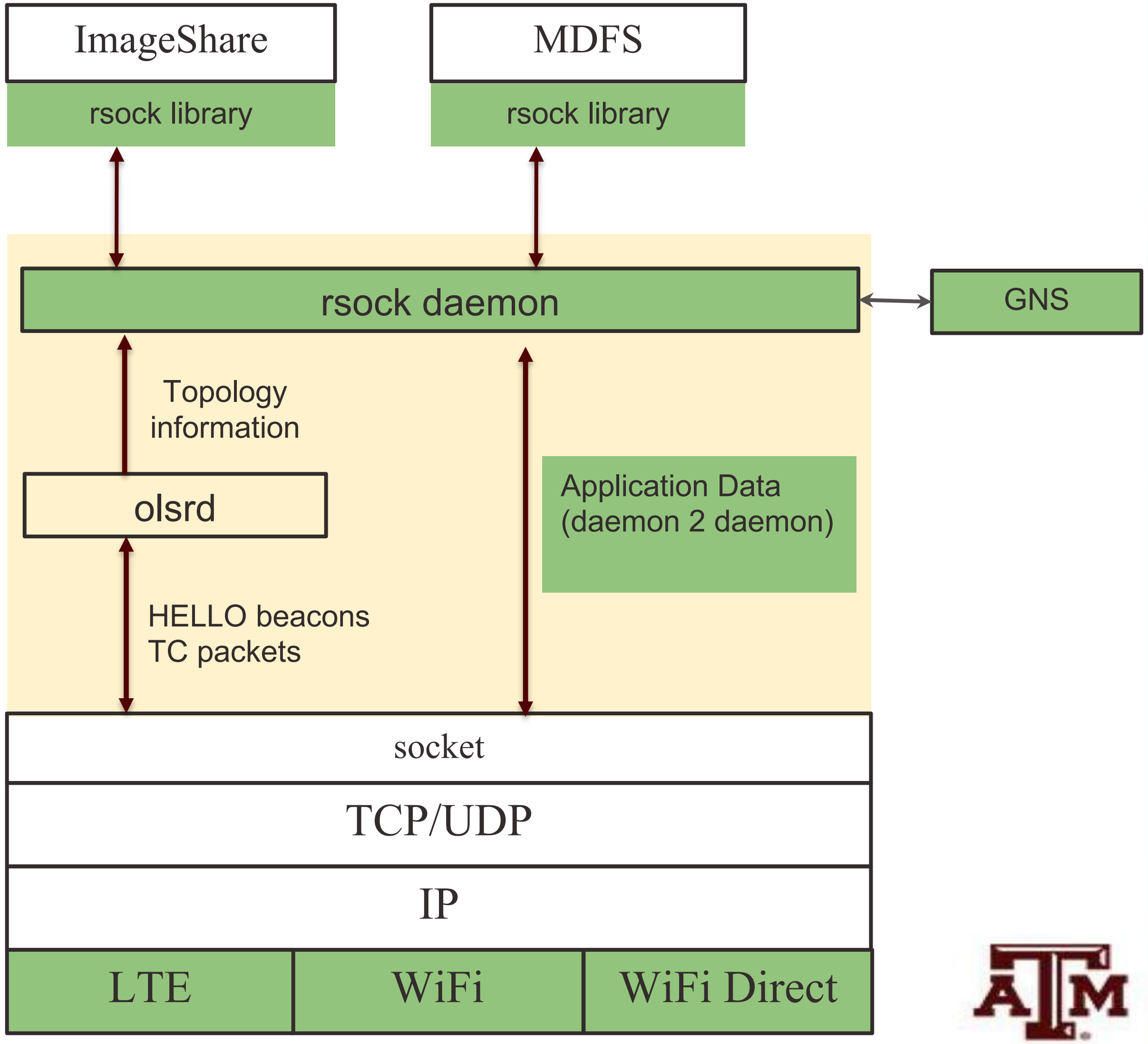
- Efficient routing in both well connected (potentially multiple hops) and sparsely connected networks.
- Network environment transparent to the applications

Resilient Socket Design

- rsock library (unified interface):
  - – Objective: network environment transparent to the applications
  - – Applications provide QoS requirements
    - E.g., **TTL**, reliably delivery, ordered delivery
- rsock daemon:
  - – Objective: ID-based routing for Mesh and DTN networks
  - – Leverages available interface (Wi-Fi or LTE) for packet delivery

Resilient Socket Status

- Finished Parts:
  - Integration with GNS (both library and daemon)
  - Integration with ImageShare and MDFS (Provides ID-based topology)
  - Seamless switching between interfaces (LTE, Wi-Fi, Wi-Fi Direct)
  - Multi-hop routing over the mesh network
- To be done:
  - DTN routing
  - Provides additional QoS (reliably delivery, ordered delivery)



Outline

- DistressNet-NG Architecture
- Resilient Sockets
- Naming and service discovery
- Mobile Distributed File System (MDFS)
- Edge Computing
  - Real Time Stream Processing (MStorm)
  - Batch Processing (MMR)

Why Naming Service is Needed?

- ID-based routing is required because:
  - Handle IP change (mobility, reconnection, etc.)
  - Leverage multi-interface (LTE and/or Wi-Fi)
  - Seamless switching among interfaces
  - Application should be transparent to interface change
  - DTN routing needs a unique ID for routing

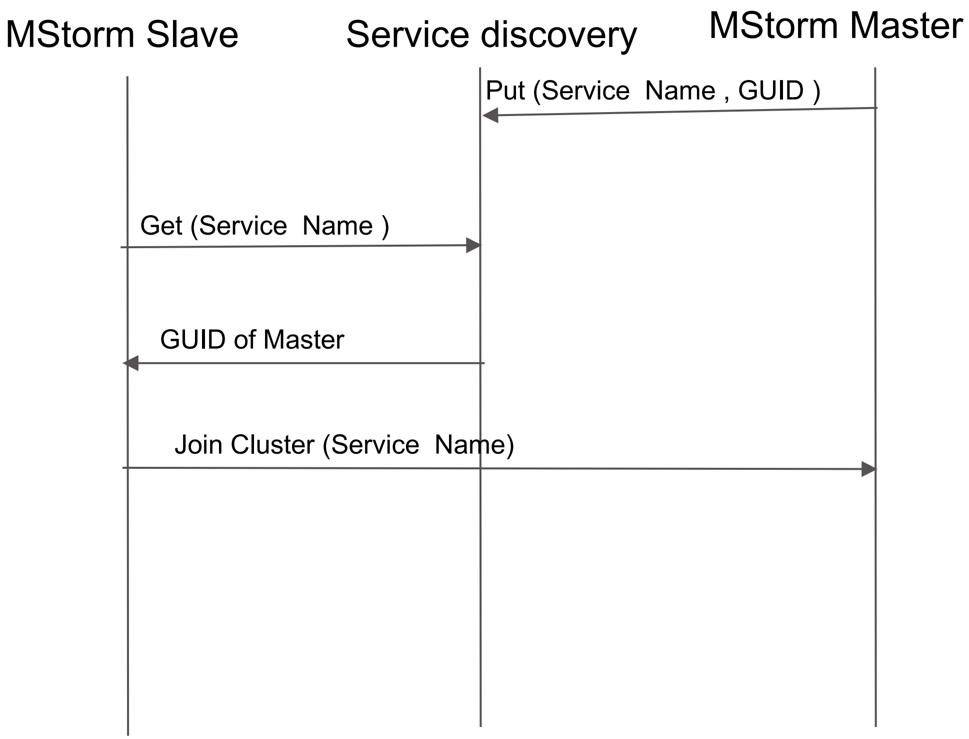
Global Naming Service (GNS)

- Current Domain Name system (DNS) is not efficient for mobility
  - frequent change in device location means more update
  - TTL based caching is not optimal
  - Hierarchical name resolution architecture is not fault tolerant
  - Static placement of servers
- Global Naming Service (GNS) is proposed as a solution* to provide:
  - Rapid translation of identity to location
  - Support seamless mobility to application
  - Scalable, geo-distributed, federated
  - Automatically controls massive name replication

*Sharma, A., Tie, X., Uppal, H., Venkataramani, A., Westbrook, D., & Yadav, A. (2015). A global name service for a highly mobile internetwork. *ACM SIGCOMM Computer Communication Review*, *44*(4), 247-258.

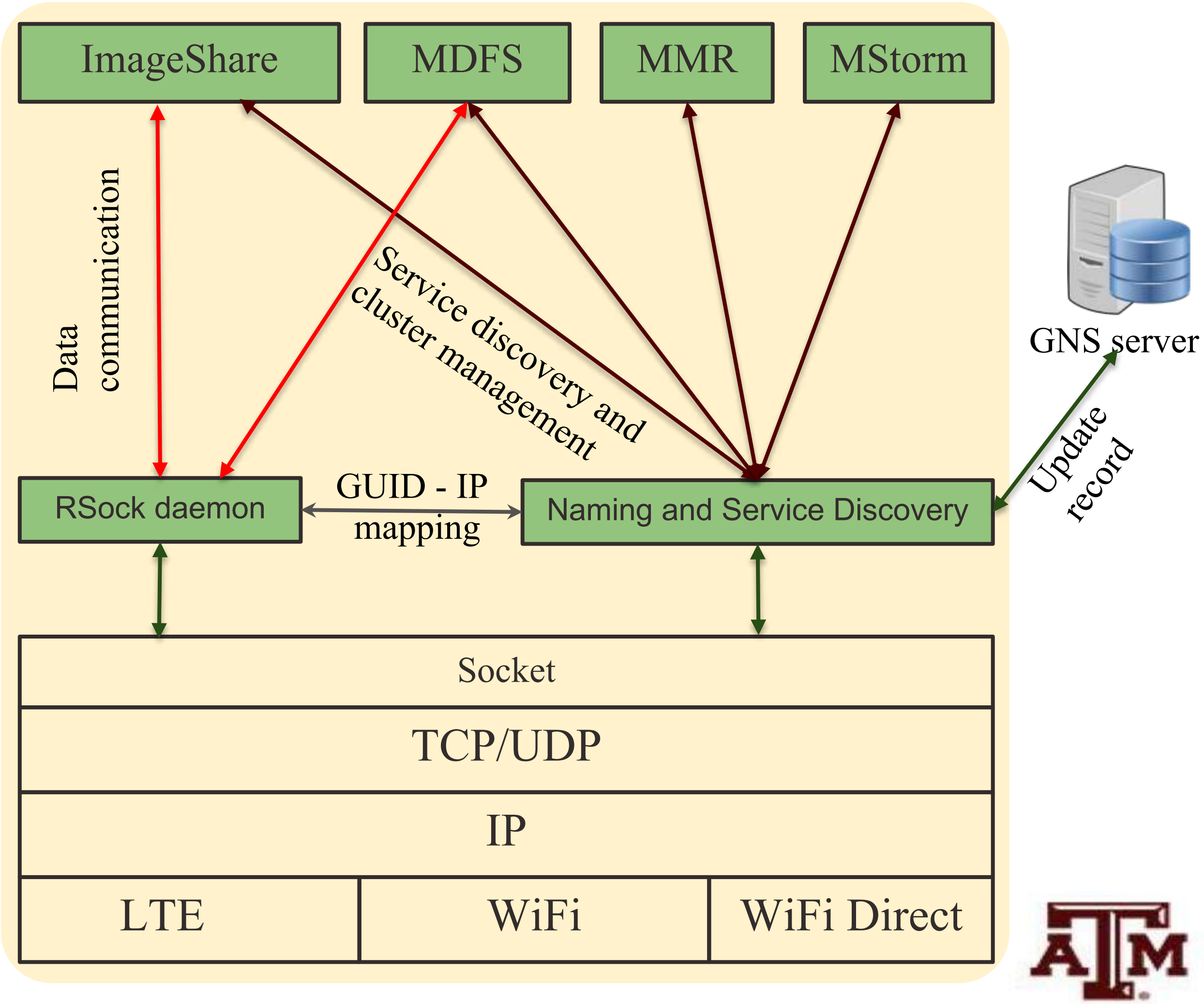
Service Discovery Use Case

- After slave boots up or relocates, it needs to join an existing cluster
  - Cluster master's ID is needed to join
    - Obtained through service discovery
- The naming and service discovery service provides interface for service discovery.
  - Runs on each device
  - Service information and role is stored in GNS server



Integration with other services

- Each device runs Naming and Service discovery service
  - IP address and service records are updated dynamically
- Service discovery and Cluster
  - Service discovery and cluster management use GUID
  - ImageShare, MDFS, MMR and MStorm use this service to find peers
- Naming and IP translation
  - RSock use this service to translate GUID to IP and IP to GUID
- DNS functionality
  - GNS runs a DNS server at the Manpack
  - Dynamically map Name and IP
  - MMR uses DNS based destination



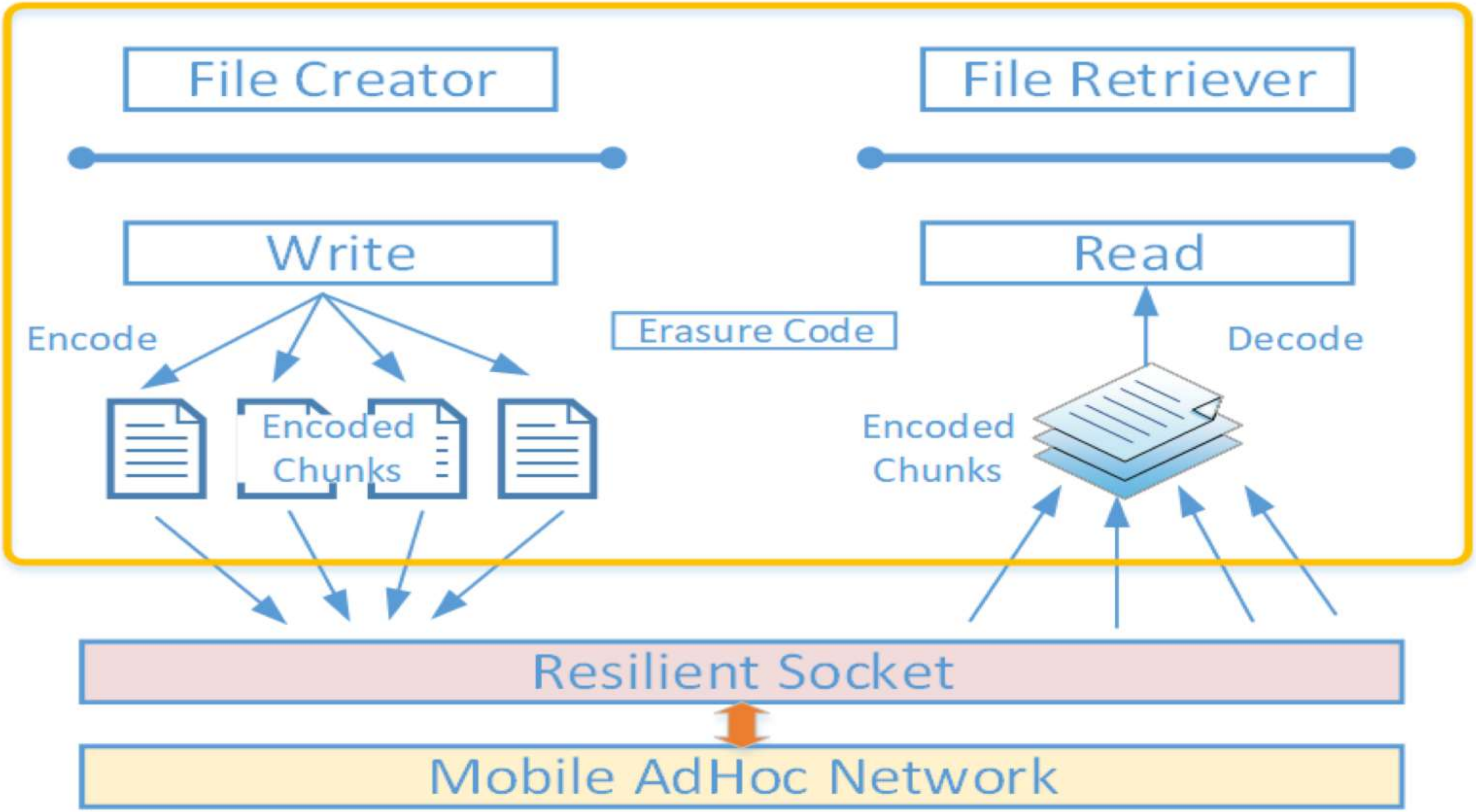
Outline

- DistressNet-NG Architecture
- Resilient Sockets
- Naming and service discovery
- Mobile Distributed File System (MDFS)
- Edge Computing
  - Real Time Stream Processing (MStorm)
  - Batch Processing (MMR)

Mobile Distributed File System (MDFS)

- Decentralized
  - No centralized NameNode (as HDFS)
  - Applicable to mobile adhoc network
- File create & retrieve (write & read)
  - k-out-of-n system
    - Retrieve any k out of n fragments = retrieve the whole file
    - Erasure code – Reed Solomon

MDFS Architecture



MDFS Status

- **Distributed Storage**
  - Implemented a robust file splitting, distributing and retrieving system.
- **Network Failure/Delay Tolerance**
  - Adopted Resilient Socket API for file creation/retrieval to support network disconnection
- **Directory Service**
  - Added distributed metadata keeper that provides directory service to nodes.
- **Access Control List**
  - Implemented UNIX-like Access Control features(owner group, world).
- **Robust Algorithm**
  - Optimum node selection algorithm for file storage, adaptive to frequent topology change..

MDFS Future Plans

- **Command Line based Interface**
  - Create and retrieve file using command line interface.

- **Robust Access Control List**
  - More permission features.

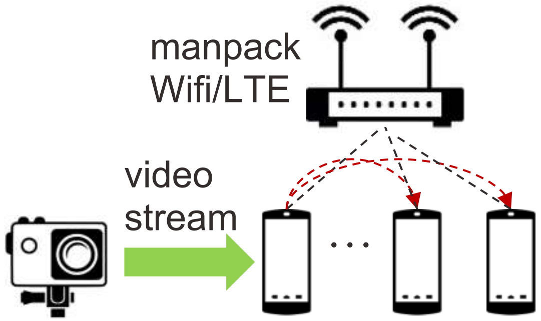
- **Debug and Large-Scale testing**
  - Aiming to deploy for test on a real disaster response scenario.

Outline

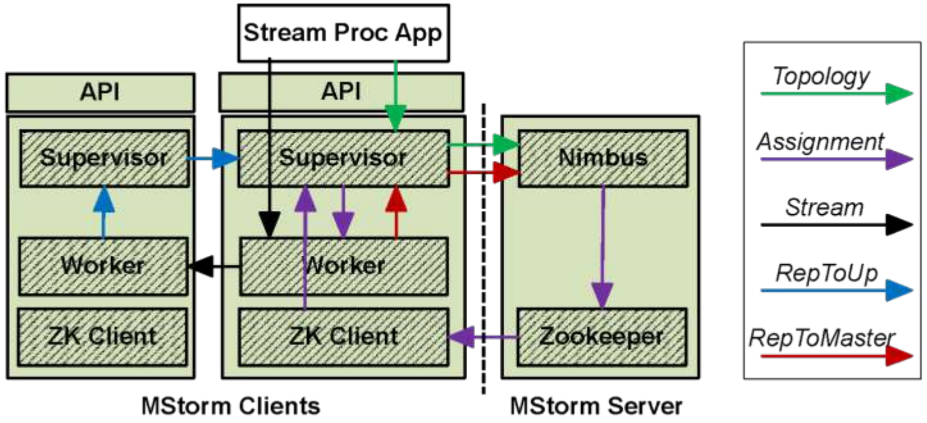
- DistressNet-NG Architecture
- Resilient Sockets
- Naming and service discovery
- Mobile Distributed File System (MDFS)
- Edge Computing
  - Real Time Stream Processing (MStorm)
  - Batch Processing (MMR)

MStorm

- **Architecture**

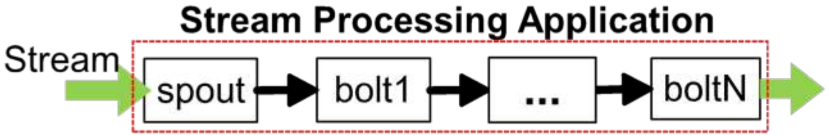


Hardware Architecture



Software Architecture

- **MStorm APP model and characteristics**

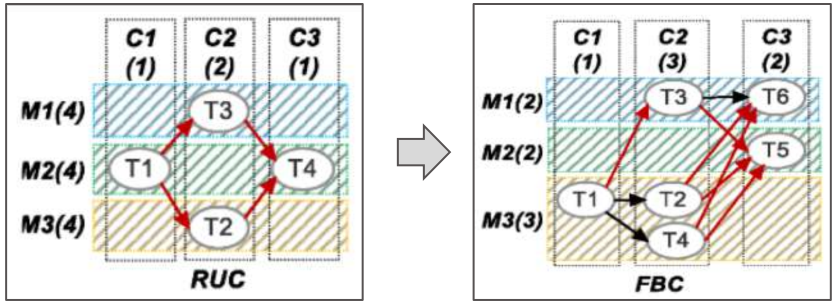


App characteristics:
- Flow-based (Network of "black box")
- Low Response Time (~1s)
- Comp. Intensive (> capacity of one dev.)
- Workload Changes (Unpredictable Input)

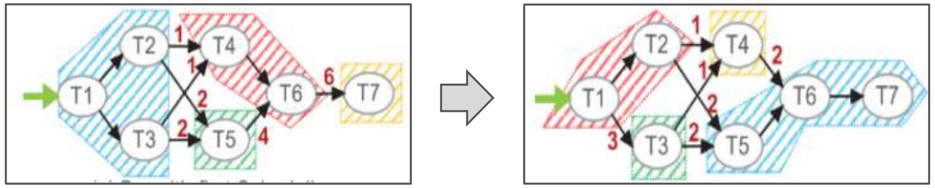
MStorm Middleware:
- Offload to other devices
- Deal with workload changes
- Deal with resource/network changes
- Easy-to-use API

MStorm Design

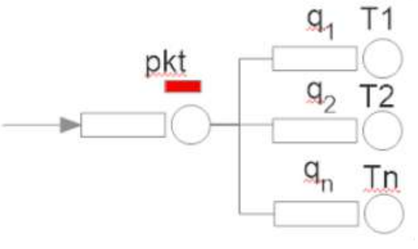
- **Feedback Based Configuration**
  - Parallelism for components and executors for devices are re-configured based on the feedback



- **Feedback Based Task Assignment**
  - Tasks are re-assigned to devices to minimize the inter-node traffic based on the feedback



- **Feedback Based Stream Grouping**
  - Tuples from upstream are dynamically sent to the downstream tasks based on the feedback



Send packet to the one with minimum expected delay !

MStorm Status

- **Finished parts:**
  - MStorm server and client that deal with workload and computing resource dynamics
  - A video face detection application (i.e., GReporter) running on MStorm
  - GReporter integrated with helmet sports camera
  - MStorm integrated with GNS for service discovery and GUID

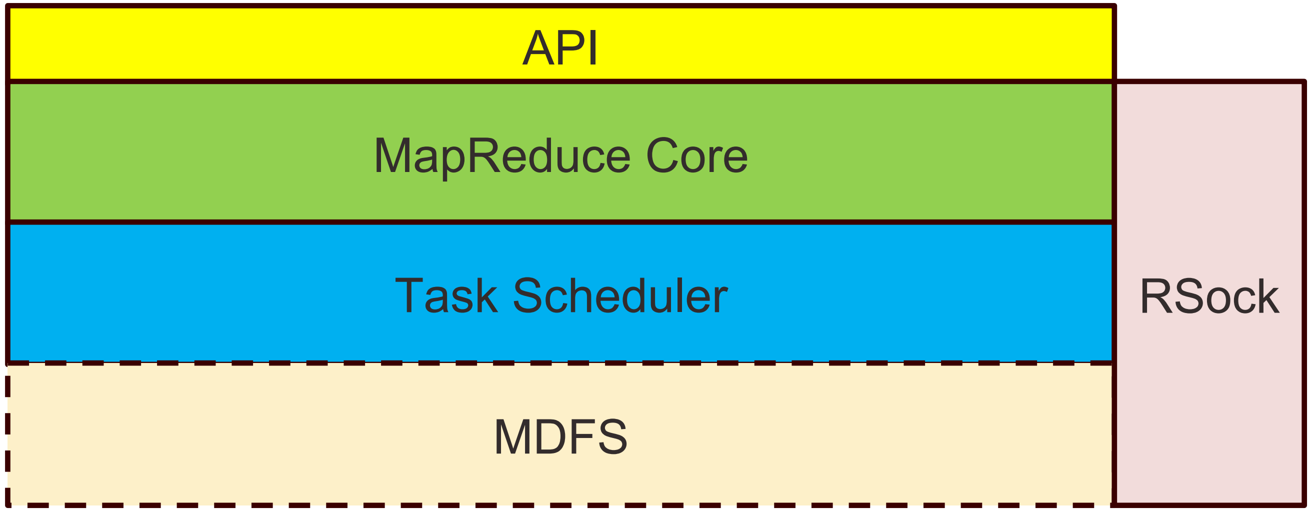
- **To be done:**
  - Robust MStorm that deals with bad network connectivity
  - Hybrid MStorm that runs on mobile devices and edge servers
  - More Applications using MStorm

Outline

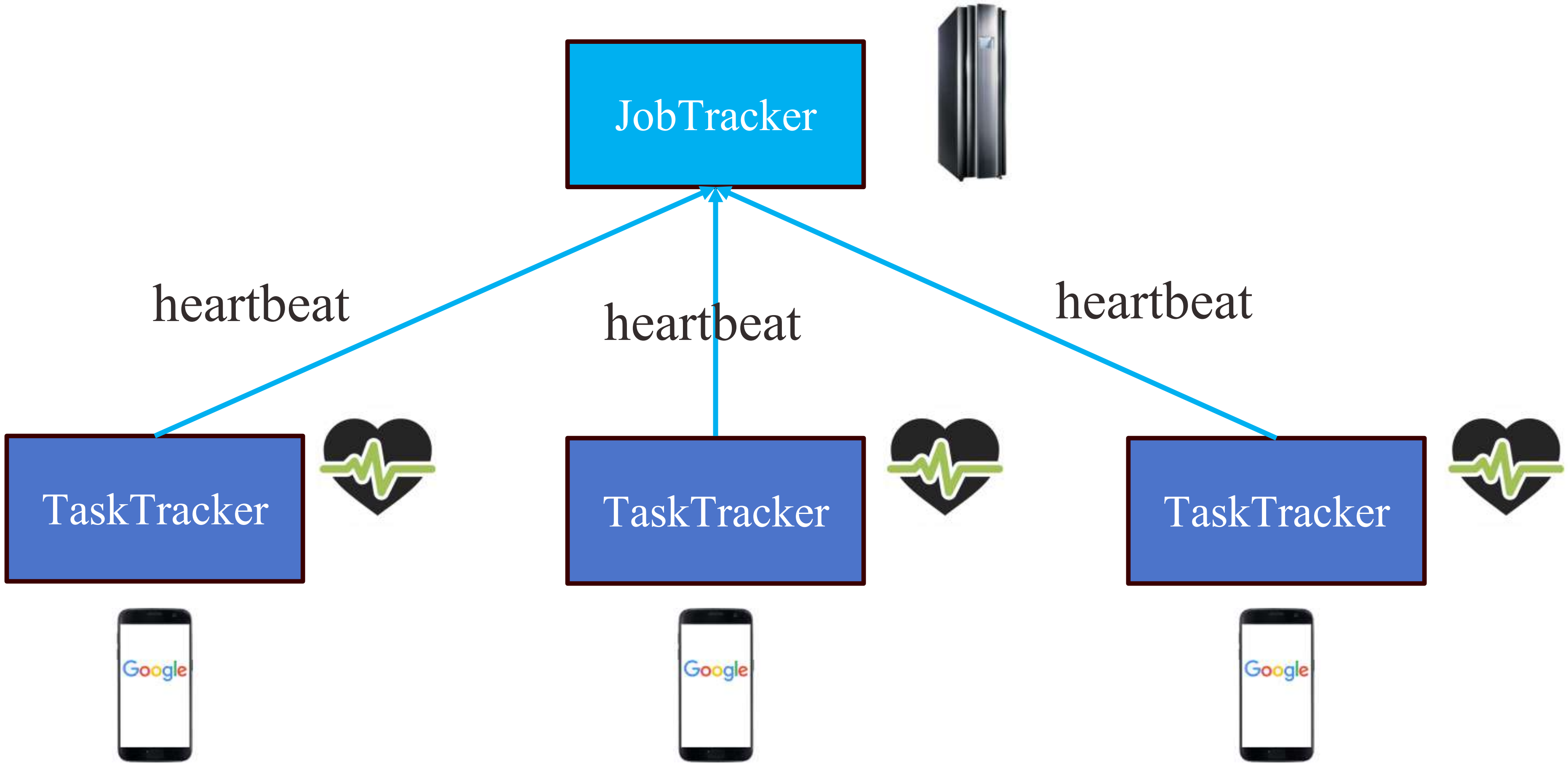
- DistressNet-NG Architecture
- Resilient Sockets
- Naming and service discovery
- Mobile Distributed File System (MDFS)
- Edge Computing
  - Real Time Stream Processing (MStorm)
  - Batch Processing (MMR)

Batch Processing (Mobile MapReduce - MMR)

- MDFS based
- Task Scheduler
  - Jobtracker
    - HPC, laptops
  - Tasktracker
    - Smartphones, tablets
- Standardized MapReduce APIs

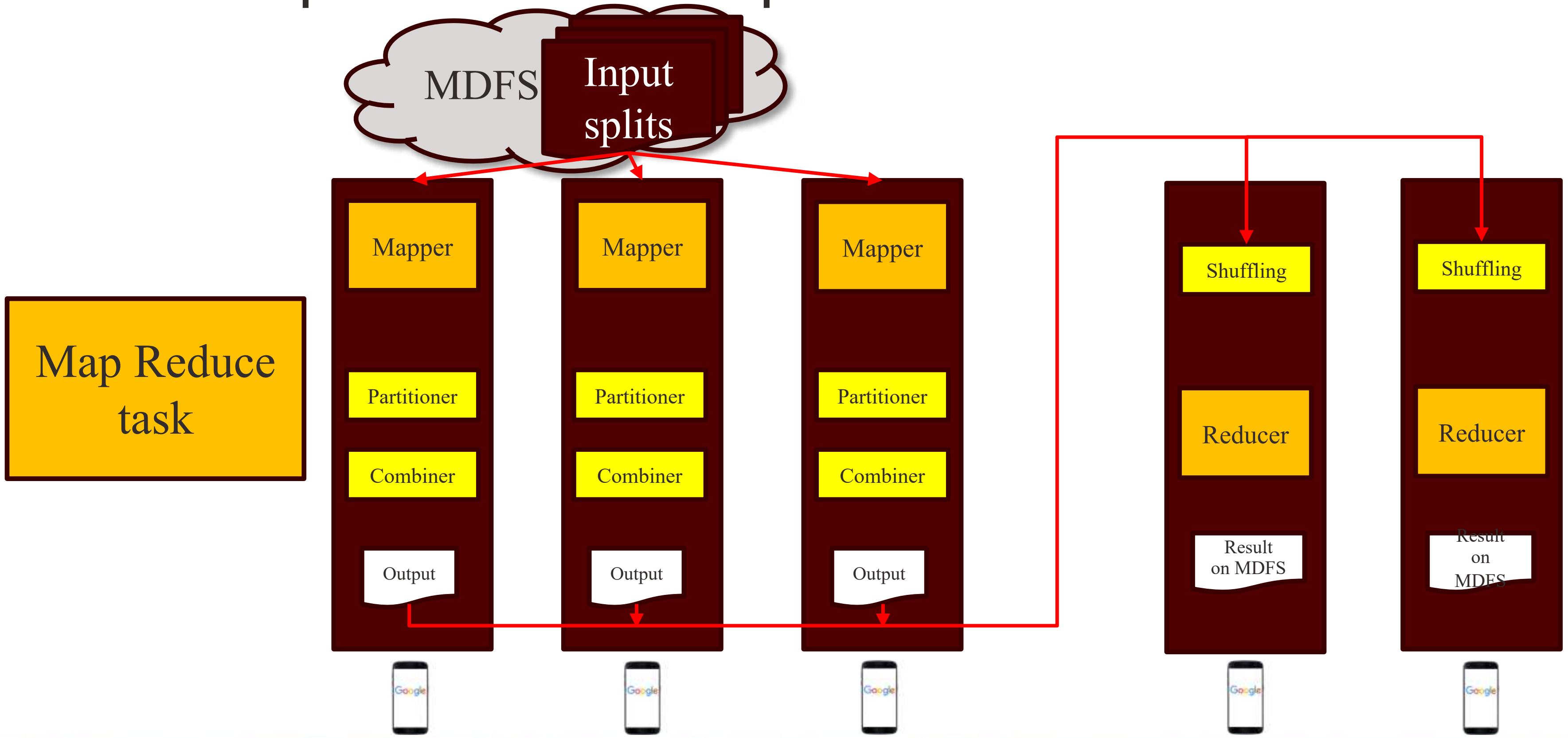


Task Scheduler



MapReduce Core

- A distributed computing framework
  - Based on MapReduce concept

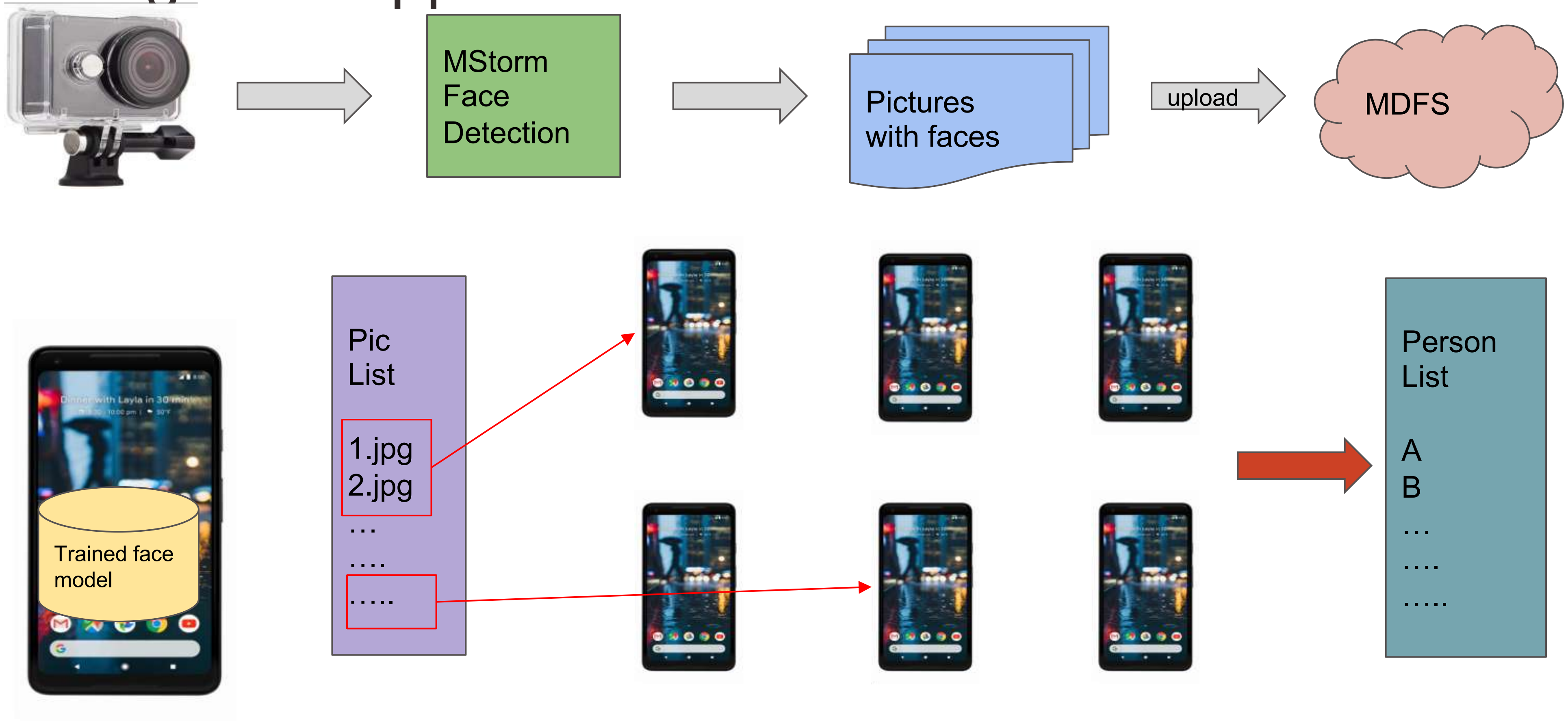


MMR Status

- Software:
  - Implemented on Android and Linux
    - A task can be executed on both Android and Linux devices
      - No task scheduling based on capabilities (cpu, battery...)
    - Centralized jobtracker
  - Use HDFS instead of MDFS
  - Integrated with GNS for automatic service discovery and disconnection issues
- Experiment during March 2019 Winter Institute at the Disaster City
  - Integration with GNS is working fine
  - Working over LTE and WiFi mesh

MMR Status (2)

Face recognition application:



MStorm Face Detection

Pictures with faces

upload

MDFS

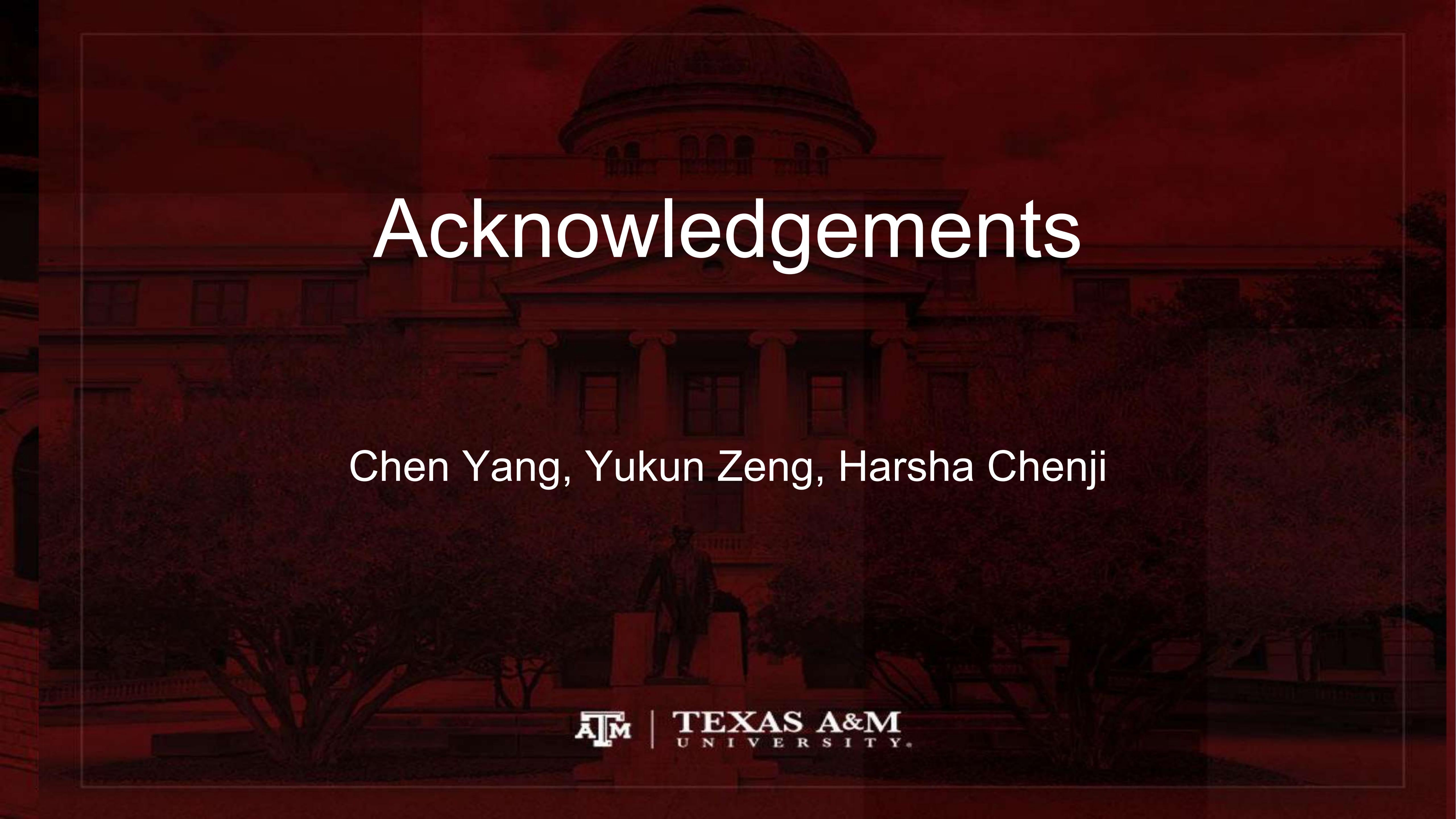
Trained face model

Pic List

1.jpg
2.jpg
…
….
…..

Person List

A
B
…
….
…..



Acknowledgements

Chen Yang, Yukun Zeng, Harsha Chenji

ATM | TEXAS A&M UNIVERSITY

DistressNet-NG: A Resilient Broadband Communication and Edge Computing Framework for FirstNet

Radu Stoleru,
stoleru@cse.tamu.edu

Wei Zhang, Ala Altaweel, Mengyuan Chao,
Suman Bhunia, Mohammad Sagor

TEXAS A&M UNIVERSITY

LTE-as-a-Service in DistressNet-NG

Harsha Chenji, Abdoulaye Saadou,
Joe Wamsley, Kevin Godenswager, Zach Shrock
Wireless Systems Research Group
Ohio University, Athens, OH

2019 PSCR Public Safety Broadband Stakeholder Meeting, Chicago, USA

Scenario

System-level resiliency and roaming of LTE functional elements

<u>Challenge:</u> reliability, Quality of Service, stringent KPI targets

In a highly dynamic environment

With heterogeneous systems

LTE Manpack

Cellular on Wheels

Nationwide Network

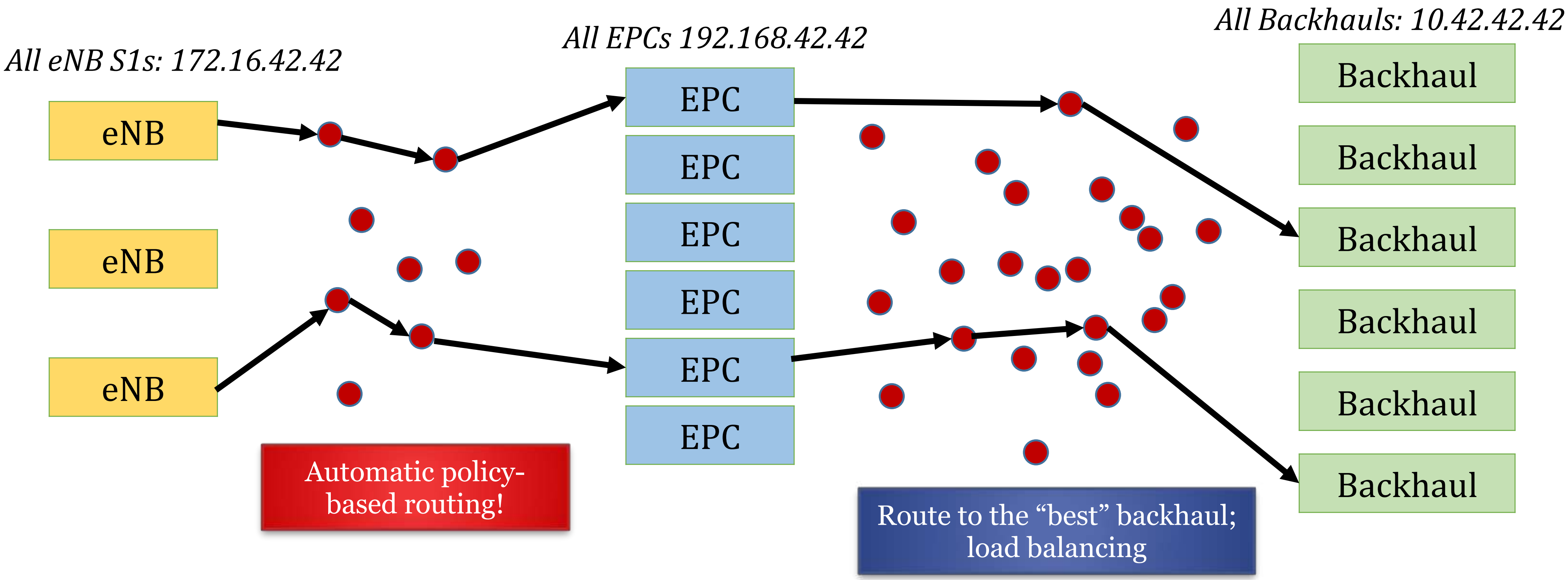
Major Goals

1. Public Safety Network Function Virtualization (PS-NFV)
   - ❑ Comms. resources from different agencies can be pooled together
   - ❑ Many LTE functional elements act as a single cell, and *autonomously*
   - ❑ Long range low power control plane, short range high rate data plane
   - ❑ Focus on application requirements (video streams, location updates)

2. Edge computing implementation over PS-NFV
   - ❑ Example: FirstNet app normally needs Amazon AWS services
   - ❑ DistressNet-NG will allow the app to discover local PS-NFV resources
   - ❑ Maintain a “shadow copy” of needed resources locally
   - ❑ Sync when backhaul available
   - ❑ Open source implementation of Amazon Greengrass/Azure IoT Edge

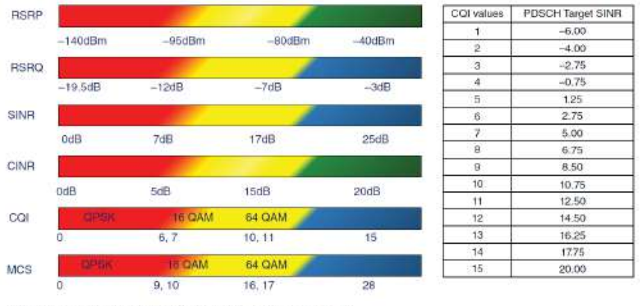
Wireless *Ad Hoc* Anycast Fabric for Decentralized LTE

1. Use anycast routing for transparency
   - ❑ Offload network administration to our software running on YOUR platforms
   - ❑ Consistent state is maintained over a control plane, PHY agnostic


All eNB S1s: 172.16.42.42
All EPCs 192.168.42.42
All Backhauls: 10.42.42.42
eNB
eNB
eNB
EPC
EPC
EPC
EPC
EPC
EPC
Backhaul
Backhaul
Backhaul
Backhaul
Backhaul
Backhaul
Automatic policy-based routing!
Route to the "best" backhaul; load balancing

Data Plane: TCP Optimization and LTE Interaction

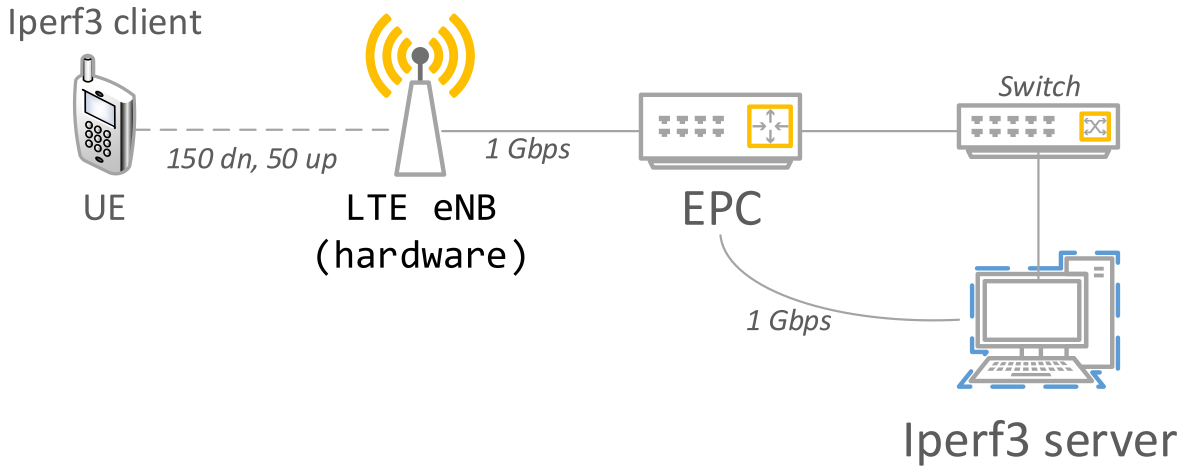
1. Enabled multiple TCP variants on mobile UE
   - ❑ Open source LineageOS
2. Measurement campaign during Winter Inst. 2019
   - ❑ Confirms lab results that downlink is never saturated with no other UEs actively sending/receiving data
3. Goal: open source kernel code for TCP optim.



CQI values	PDSCH Target SINR
1	–6.00
2	–4.00
3	–2.75
4	–0.75
5	1.25
6	2.75
7	5.00
8	6.75
9	8.50
10	10.75
11	12.50
12	14.50
13	16.25
14	17.75
15	20.00

Figure 2.28 LTE RSRP, RSRQ, CQI, MCS, CINR, and SINR ranges.

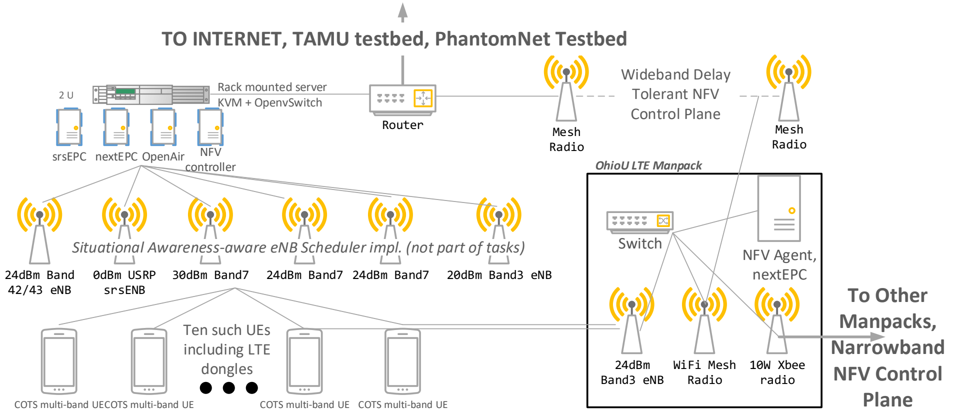
	downlink	uplink
reno	67.3	40.74
cubic	39.2	40.89
westwood	52	41.08
bic	10.945	41.005
htcp	101.5	40.98
illinois	27.1	38.96
highspeed	109.5	40.455
hybla	30.25	40.845
scalable	122.5	41.2
lp	8.965	40.03
veno	98.65	37.64
dctcp	122	41.165
cdg	11.85	40.865
bbr	17.75	40.995
nv	18.35	41.145



Results so far

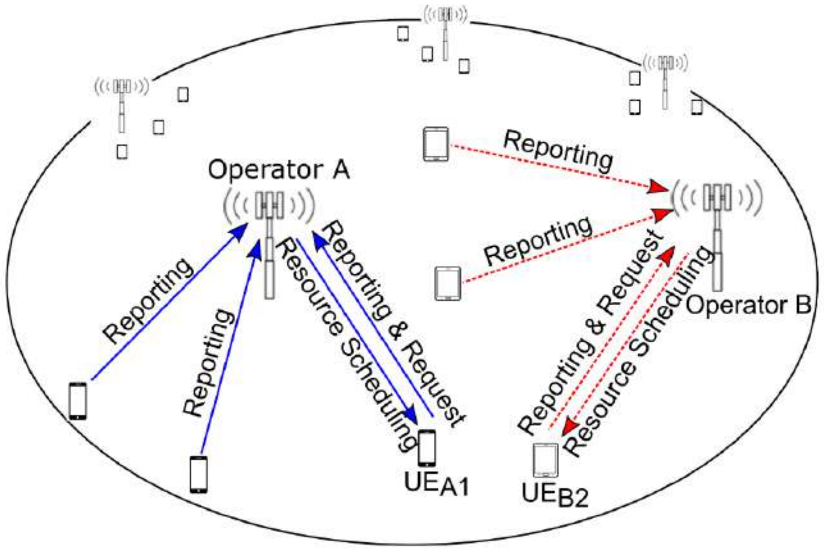
1. Built and demonstrated at Winter Institute in Mar 2019
   - ❑ Distributed EPC+HSS database
   - ❑ Service discovery using DNS (local)
   - ❑ Operation when disconnected from macro cell (simulated FirstNet)
   - ❑ Portable manpack with LTE bubble and services. < $1000, 8-hr. life, < 10 lbs.

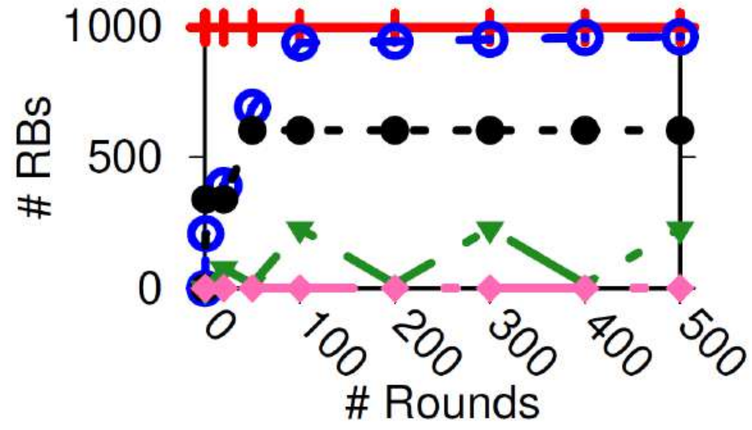
2. Feedback: too bulky but lightweight
   - ❑ We're reducing the size to a textbook

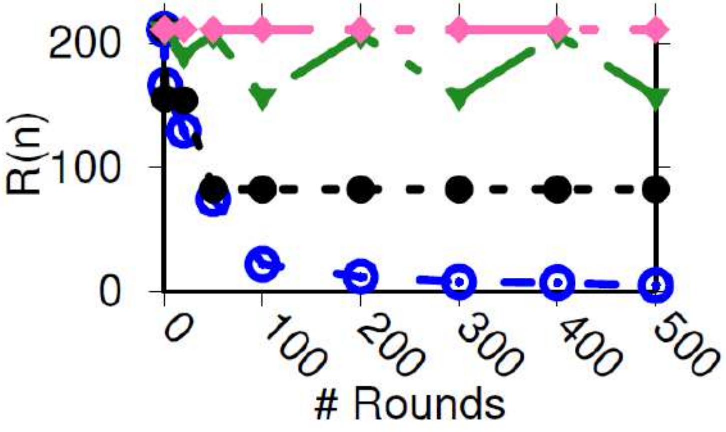


Results so far

1. Theoretical results
   - ❑ Several COO eNBs can coordinate in Band 14 without user intervention
   - ❑ Autonomous allocation of resource blocks
   - ❑ System objective is fairness w.r.t. demand generated by UEs
   - ❑ Practical and efficient for SDR implementation
   - ❑ Fairly fast convergence

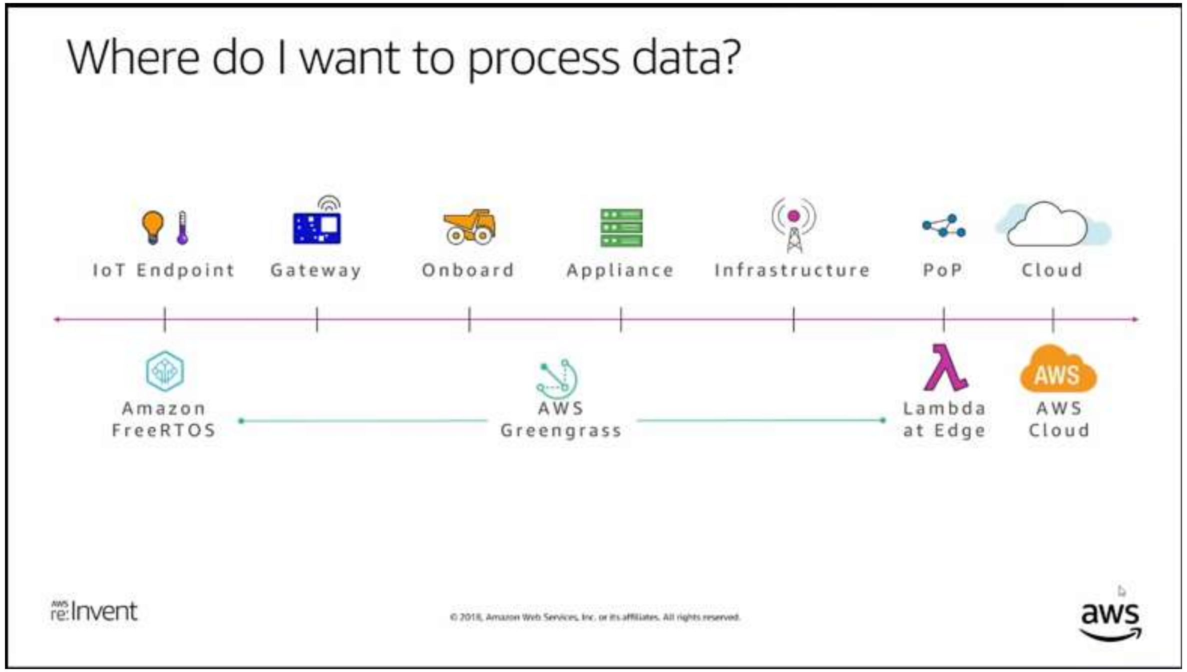






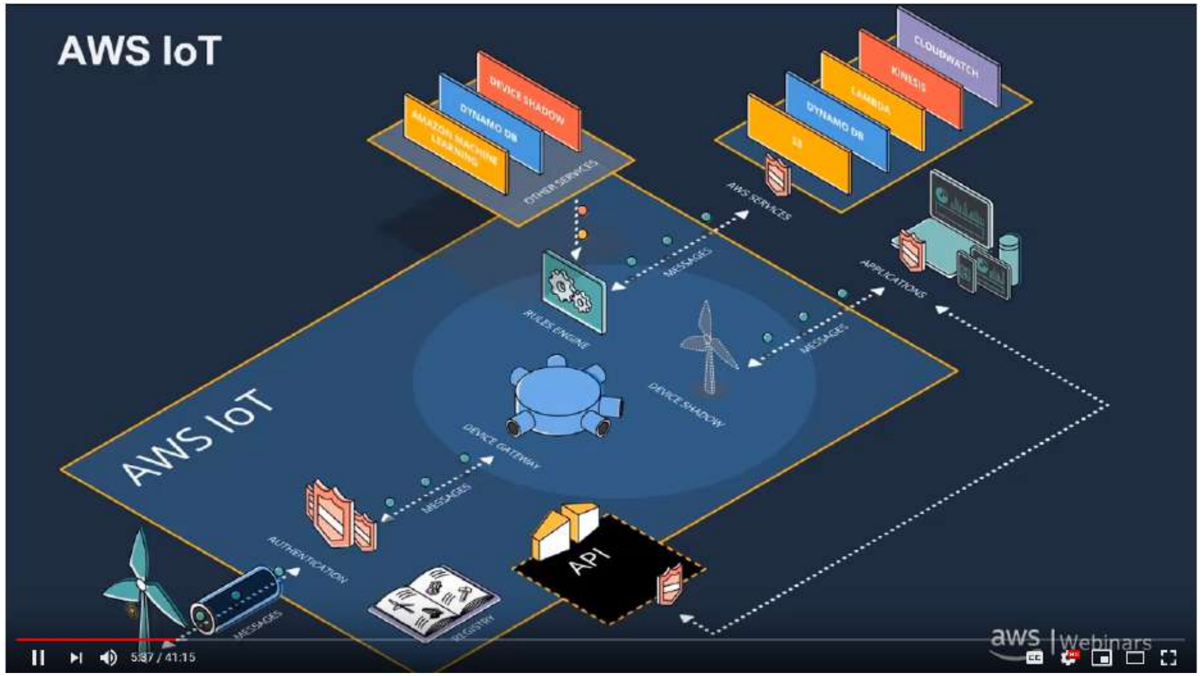
Ongoing Work

1. Edge computing for Public Safety NFV in the field
   - ❑ Looking for insight from app developers on how a cloud is used
   - ❑ Enable your app to discover our DistressNet-NG resources transparently



Where do I want to process data?

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Questions

chenji@ohio.edu

#PSCR2019

Get your hands on the tech!

Demos Open