The role of Device to Device & Edge Compute For IOT Applications

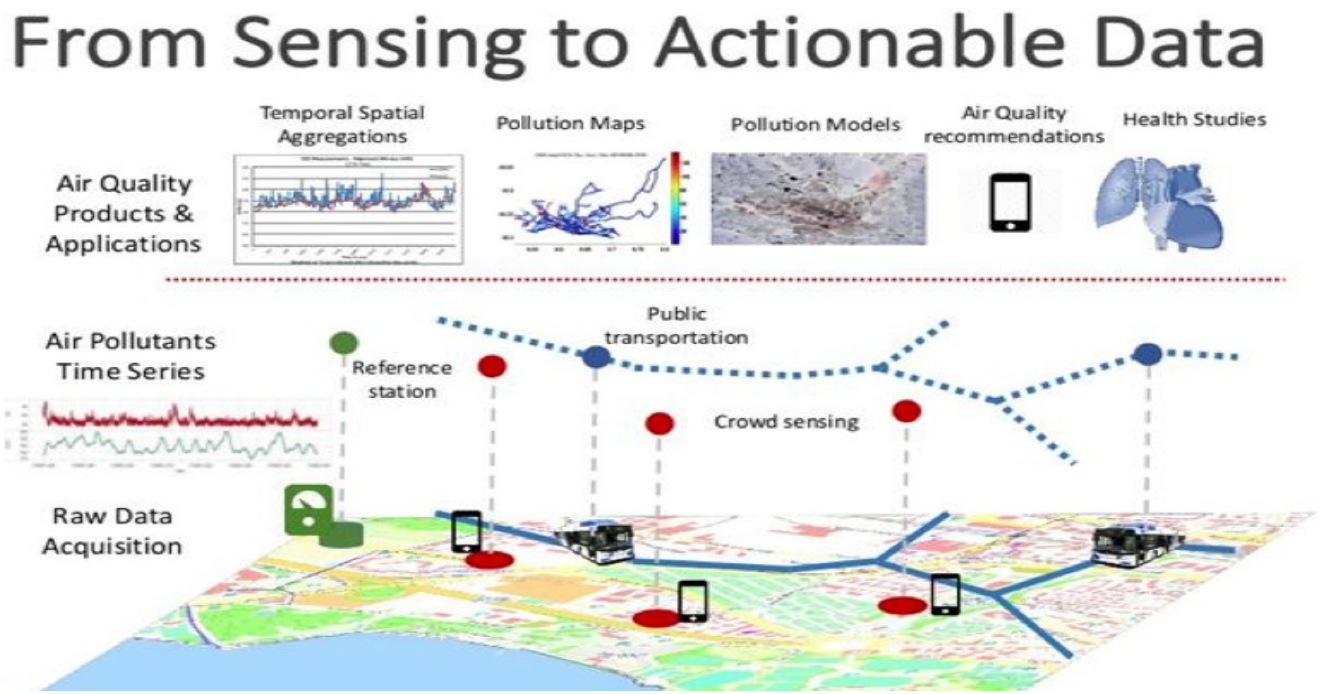
Huzur Saran

Department of Computer Science & Engg.

Indian Institute of Technology Delhi

New Delhi, India

IoT Framework



5G standards based Interfaces and protocols available for IoT app & device developers on 5G network
IoT apps like Air pollution monitoring, health care (will be explored with AIIMS) on network setup within IITD and will be made available to developers and manufacturers

Mobile Edge Computing

- IoT devices are resource limited, yet many applications require large levels of compute power
- Machine learning/deep learning
- Video encoding and streaming
- Vehicular applications Data aggregation and processing for roadside traffic assistance
- Mobile Edge Computing (MEC) has the potential to address IoT compute offload
- Being introduced in 5G cellular networks
- Brings computing power to within the radio access network (RAN)
- Low latency, ability to leverage 5Gnetwork virtualization

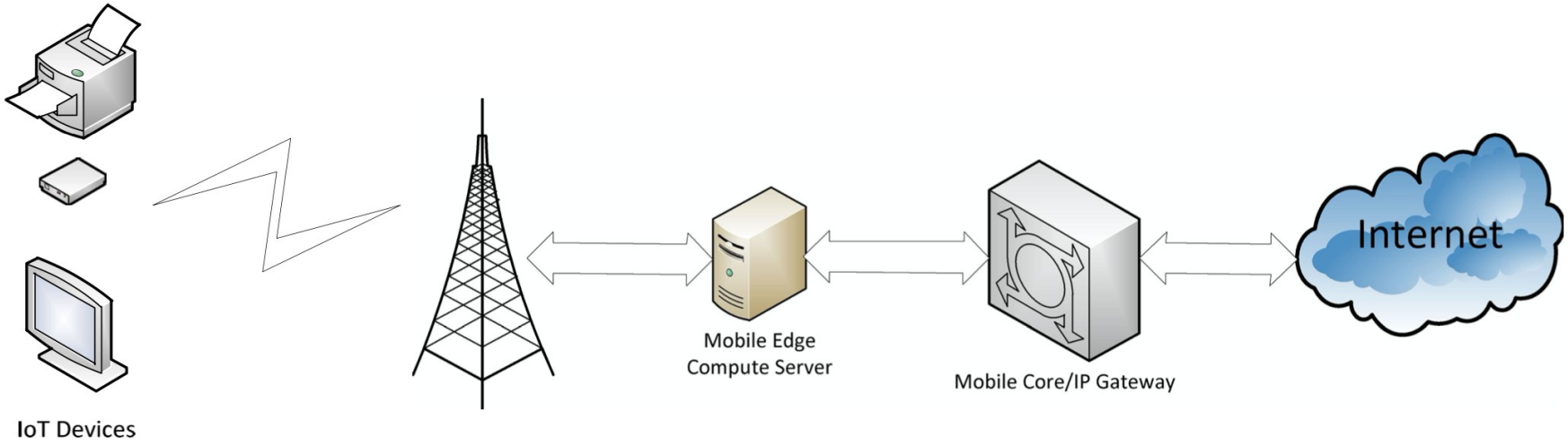
Cloud Computing versus Edge Computing (derived from [10])

Feature	Conventional Cloud	Edge Computing/Edge Cloud
Availability	Small no. of high-capacity datacenters	Large no. of small capacity datacenters
Proximity of compute resources	Usually remote	At edge, close to IoT devices
Latency (E2E)	Can be high, due to geographical distance	Low, due to proximity
Backbone network throughput	High, due to large data transfer to datacenters	Low, since most data processing is at edge
Scalability	Scalable at data center	Scalable at datacenter and edge
Security	Data vulnerable due to remote transmission, potentially through intermediaries	Attacks possible for data en route, but must be localized. Difficult to scale.

[10] Pan, J. and J. McElhannon. "Future Edge Cloud and Edge Computing for Internet of Things Applications." *IEEE Internet of Things Journal*. Vol. 5. No. 1. February 2018. pp. 439-449.

MEC Architecture

Derived from European Telecommunications Standard Institute [11]



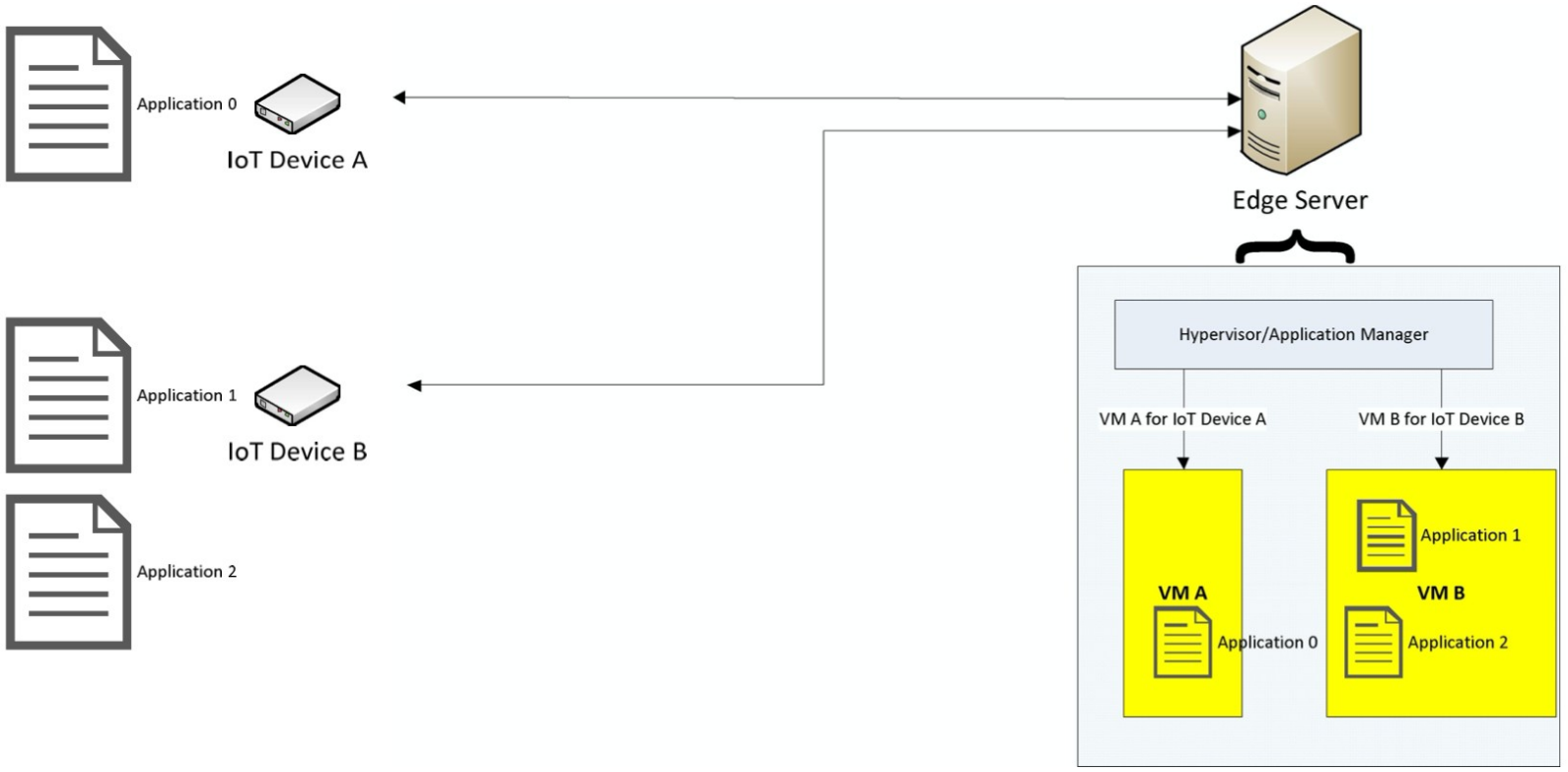
[11] Y.-C. Hu et al. "Mobile Edge Computing: A Key Technology Towards 5G." *ETSI White Paper*. No. 11. September 2015.

MEC - Challenges

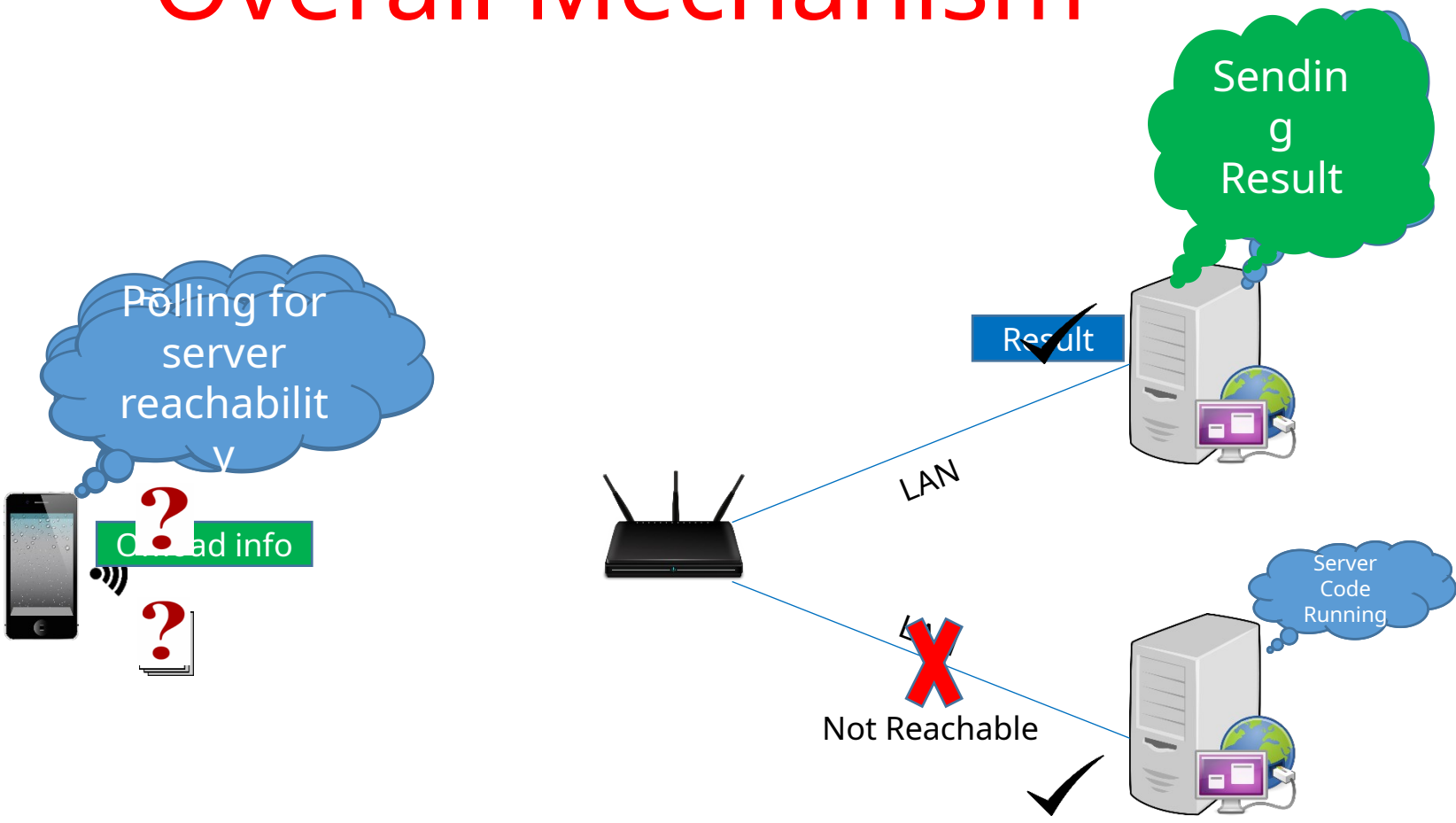
- Performance
  - Active Probing for QoS
  - Low Latency Transport
  - When and what to offload?

- Security: Edge Nodes should satisfy certain requirements
  - Tamper resistance
    - e.g. physical enclosures, disabling manufacturer debug/testmodes
  - Tamper (intrusion) detection
    - Detecting changes in normal operating conditions
  - Secure boot
    - Verify that all firmware in node is at latest version, code has not been modified

Edge Virtualization

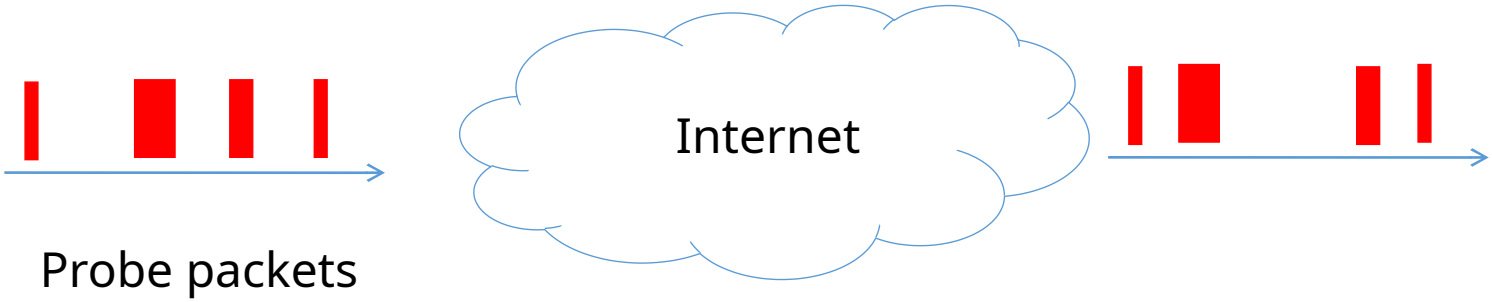


Overall Mechanism



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

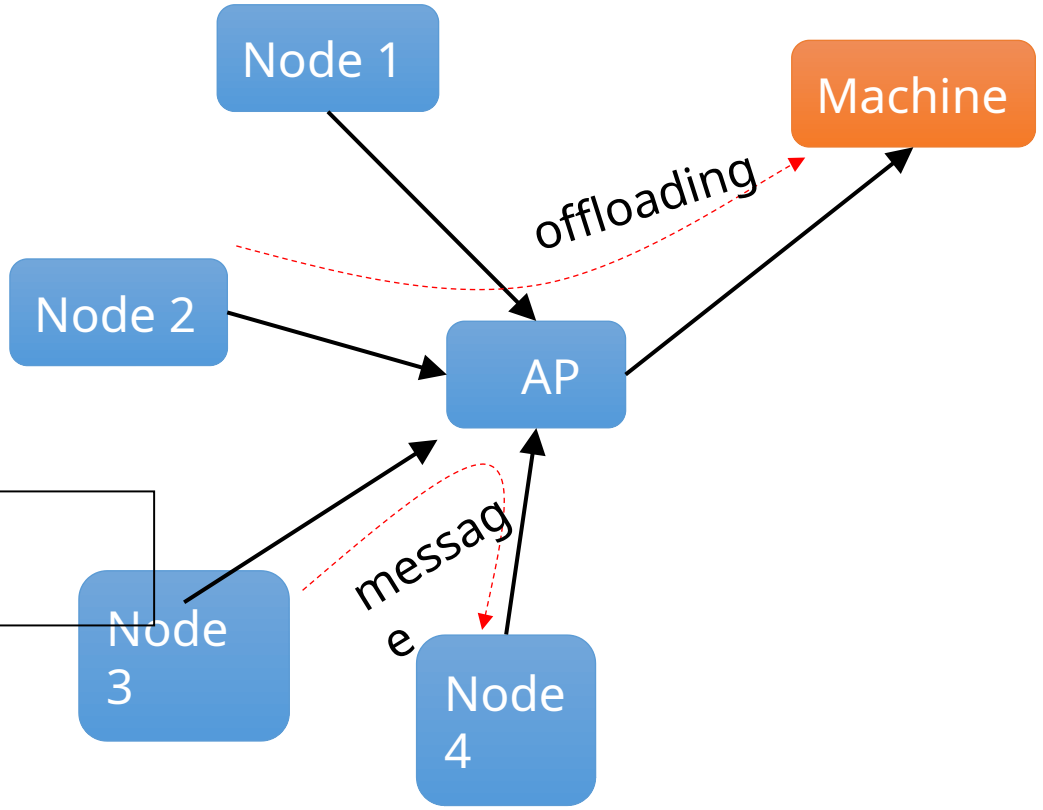


- Variation in probe packet gaps, or packet loss reveal information about e2e path
- Our finding: Existing probing tools (Spruce, Wbest, pathchirp, pathload etc.) perform poorly for 3G/4G networks
- Developed own app



Low Latency Transport Protocols

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



Time to compute locally > Time for offloading

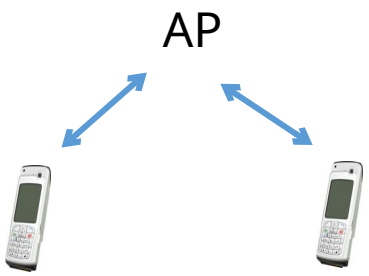
*Star Shape Topology

Time for offloading = Time to send + Time to compute

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



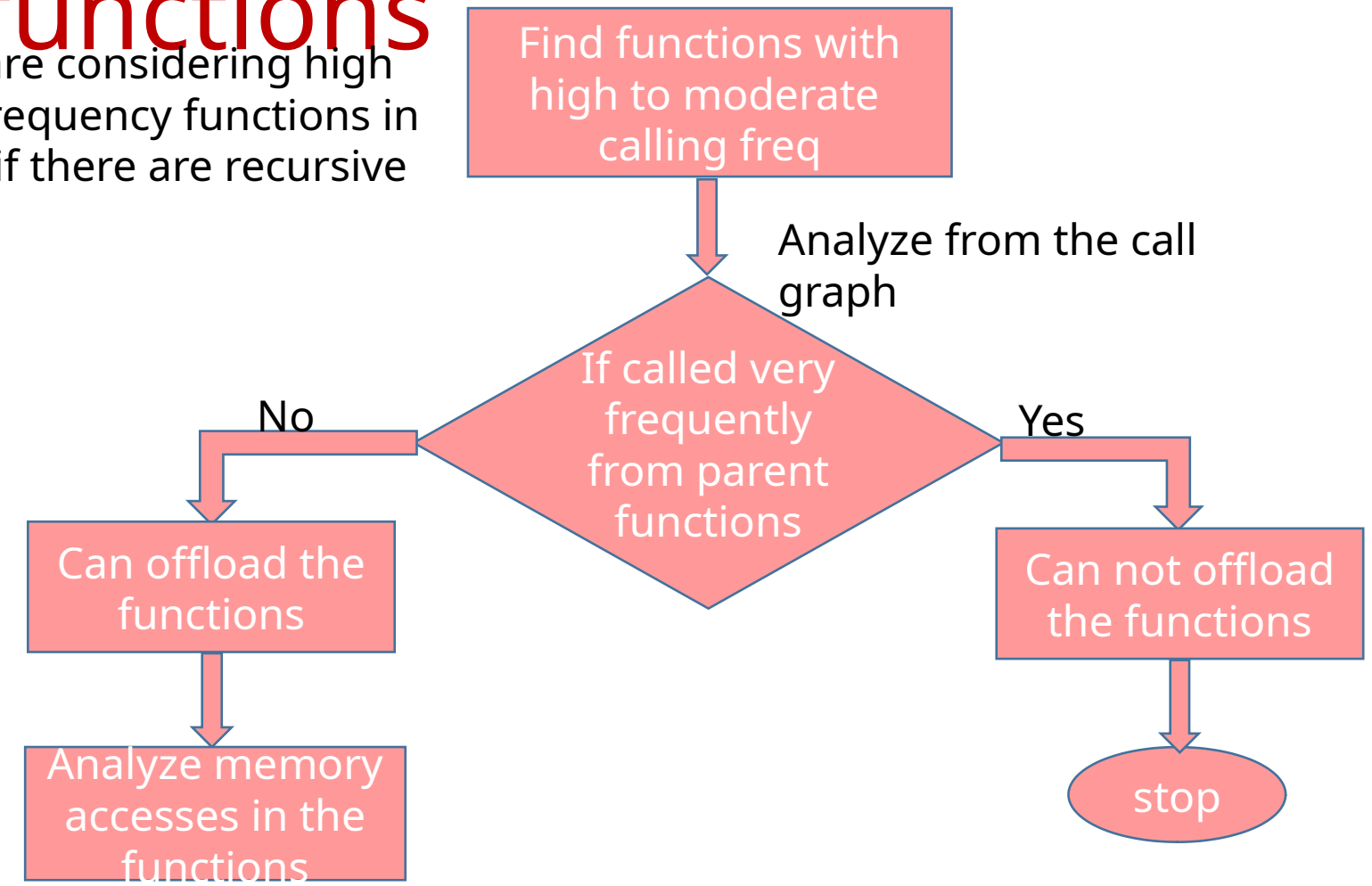
Del Ack parameter "d"	Throughput in Mbps
0	10.34
2	9.21
4	9.92
dynamic	13.58

Lossy environment (4-5%)

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

Detecting Offloadable functions

(We are considering high call frequency functions in case if there are recursive calls)



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 S(,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).



VIDEO STITCHING BENCHMARK

- Video Stitching is the process of combining the images/video frames with overlapping fields of view to produce a panorama image.



- In this benchmark, the video is captured and finally the stitched image is displayed.

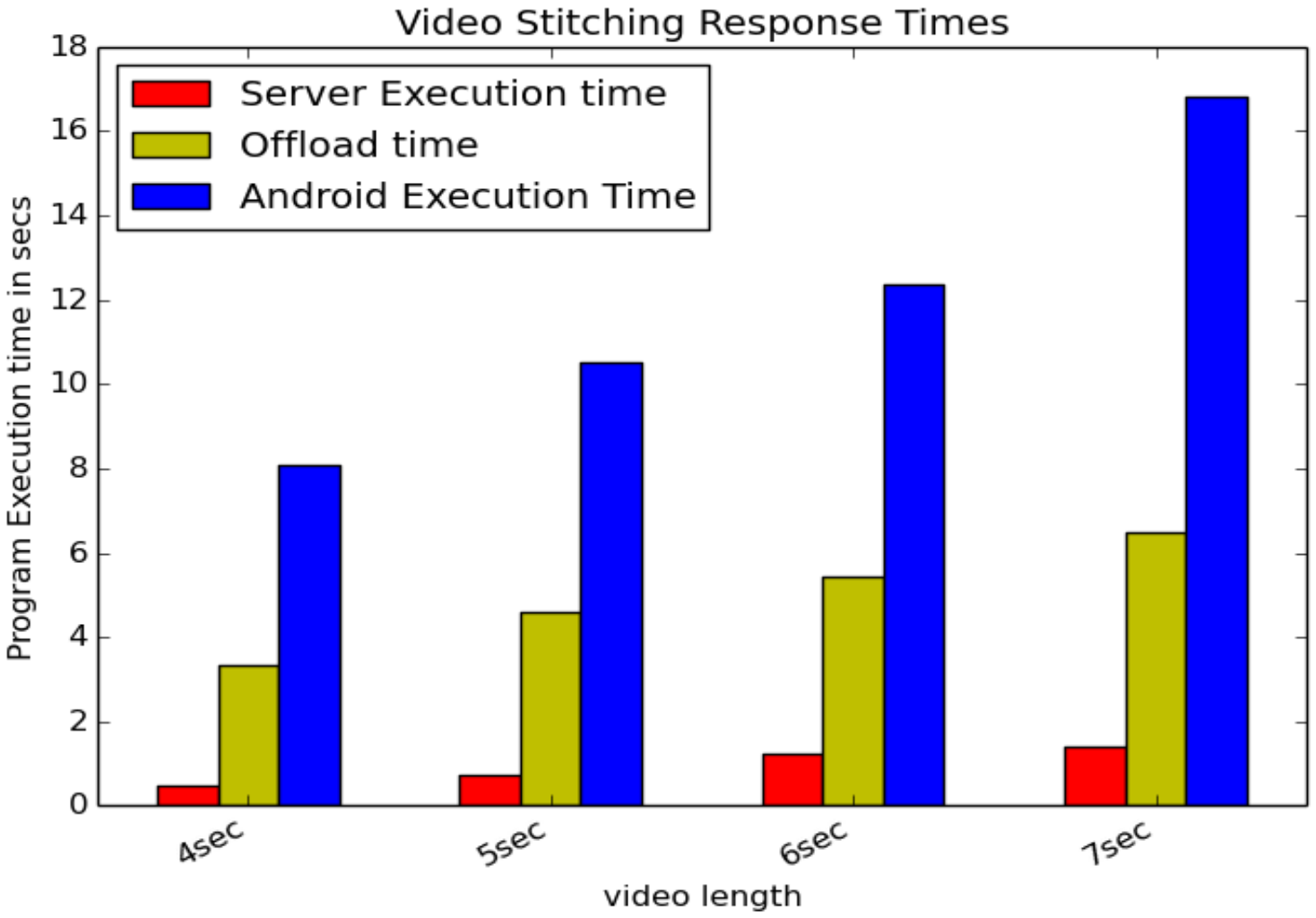


- Based on the video frame rate, certain frames are filtered out for stitching.

VIDEO STITCHING CONTD..

- Next, If all the offloading criteria are satisfied then the frames will be sent to the server and perform stitching at server side and send back the resulting image.
- Otherwise computation is performed at the device.
- Stitching computation time varies based on the video length, video resolution.
- It has applications such as satellite imaging and medical imaging

Video Stitching Benchmark Results



Time breakup of offloading for Video Stitching Benchmark

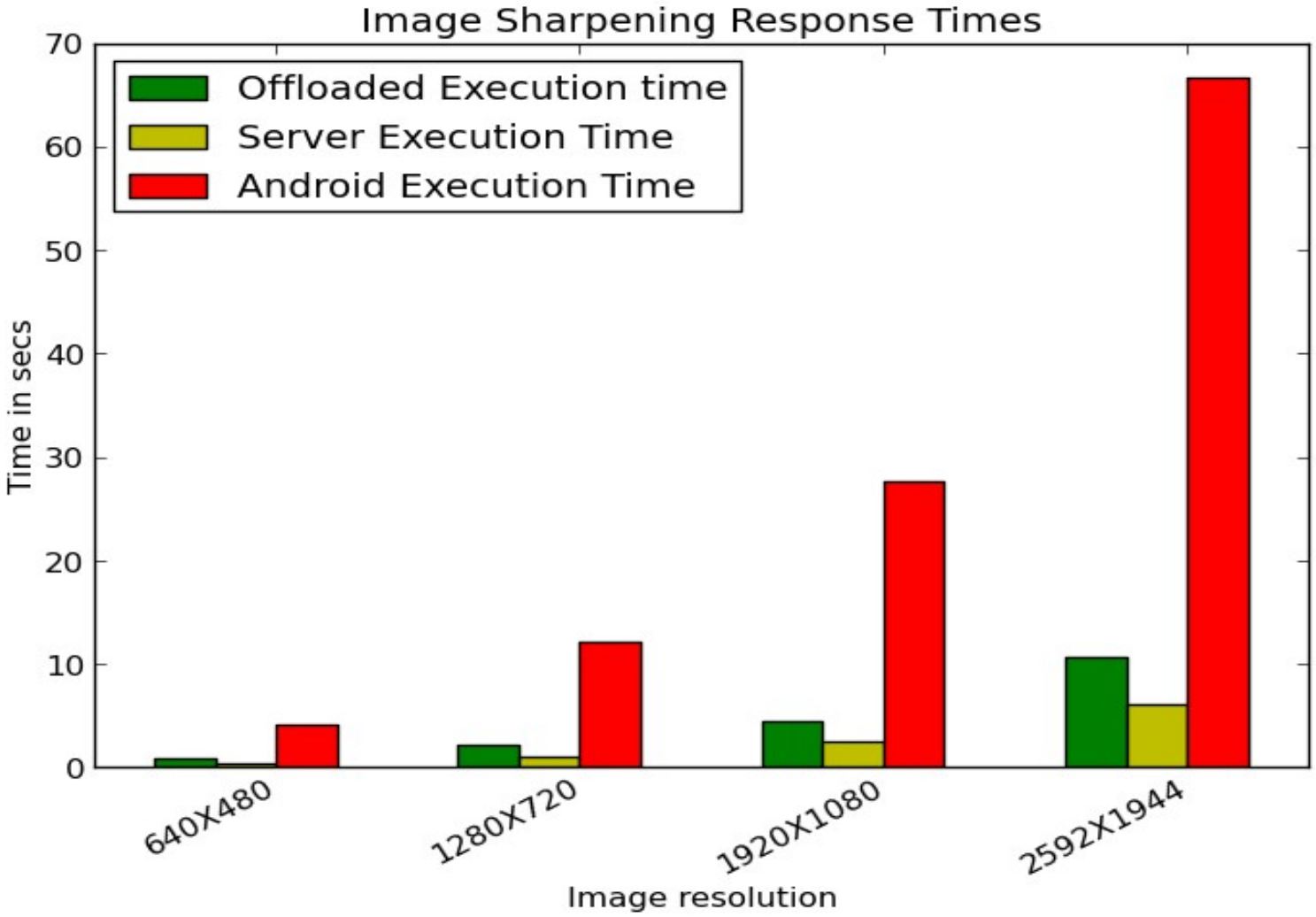
Video resolution	Video length	Total offload time	TCP handshake	Communication time	Server computation time
480 X 320	6 sec	5.80 sec	8 ms	2.3 sec	3.5 sec
640 X 480	6 sec	7.29 sec	10 ms	3.5 sec	3.78 sec
720 X 480	6 sec	8.79 sec	20 ms	4.37 sec	4.4 sec

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



Mean and Std deviation values

	ARM processor offload time		Atom processor offload time	
Image resolution	Mean time (s)	Std deviation (s)	Mean time (s)	Std deviation (s)
640 X 480	0.85	0.07	0.73	0.03
1280x720	2.51	0.08	1.93	0.08
1920x1080	4.50	0.06	4.25	0.16
2592x1944	10.63	0.09	10.03	0.26