The Road to 5G Wireless: The role of Device to Device Sharing

Huzur Saran

Department of Computer Science & Engg.

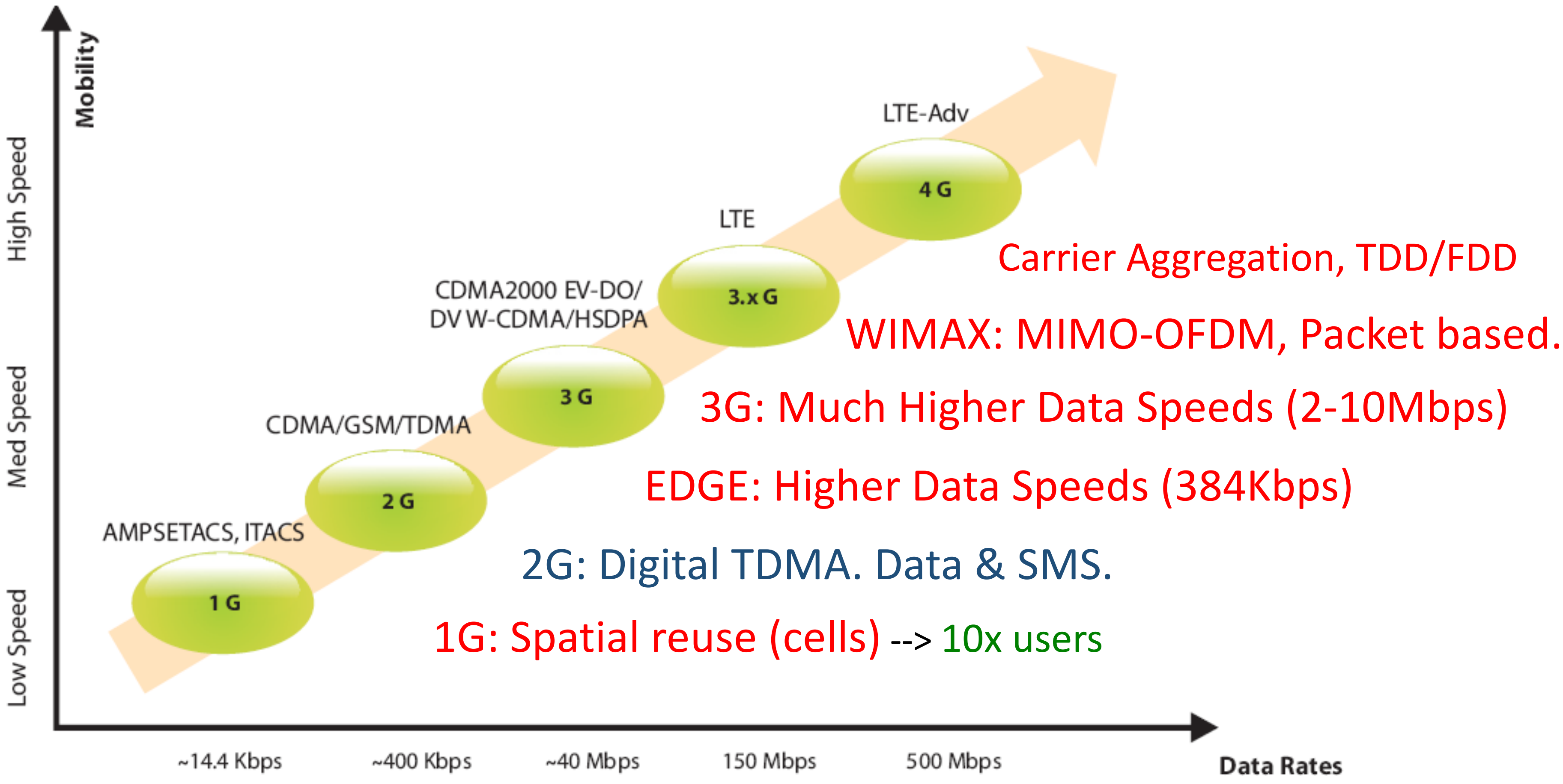
Indian Institute of Technology Delhi

New Delhi, India

overview

- We will go through the evolution of wireless networks
  - 1G , 2G, 3G etc. Currently 4th Generation systems are being deployed.
- 5G Research directions
  - Need for spectrum
  - Wifi Offload
  - Millimitre Wave
  - Co-Ordinated Multipoint: Edge-less Cell
  - D2D
- Our work on Device-2-Device Sharing
  - Network Sharing
  - Caching
  - Compute offload

Evolution up to 4G



4G LTE: Key features

- Sophisticated multiple access schemes
  - DL: OFDMA with Cyclic Prefix (CP)
  - UL: Single Carrier FDMA (SC-FDMA) with CP
- Adaptive modulation and coding
  - QPSK, 16QAM, and 64QAM
  - 1/3 coding rate, two 8-state constituent encoders, and a contention-free internal inter-leaver
- Advanced MIMO spatial multiplexing
  - (2 or 4) x (2 or 4) downlink and uplink
- Supports Multiple Bands both TDD & FDD version
- Data Rates of 300Mbps peak supported with Carrier Aggregation

Why 4G & Beyond?

- India went from 30 million landlines to 1000 million Cellphone users in 15 years
- 30x - 100X increase in connected devices
- Embedded sensors & computers everywhere (IOT/ M2M)
- DIGITAL India, Electronic Payments, E-Health
- E-Education, Video Distribution & Messaging
- Expected 100X Traffic Increase in Next 15 years
- Digital Divide – coverage for remote areas

Coverage & Cell size (data density) define user experience

	Iridium Satellite	Rural	Urban macro	Urban micro	Pico	Wi-Fi Hotspot
Coverage	World	Rural	Urban	Urban	Metro	Home/Office
Mobility	Perfect	V Good	V Good	Good	Fair	Nomadic
Cell radius	1500 km	30 km	3 km	300 m	30 m	10 m
Cell area km^2	7,700,000	2826	28	0.28	0.0028	0.0003
Efficiency bps/Hz	0.6	1.0	1.0	1.5	1.5	1
Data density $Mbps/km^2$/MHz	0.00000008	0.00035	0.035	3.5	350	3000
Total cells	66	500 k	1M	5M	50 M	1B
System capacity/MHz	40 Mbps	500 Gbps	1 Tbps	7.5 Tbps	75 Tbps	1000 Tbps

5th Generation Vision

- Planned for 2020 and Beyond IMT-2020
- User data rates starting from 1Gbps +
- anytime anywhere connectivity
  - Enhanced Coverage & Spectrum efficiency
  - High mobility – expected to work on highways and planes etc
  - Massive increase in aggregate traffic
- Internet of things – Low power devices & Sensors
- Energy Efficiency: 90% reduction in network energy usage
- Increase in Device to Device (D2D) communication

5G: Technology Directions

- Network Densification through small cells
- Software Defined Networking and Network Virtualization
- Leverage Device-to-Device (D2D) communication
- Co-ordinated Multi-Point/Multi-User MIMO
- Dynamic spectrum allocation/sharing techniques
- Use additional spectrum Beyond 6 Ghz

5G: Evolutionary Enhancements

- Voice over LTE
- Software Defined Networking and Network Virtualization
- Device-to-Device (D2D) enhancements
- Dynamic spectrum allocation/sharing techniques
- Machine-to-Machine (M2M) enhancements
- Higher order modulations (HOMs) in downlink and uplink, better Codecs
- Cloud RAN (C-RAN) and Remote Radio Head (RRH)

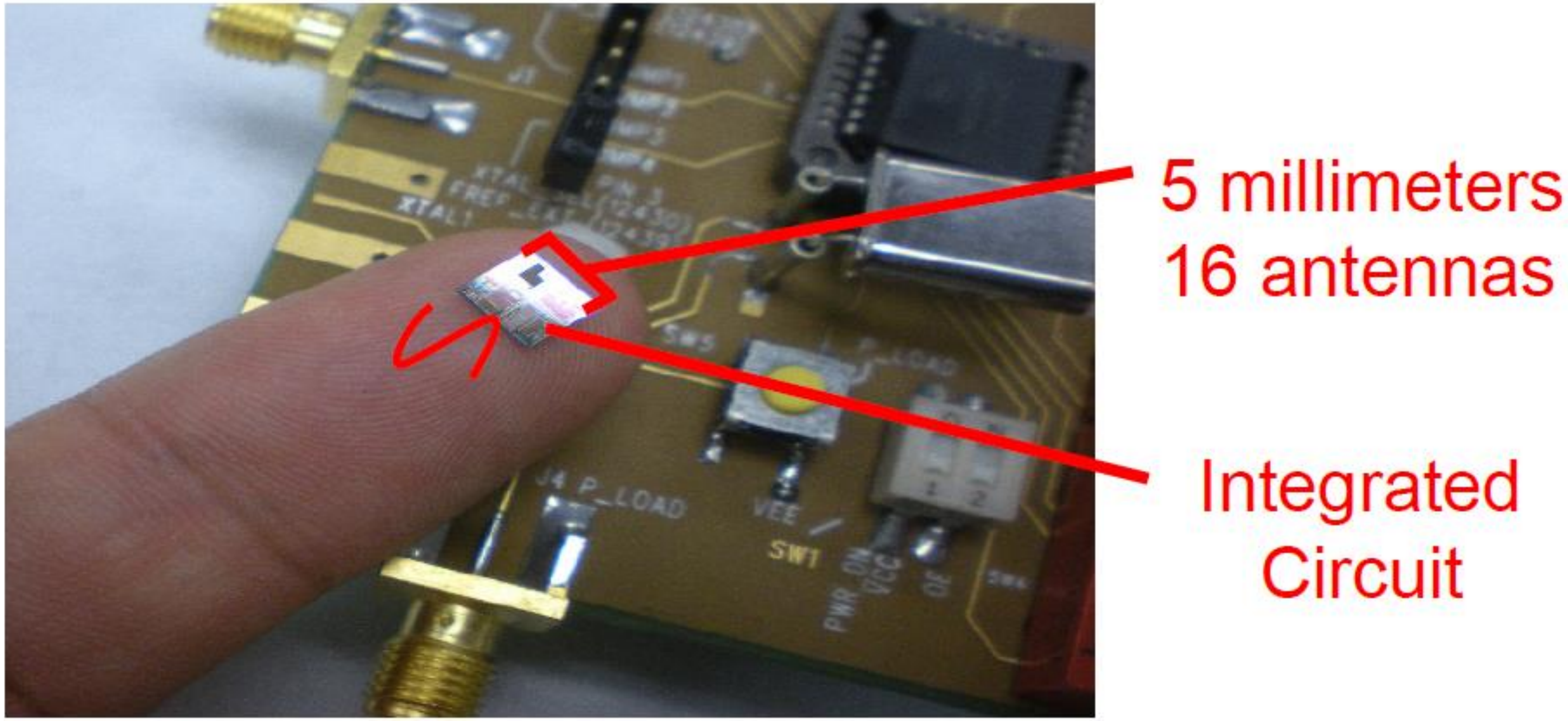
5G: Advanced Wireless tech (Millimiter Wave, Massive MIMO)

- Beyond 6Ghz e.g. 28, 60, 73 Ghz
- WiGiG 802.11ad at 60Ghz standardized
- Highly directional; Large Antenna Arrays possible.
- Distance 1-10m Capacity 4-16 Gbps
- With 3D Massive MIMO and Beamforming
  - 200m range shown in experiments
- Others: In-Band Full Duplex (IBFD)

POLYTECHNIC SCHOOL OF ENGINEERING

mmWave Wavelength Visualization – 60GHz

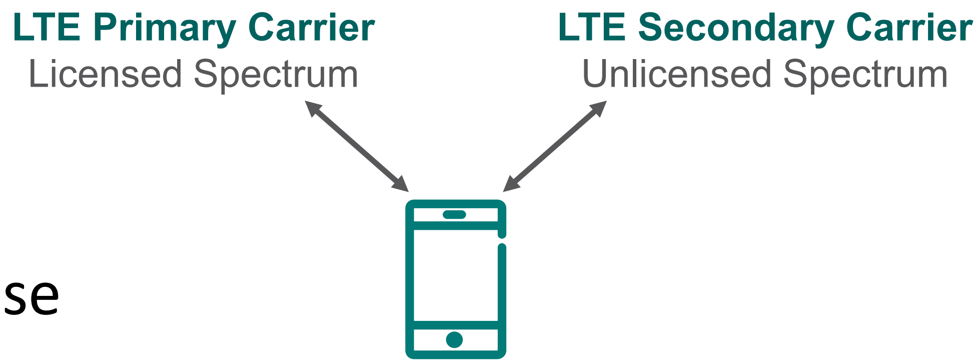
NYU WIRELESS



e: [Millimeter Wave Wireless Communications - The Renaissance of Computing ommunications] – Ted Rappaport, ICC 2014

5G: Network Densification

- Carrier Aggregation
- Small Cells
  - Smaller cells, Higher Reuse
- Leverage Unlicensed Spectrum
  - Wifi-offload
  - Licensed-Assisted Access LTE (LAA)



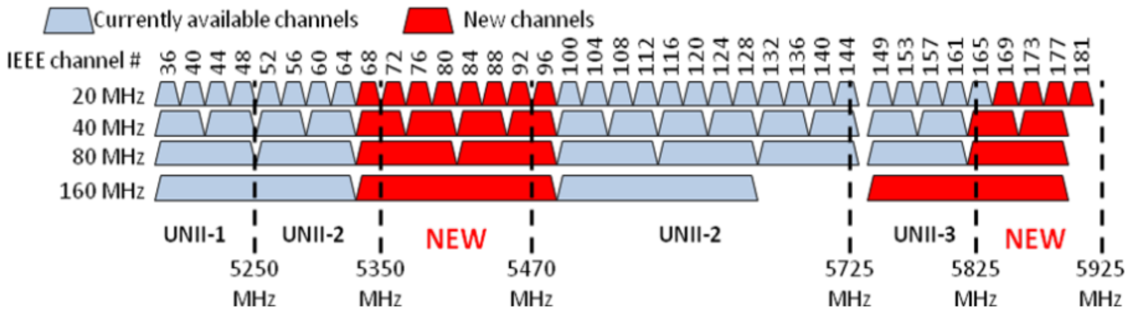
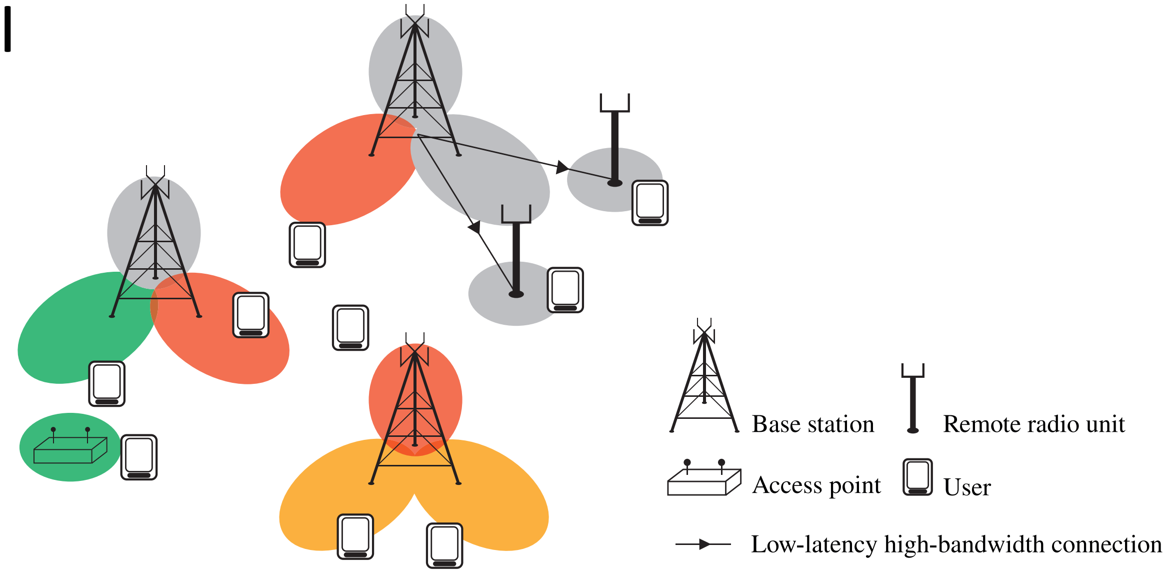


Image source: RevolutionWiFi

Wifi 5Ghz unlicensed spectrum

Coordinated Multi-Point (CoMP)

- *Coordinated transmission/reception* of data among several transmission/reception points to ***reduce or even exploit interference***
- Low Latency & High Capacity Backhaul
- Estimation of CSI

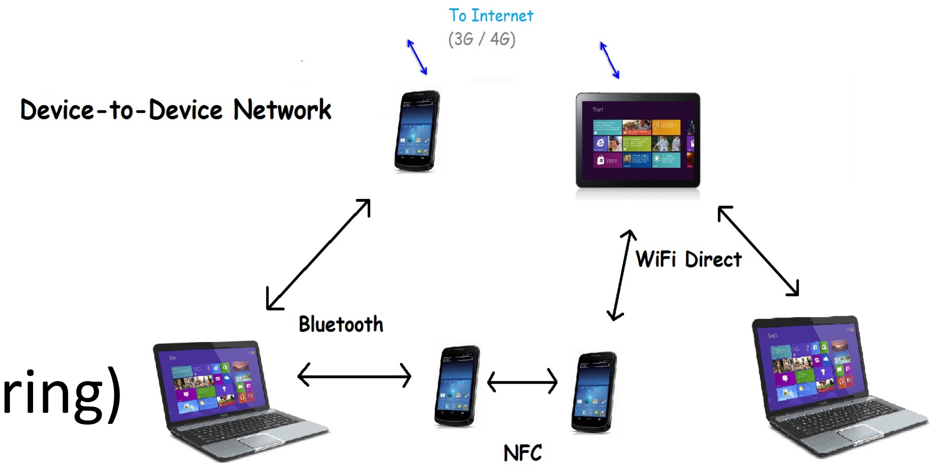


HetNet. . . heterogeneous network, RRH. . . remote radio head

D2D Network

- Extend the Networking Hierarchy to Leverage large number of connected devices & sensors in a small cell
- Form a network among devices – any two devices can communicate over it
- Resource sharing
  - Compute sharing
  - Network sharing

  (bandwidth pooling, file sharing)



Network Creation Using Wifi-direct Technology

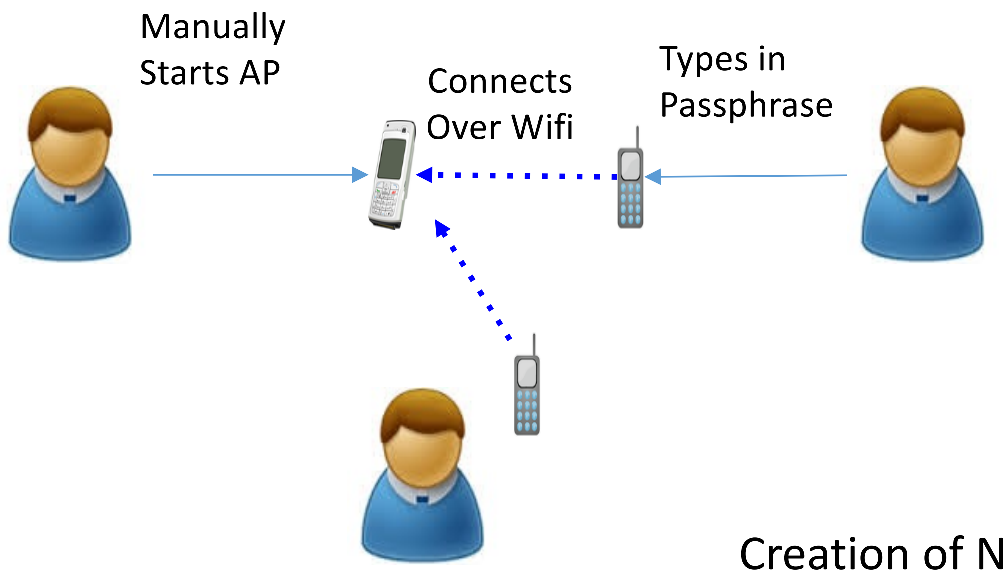
- Wifi-direct allows to turn a node into an access point

- Other devices can connect over wifi using **SSID** and **PASSPHRASE**

Creation of Network

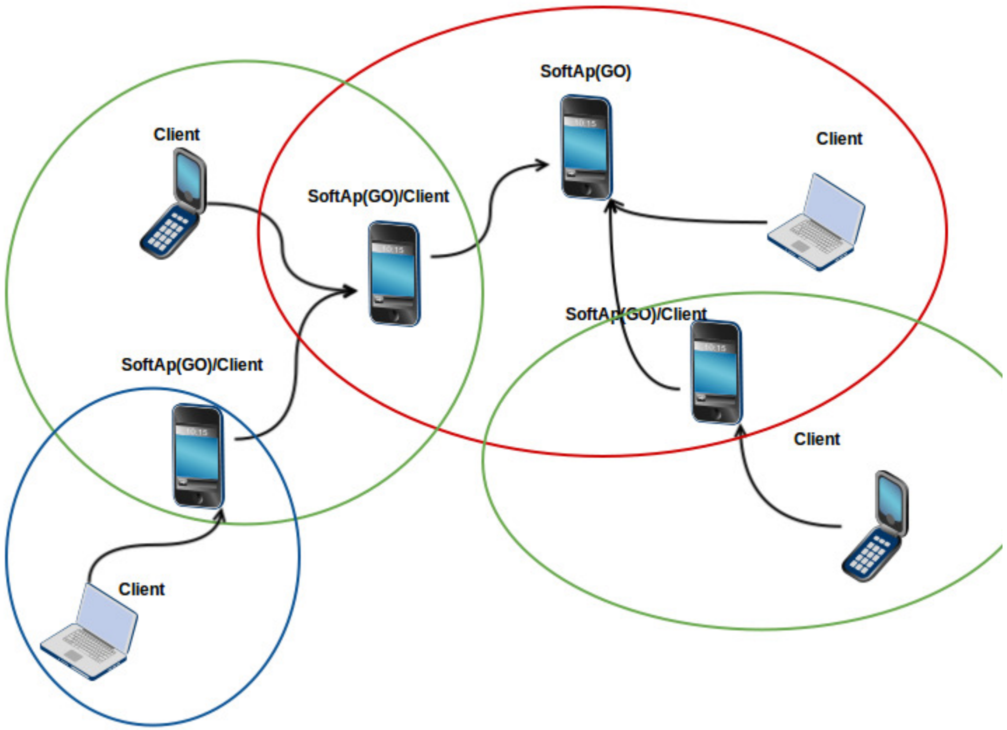
D2D Network Creation

- Wi-Fi ubiquitous, fast, stable technology
- Two-hop star topology
  - (option 1) Through existing Wi-Fi AP
  - (option 2) One device volunteers to become anchor node (SoftAP), join using SSID and passphrase



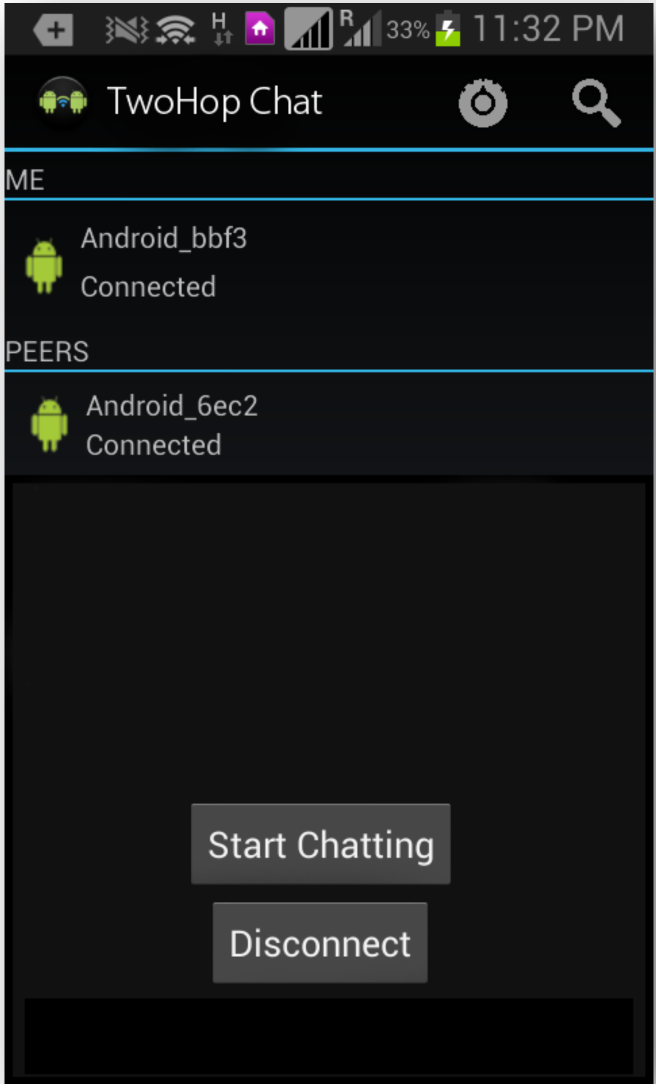
Creation of Network

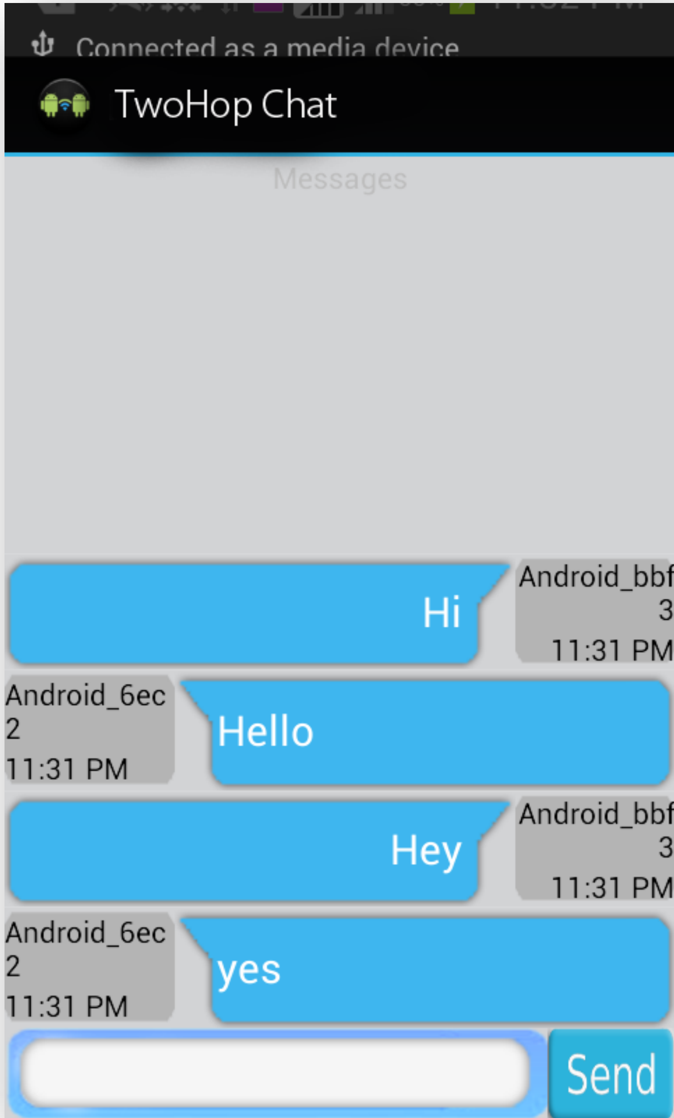
How to Build Multihop Network?



- node becomes both a client and SoftAP
- However IP addresses in different groups can be same: Need Higher layer
- Also performance degradation with multiple hops
- OUR FOCUS: Max 2Hops

Screenshots





Internet Sharing

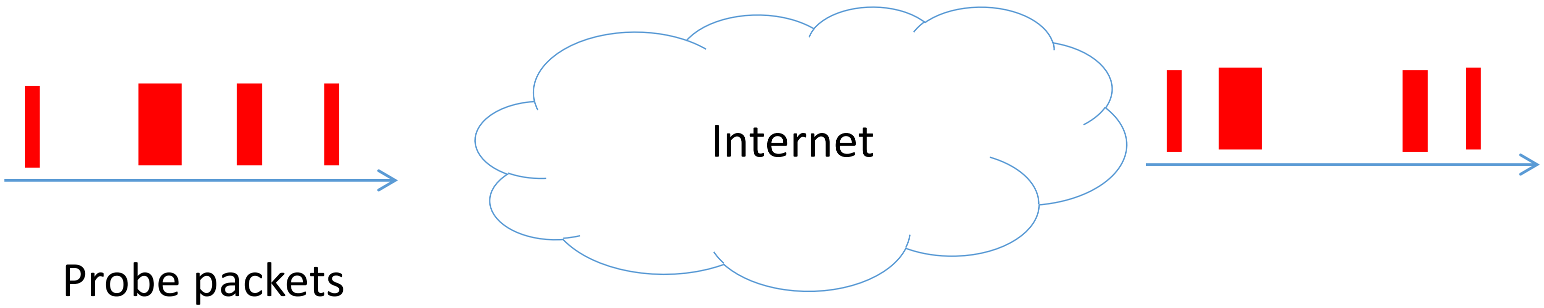
- Shortcomings of Wi-Fi hotspots
  - No limit on bandwidth usage of users
  - Single node sharing bandwidth

- Developed applications overcoming these shortcomings

- Some of the ideas included in Intel’s PEG/WPRD/WINS product

Active Probing for QoS

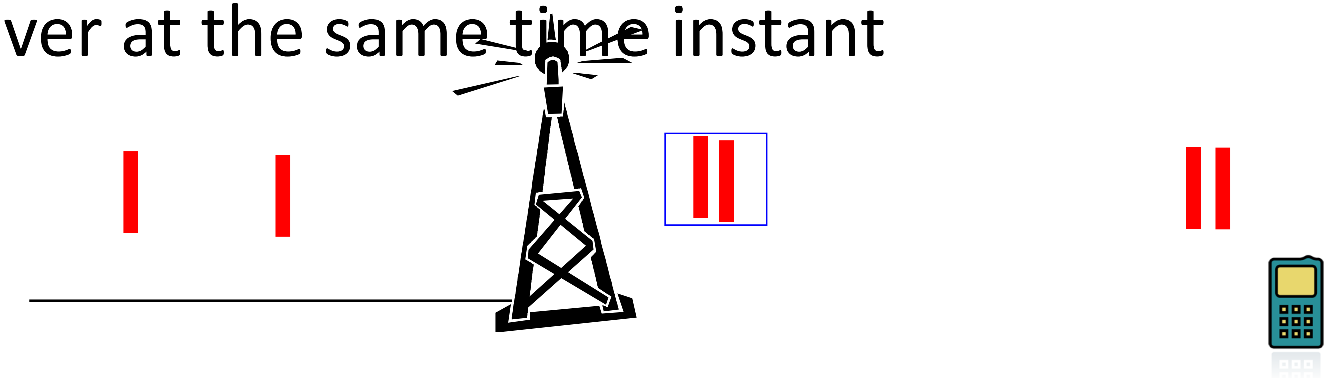
- Achieved QoS can be very different from advertised QoS
- Estimate QoS without using up precious bandwidth resources
- Active but non-invasive probing (using few packets) tools exist
  - E.g. : for available bandwidth



- Variation in probe packet gaps, or packet loss reveal information about e2e path
- Our finding: Existing probing tools (Spruce, Wbest, pathchirp, pathload etc.) perform poorly for 3G/4G networks
- Developed own app

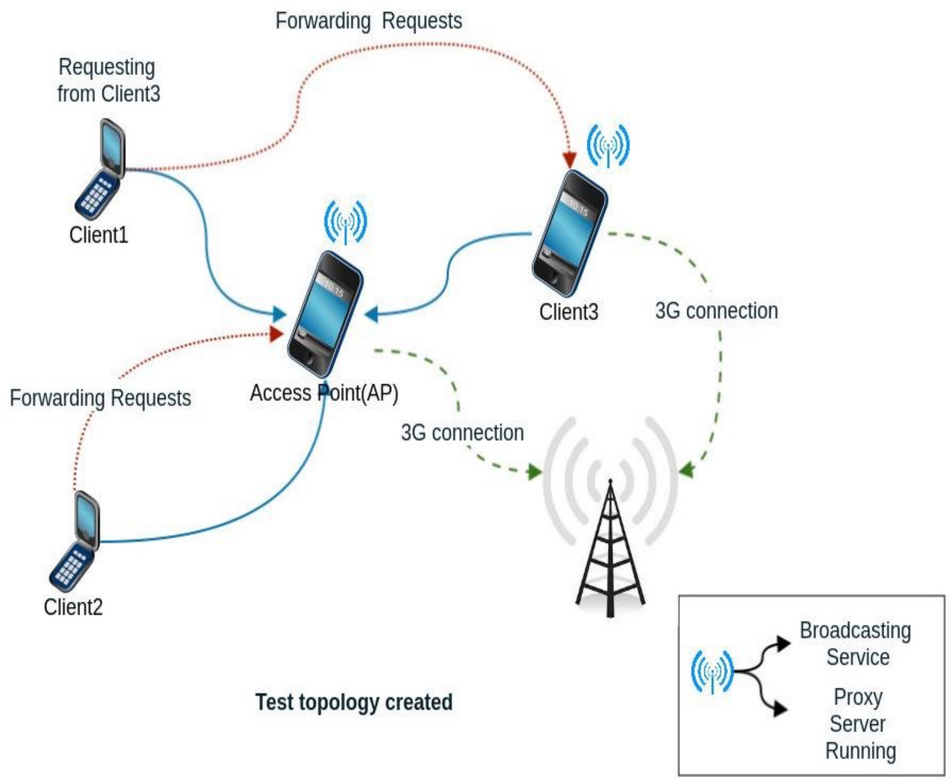
Failure of Existing Available Bandwidth Tools on 3G/LTE networks

- 3G and LTE transmit data in “transport blocks”
- Several probe packets typically get bunched together into the same transport block at the base station
- All packets in a transport block appear to reach the receiver at the same time instant



Internet Sharing: QOS based GW Selection

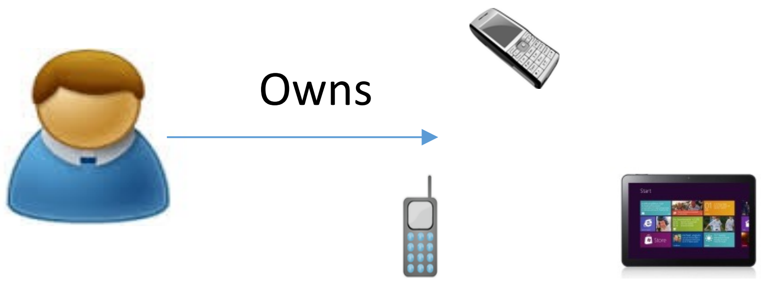
- **Improve QOS** via Dynamic Internet Gateway Selection
  - Parameters: ping RTT, Signal strength
- **Reduce cost** : provider selection based on policy parameters
- Gateway **Advertise & monitor** QOS parameters



Test topology created

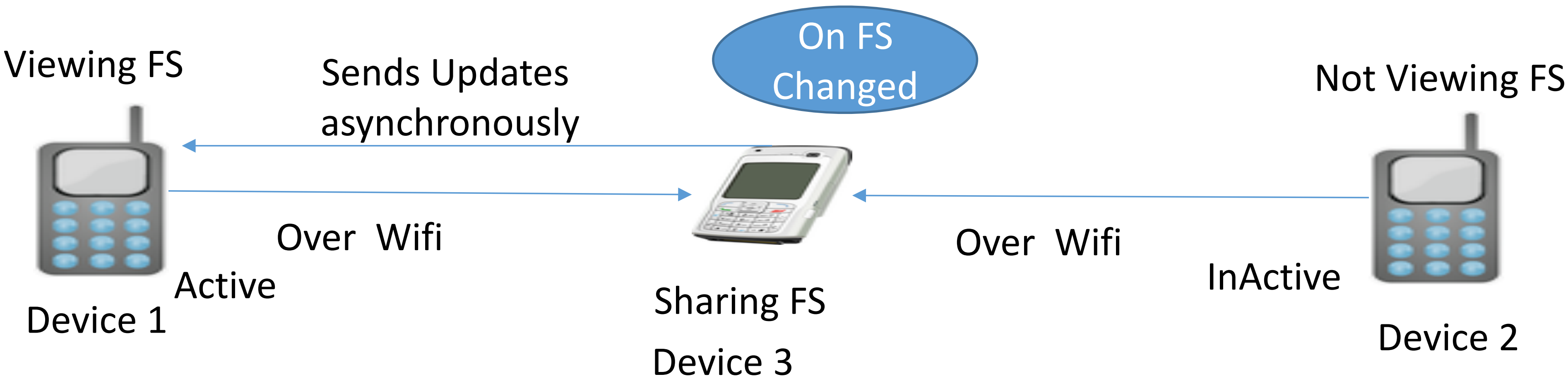
File system sharing

- Seamlessly share files in D2D network
- Using cloud can be expensive (cost, time) and cumbersome (all users must have access)
- Use cases
  - Single owner of multiple devices, wants to view large video on one device (e.g. tablet) which is stored on another device (e.g. mobile phone)
  - Classroom/meeting where users share material



File sharing Android app

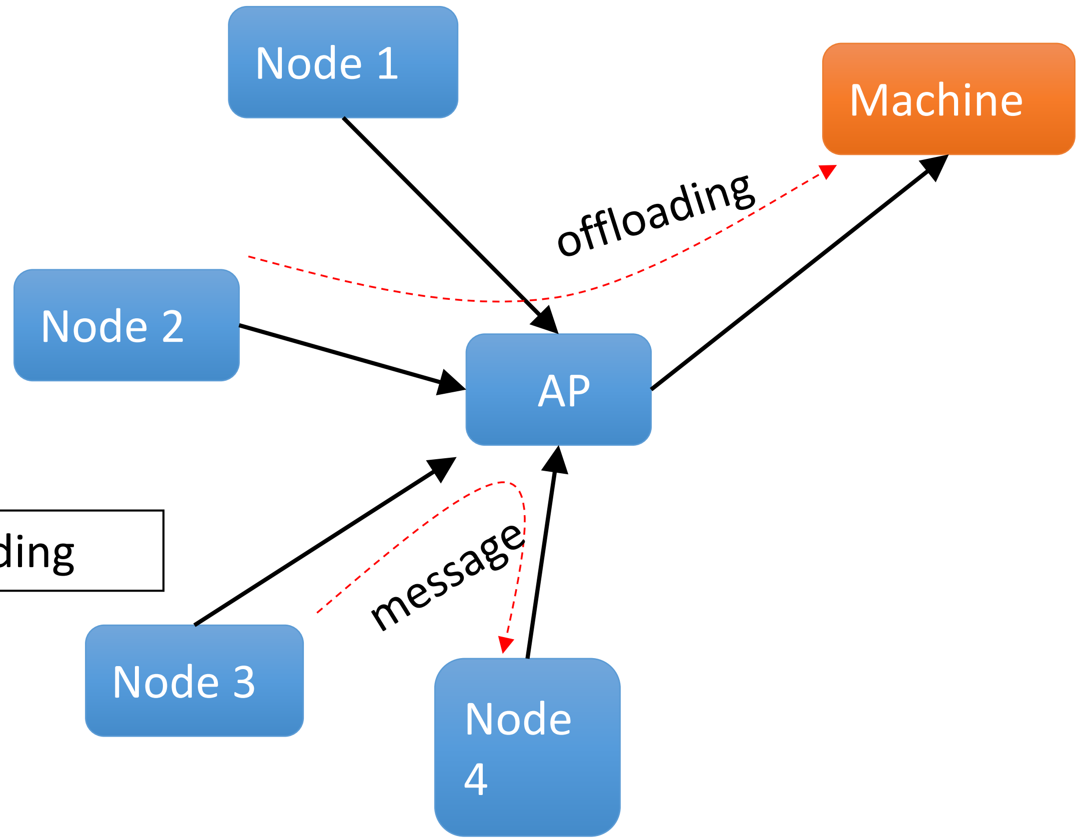
- Tree based structure for FS content
- Efficient sharing of meta data (push/pull)
  - changes on one device pushed only to active users
  - On becoming active users pull metadata



- Users can download/stream files of other devices

Low Latency Transport Protocols

- Important for applications such as compute offload
- Enhances network performance



Time to compute locally > Time for offloading

*Star Shape Topology

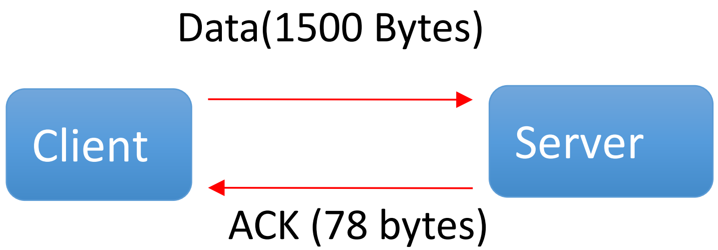
Time for offloading = Time to send + Time to compute

TCP vs. UDP

- 1 hop, lossless, no interference in 802.11g:

UDP performs by ~46% better compared to TCP

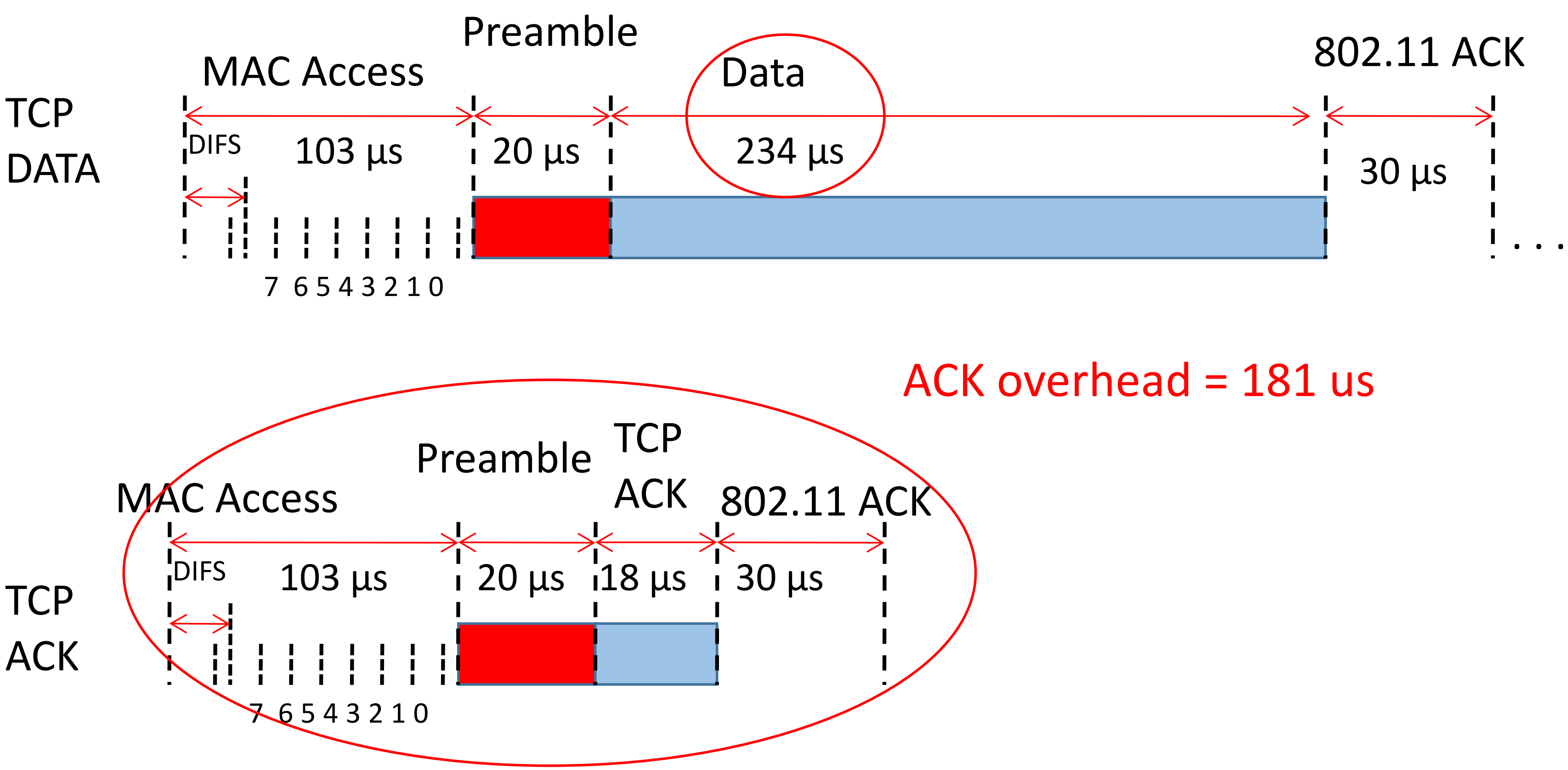
WiFi uses a half-duplex channels



- TCP throughput is limited by overheads
  - MAC layer overheads(inter guard interval time, headers,etc)
  - TCP overheads (acknowledgements, timeouts,etc.)

ACKs is only 5% of Data packet !!

Wi-Fi Overheads

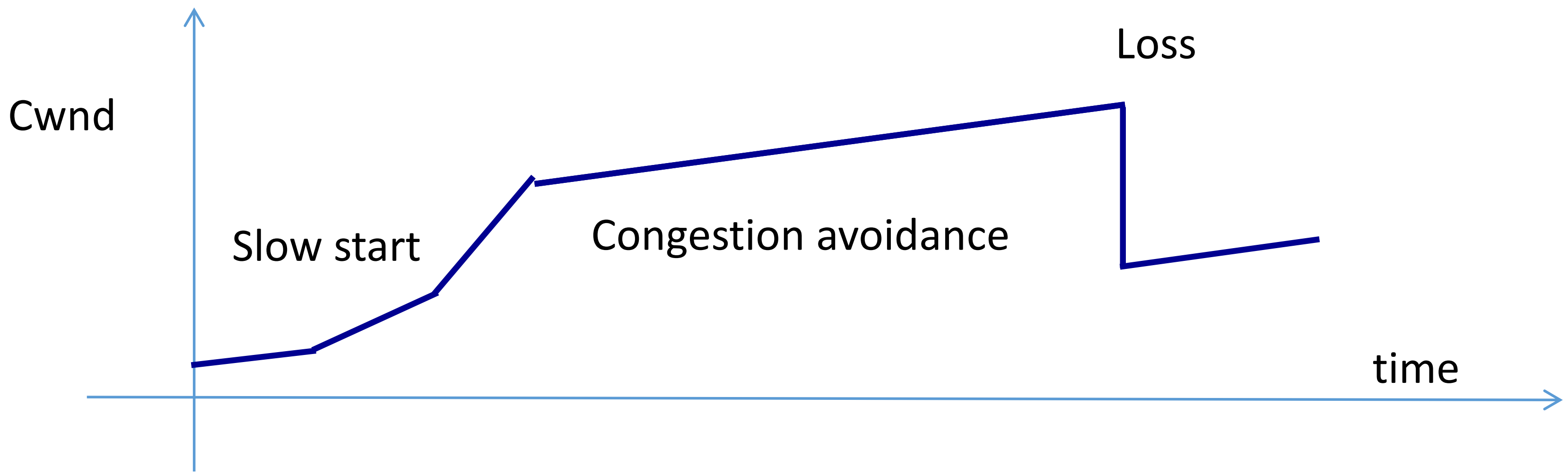


TCP Window Increase

- Cwnd: TCP congestion window, max. number of unACKed packets

- Data rate = cwnd/RTT; RTT: round trip-time

- Cwnd Increase (per ACK)
  - Cwnd = Cwnd + 1 (slow start mode, exponential increase)
  - Cwnd = Cwnd + 1/Cwnd (congestion avoidance, linear increase)



Delayed Acks

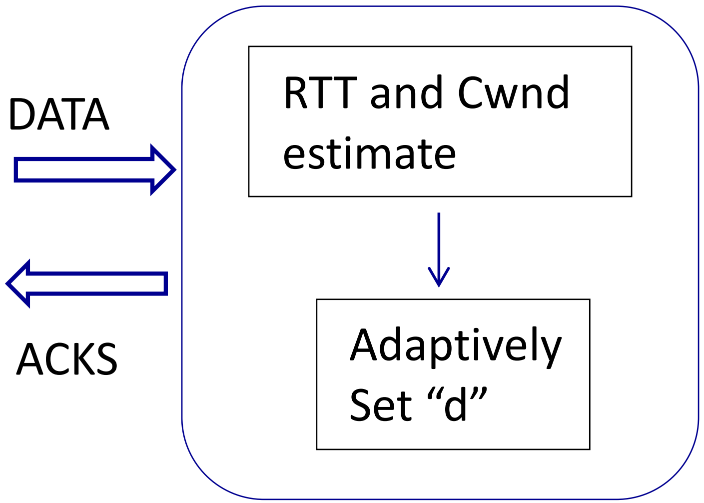
- Reduce number of TCP Acks
  - send one ACK for every “d” packets received

- Larger “d” leads to less overhead

- Constraint: $d < cwnd$ (else timeouts can occur)

- Our solution: estimate cwnd at receiver and set “d” accordingly

TCP Proximity



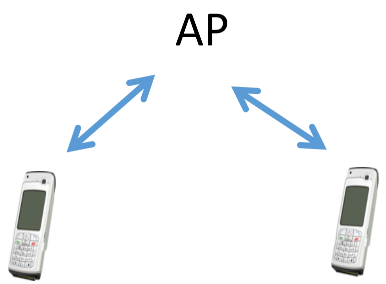
Receiver-side modification

- At receiver: estimate cwnd using seq. numbers of DATA packets, ACKs being returned, RTT estimates
- Ensures that d < (estimate of cwnd)

Performance improvements (ns-3)

Del Ack parameter "d"	Throughput in Mbps
0	10.49
2	12.29
4	12.77
dynamic	14.26

Lossless environment



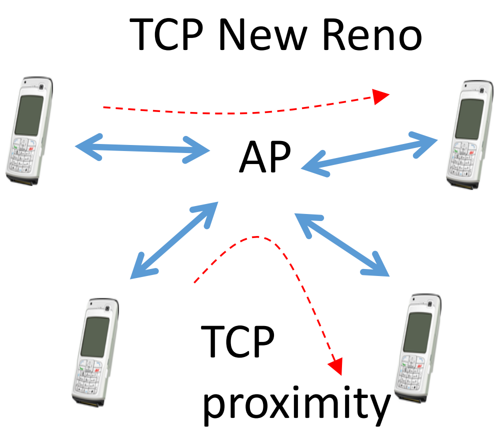
Del Ack parameter "d"	Throughput in Mbps
0	10.34
2	9.21
4	9.92
dynamic	13.58

Lossy environment (4-5%)

- Lossless case: TCP proximity gives 35% throughput improvement
- Robust in the lossy case

Shortcomings

- TCP proximity loses out when competing with conventional TCP (without delayed ACKs)



TCP type	Throughput in Mbps
New Reno	7.7
Proximity	6.1

- Cwnd grows with each ACK sent back: fewer ACKs leads to slower Cwnd growth

- Will require sender-side modifications

UDP based transport

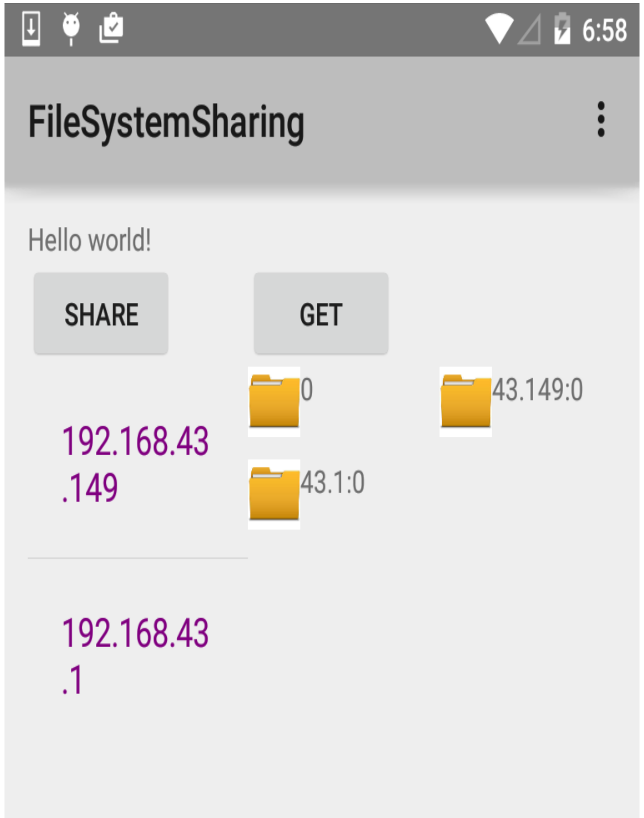
- Requiring both sender and receiver side modifications to TCP will hinder wide adoption

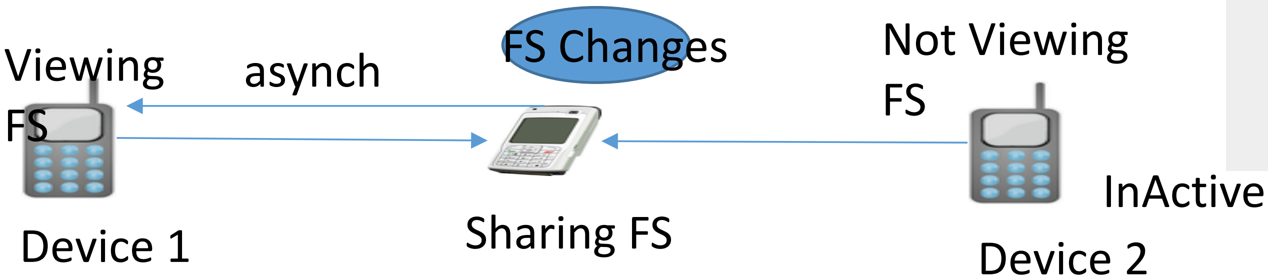
- Current work: UDP-based transport protocol for compute sharing app

- Rich and concise information feedback to reduce ACK overheads

D2D File System Sharing

- Nodes share their File system service using broadcast messages
- Clients listen to broadcasts to browse their file-system





D2D File System Sharing

- File-System Representation
  - Created Tree based structure for FS content
  - Structure is completely serializable
  - The serialization done using Google gson library.
- File Metadata Exchange & Security
- Background File Transfer
- Simple User Interface

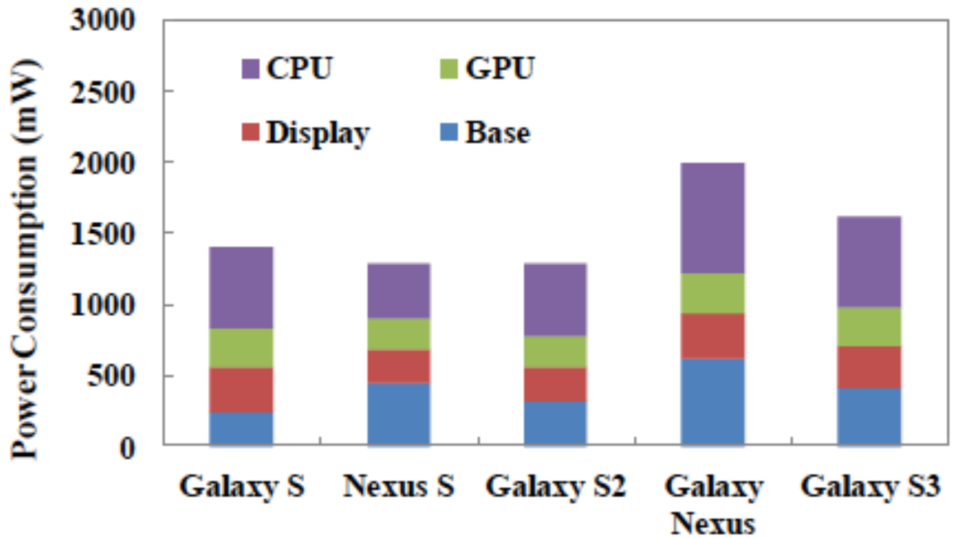
D2d File Sharing: Security: Why is it needed?

- We need secure transfer of data/meta-data in between the nodes.
- Various attacks are possible such as IP Spoofing, MITM.
- We need to define different access levels for different type of sharing in the network.
- The key used to join the network is entirely different. We are assuming that the network can be big enough and we cannot trust all the nodes in the network. Eg: Classroom scenario.

MOTIVATION: SHARING CPU CYCLES

- Sharing CPU cycles
  - Increase battery life 
  - Compute in time: meet delay requirements. 
  - Example applications: Image processing, video format conversion

- *"Smartphone games consume a large amount of power. This power is mainly consumed by background computation …."* – *Chen et al. ,* "How is Energy Consumed in Smartphone Display Applications?", *ACM HotMobile'13 [Chen13]*



Power breakdown in a video game
Source: [Chen13]

Challenges

- Identifying methods correctly based on profiling information analysis to be suitable candidates for offloading.

- Detect dependencies between methods by using call graph analysis of the application.
- Refactor the calls to candidate methods to be able to decide dynamically about whether to offload it to a server or execute on the device.

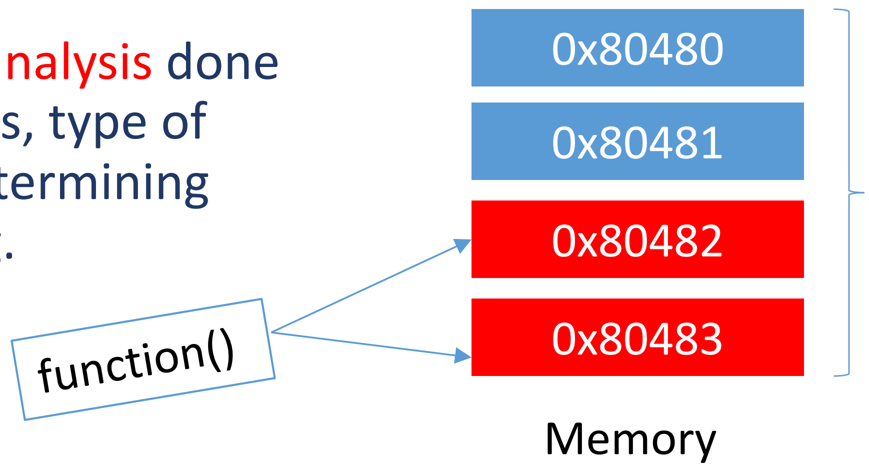
- Modifying method calls using java source code instrumentation and additions for handling server/client code.

Offloading Native Applications

- Native C/C++ code can be run on Android devices using the Android Native Development Kit (NDK).

- Dynamic source code instrumentation done for runtime profile details like number of calls, memory references, total number of instructions on a per function basis using PINdroid.

- Memory filtering and Call graph analysis done to detect functional dependencies, type of memory accesses. Thereafter, determining suitable candidates for offloading.



Experimental Setup

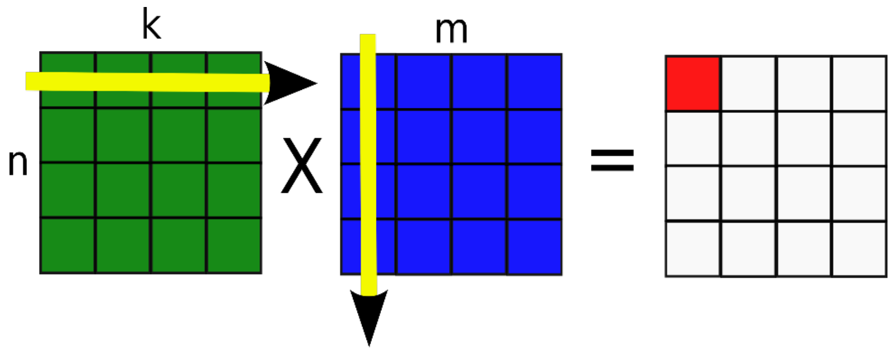
- Server-Side Specification
  - x86-64 architecture (Intel Core-i7-3770S) ,3.1 GHz frequency, CPU Cache 8192K, 8GB RAM running Linux.

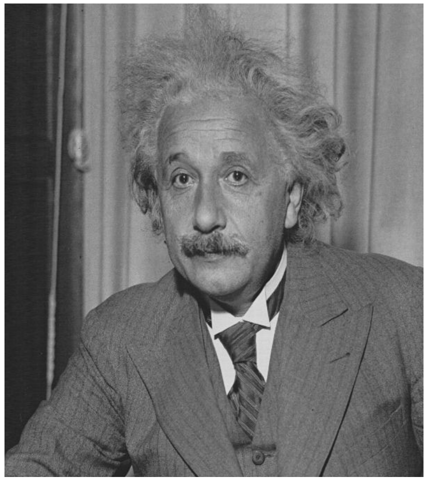
- Client-Side Specification
  - Android Device (Google nexus 4) with ARM v7 Instruction set (Qualcomm Snapdragon S4 pro SOC) ,1.7GHz frequency, 2GB RAM, CPU Cache(L0: 4 KB + 4 KB, L1: 16 KB + 16 KB , L2: 2 MB) running on Android 5.1.

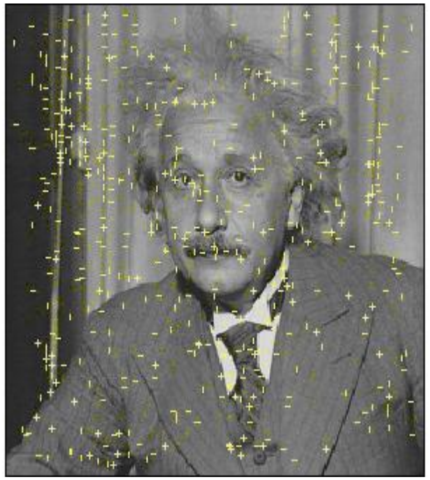
- Network Specification
  - Institute WI-FI Network (802.11b/g).

Benchmarks

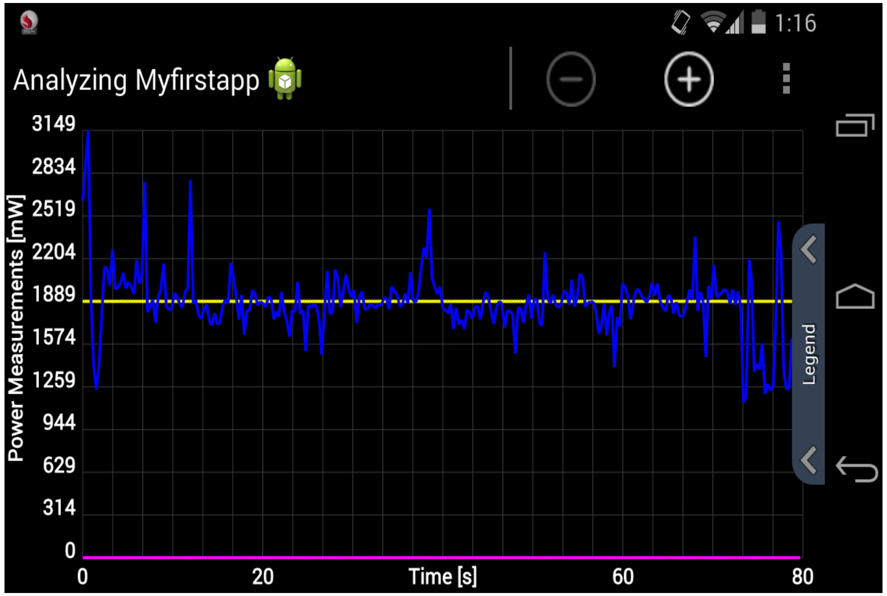
Dense LU factorization	Extensively used in scientific applications involving solving a system of linear equations.
Matrix Multiplication	Image processing applications like scaling, translations and rotations.
Fast Fourier Transformation	Spectral analysis, Data compression, Partial differential equations, Polynomial multiplication etc.
SIFT	Image matching, Face recognition



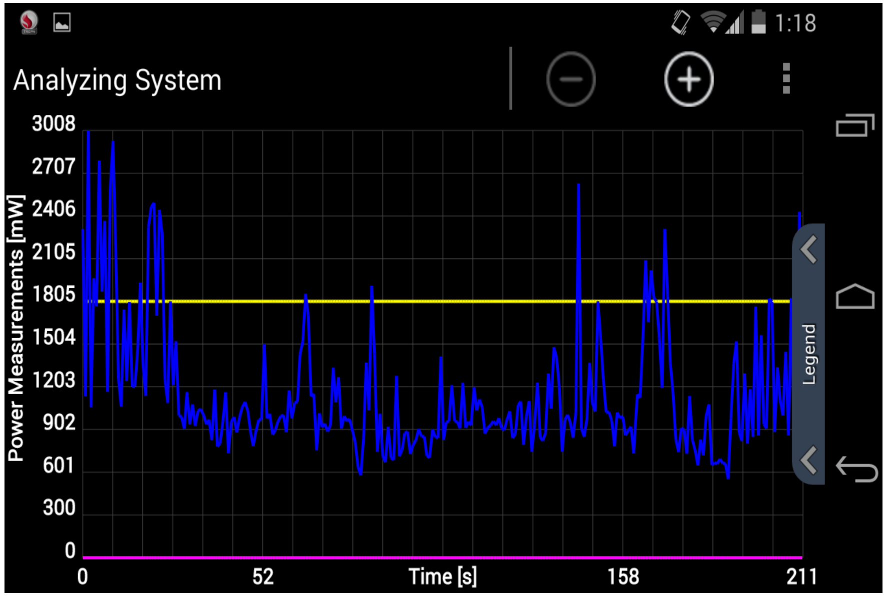




Device Power Consumption



Power Profile of Matrix Multiplication running on the Android Device without offloading (Average Consumption: 1864 mW)



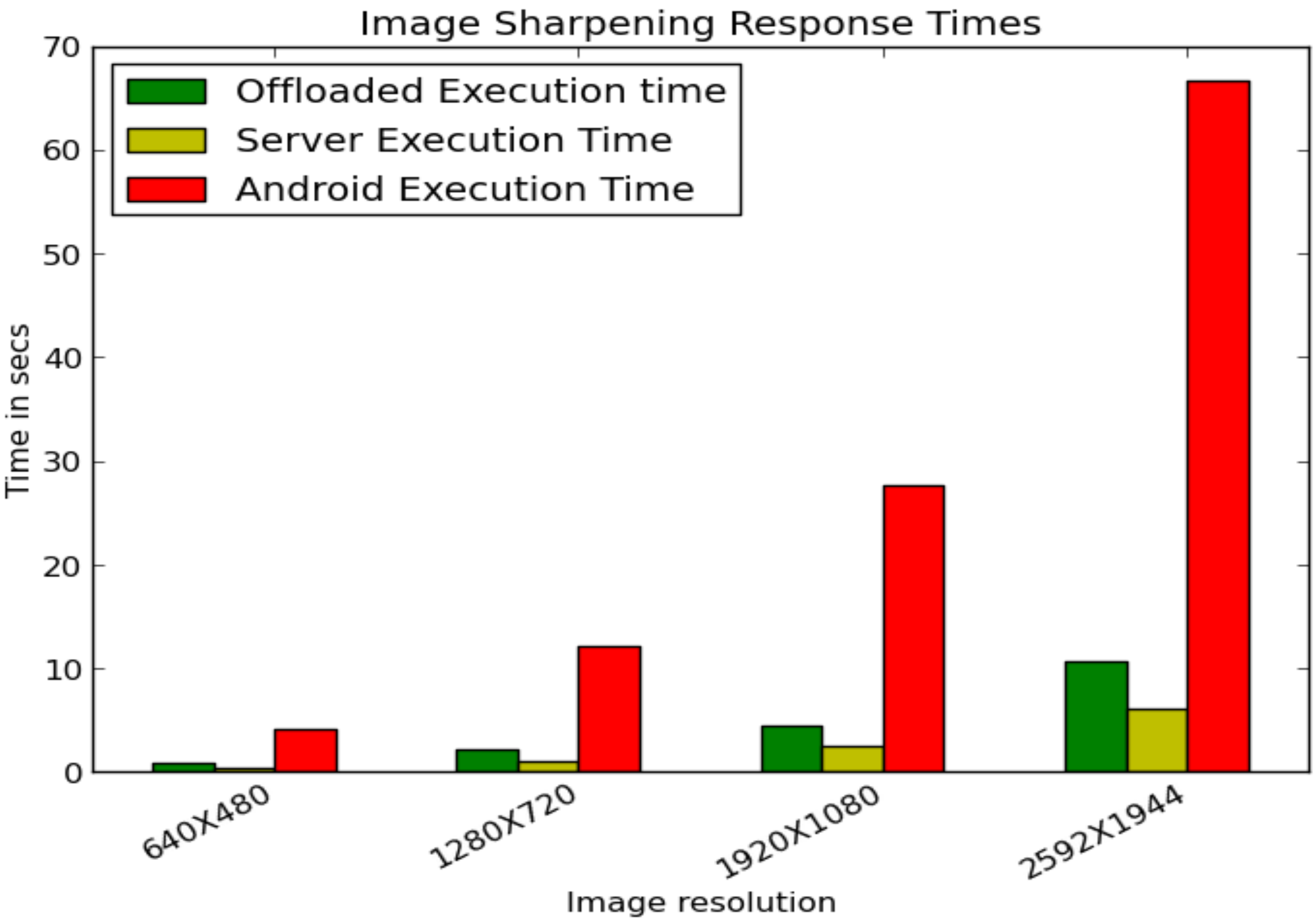
Power profile of Matrix Multiplication with offloading (Average consumption : 1073mW)

Improvement of 42% with offloading

Image Sharpener Benchmark

- Image Sharpening refers to enhancing the visual quality of the image by increasing the high frequency components in it.
- For this benchmark we first performed program analysis using SOOT, and detected the off loadable functions.
- Then the original application was modified to add the offload automation code.

Image Sharpener Results



Future Plans

- Implement a method in the operating system to identify the system load → network usage, signal strength, CPU usage, memory overhead
- Before offloading the user application calls the OS method, *shouldOffload* (via a library call or system call). Parameters → expected memory usage, duration of the job, and network usage.

- Plan: show performance benefits with a suite of real world applications on an Android based system.