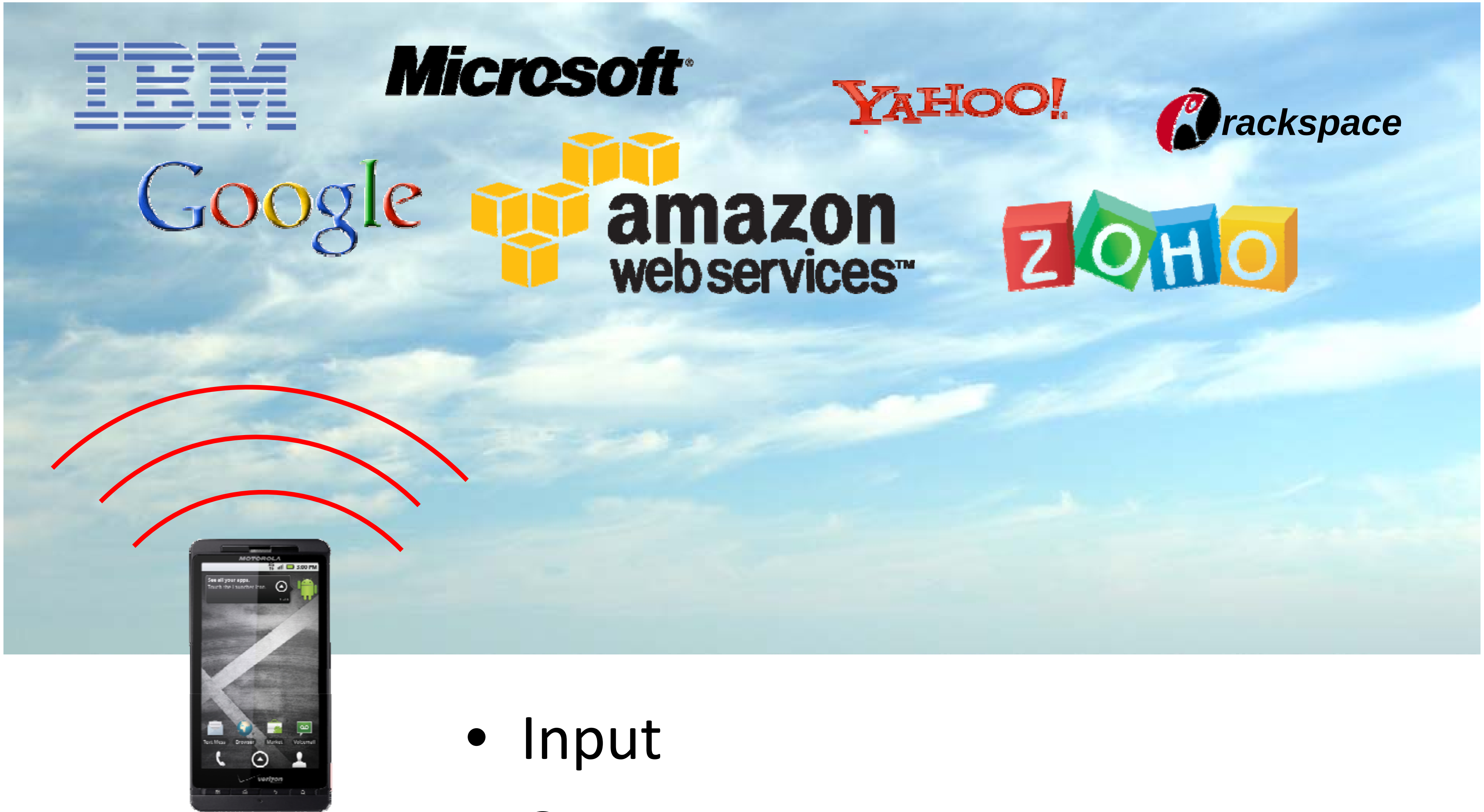
Mobile System Concerns in the Cloud Age

Lin Zhong

Rice Efficient Computing Group (recg.org)

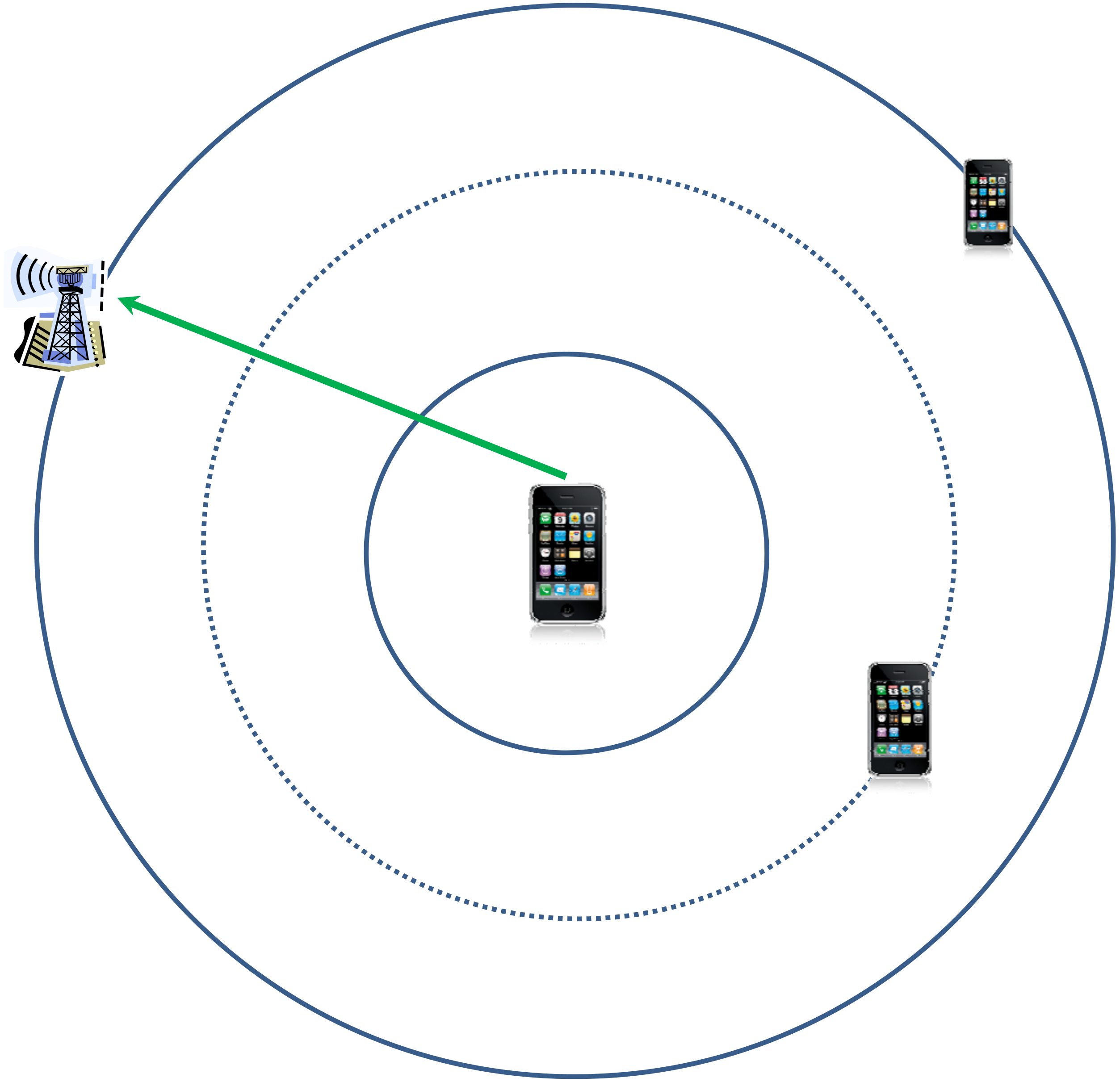
Rice University



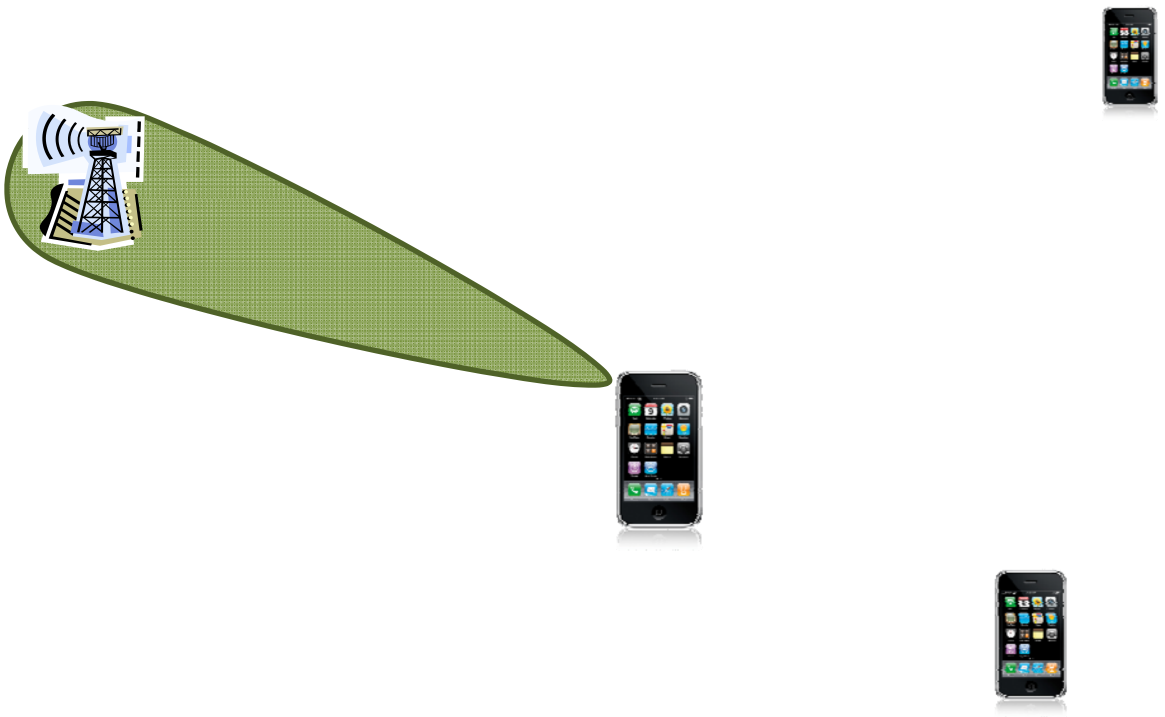
- Input
- Output
- Wireless connectivity


ARMATO ICE & FIREWOOD
Levi's
PALACE
PARK
nyc

Omni directional transmission a key bottleneck



Ongoing project with Ashutosh Sabharwal

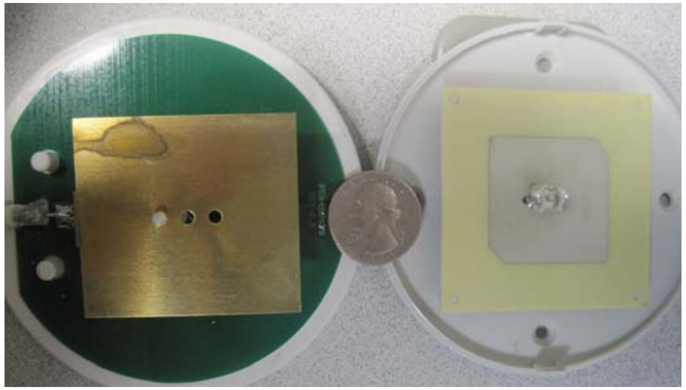


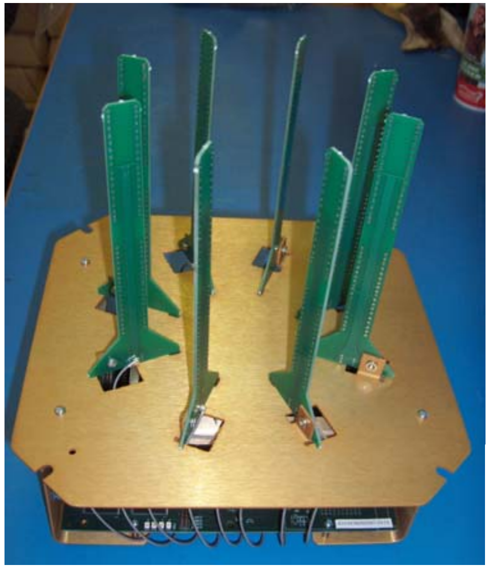
ISLPED'10 and MobiCom'10

Two ways to realize directionality

- Passive directional antennas
  - Low cost
  - fixed beam patterns
  - MobiCom'10
- Digital beamforming
  - Flexible beam patterns
  - High cost

8 and 5dBi antennas

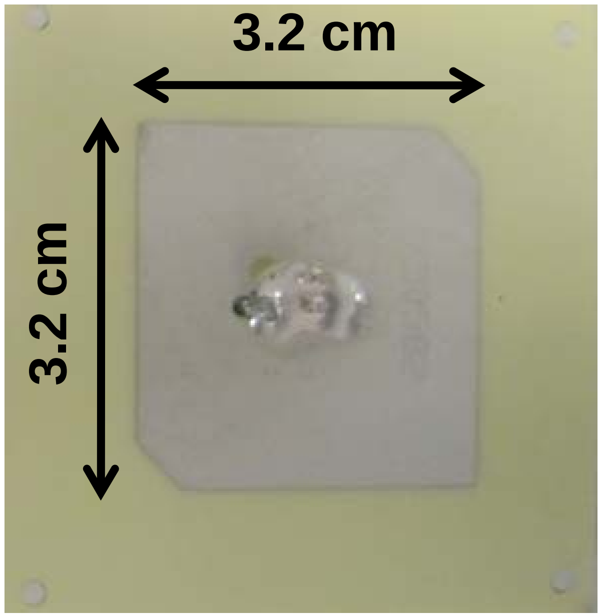


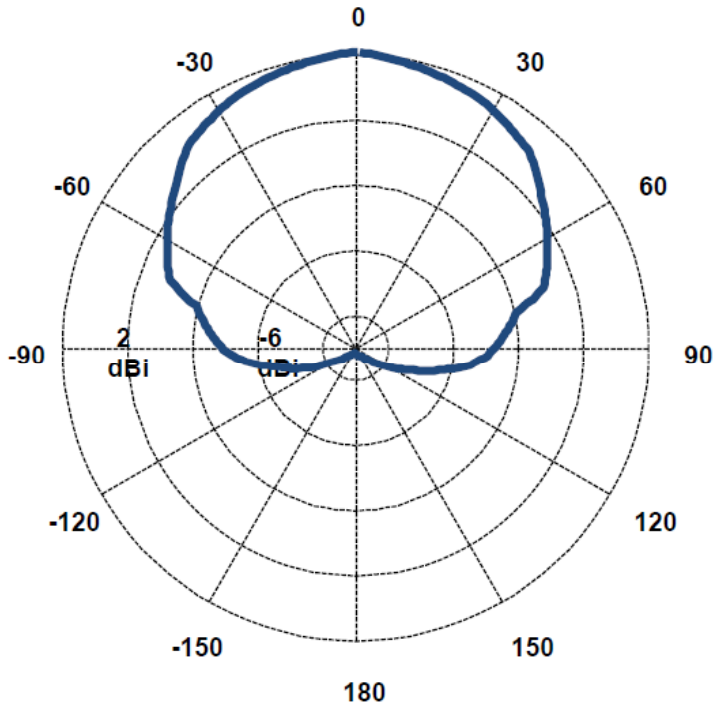


Phased-array antenna system from Fidelity Comtech

Passive directional antennas

- Microstrip antenna (5dBi peak gain)





Challenges

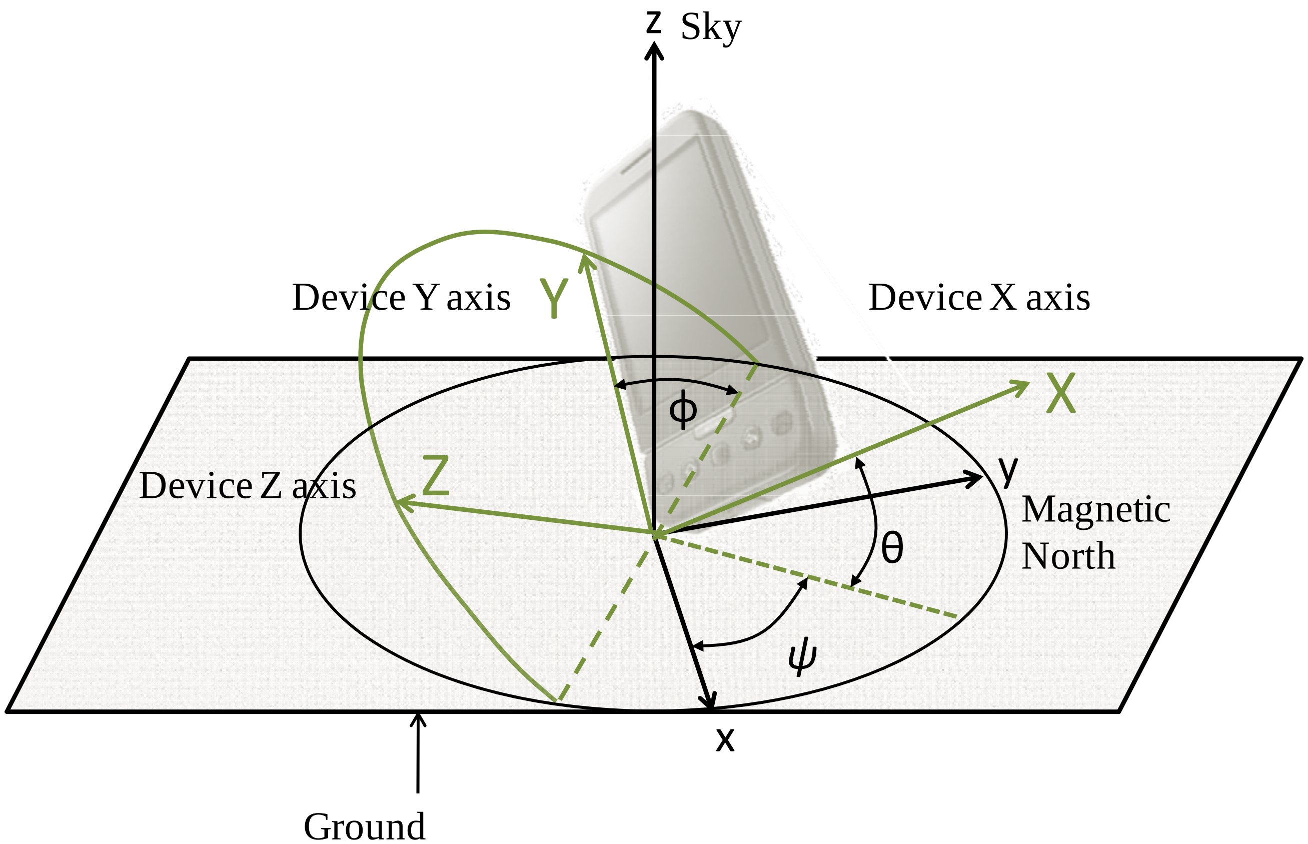
- Multipath effect
  - Hard to find the best transmit direction
- Mobility and rotation
  - Destroys the already found best direction
  - Rotation is most challenging

Questions we try to answer

1) How do smartphone-like mobile device rotate during wireless access?
2) How do directional antennas behave with indoor and non-line-of-sight (NLOS) propagations?
3) How can a device dynamically select the best antenna?

Orientation estimation using Euler Angles

- θ and ϕ based on tri-axis accelerometer
- ψ based on tri-axis compass and θ and ϕ

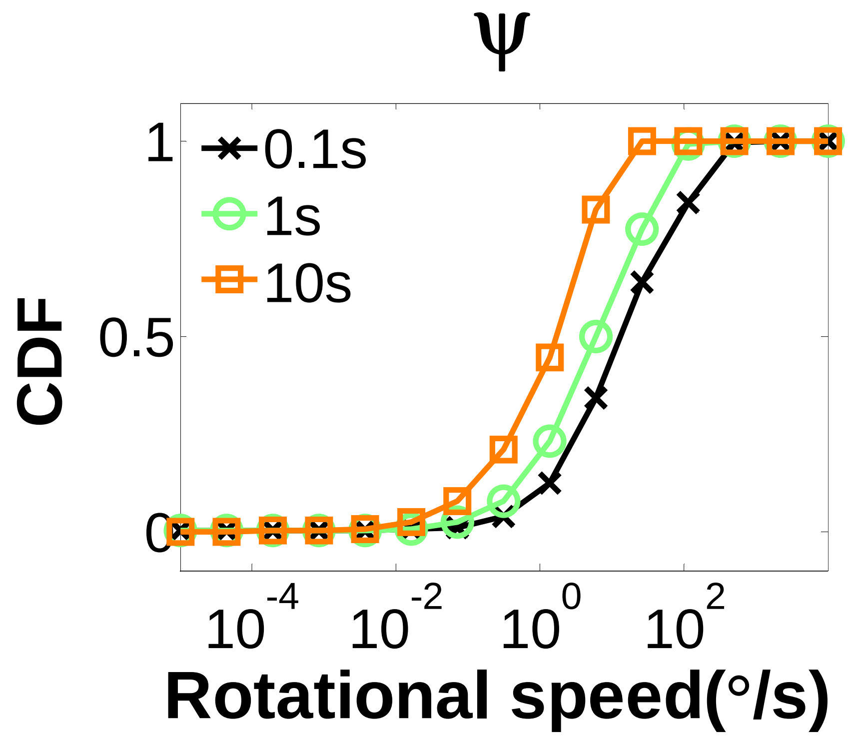


User study

- Collecting accelerometer/compass data
- 11 users with G1 android phone
- One week of continuous measurement for each user

- Data is made open access
  - http://www.ruf.rice.edu/~mobile/downloads.htm

Device rotates slowly

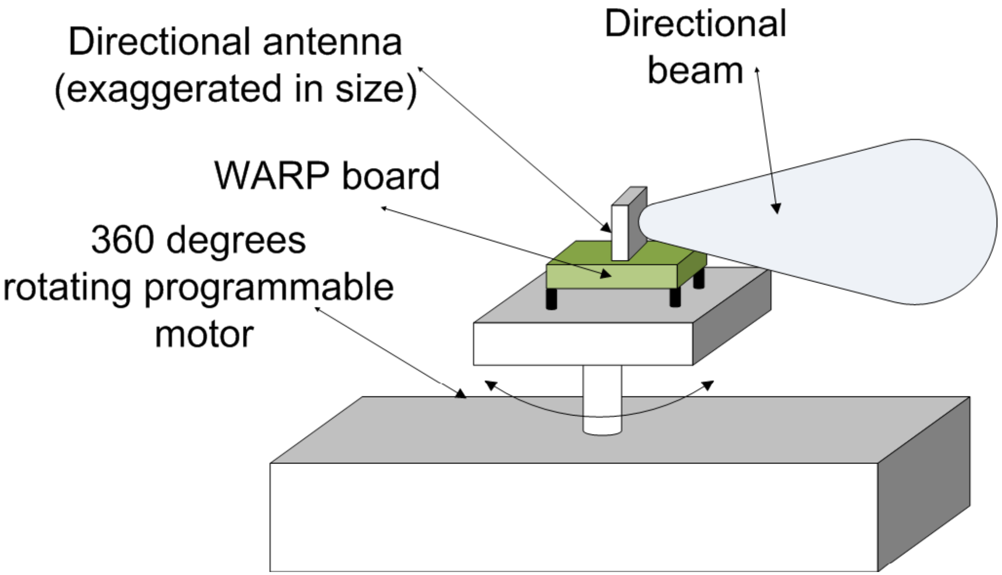


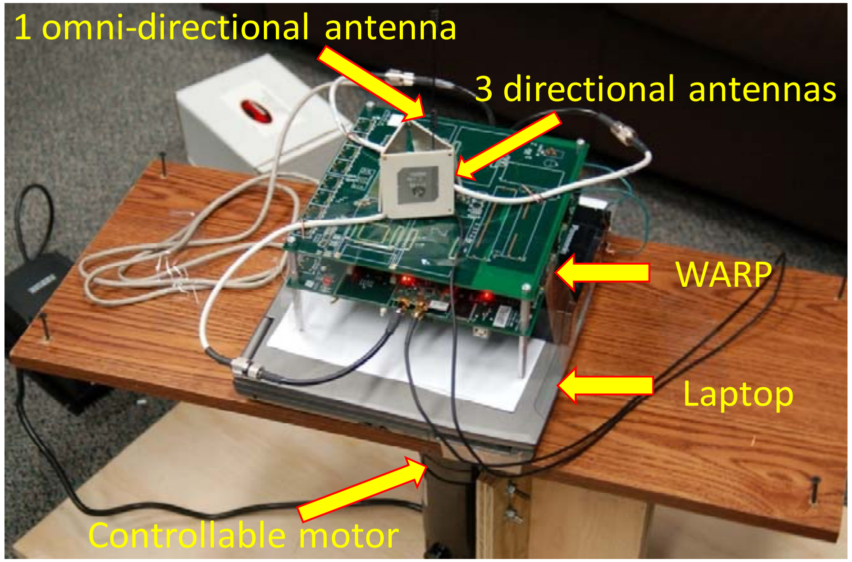
Slower than 120°/s for 90% of the time

Questions we try to answer

1) How do smartphone-like mobile device rotate during wireless access?
2) How do directional antennas behave with indoor and non-line-of-sight (NLOS) propagations?
3) How can a device dynamically select the best antenna?

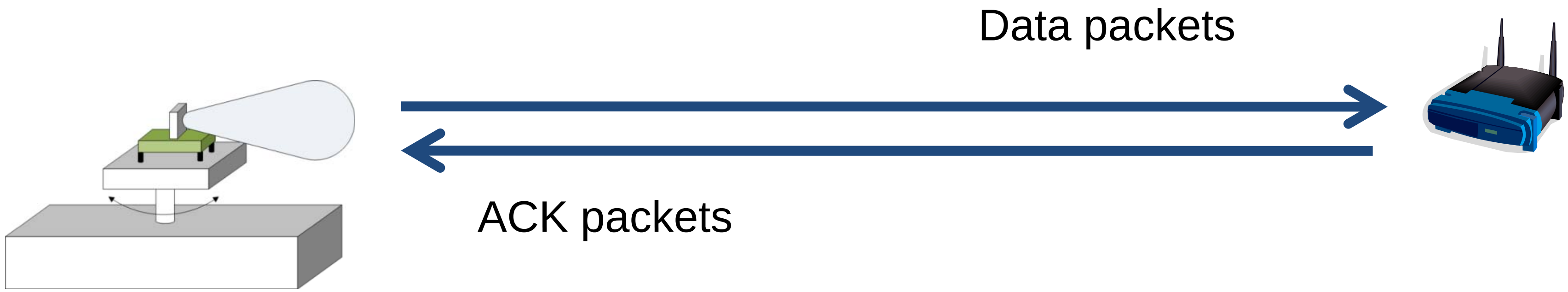
Measurement setup





First measurement

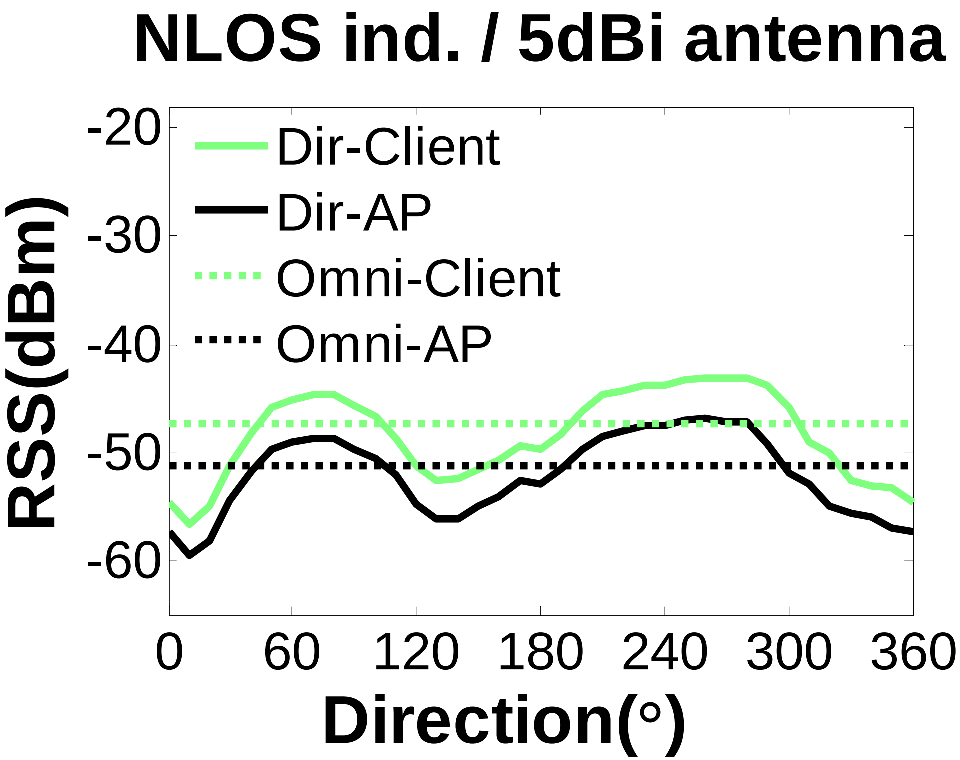
- RSSI measured at both ends



Different directions in the horizontal plane

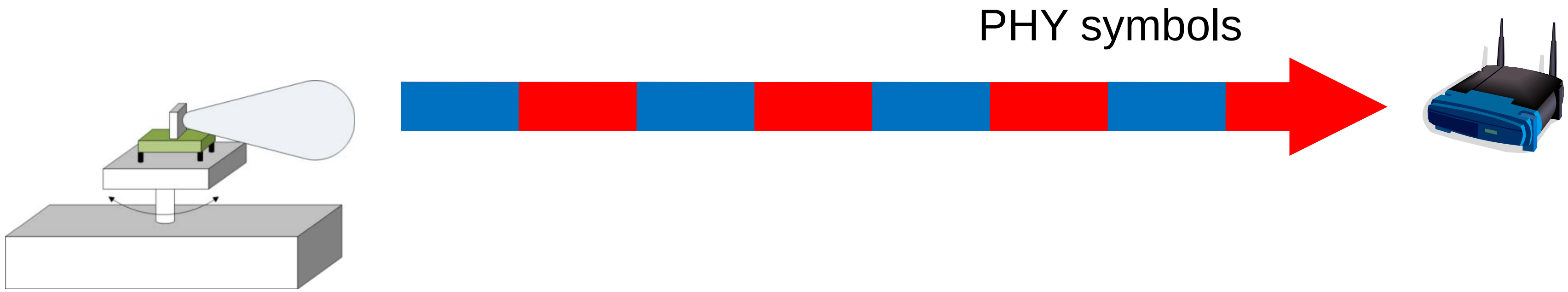
Directional antennas outperform omni in big angular regions

Channel reciprocity holds



Second measurement

- RSSI measured at AP



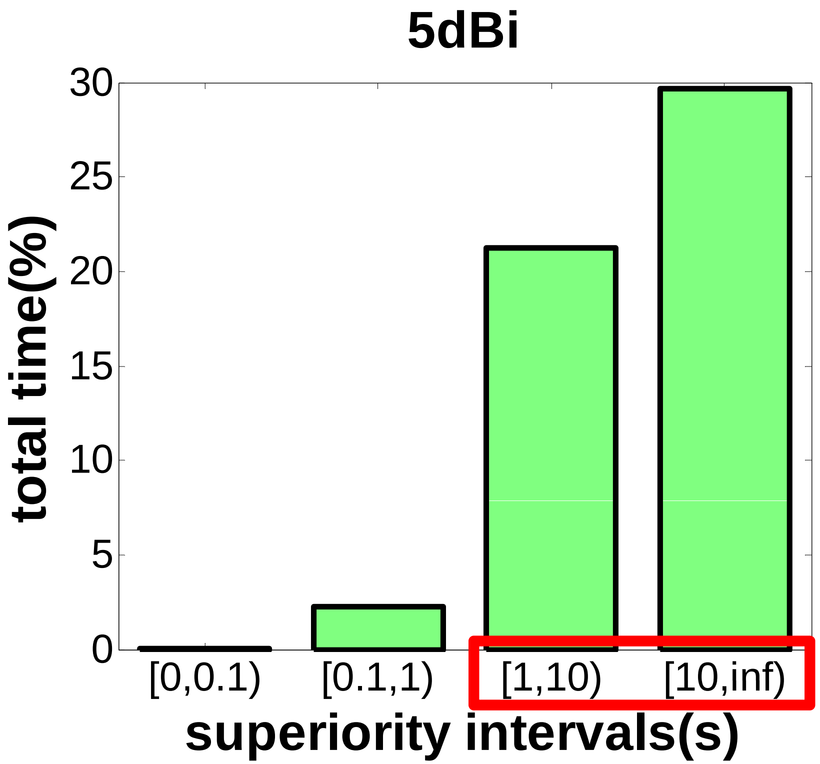
One hour of rotation according to user study traces

Sent from omni antenna

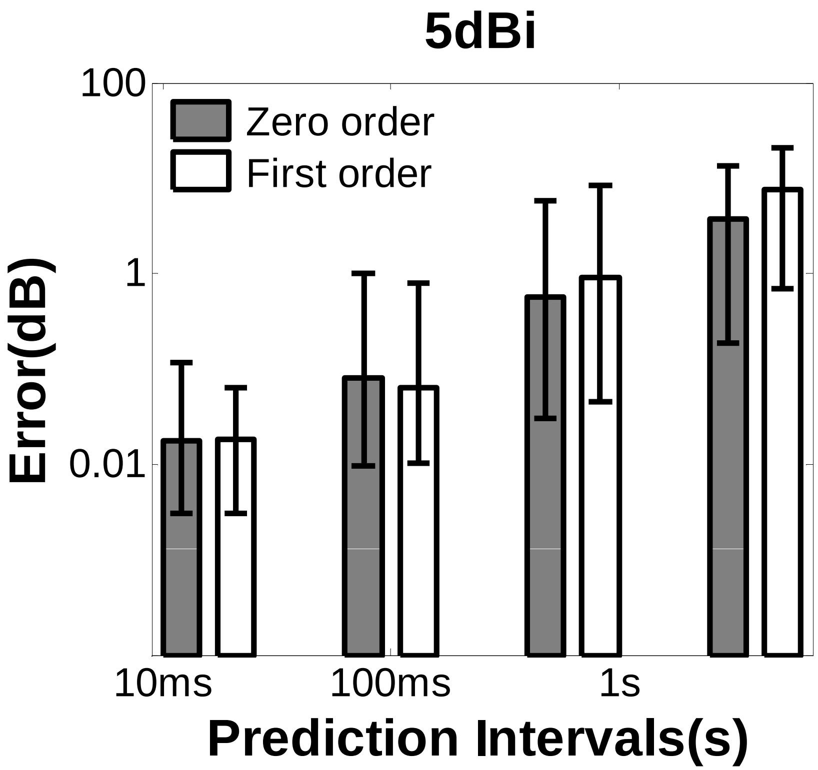
Sent from directional antenna

Directional outperforms omni for ~50% of the total time

Superiority intervals > 1 second



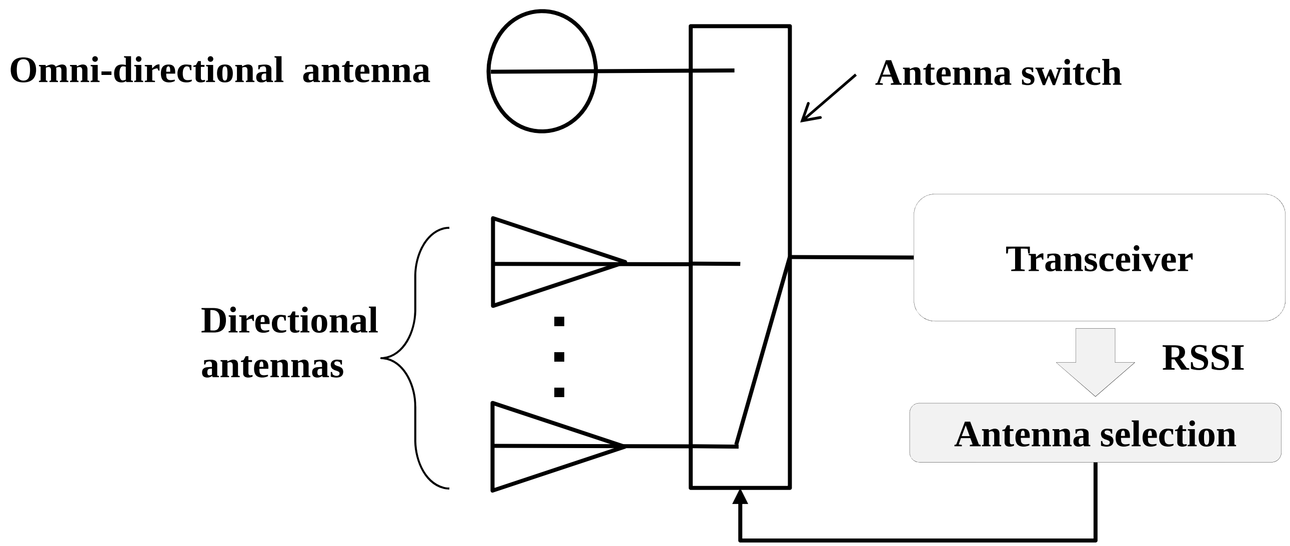
RSS is predictable



Less than 0.1 dB error for short intervals

Multi antenna design (MiDAS)

- Only one RF chain
  - One active antenna at a time
- Directional antennas used only for Data transmission and Ack reception



Questions we try to answer

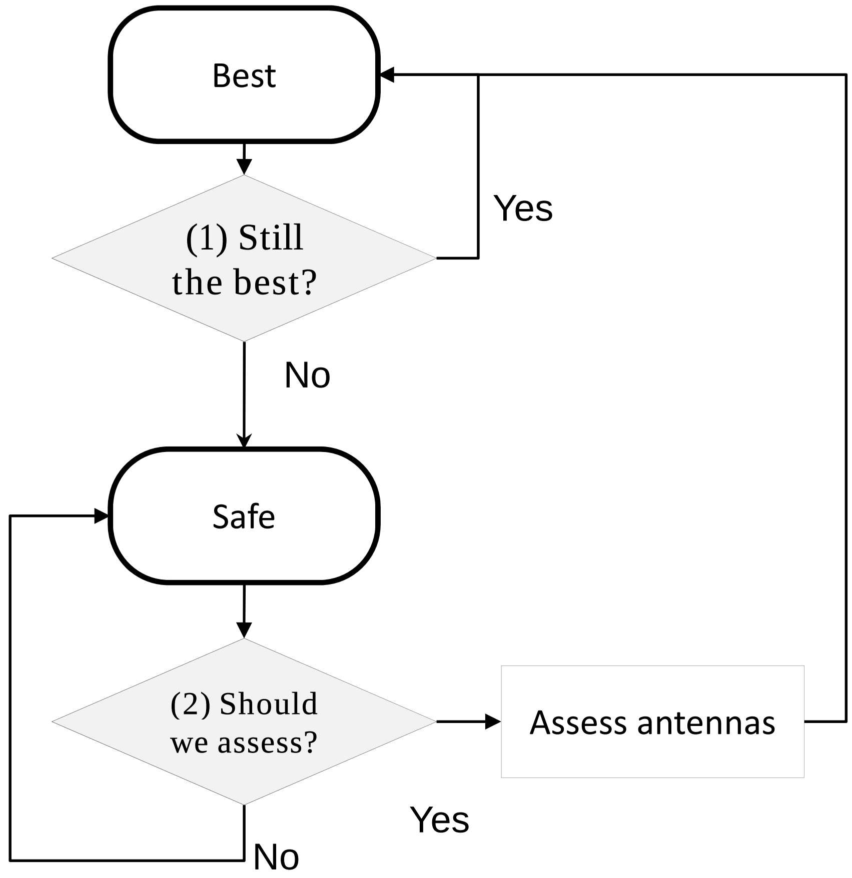
1) How do smartphone-like mobile device rotate during wireless access?
2) How do directional antennas behave with indoor and non-line-of-sight (NLOS) propagations?
3) How can a device dynamically select the best antenna?

Packet-based antenna selection

- Works for legacy networks
- Antenna assessment based on ACK packet
  - Uses channel reciprocity
- One antenna assessment per packet
  - Antenna assessment is expensive

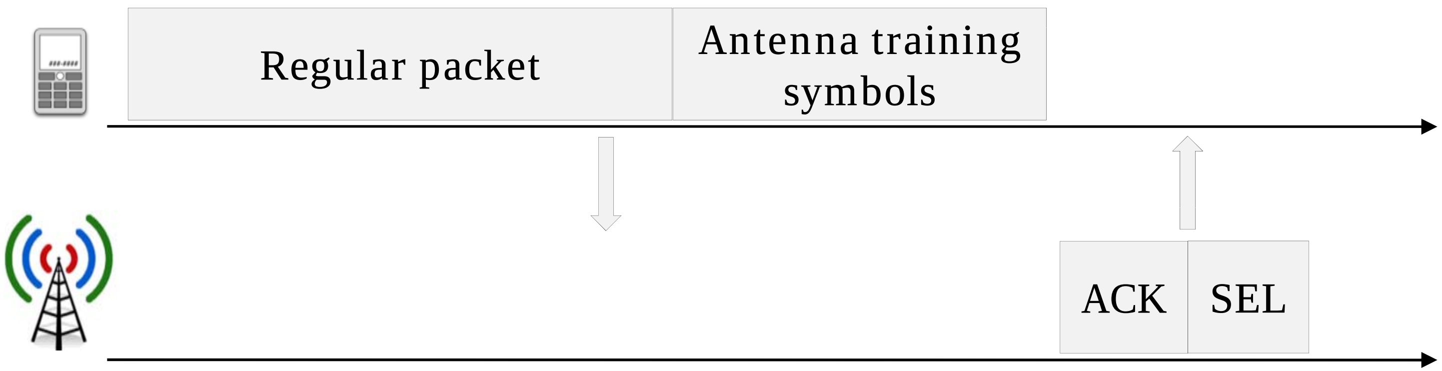
Heuristic antenna selection

- **Best mode:** Uses the previously selected right antenna
- **Safe mode:** Uses omni antenna when RSS changes rapidly



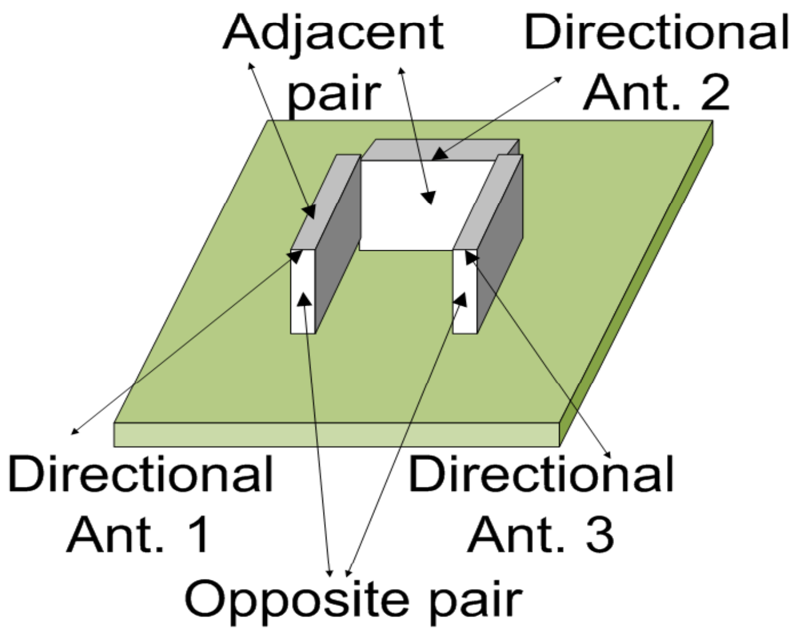
Symbol-based antenna selection

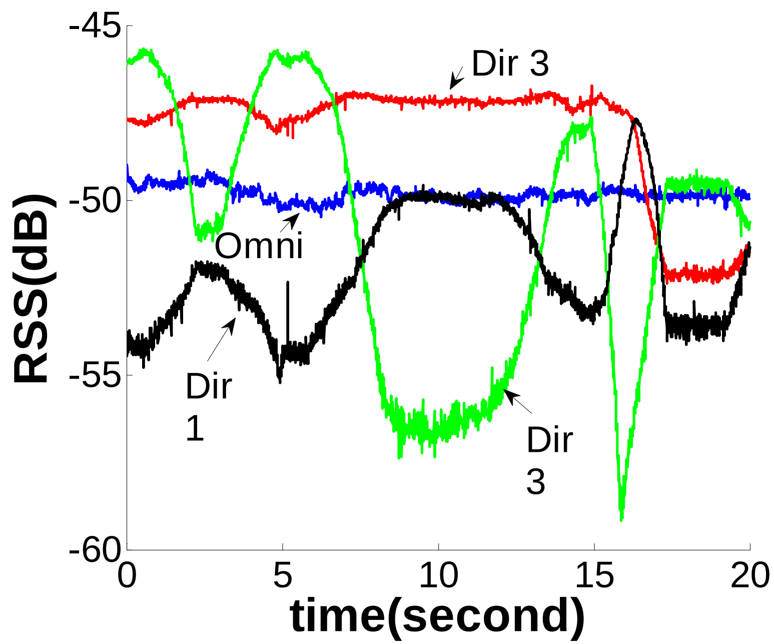
- Needs change to the PHY layer
- Data packet-based assessment
  - Does not depend on channel reciprocity
- Assess all antennas per packet



Trace based evaluation

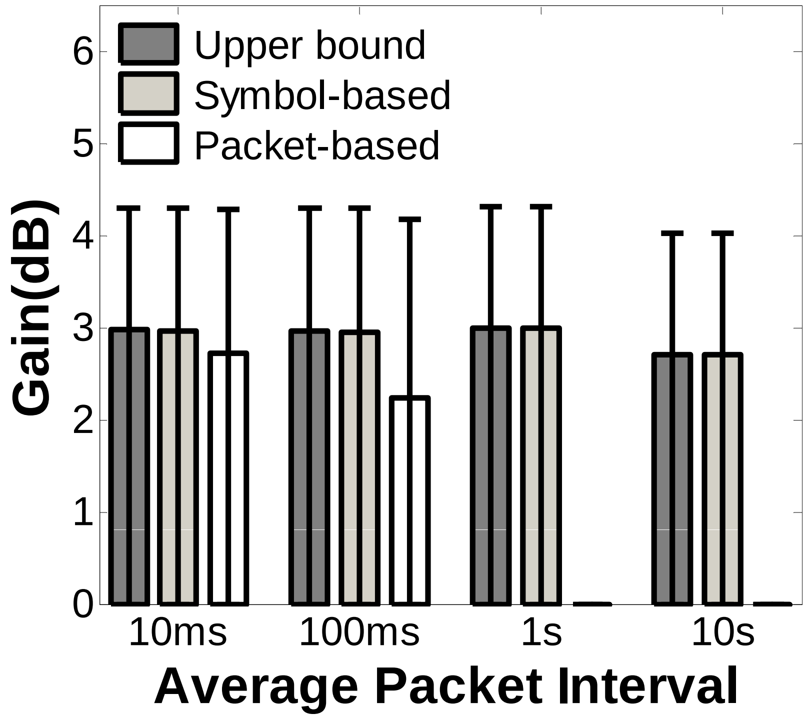
- Rotation traces replayed on the motor
- RSSI traces collected for all antennas
- Algorithms evaluated on traces offline





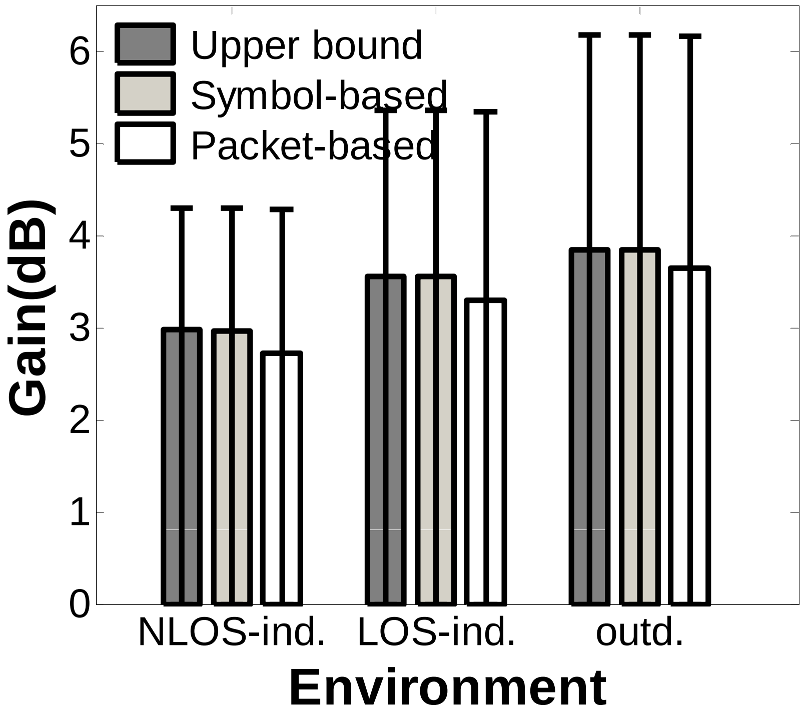
Both selection methods work well for short average intervals

~3dB possible link gain



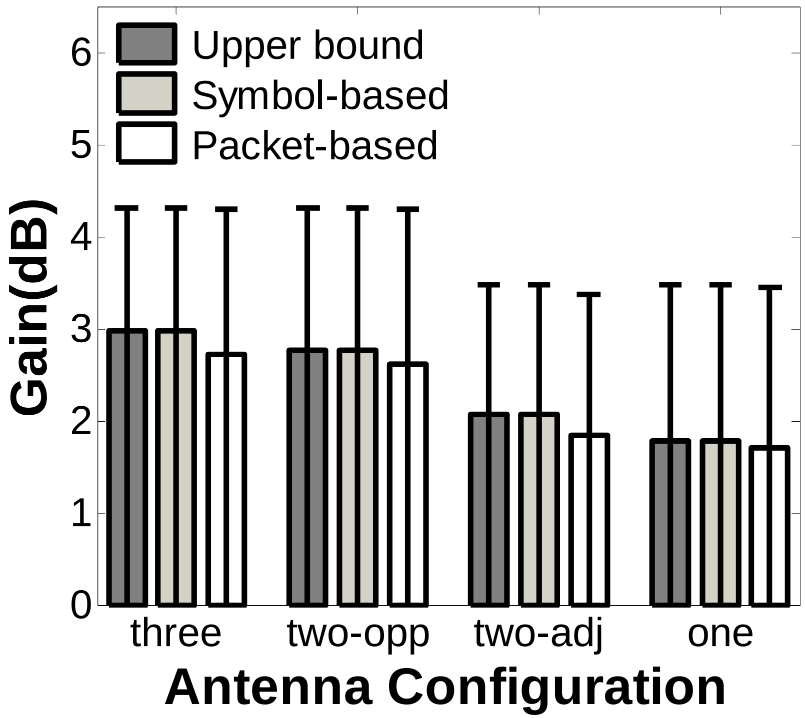
- Poisson traffic
- Three directional antennas and one omni
- 5dBi directional antennas
- NLOS indoor environment

Good performance in all environments



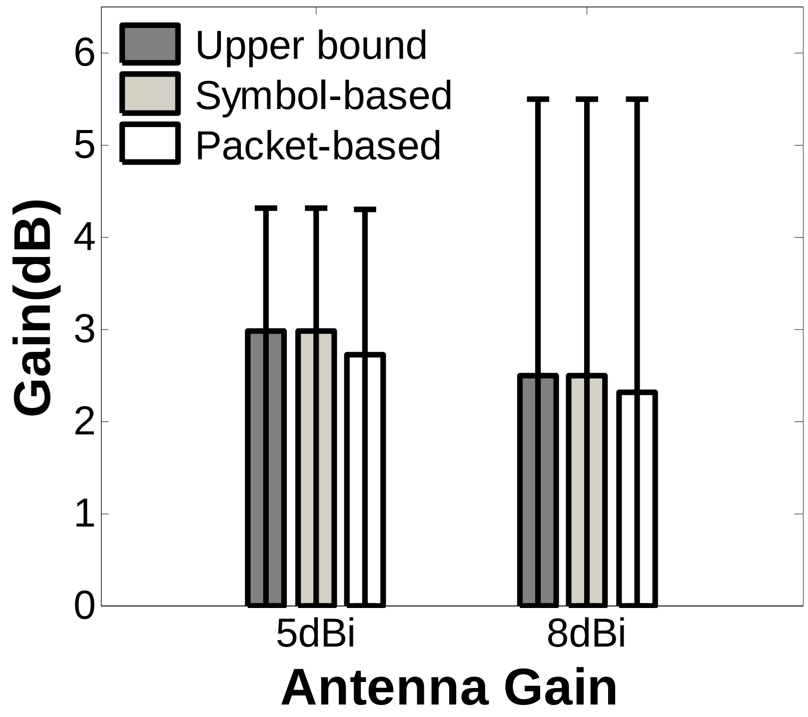
- Poisson traffic with 10ms average interval
- Three directional antennas and one omni
- 5dBi directional antennas

Two directional antennas are enough



- Poisson traffic with 10ms average interval
- 5dBi directional antennas
- NLOS indoor environment

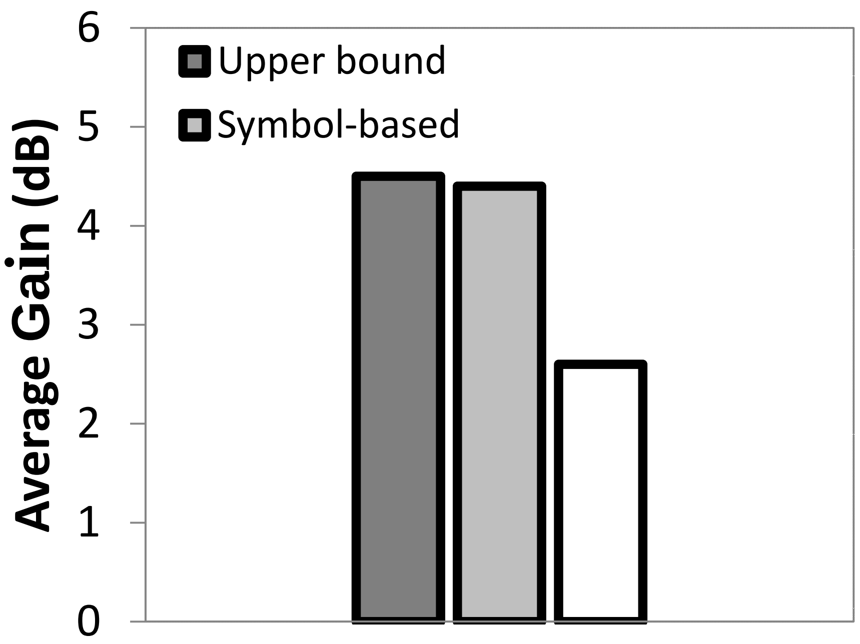
More focused beam is not necessarily better



- Poisson traffic with 10ms average interval
- Three directional antennas and one omni
- NLOS indoor environment

MiDAS handles mobility too

- MiDAS client rotating according to the traces
- Omni AP moves around randomly

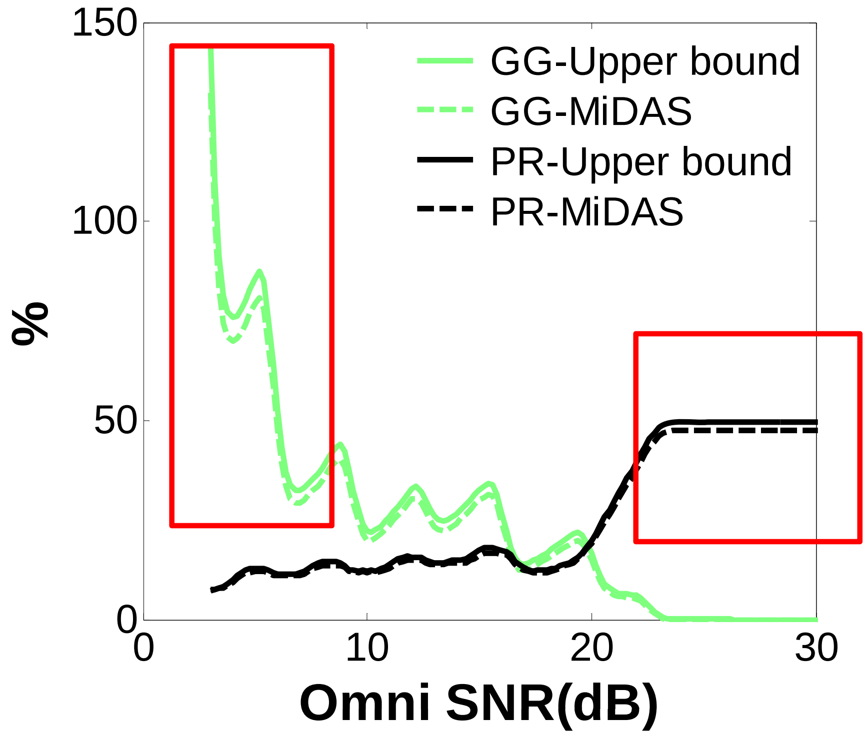


Adding Rate Adaptation & Power Control

- Why?
  - Realize the gain of MiDAS in terms of goodput and power saving
- How?
  - SNR-triggered rate adaptation
    - Uses goodput-rate table of the wireless card
  - Pick the highest rate possible, given SNR
  - Reduce the transmit power as much as possible not to hurt the chosen rate more than a threshold

Goodput gain for weak connections

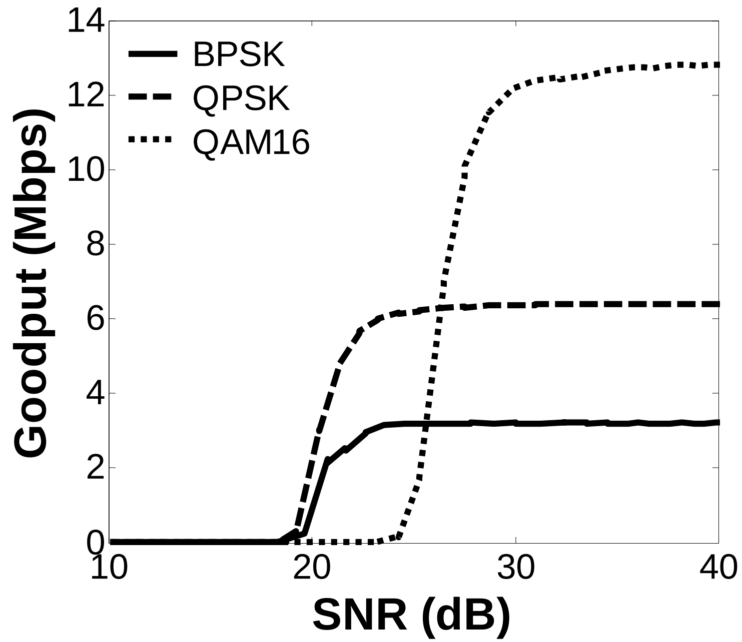
Power saving for strong connections



Simulated for 802.11a with 8 rates

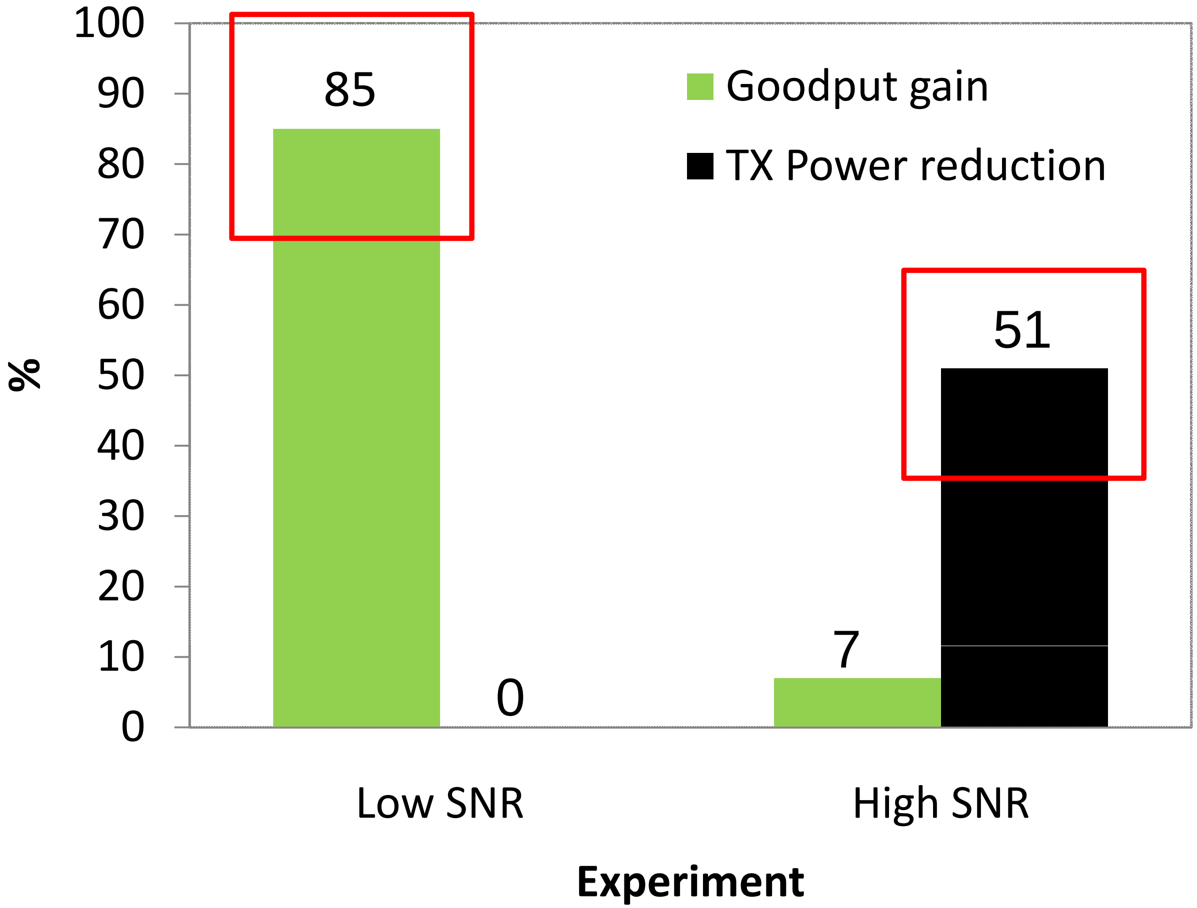
Real-time experiment

- WARP goodput-rate table



Goodput gain for weak connections

Power saving for strong connections

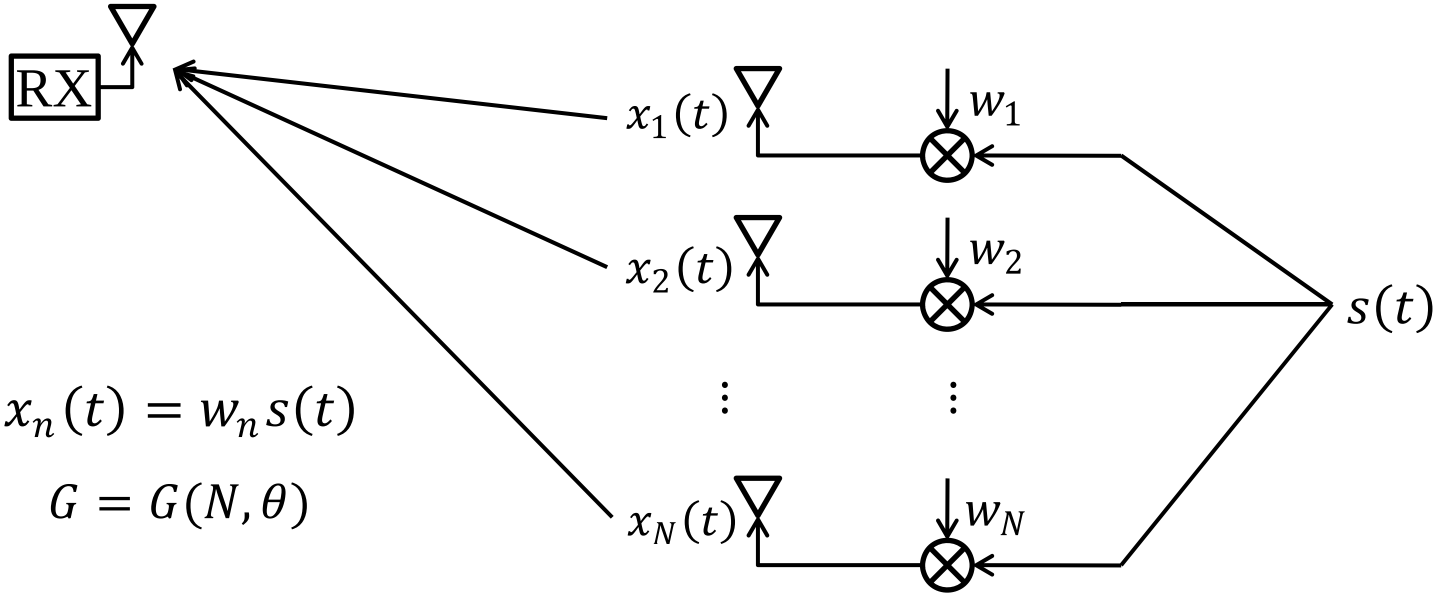


Conclusions

- Characterizations
  - Smartphones rotate relatively slow
  - The channel of directional antennas is reciprocal and predictable in short intervals
- MiDAS
  - Effectively employs directional antennas on smartphones to increase link gain by ~3dB

Beamforming

- A group of omni-directional antennas
- Multiple RF chains
- Baseband signal is weighted, and multiplexed to different RF chains and antennas

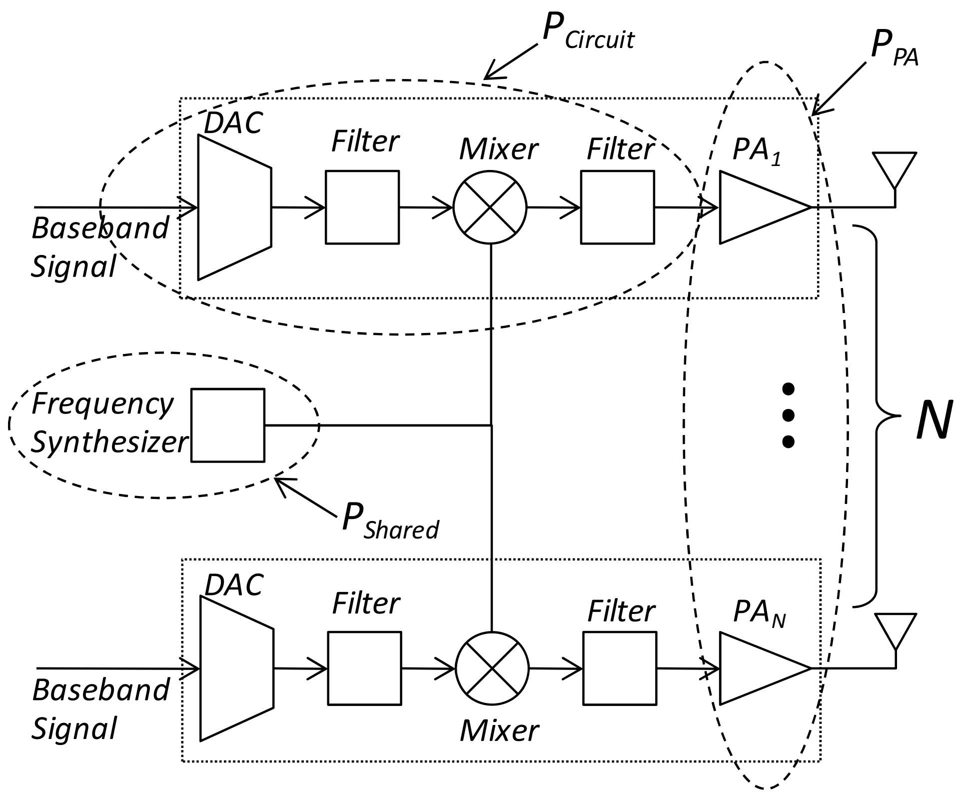


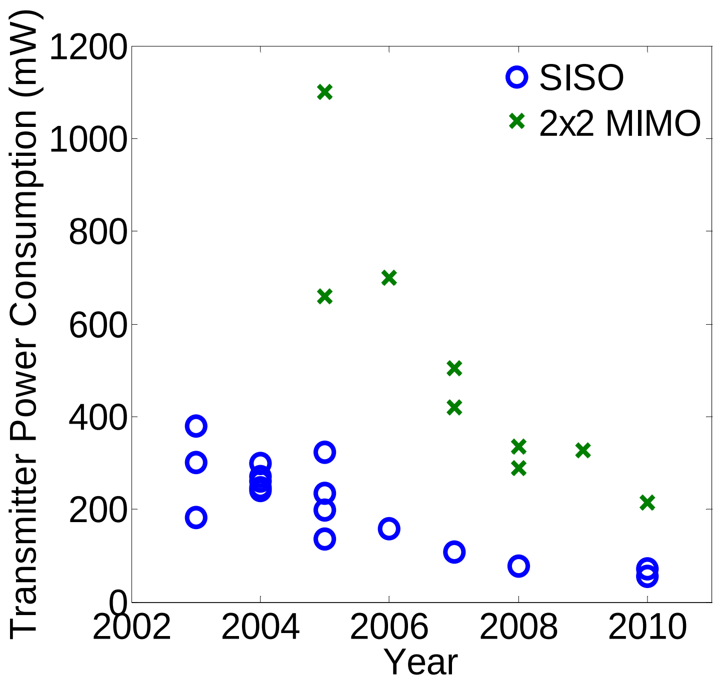
$$x_n(t) = w_n s(t)$$

$$G = G(N, \theta)$$

Power constraint

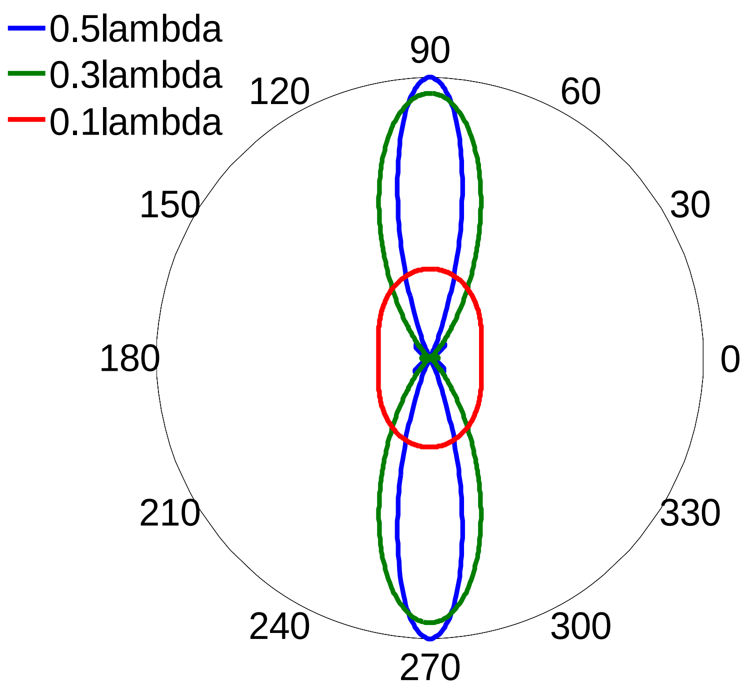
- Circuit power increasingly small

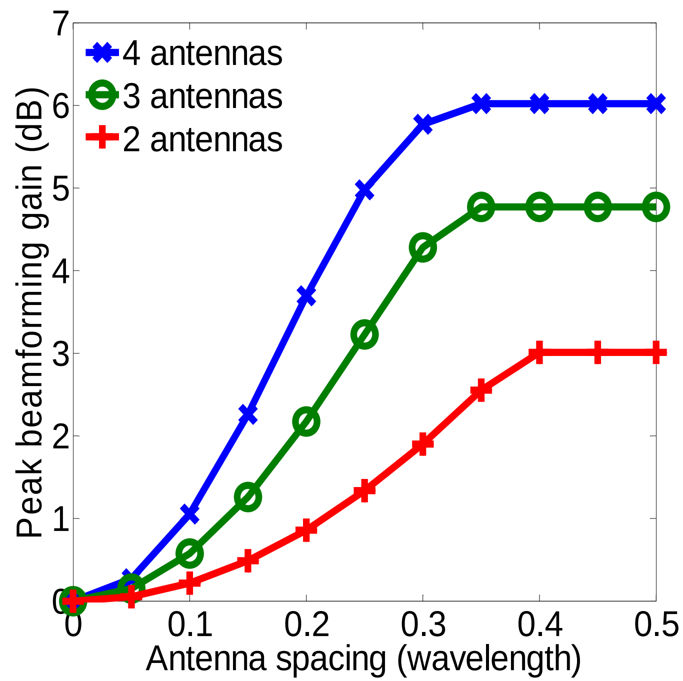




Data collected from JSSC and ISSCC

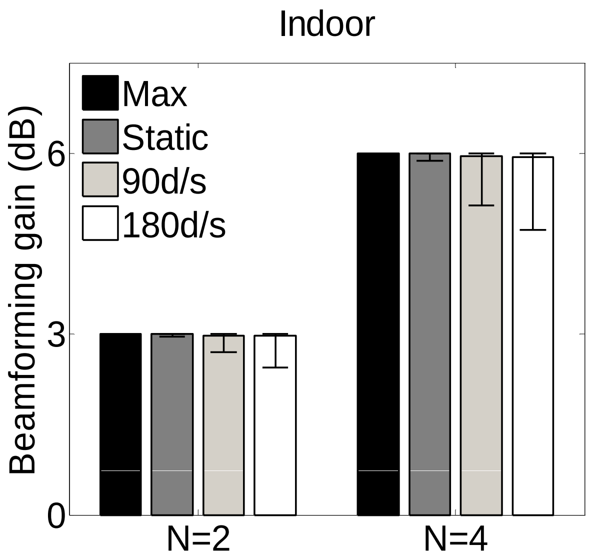
Form factor constraint

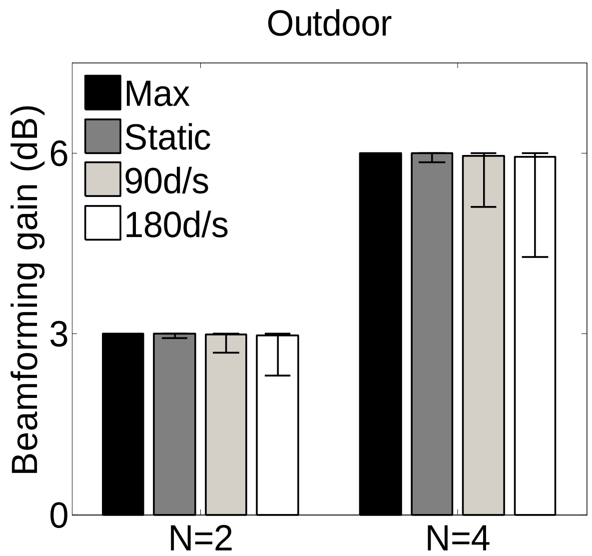




Mobility Constraint

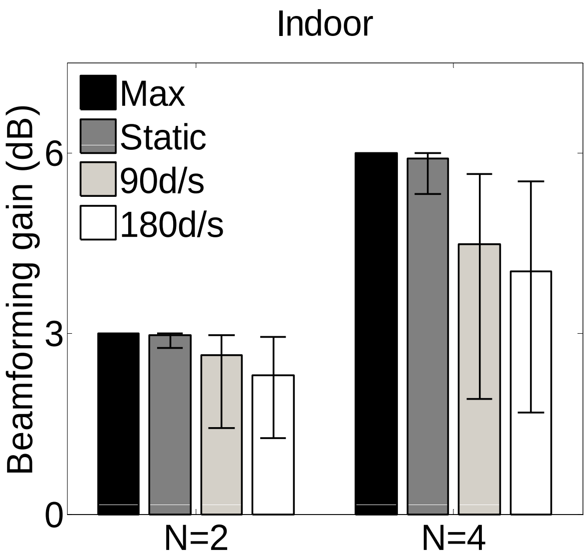
- Beamsteering gain under CSI estimation (per 10ms)

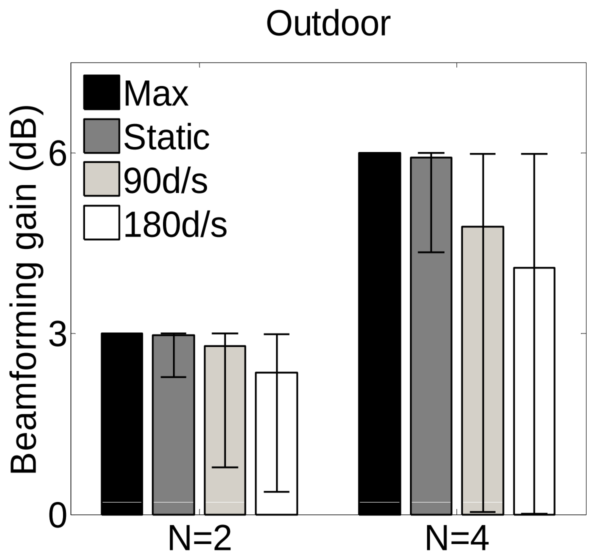




Mobility Constraint (Contd.)

- Beamsteering gain under CSI estimation (per 100ms)



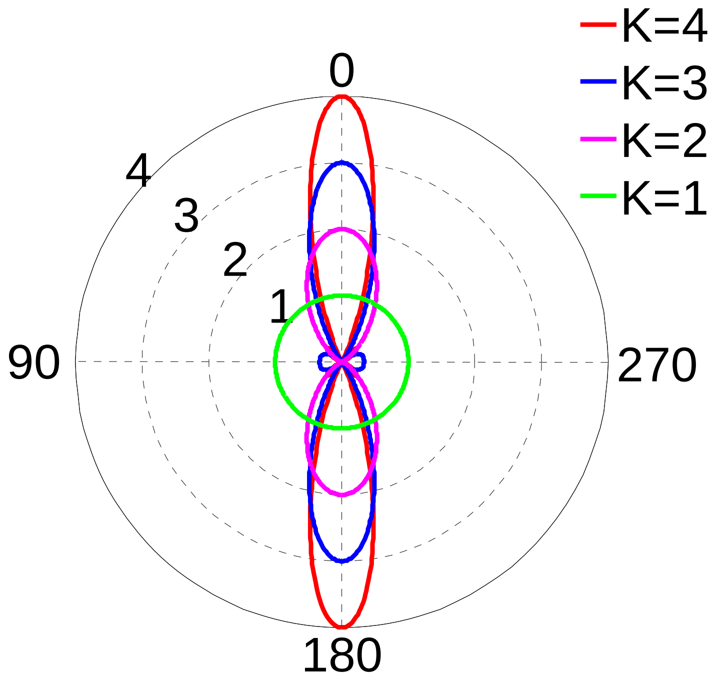


Mobility Constraint (Contd.)

- Key conclusions
  - Beamsteering gain is more stable indoor
  - Static client can always accurately track the channel
  - **CSI estimation per 10ms guarantees near-perfect tracking of the channel**
- Typical frame length
  - UMTS/LTE: 10ms
  - 802.11: tens of ms

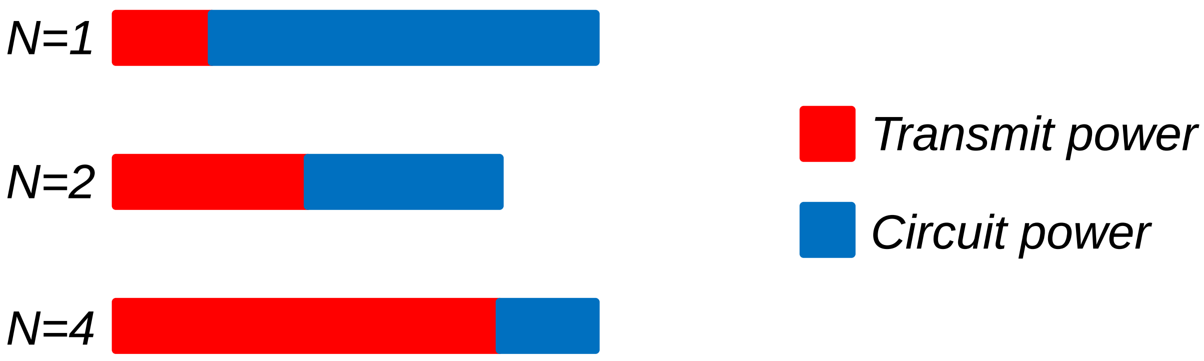
Key tradeoffs

- Circuit and transmit power
- Device efficiency and network capacity



Tradeoff by Beamsteering

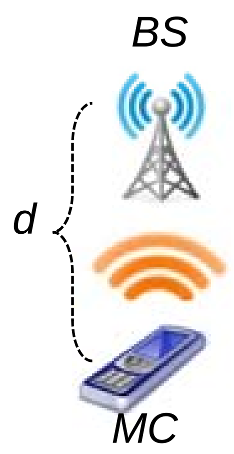
- $P = P_{PA}\downarrow + NP_{Circuit}\uparrow + P_{Shared}$

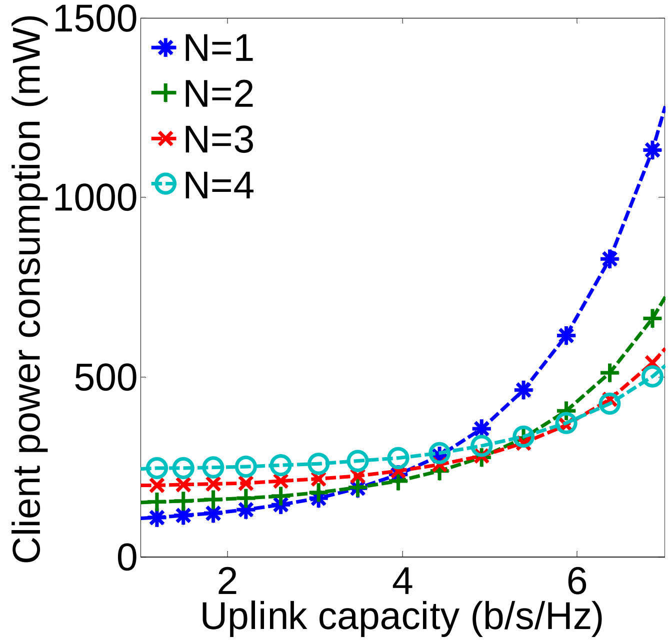


- Single-link scenario
- Two-link scenario

Single-link Scenario

- An uplink channel between a MC and a BS



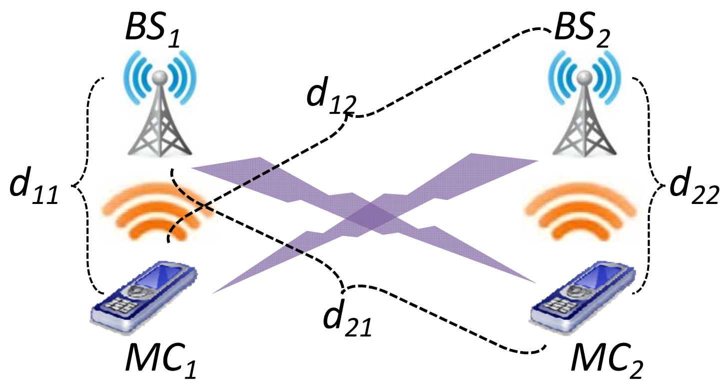


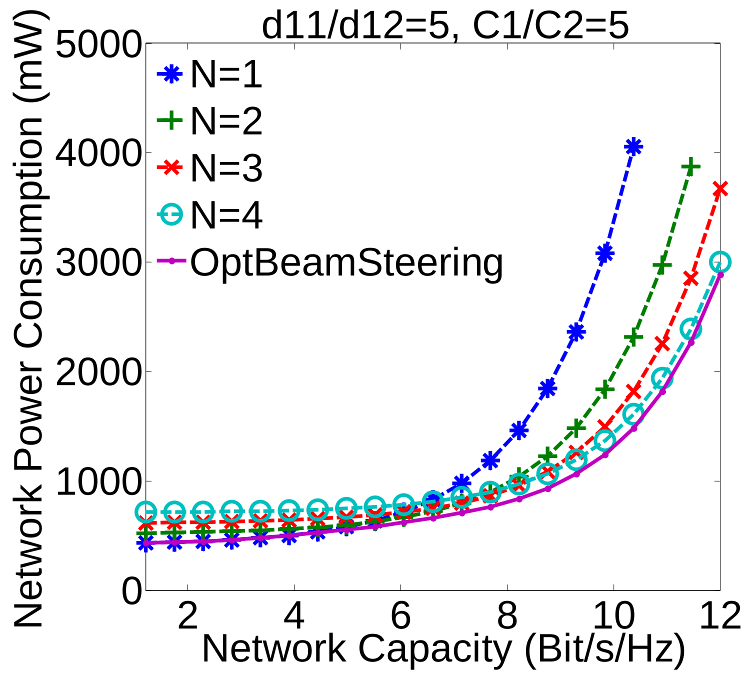
Single-link Scenario (Contd.)

- Key conclusions
  - Beamsteering can be more power efficient than omni-directional transmission
  - The larger the uplink capacity, the larger the optimal beamsteering size
  - Optimal beamsteering size $N_{opt} = \sqrt{(1+\alpha)P_O/P_{Circuit}}$

Two-link Scenario

- Two uplink channels between two MCs and two BSs





Two-link Scenario (Contd.)

- Key conclusions
  - Capacity is determined by inter-link interference
  - Beamsteering can be more power efficient than omni-directional transmission
  - With the same power, beamsteering achieves higher network capacity
  - Beamsteering can achieve network unattainable by omni-directional transmission
  - The optimal beamsteering sizes need to be jointly decided

BeamAdapt

- Optimal use of beamsteering on mobile devices
  - Optimal tradeoff between power efficiency and network capacity
- Theoretical formulation
- System design
- Evaluation

Theoretical Formulation

- Minimum power consumption to achieve certain network capacity

$Minimize$

$$P_{Network} = \sum P_i(P_{TX,i}, N_i)$$

$s.t.$

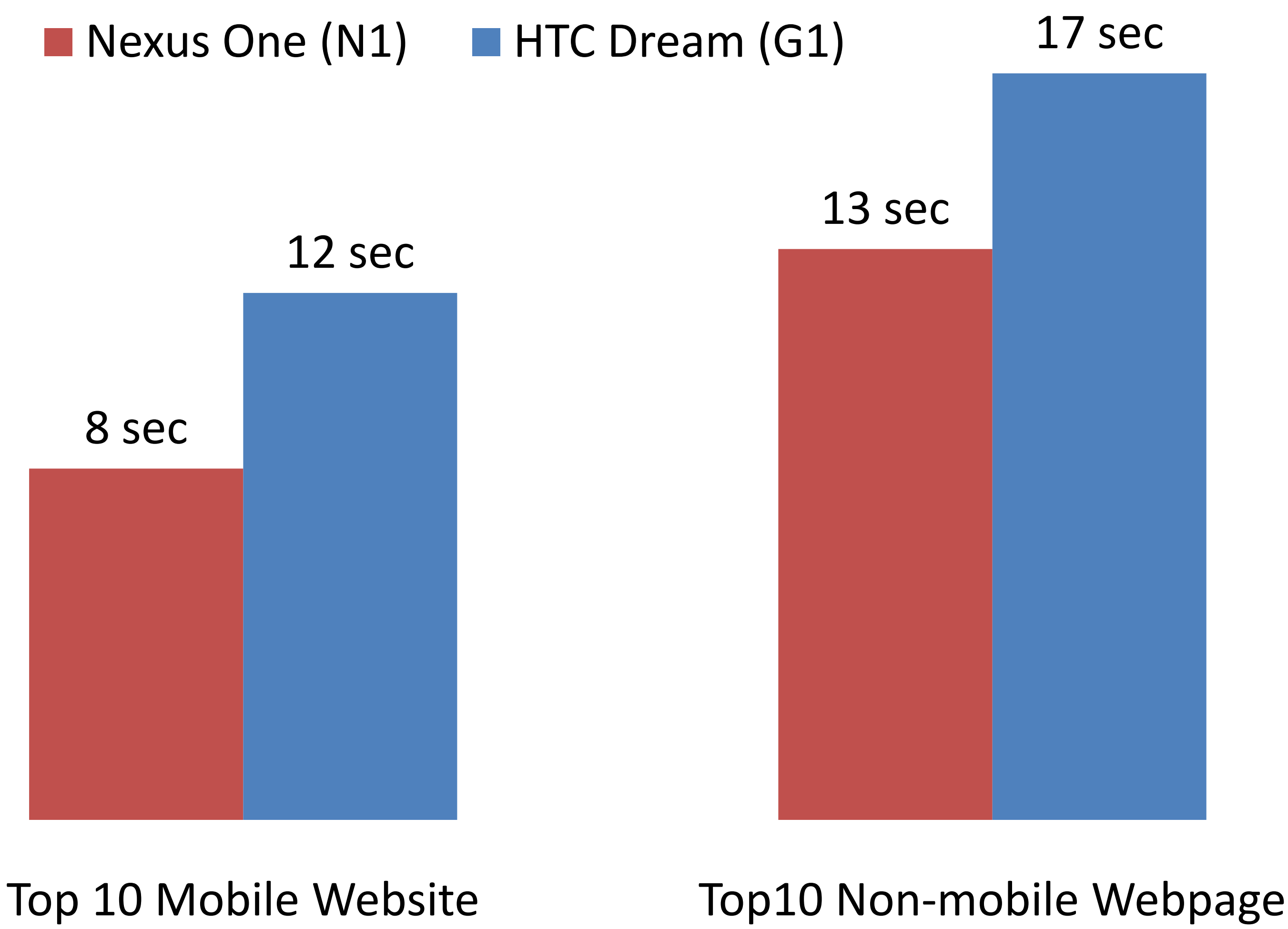
$$SINR_i(\boldsymbol{P}_{TX}, \boldsymbol{N}) \geq \rho_i, 1 \leq N_i \leq N_{i,max}, \forall 1 \leq i \leq M$$

Difficult to Solve

- No closed-form formulation of the beamsteering gain $G = G(N, \theta)$
- Non-convex
- Integer constraint of *N*
  - NP-hard mixed-integer-programming (MIP) problem
- High complexity of brute-force search
  - $O\left(\prod_{i=1}^{M}(N_{i,max})\right)$

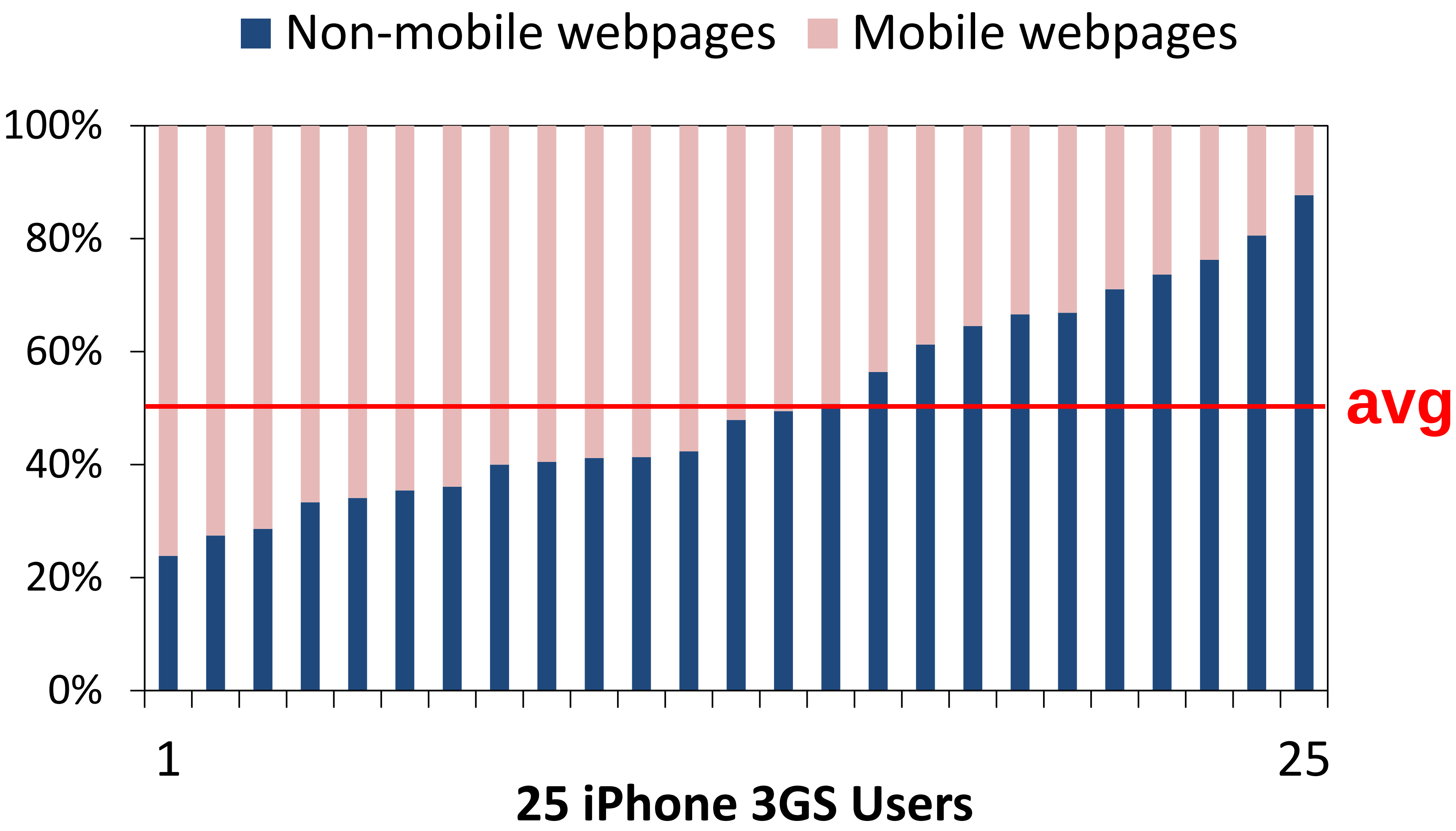
WINDOW TO THE CLOUD

Mobile browsers are slow

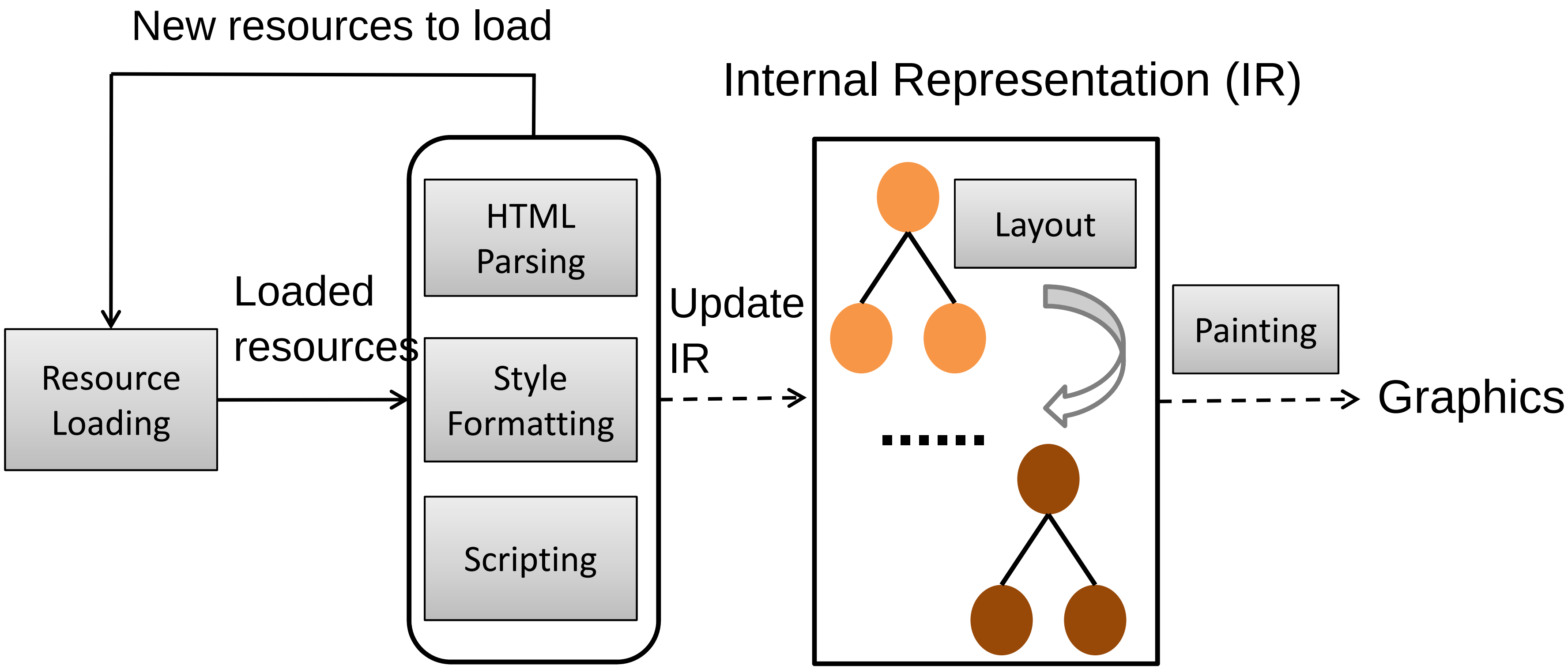


- What does the browser show?
- How does the browser work?
- Where is the bottleneck?

What does the browser show?



How does the browser work?



IR operations: Parsing, Style, Scripting, Layout, Painting

Where is the bottleneck?

- Existing work on PC browsers
  - Layout
  - Style formatting
  - Scripting

1. C. Stockwell, "IE8 Performance," http://blogs.msdn.com/b/ie/archive/2008/08/26/ie8-performance.aspx, 2008.
2. L. A. Meyerovich and R. Bodik, "Fast and parallel webpage layout," in *Proc. Int. Conf. World Wide Web (WWW) Raleigh, North Carolina, USA: ACM, 2010.*
3. K. Zhang, L. Wang, A. Pan, and B. B. Zhu, "Smart caching for web browsers," in *Proc. Int. Conf. World Wide Web (WWW) Raleigh, North Carolina, USA: ACM, 2010.*

Is it true for mobile browsers?

Layout, Style, Scripting

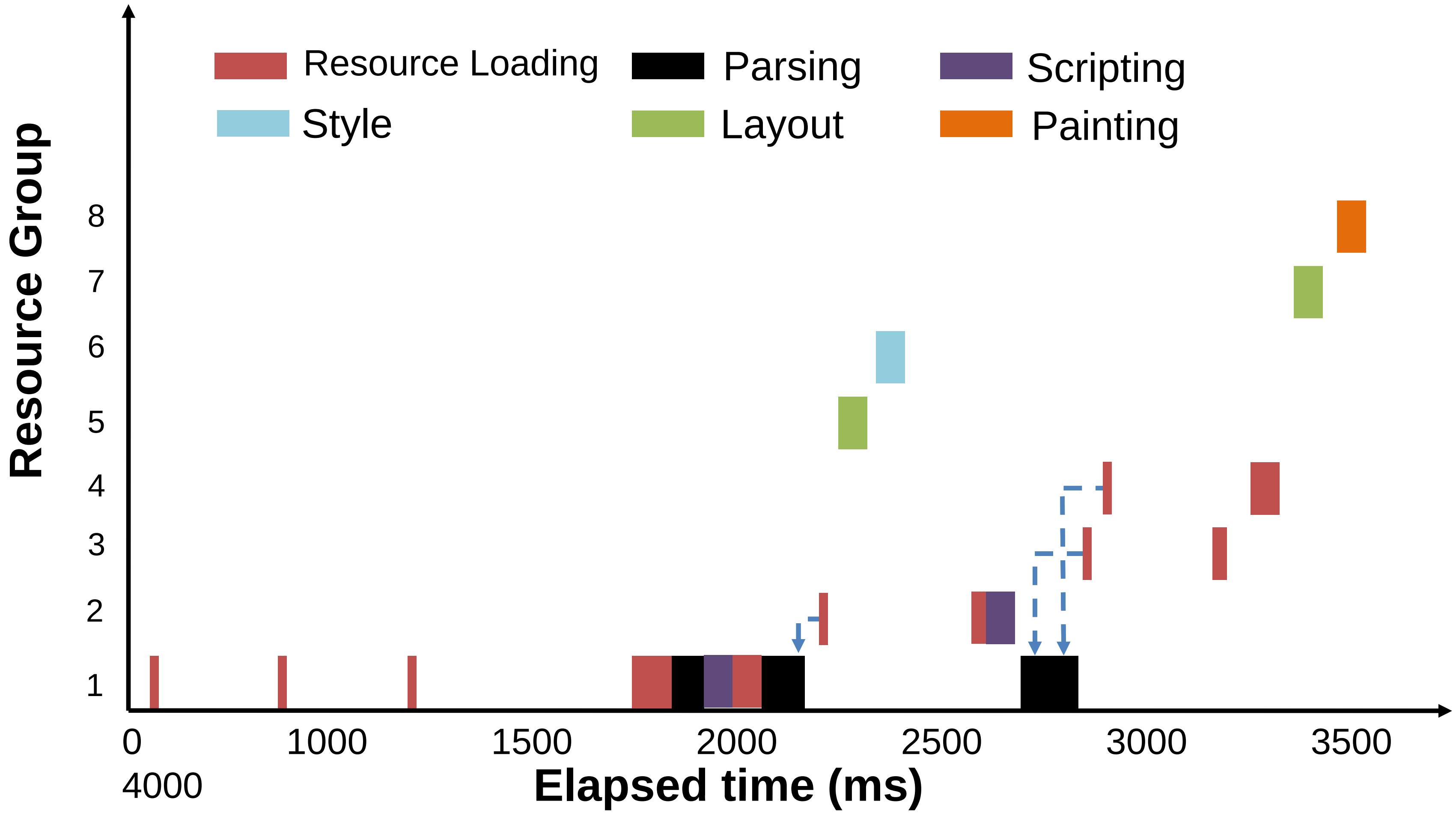
Performance characterization

- Metric: *browser delay*
  - *Starting point*: when the user presses the “GO” button of the browser to open an URL.
  - *End point*: when the browser’s page loading progress bar indicates 100%.

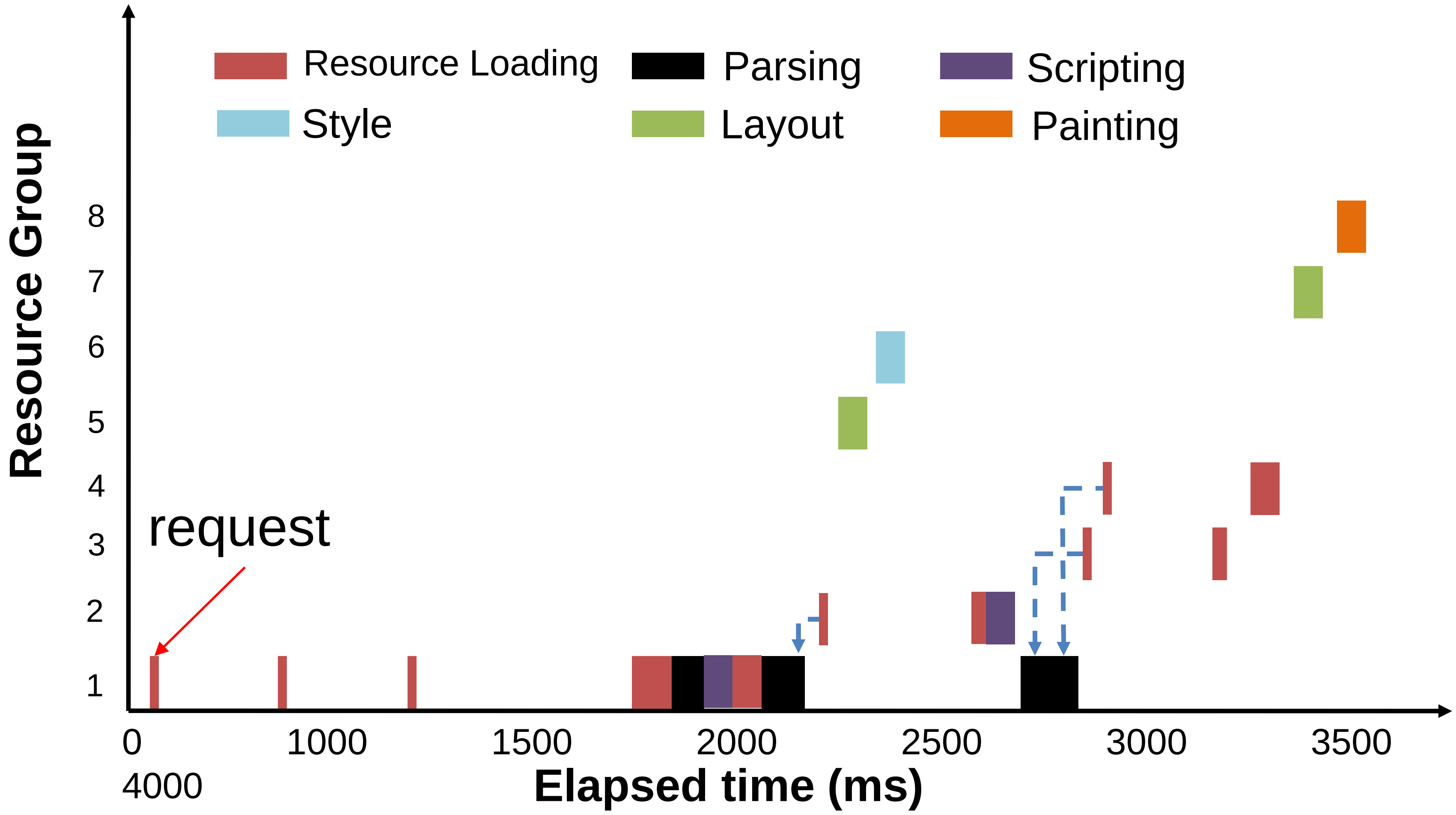
Performance characterization

- Dependency timeline characterization
- What-if analysis

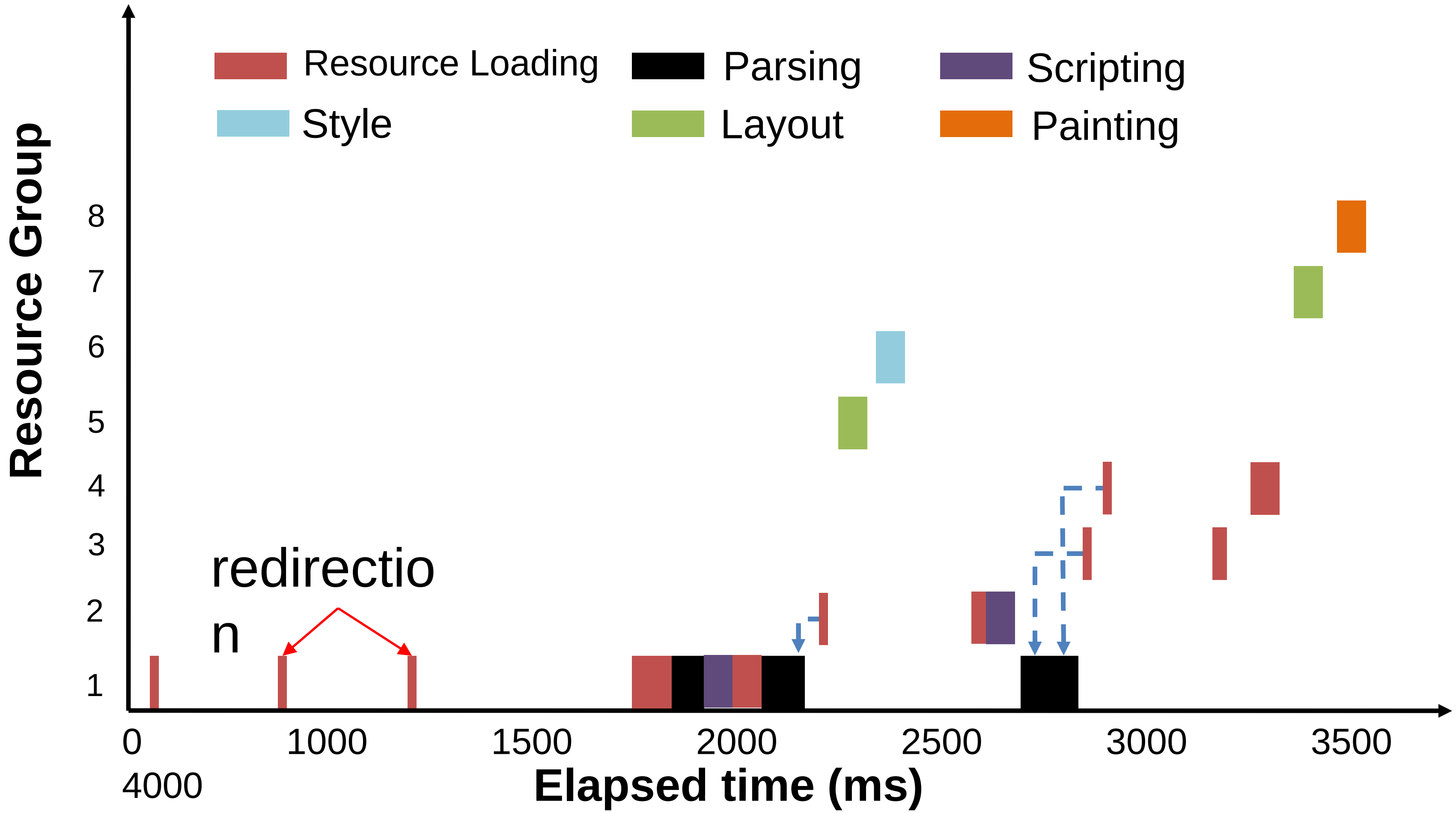
Dependency timeline characterization



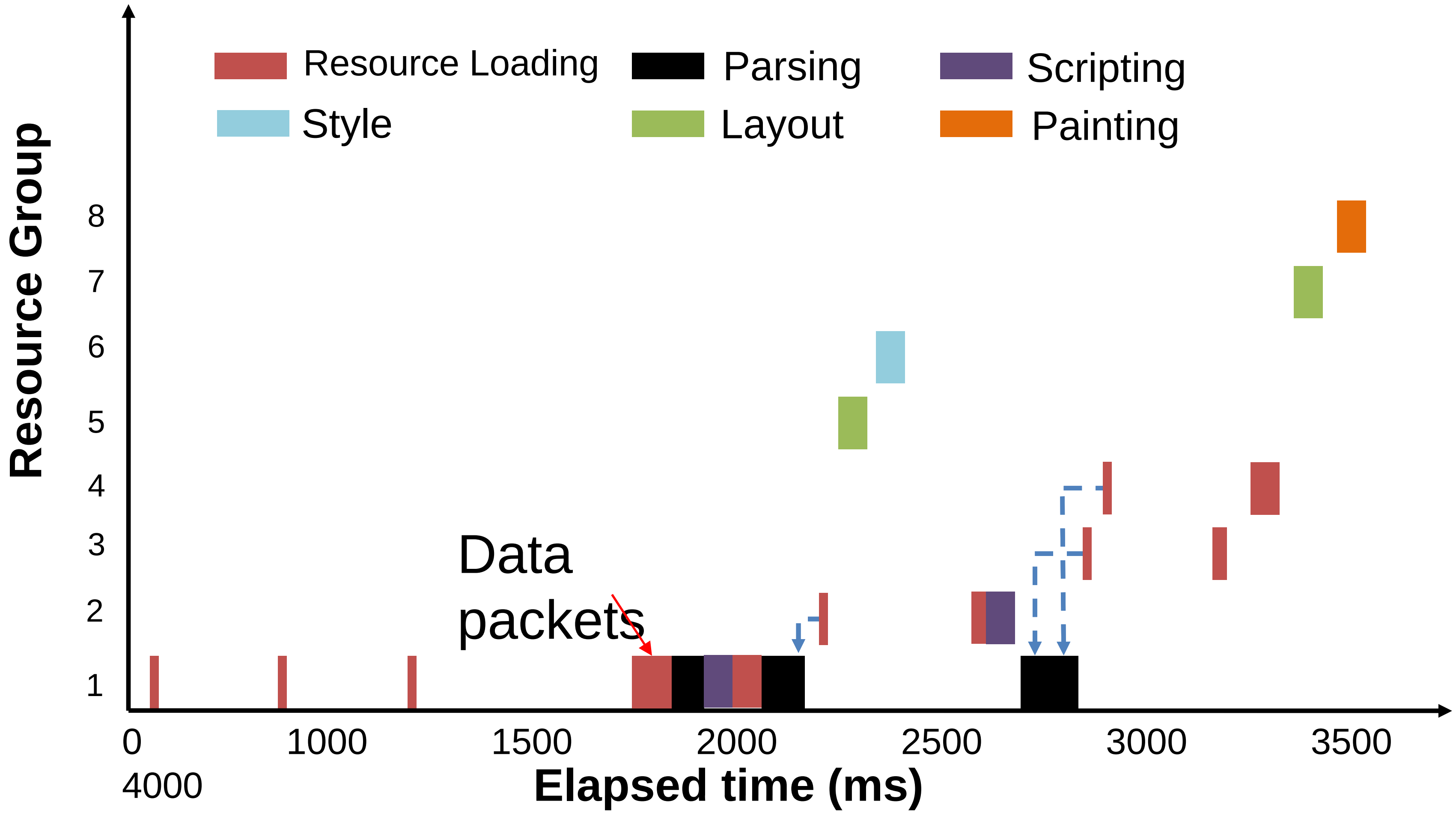
Dependency timeline characterization



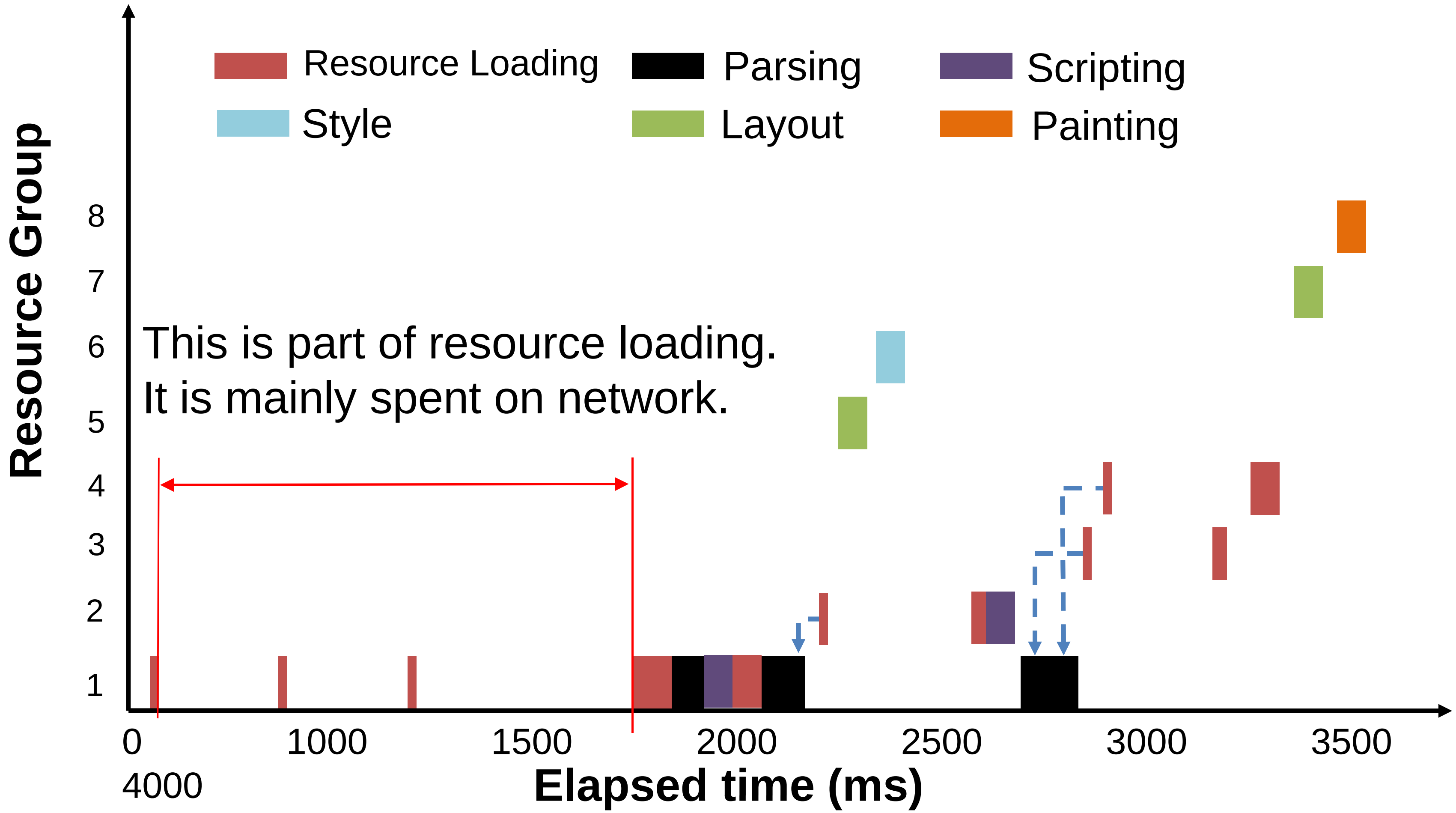
Dependency timeline characterization



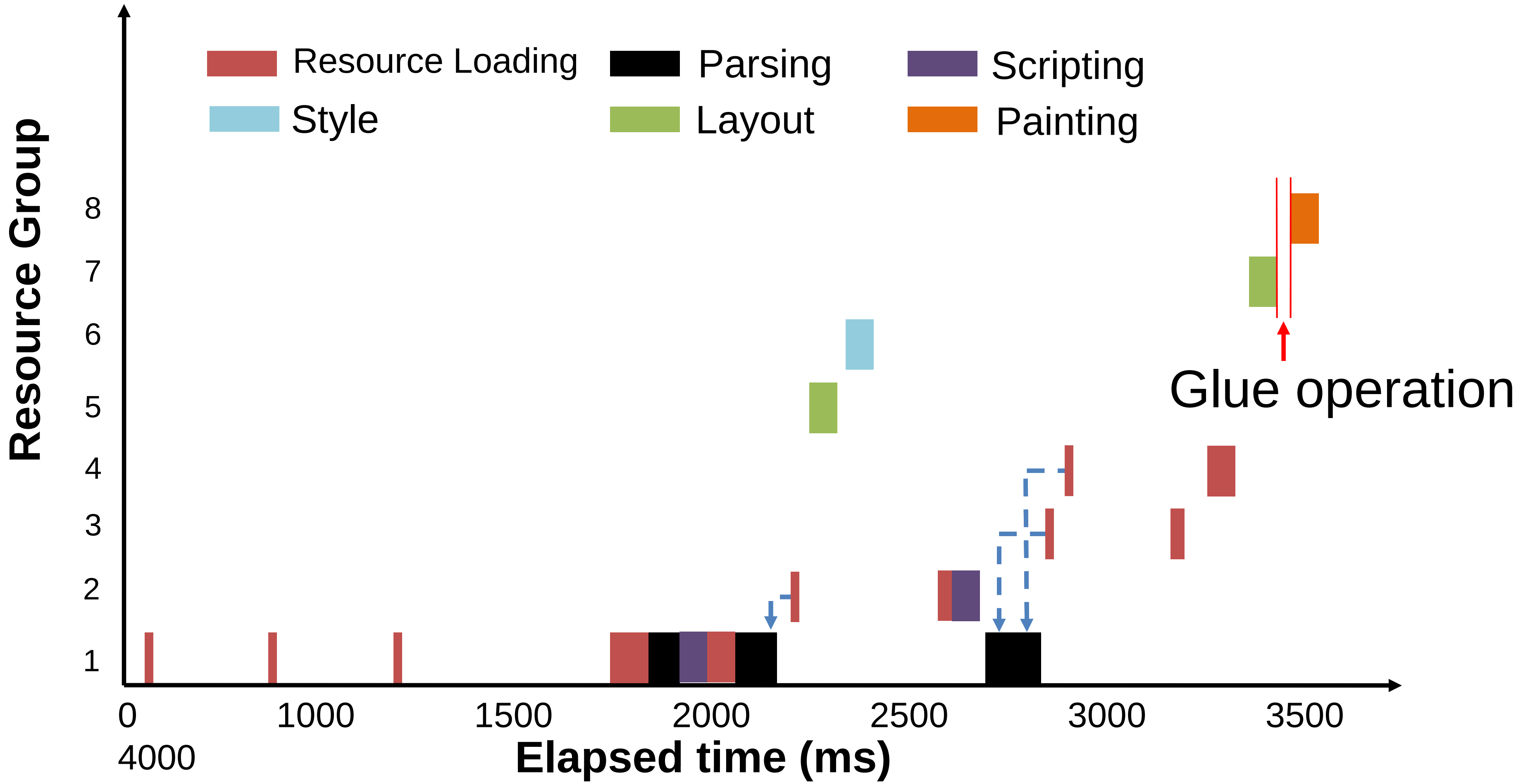
Dependency timeline characterization



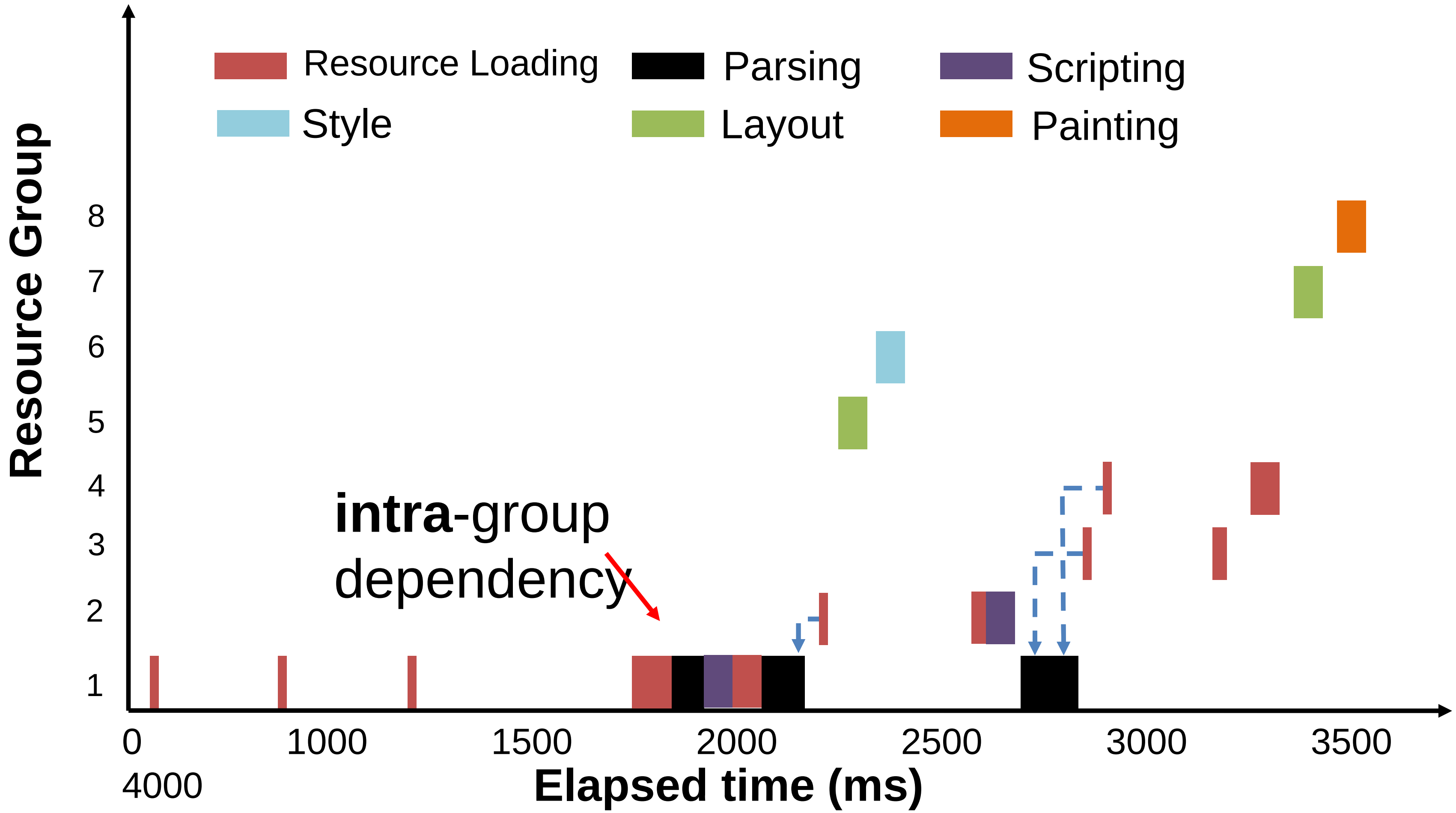
Dependency timeline characterization



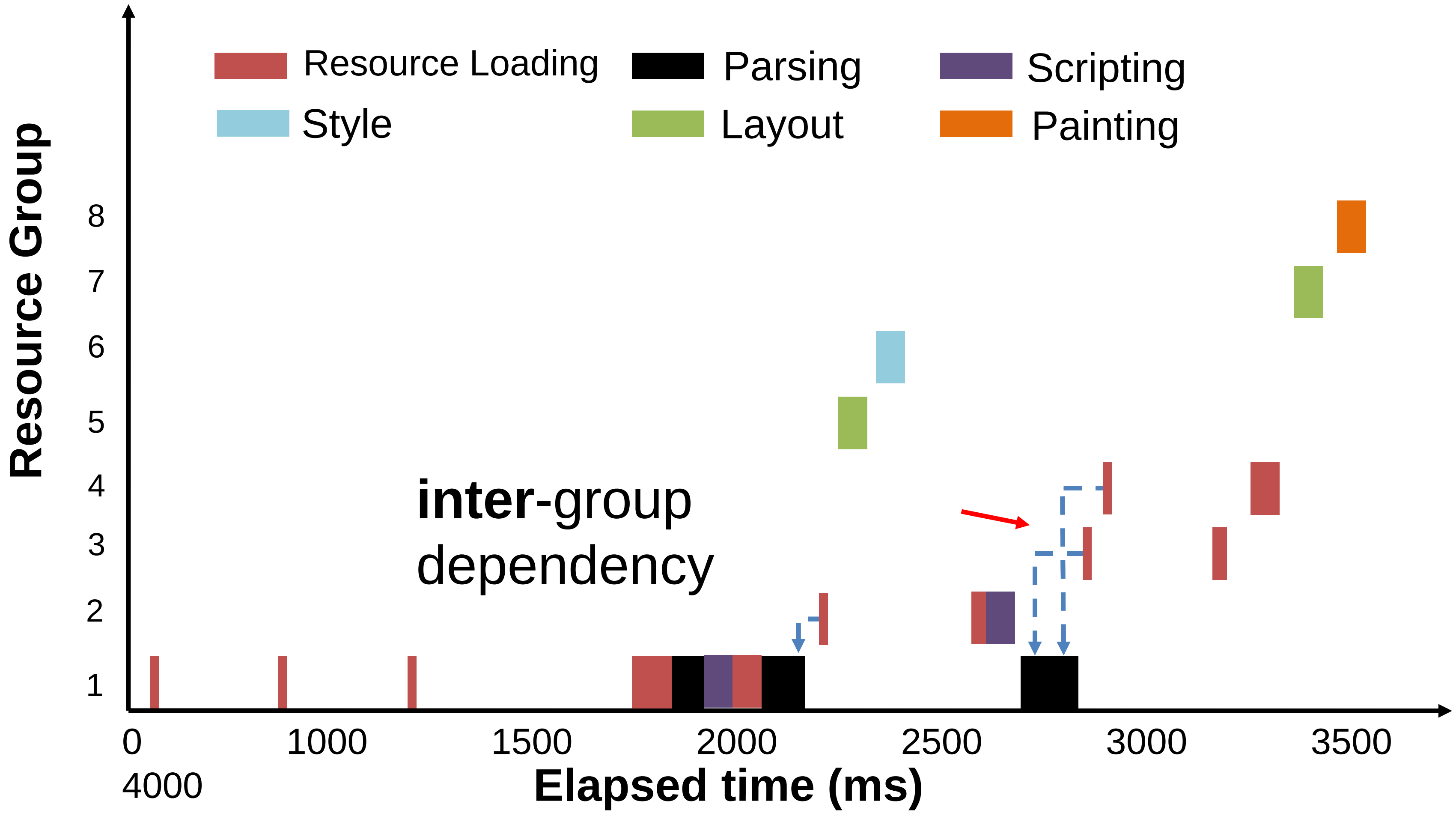
Dependency timeline characterization



Dependency timeline characterization

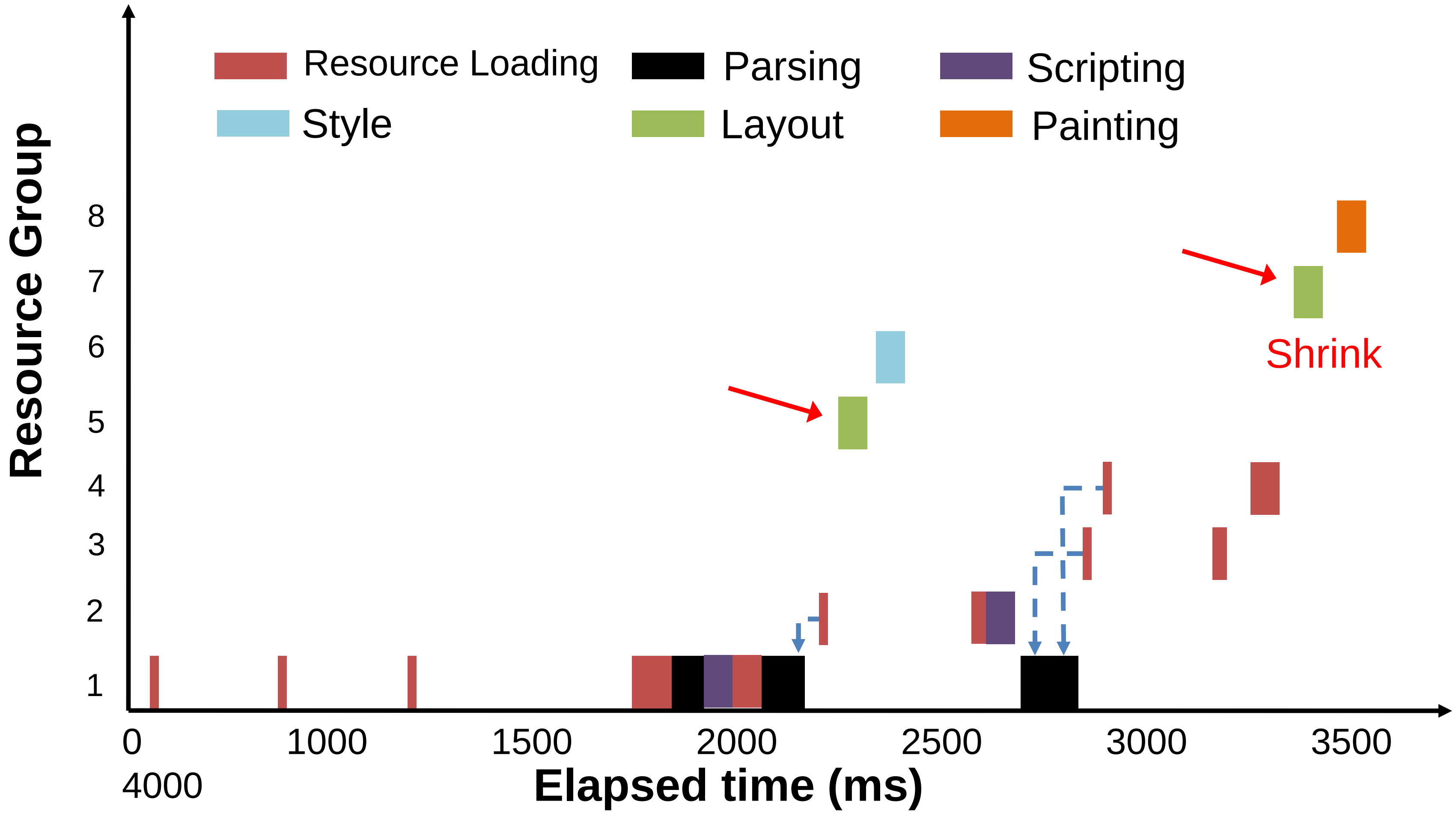


Dependency timeline characterization

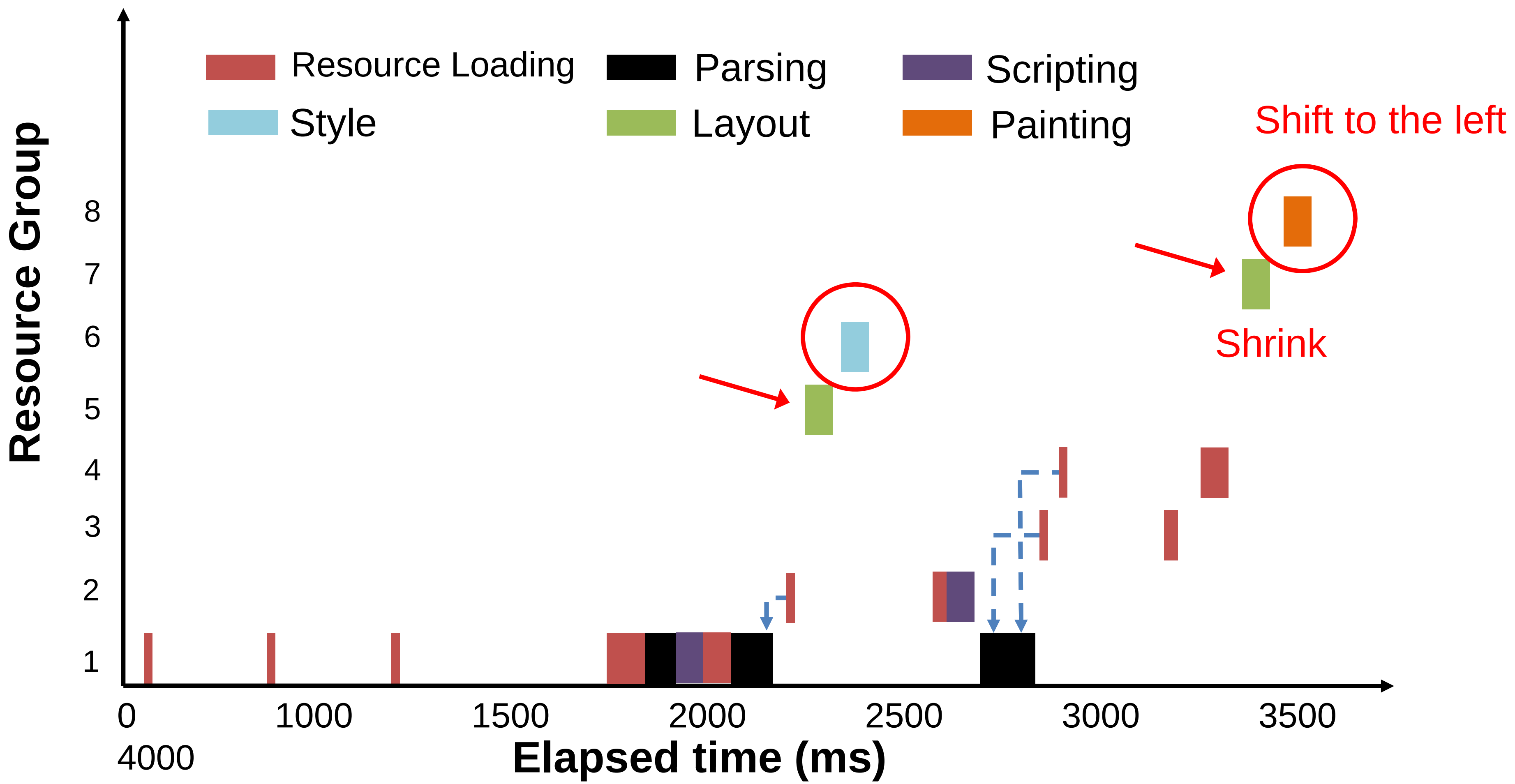


What overall performance gain will be achieved *if* a browser operation is accelerated?

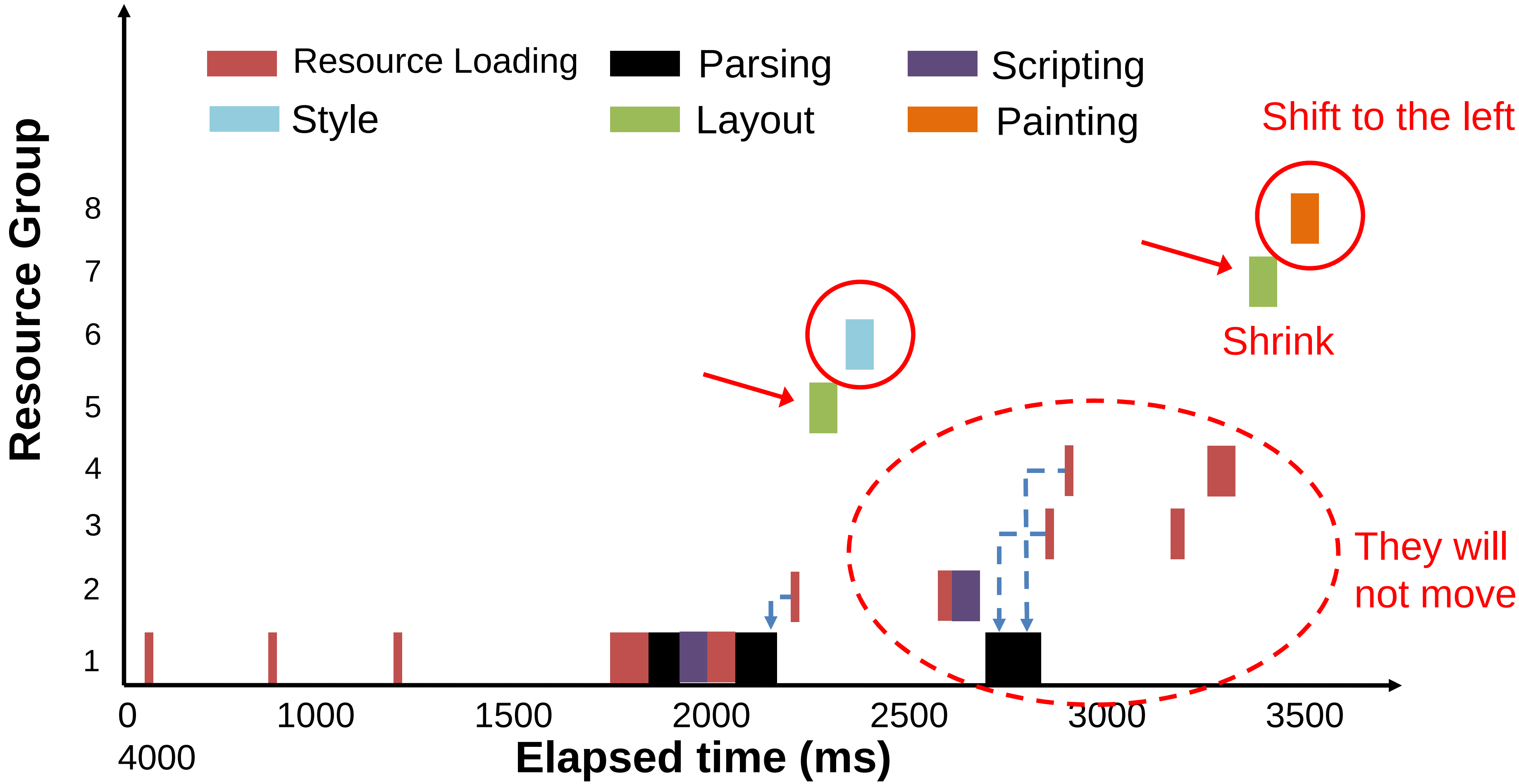
What-if analysis



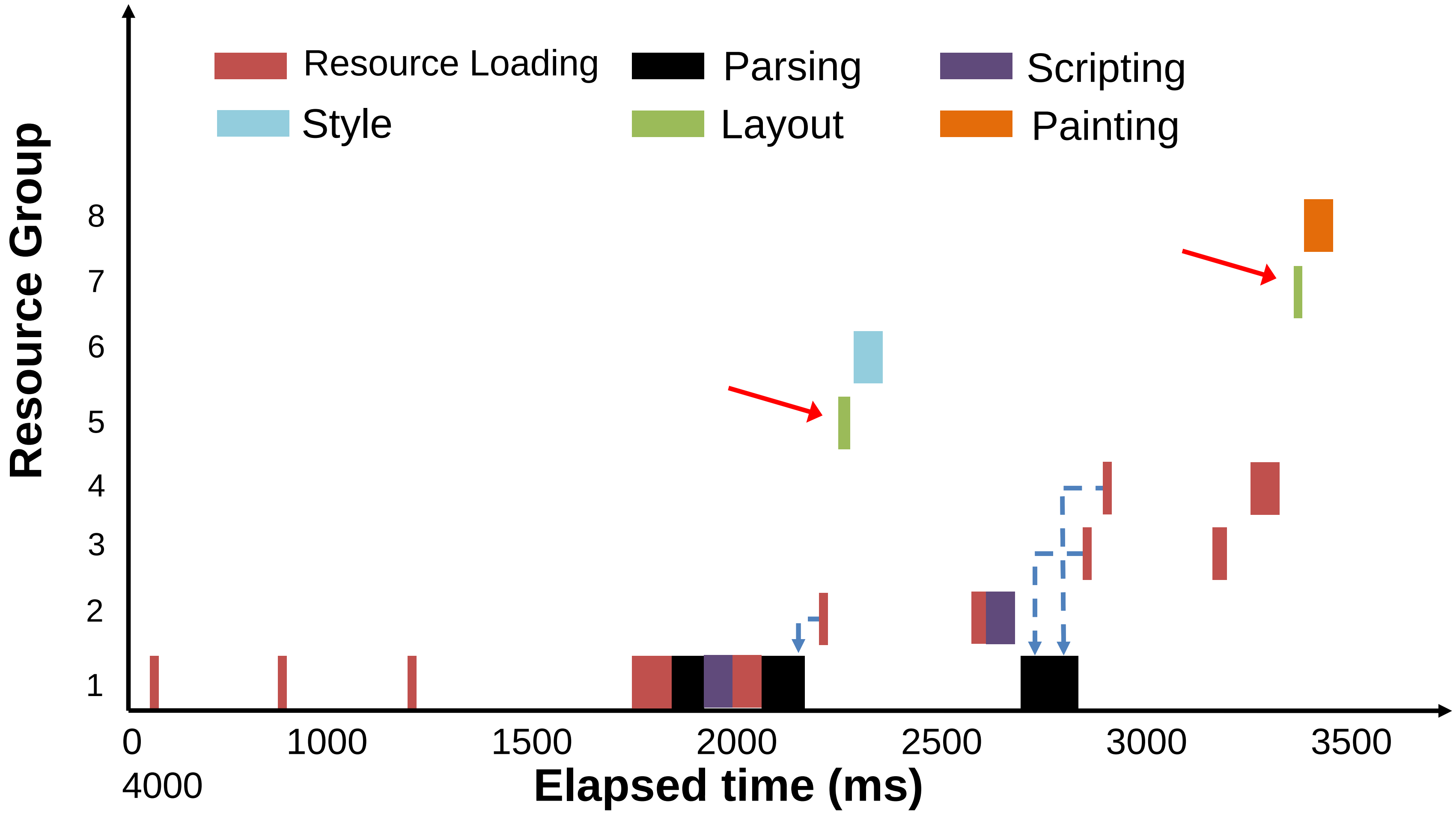
What-if analysis



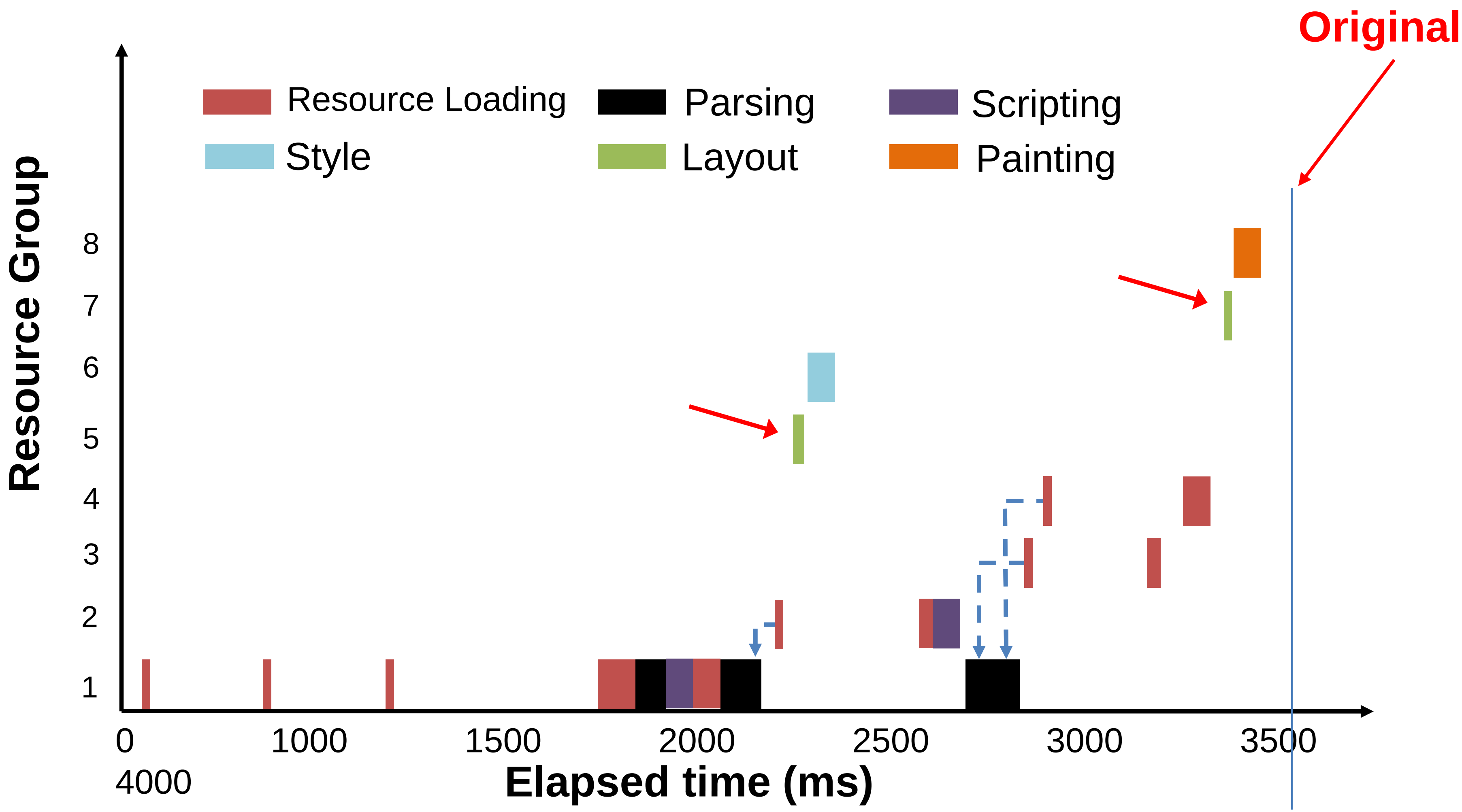
What-if analysis



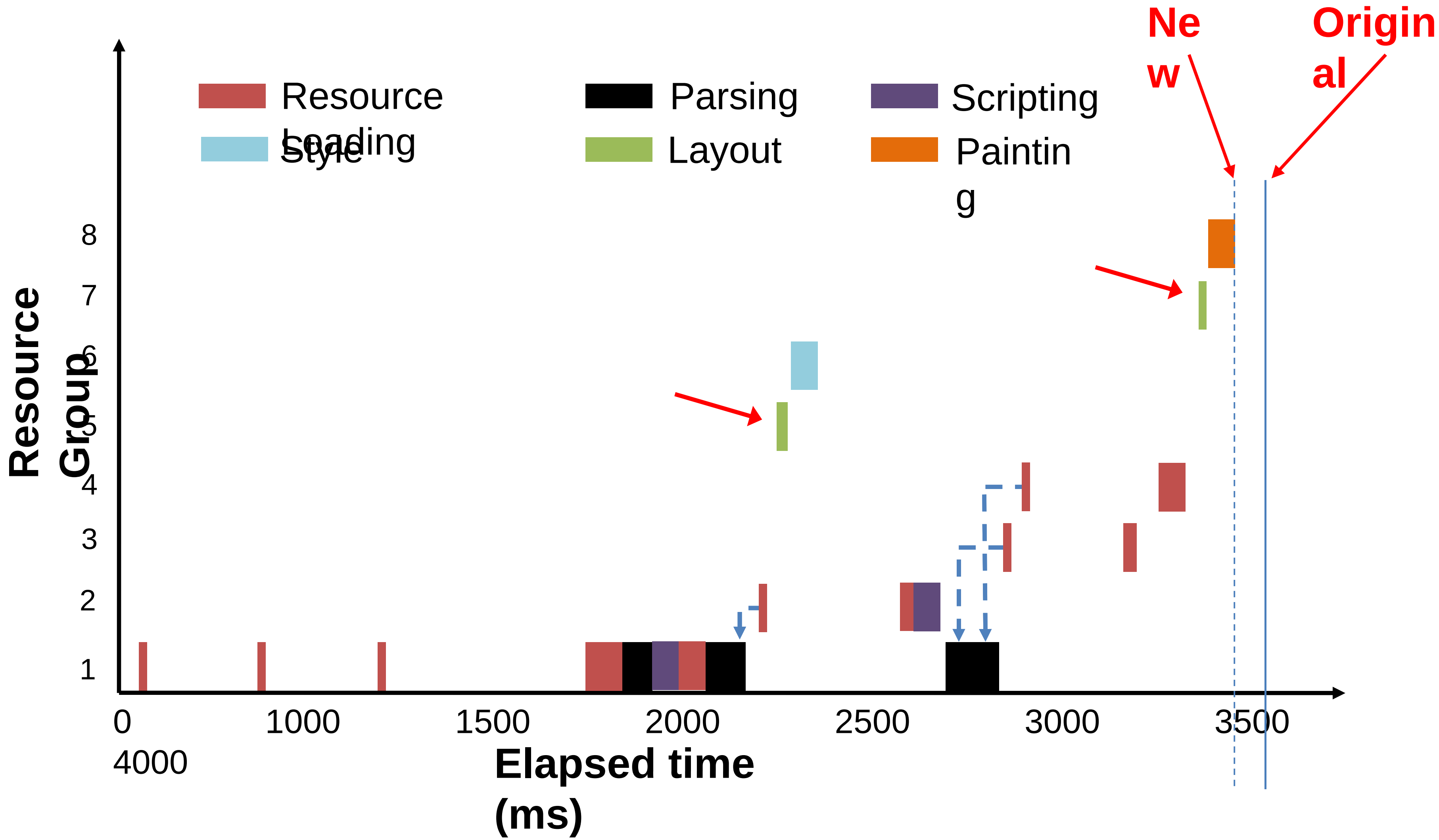
What-if analysis



What-if analysis



What-if analysis



Experimental setup

- Platform:
  - HTC Dream (G1): 528MHz
  - Nexus One (N1): 1GHz

- Operating System
  - Android 2.1 (Eclair)
- Benchmark Webpages:
  - Top 10 mobile websites
  - Top 10 visited non-mobile webpages from LiveLab

1.Nielsen.com, "Top mobiel phones, sites and brands for 2009," http://blog.nielsen.com/nielsenwire/online_mobile/top-mobile-phones-sites-and-brands-for-2009/, 2009.
2.C. Shepard, A. Rahmati, C. Tossell, L. Zhong, and P. Kortum, "Live- Lab: Measuring Wireless Networks and Smartphone Users in the Field," in *Proc. Workshop on Hot Topics in Measurement & Model-ing of Computer Systems, June 2010.*

Experimental setup

- We used three network conditions:
  - Emulated enterprise Ethernet (no traffic control)
  - Typical 3G network (T-mobile)
  - Emulated adverse network
    - First-hop RTT: 400ms
    - Bandwidth (downlink/uplink): 500Kbps/100Kbps

Logging information

- Time stamp for browser operations
  - Overhead: <1%
- Tcpdump
  - Overhead: <2% (CPU); <0.4% (MEM)

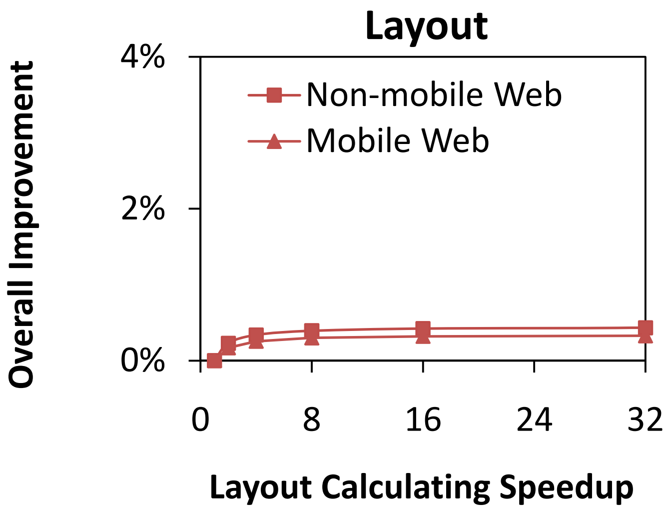
Results

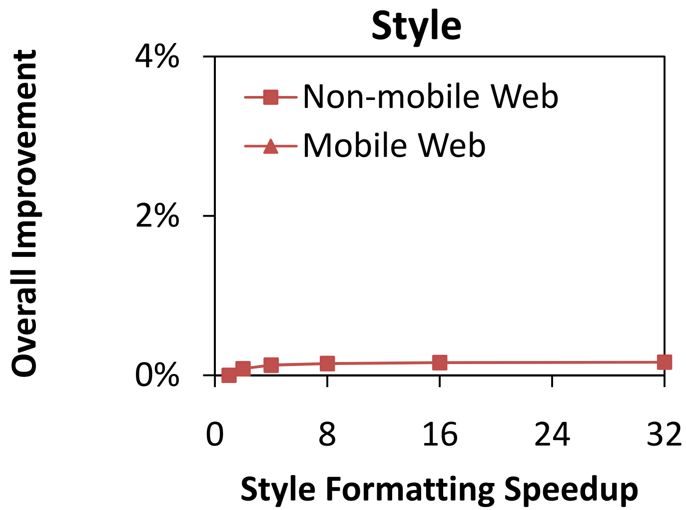
two take-away messages

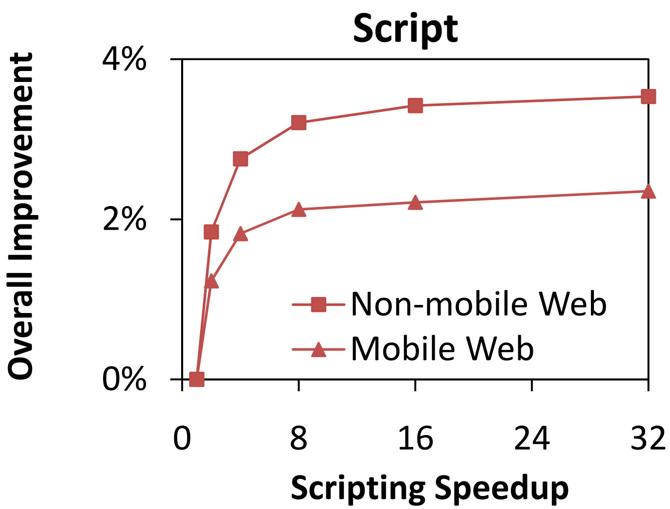
IR operations do not matter much!

Parsing, Style, Scripting, Layout, Painting

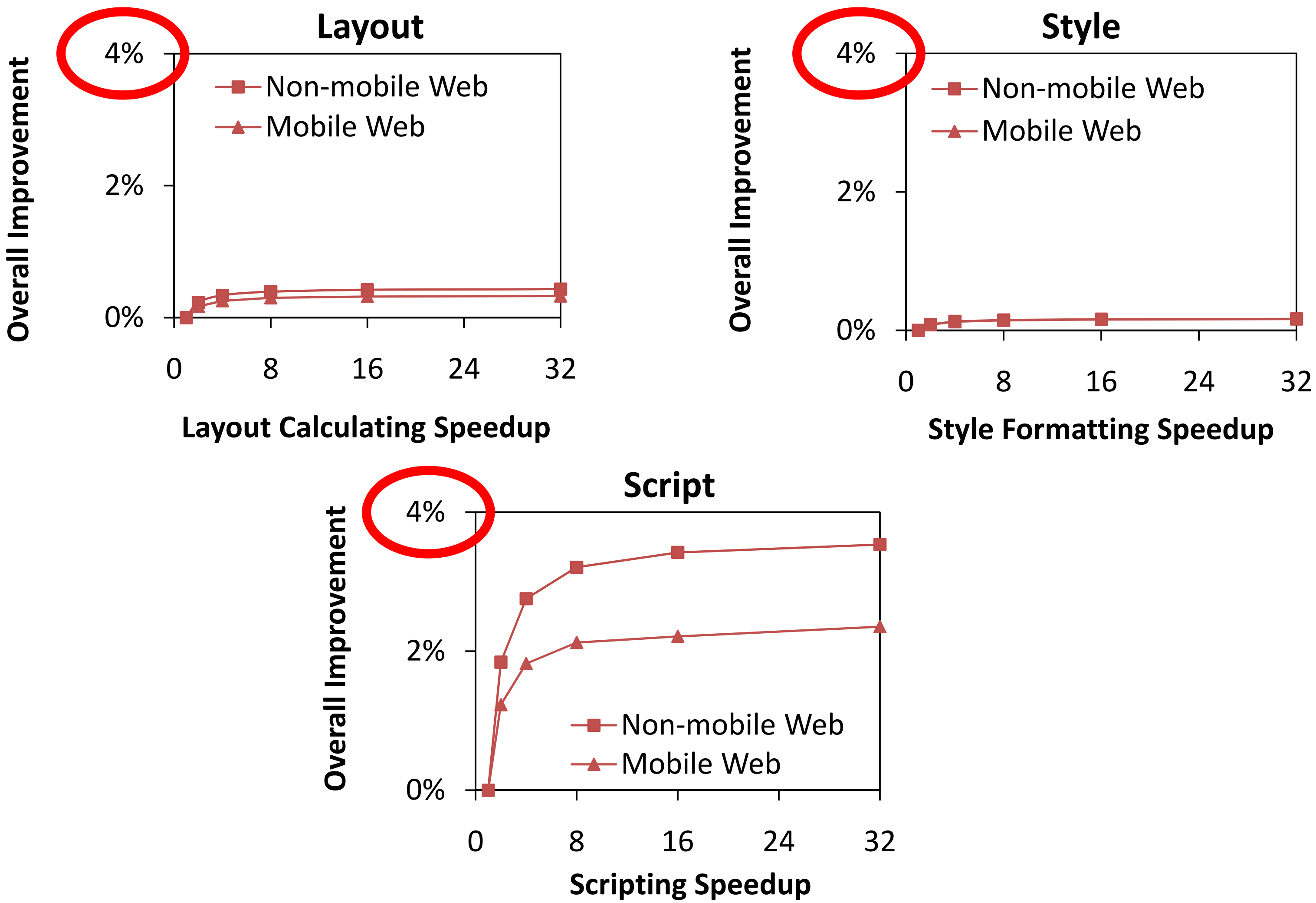
IR operations do not matter much



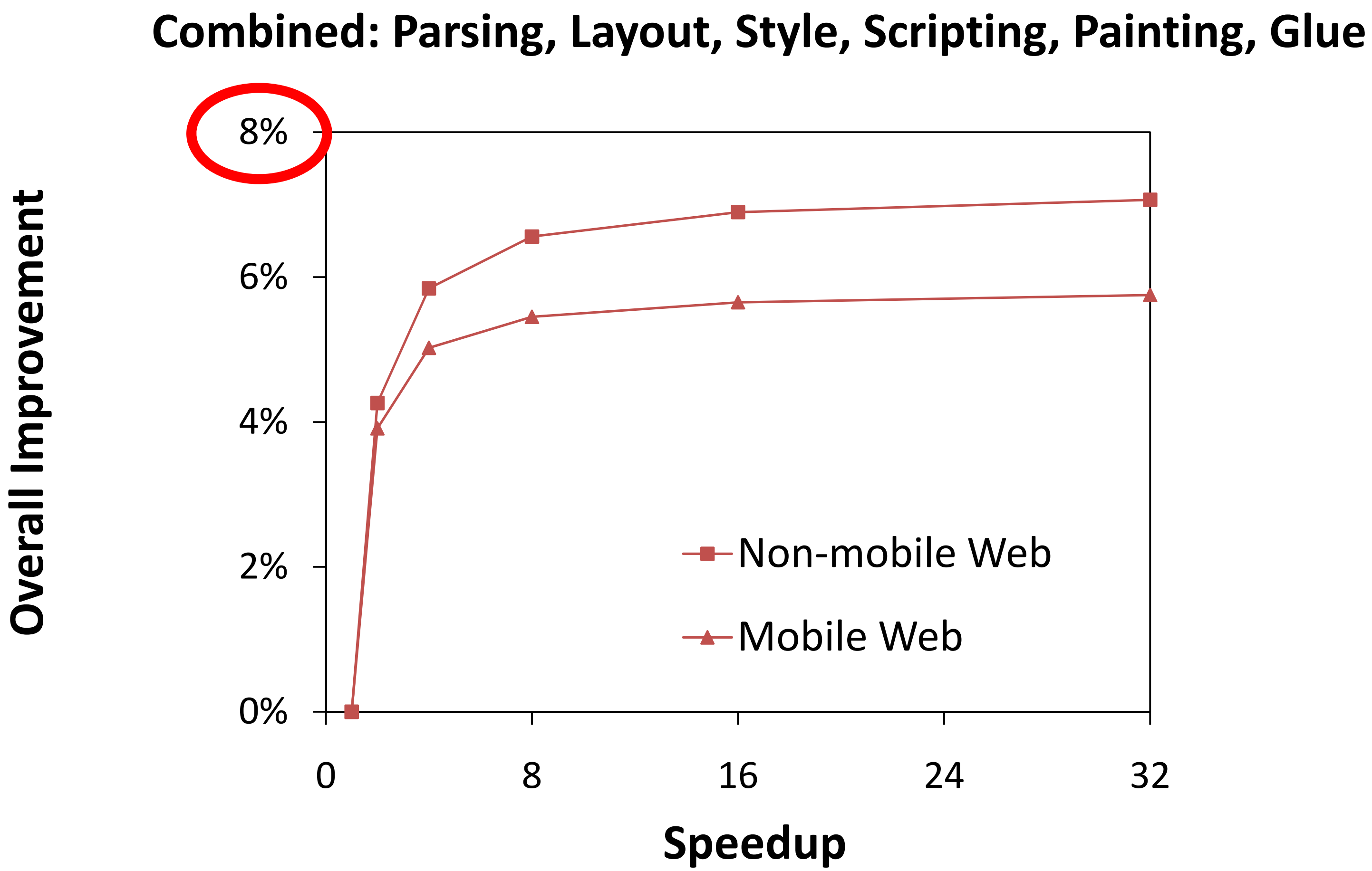




IR operations do not matter much

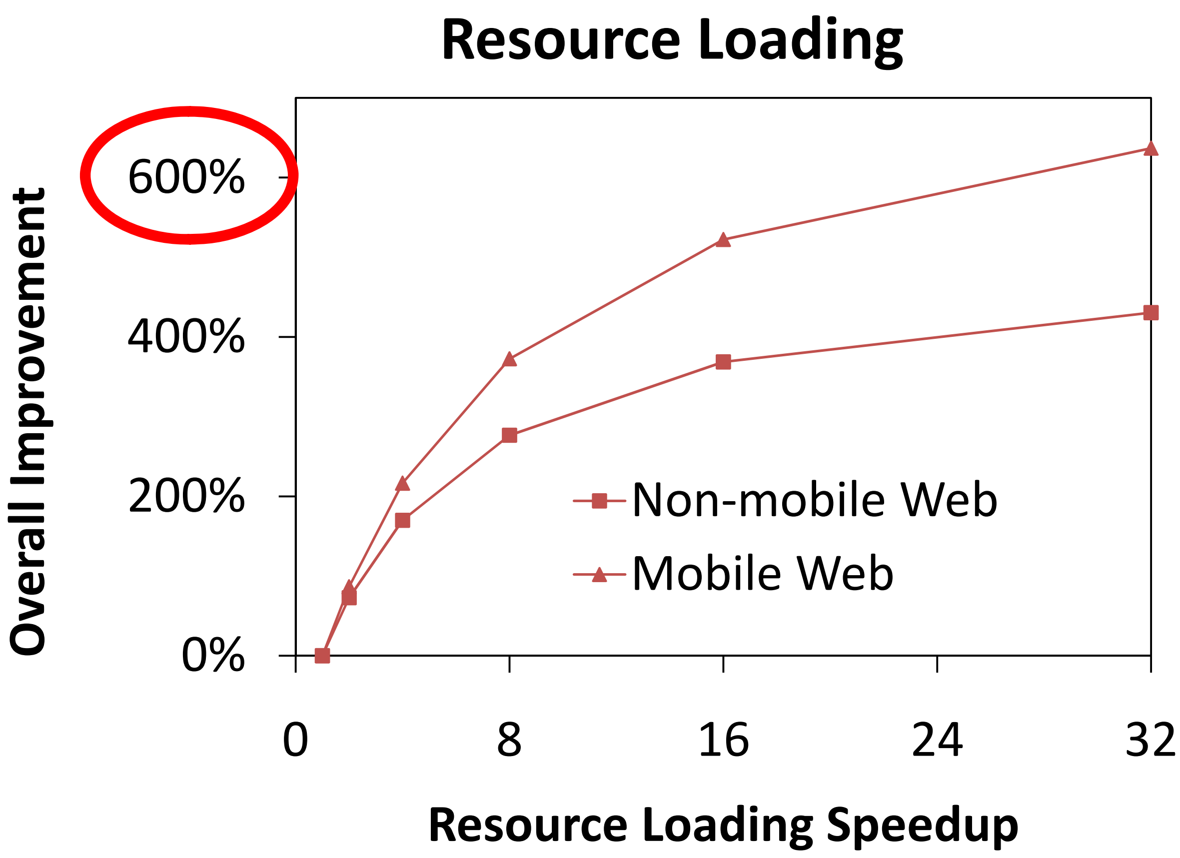


IR operations do not matter much

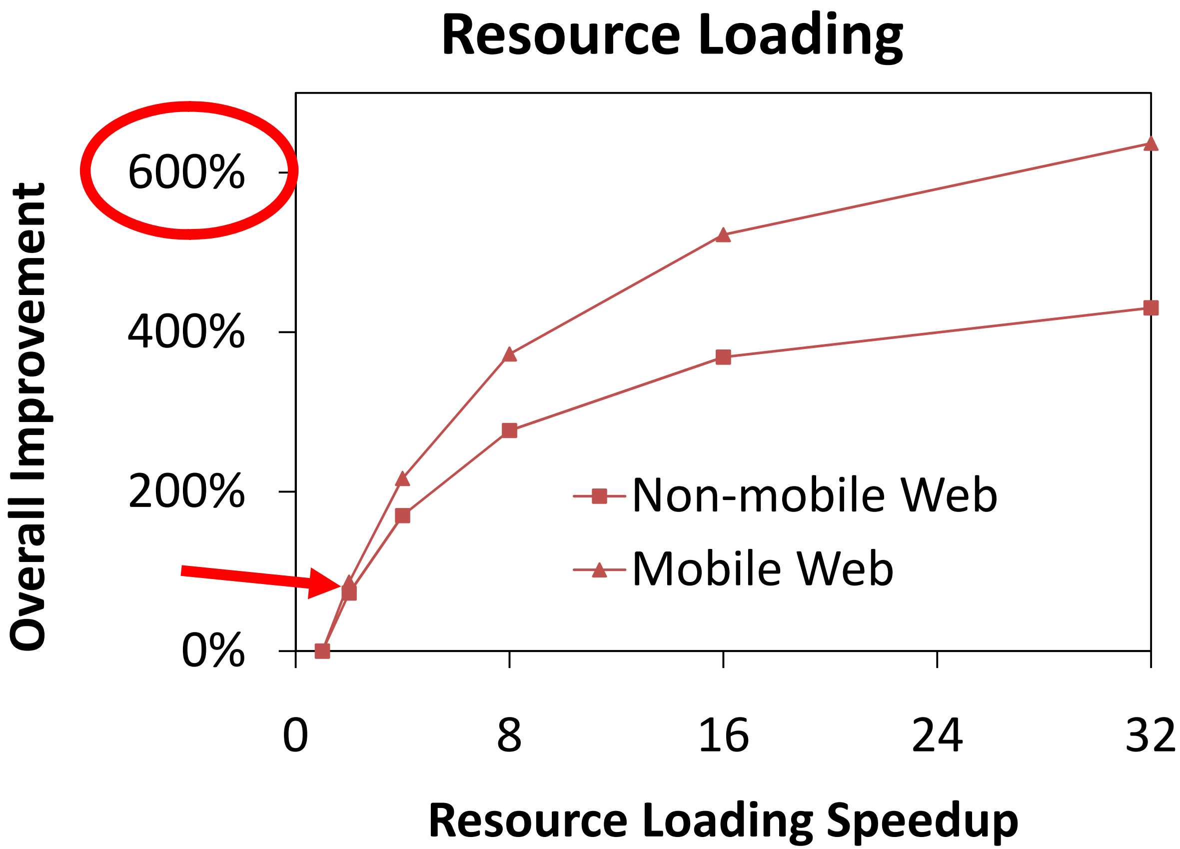


Resource loading is the bottleneck!

Resource loading is the bottleneck



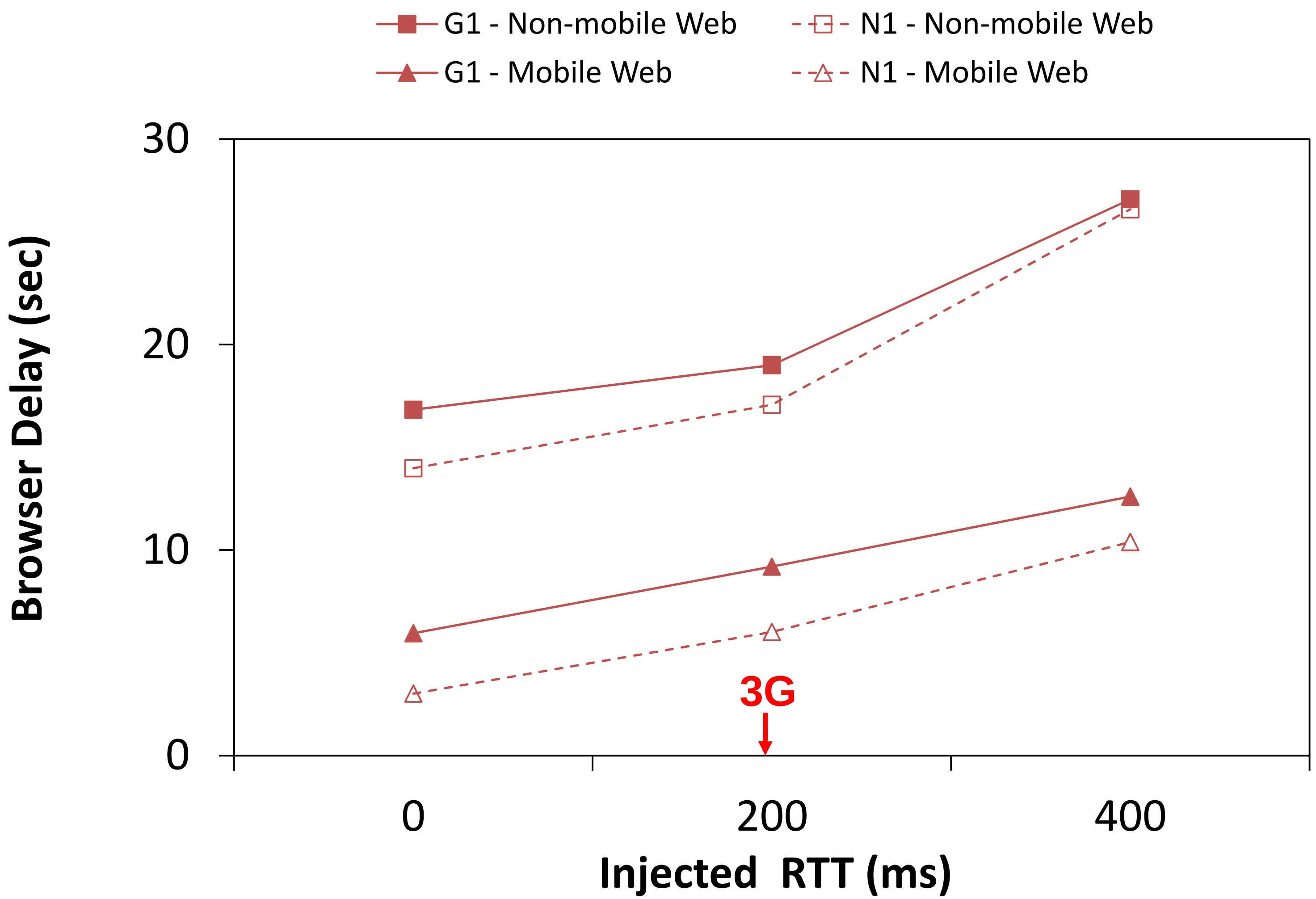
Resource loading is the bottleneck



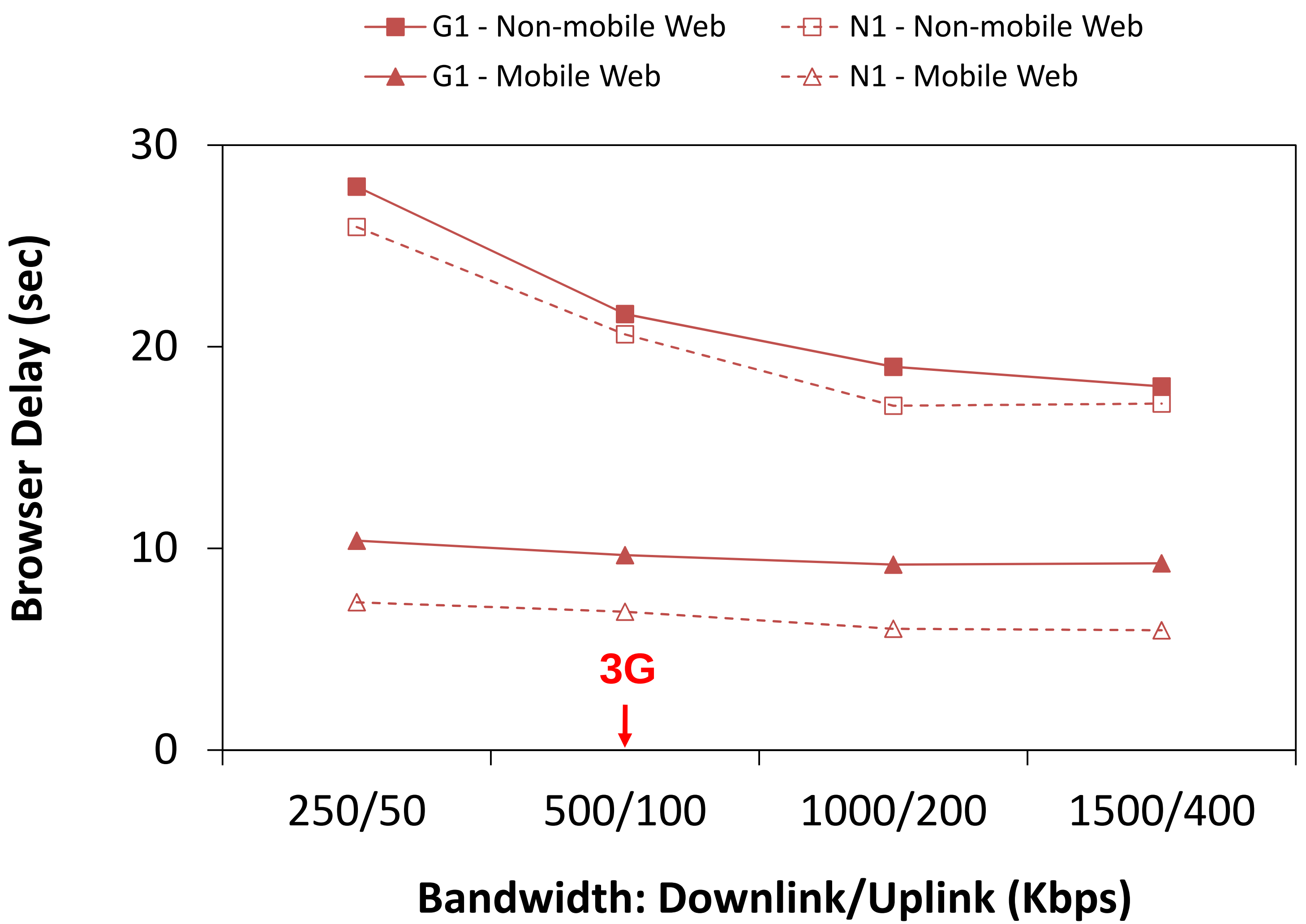
Resource loading is the bottleneck

- Network RTT
- Network Bandwidth
- Browser loading procedure
- Processing power

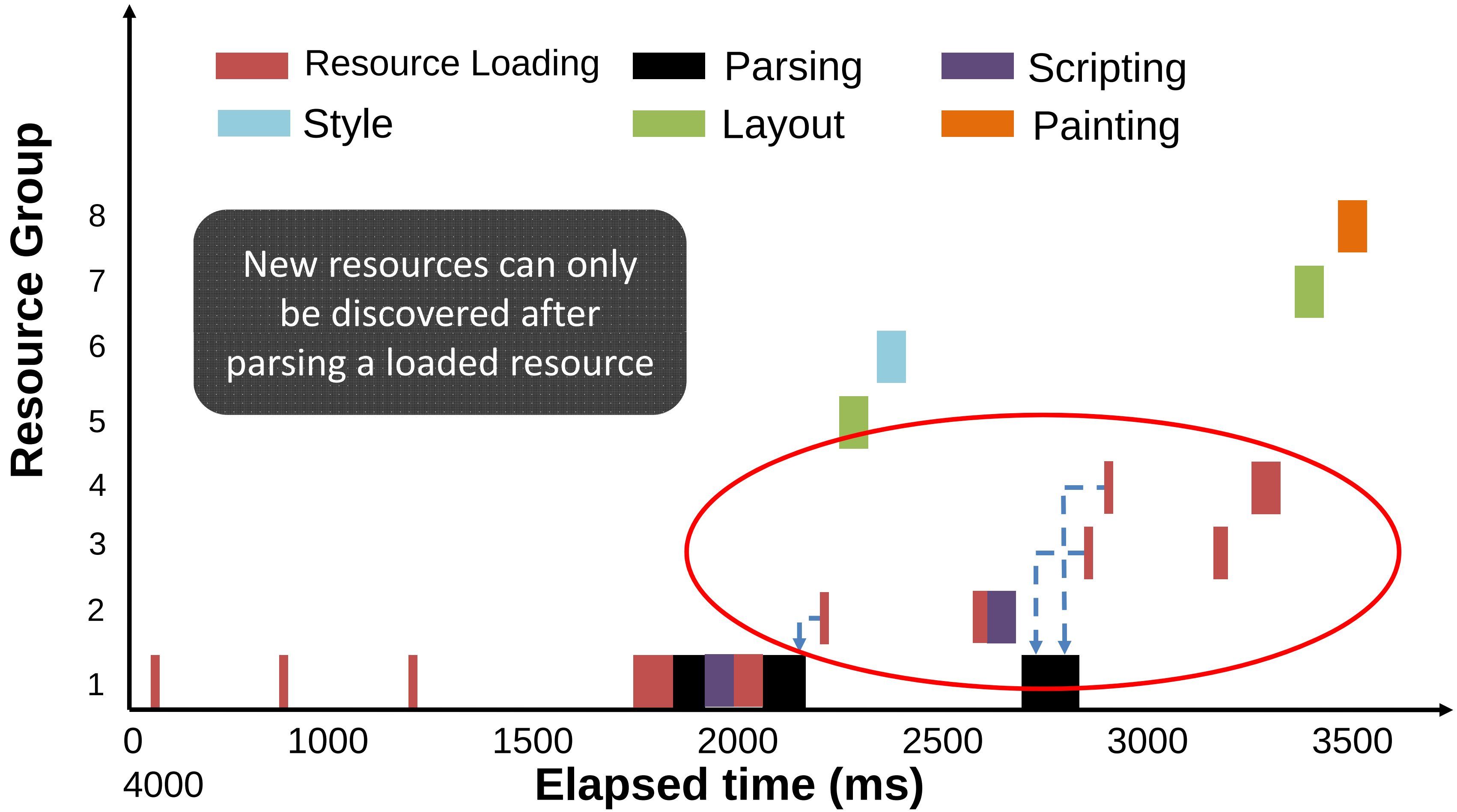
Network RTT matters



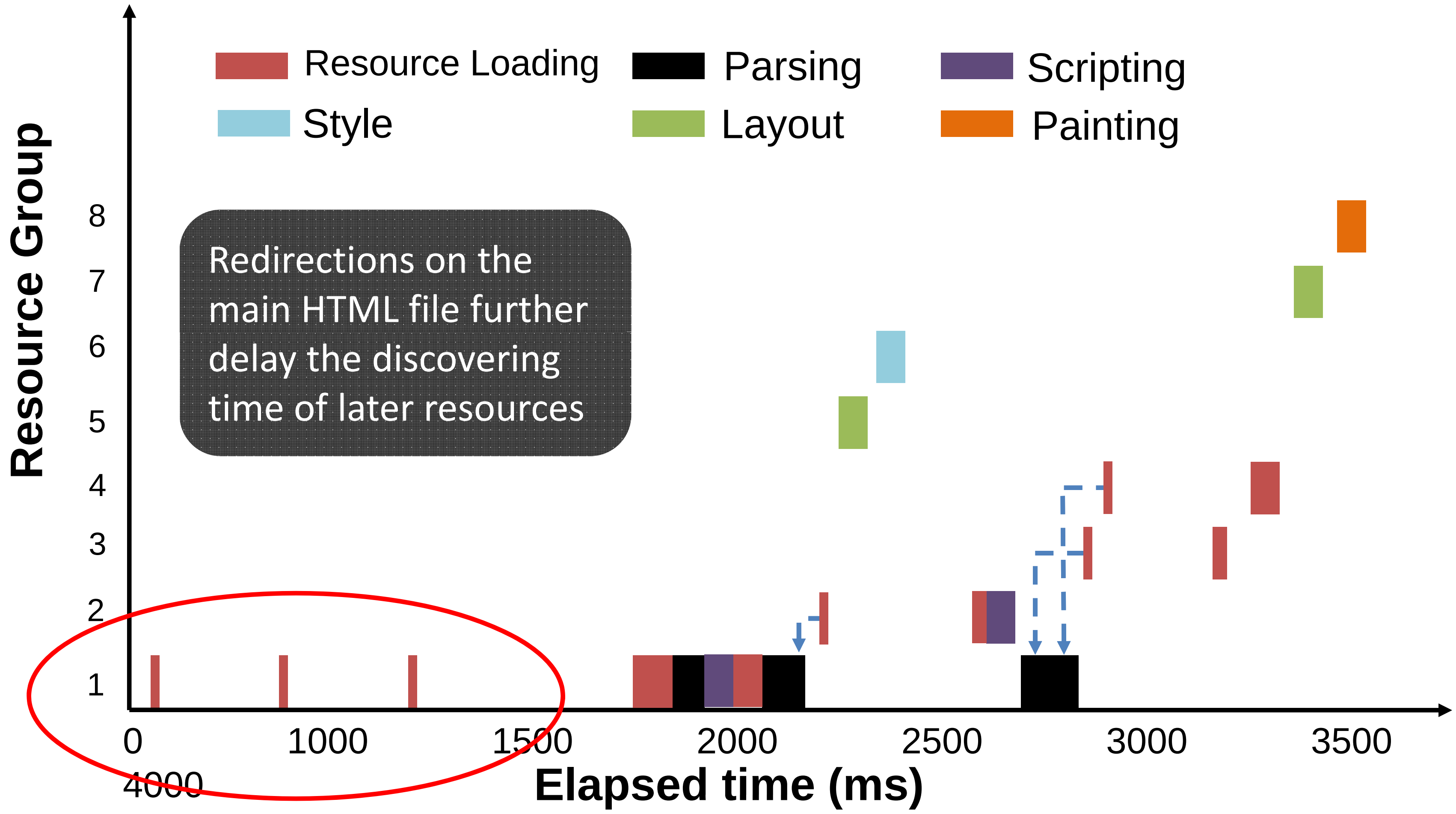
Network bandwidth doesn't matter



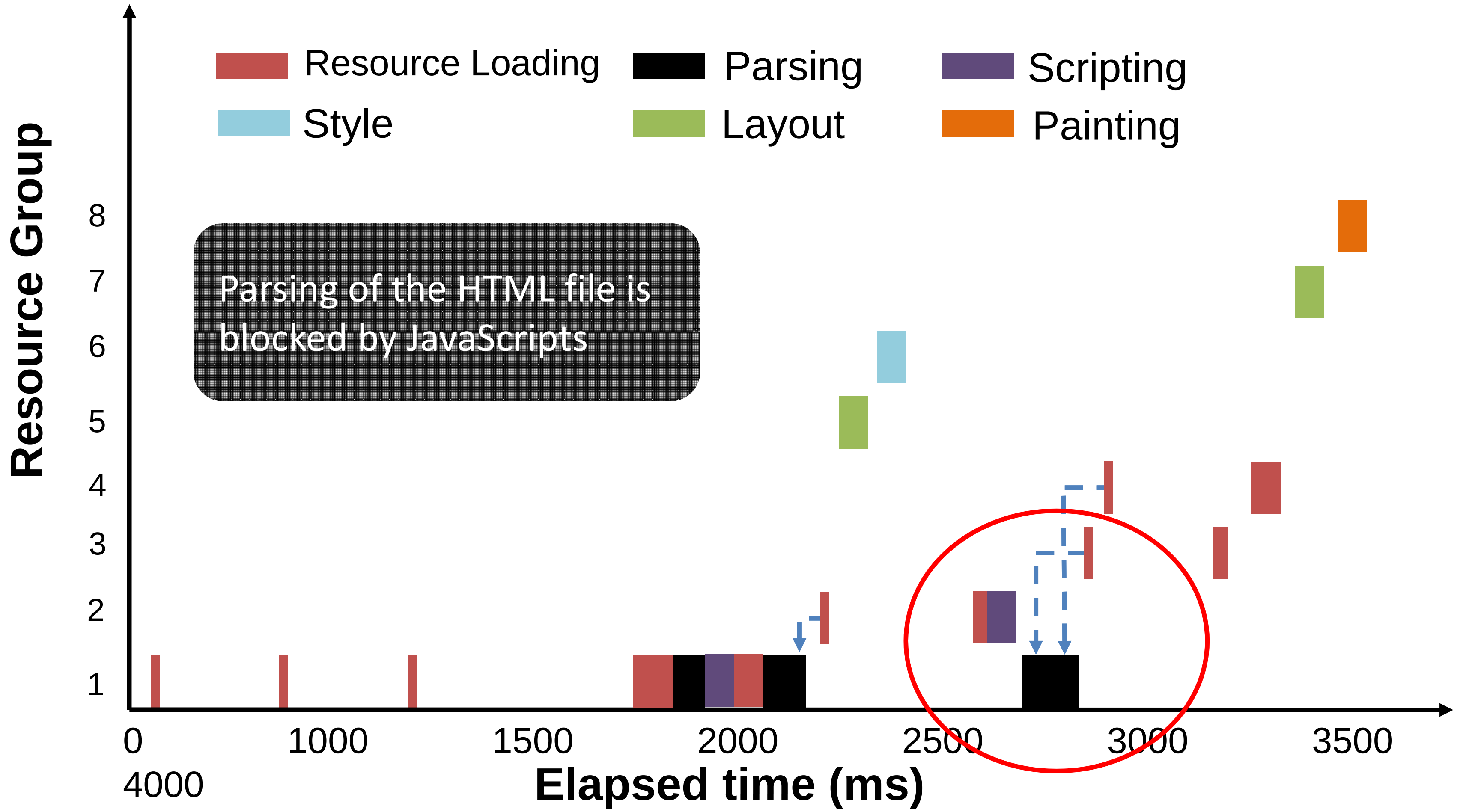
Browser loading procedure



Browser loading procedure



Browser loading procedure



Browser loading procedure

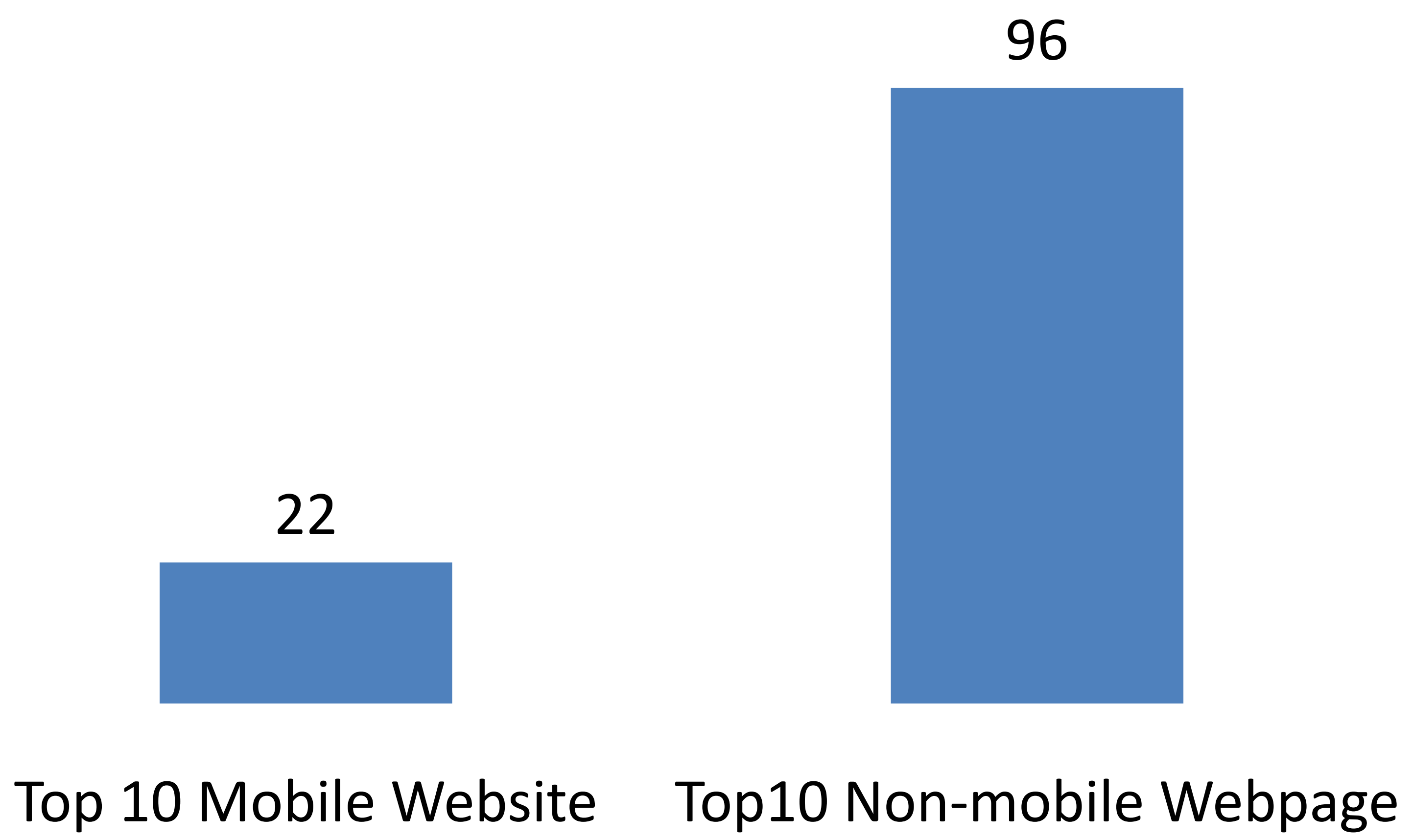
- Up to 25 concurrent requests for top mobile/non-mobile webpages

- Constrains on concurrent TCP connections

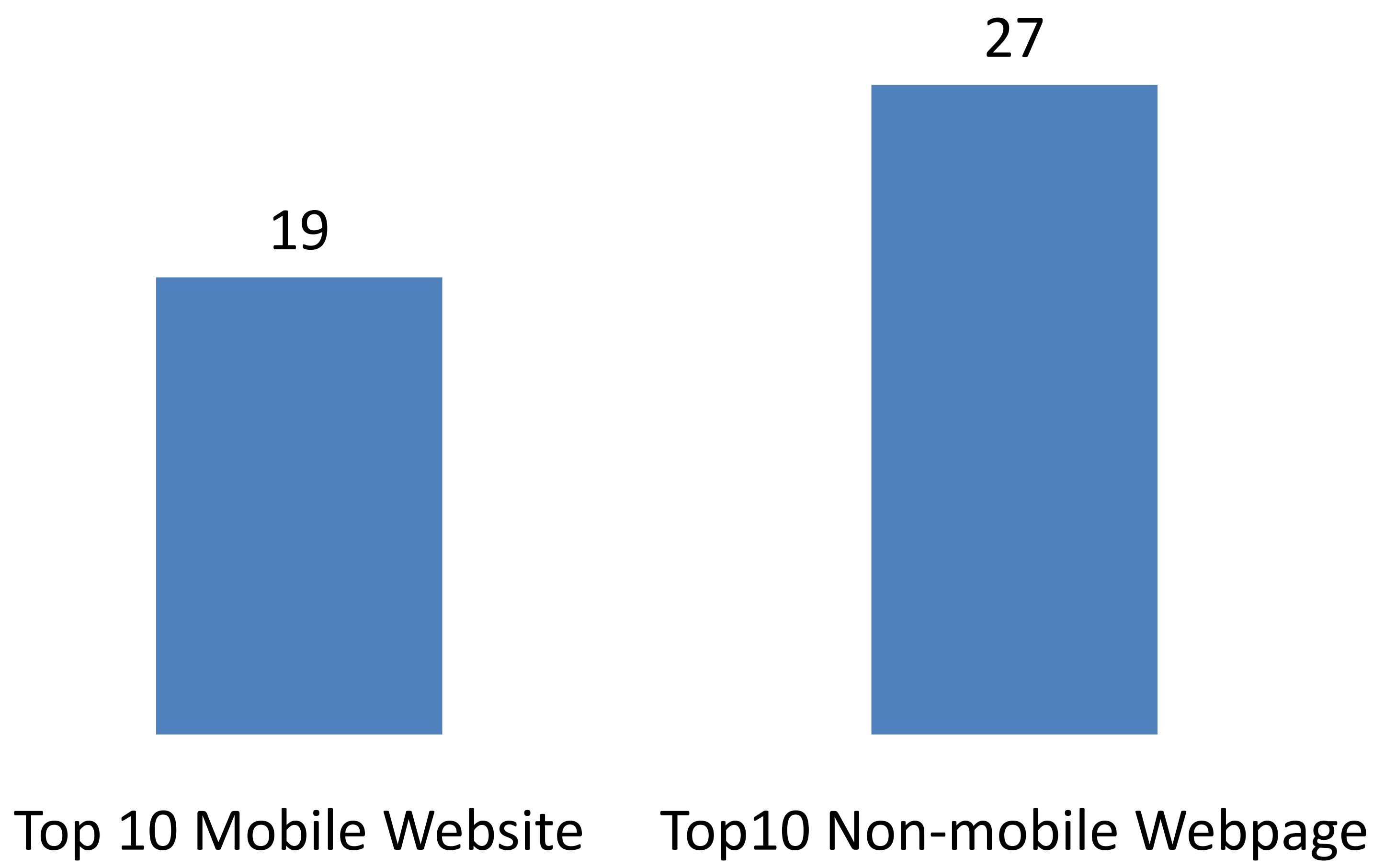
Mobile Browser	Connections/hostname	Maximum connections
Android	4	4
iPhone 4.3	6	35
Blackberry 9700	4	16
Opera Mobile	4	4
Opera Mini	10	60

http://www.browserscope.org/

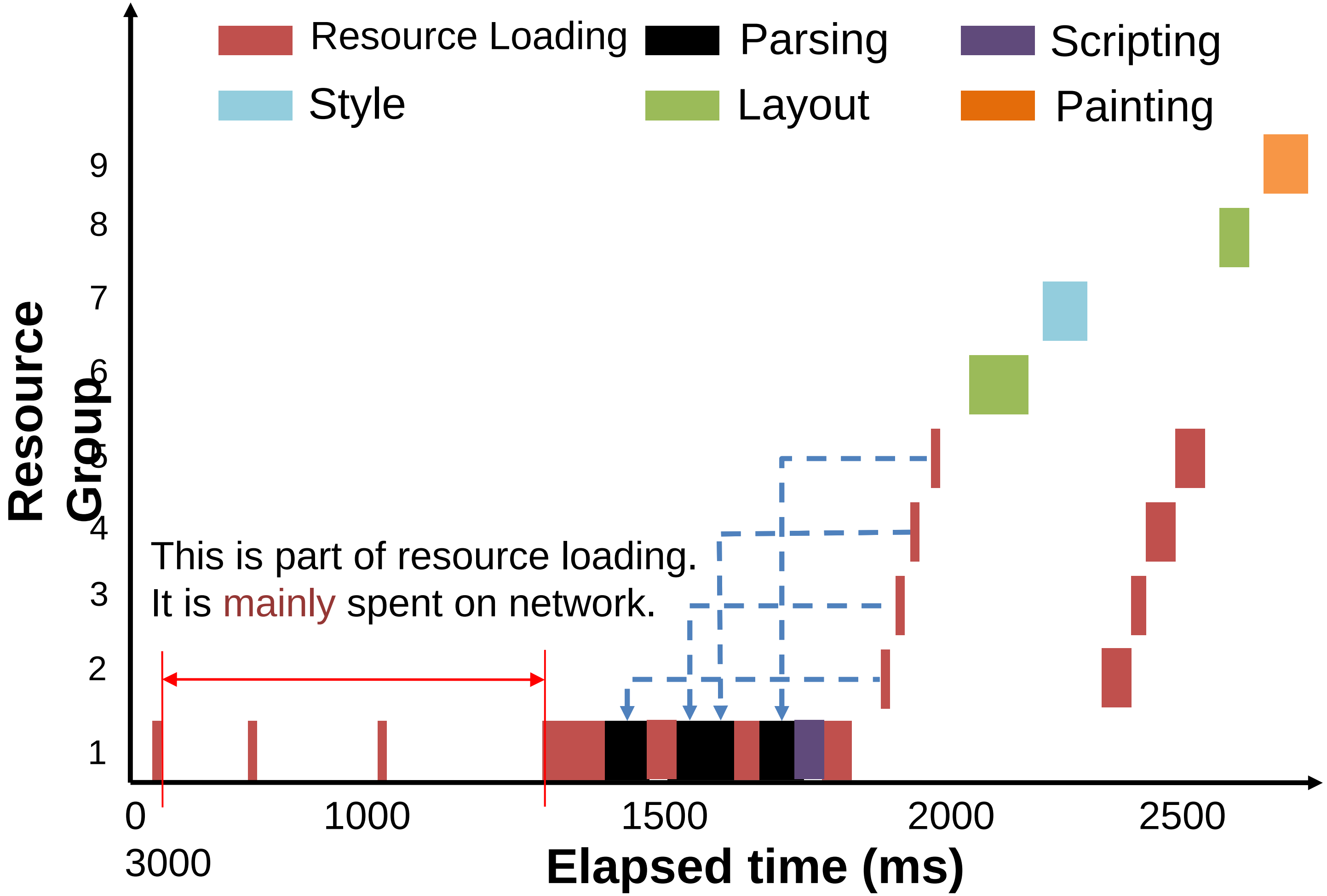
Average number of resources



Average number of network round trips

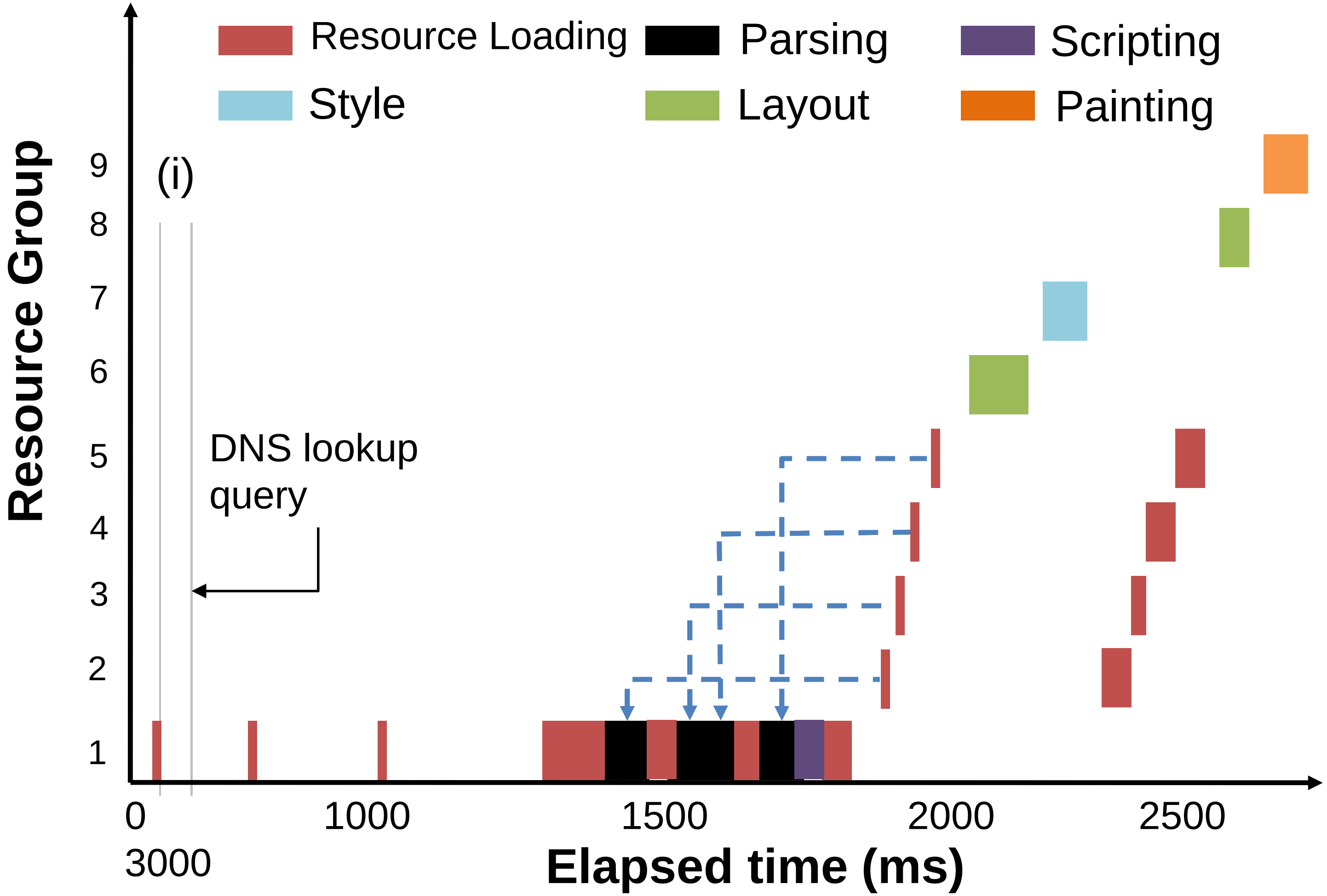


Processing power in resource loading



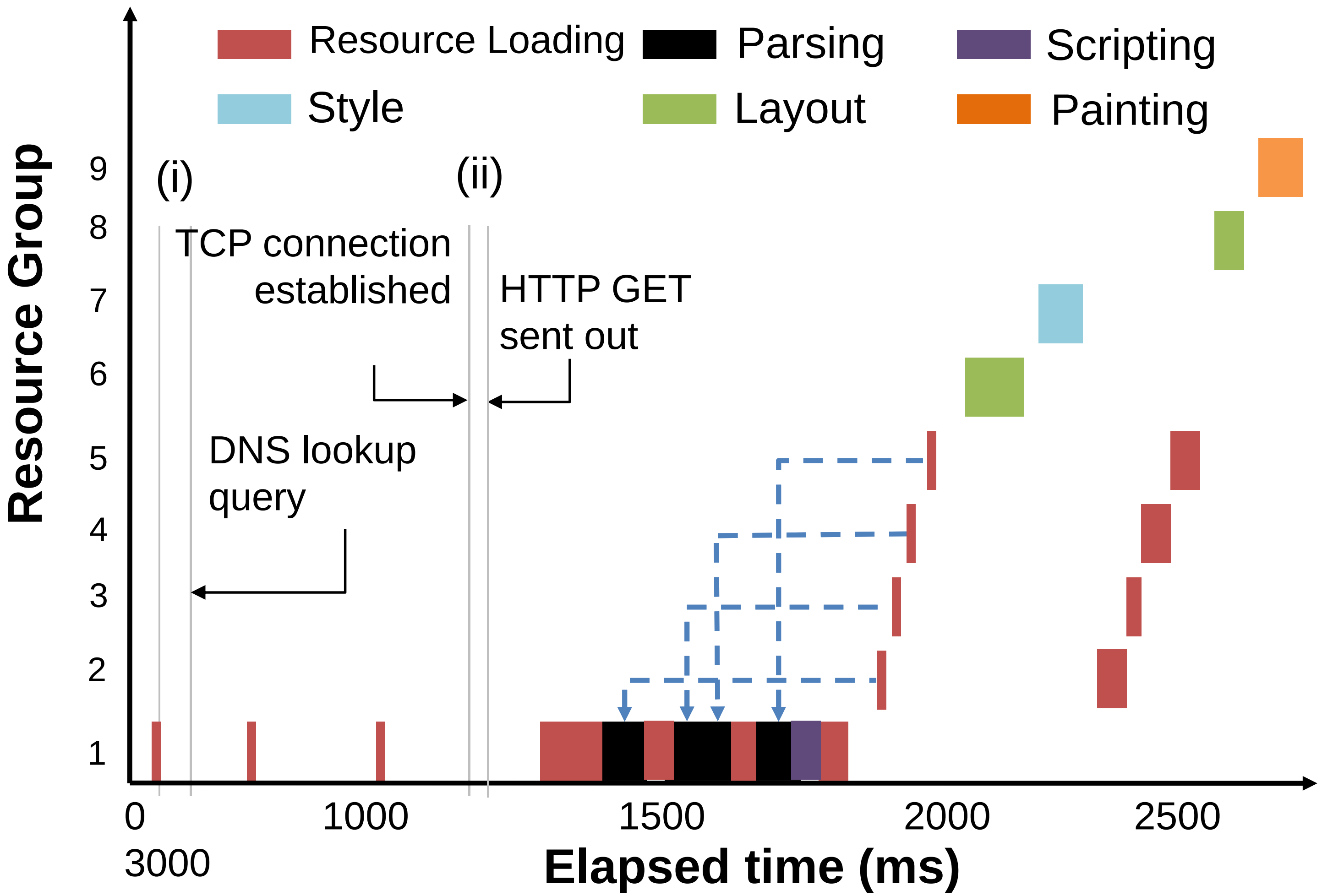
http://mail.yahoo.com

Processing power in resource loading



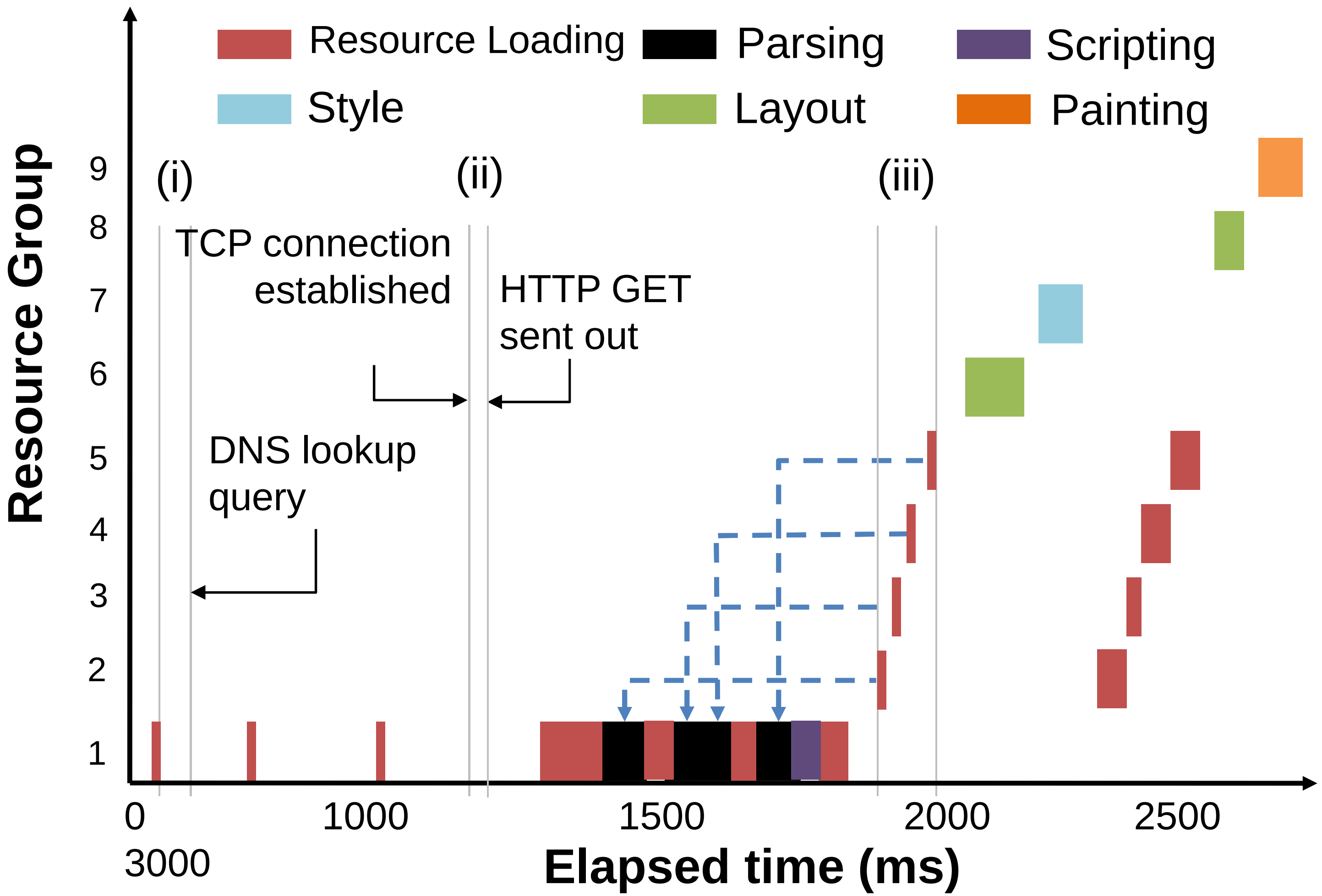
http://mail.yahoo.com

Processing power in resource loading



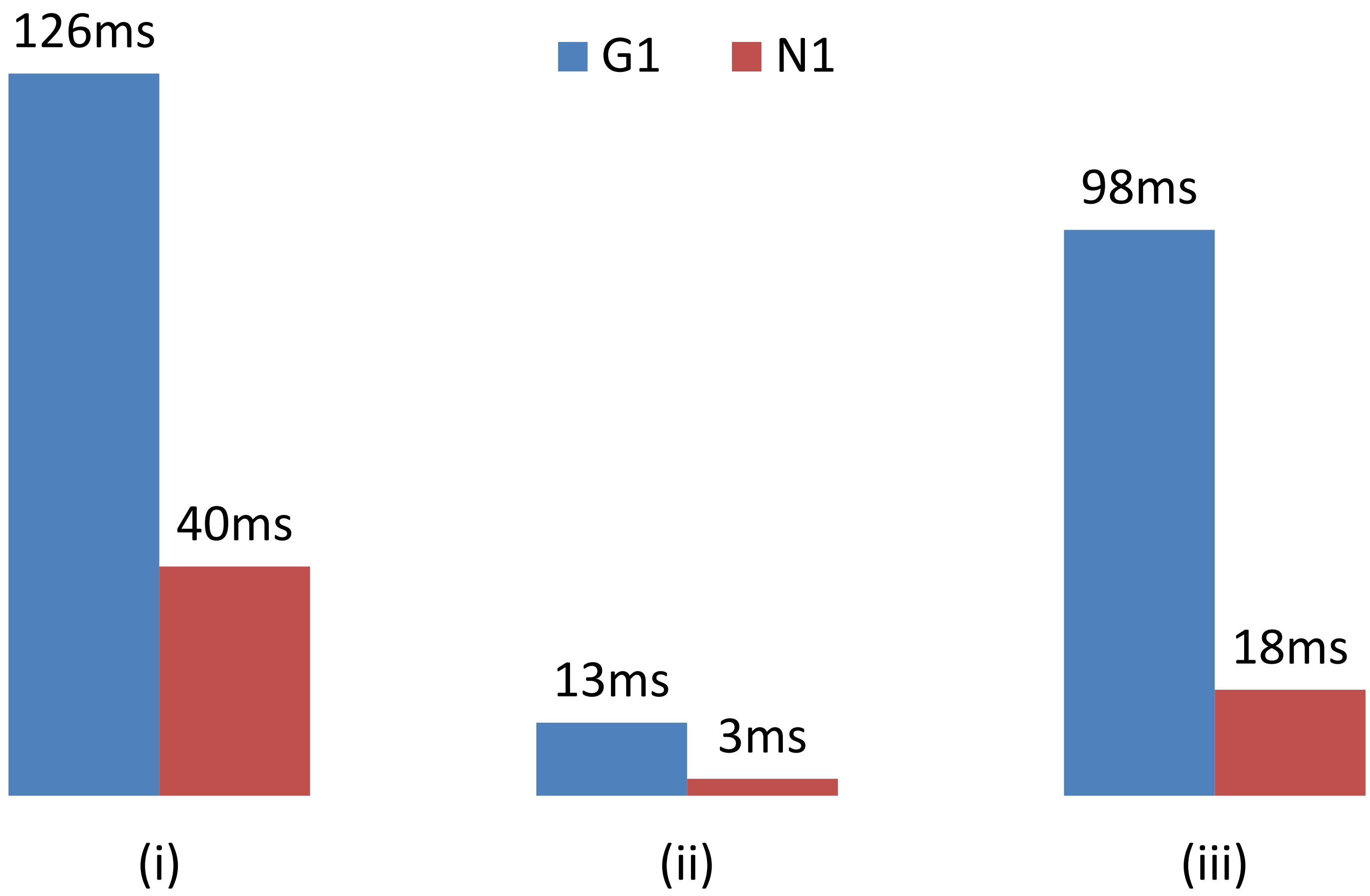
http://mail.yahoo.com

Processing power in resource loading

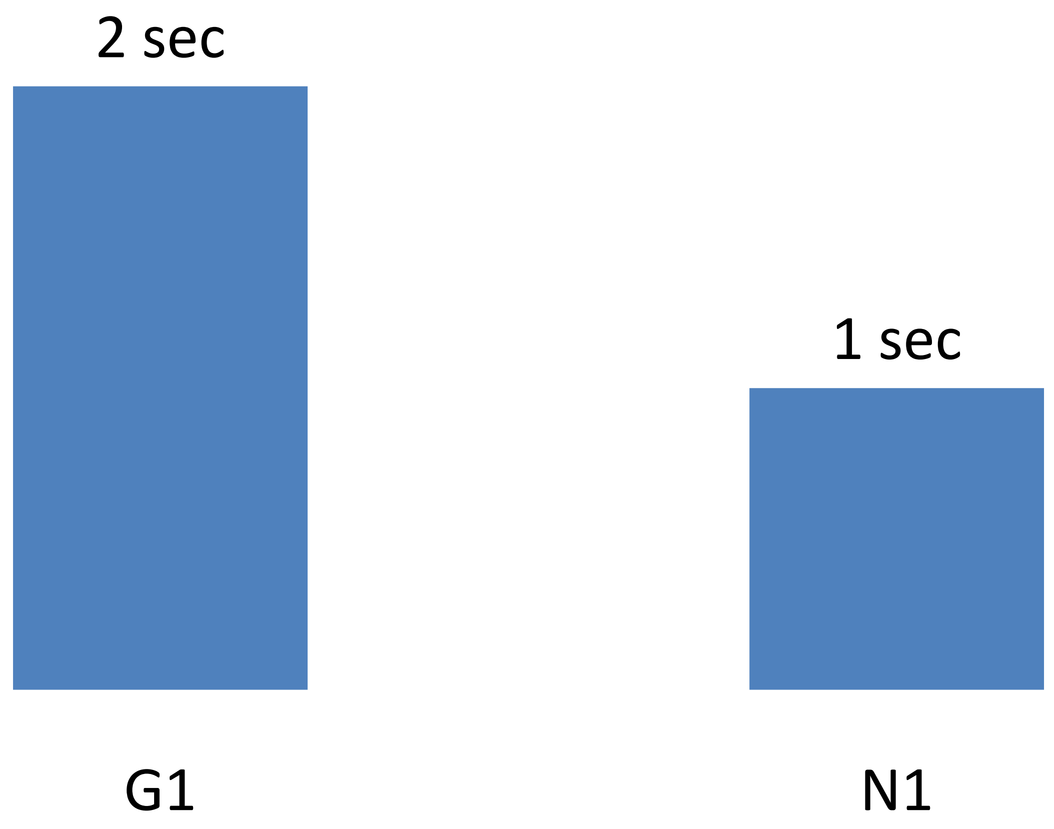


http://mail.yahoo.com

Time spent by G1 and N1 for those three cases



Total time spent in the three cases on average when opening a mobile webpage



Processing power in resource loading

- Other uncategorized processing
  - The OS moves the data from network stack to browser after receiving data packets
  - Computation for secure connection (HTTPS)

More powerful hardware improves the browser delay mainly through faster OS services and network stack instead of faster IR operations.

Performance characterization results

- IR operations do not matter much

- **Resource loading is the bottleneck**
  - Network RTT (**X**)
  - Network bandwidth
  - Browser loading procedure (**X**)
  - Processing power (**X**)

How to improve mobile browser's performance?

Reduce RTT

- Cloudlet
- Data staging

Reduce # of Round Trips

- Web Pre-fetching
- Resource batching
- Data URI scheme
- *Speculative resource loading*

1.M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing, vol. 8, pp. 14-23, 2009.*

2.J. Flinn, S. Sinnamohideen, N. Tolia, and M. Satyanaryanan, "Data Staging on Untrusted Surrogates," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies San Francisco, CA: USENIX Association, 2003.*

3.V. N. Padmanabhan and J. C. Mogul, "Using predictive prefetching to improve World Wide Web latency," *SIGCOMM Comput. Commun. Rev., vol. 26, pp. 22-36, 1996.*

4.Skyfire: http://www.skyfire.com/.

5.L. Masinter, "The "data" URL scheme," http://tools.ietf.org/html/rfc2397, 1998.

On-going work

- Speculative mobile browser design
- Fully understand the impact of hardware
- OS and network service acceleration

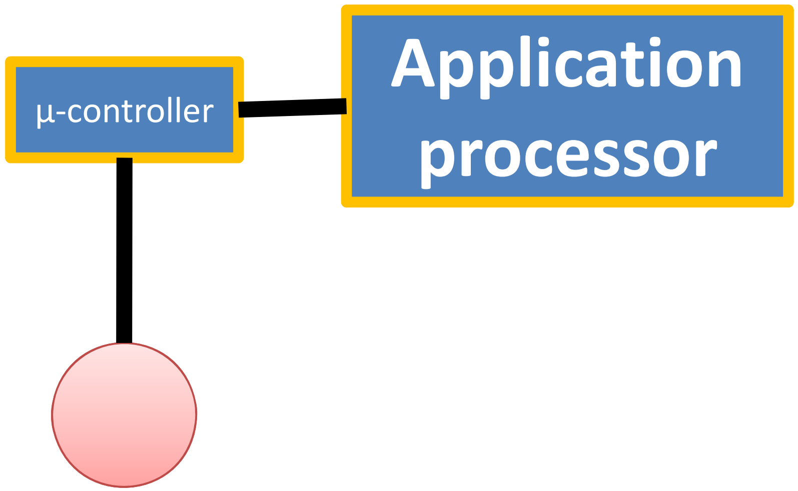
http://www.owlnet.rice.edu/~zw3/projects_Tempo.html

Today's smartphone

Application processor


IBM
Microsoft
YAHOO!
rackspace
Google
amazon web services
ZOHO

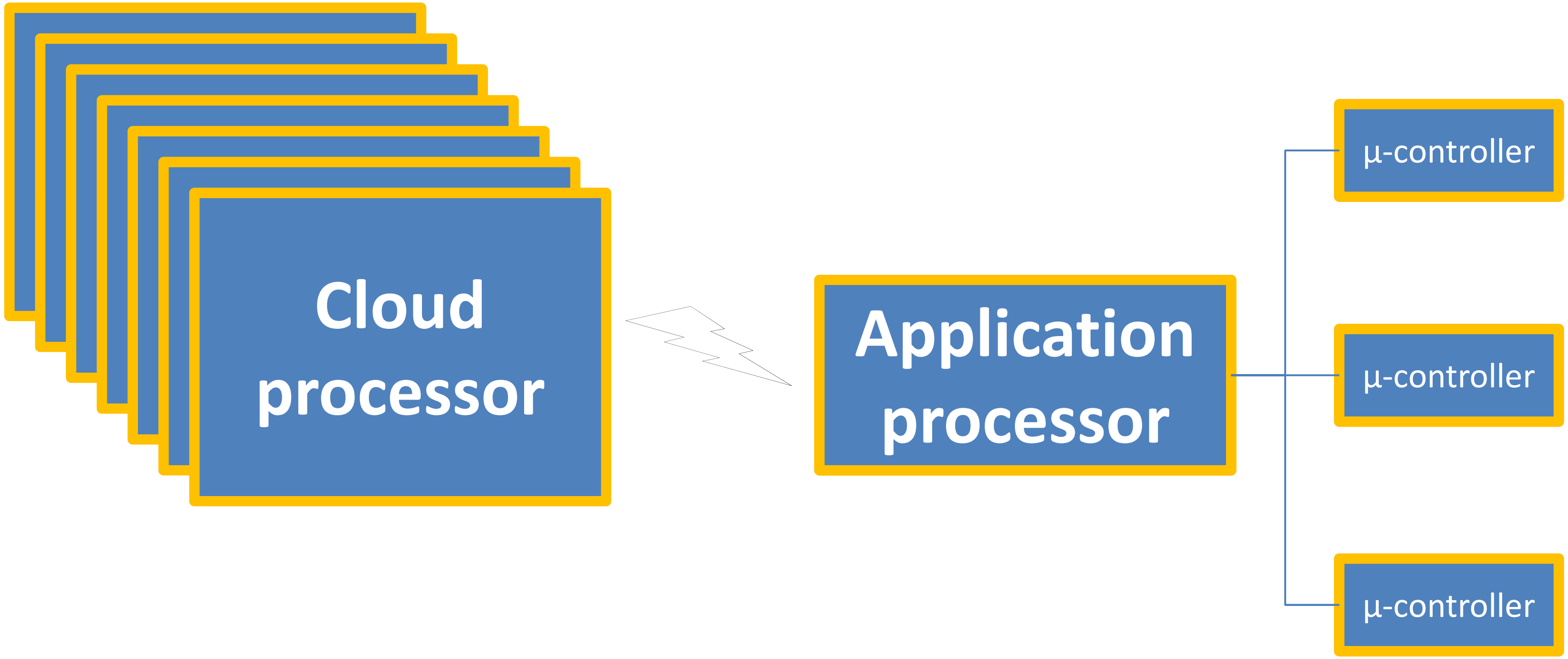
Heterogeneous multiprocessor



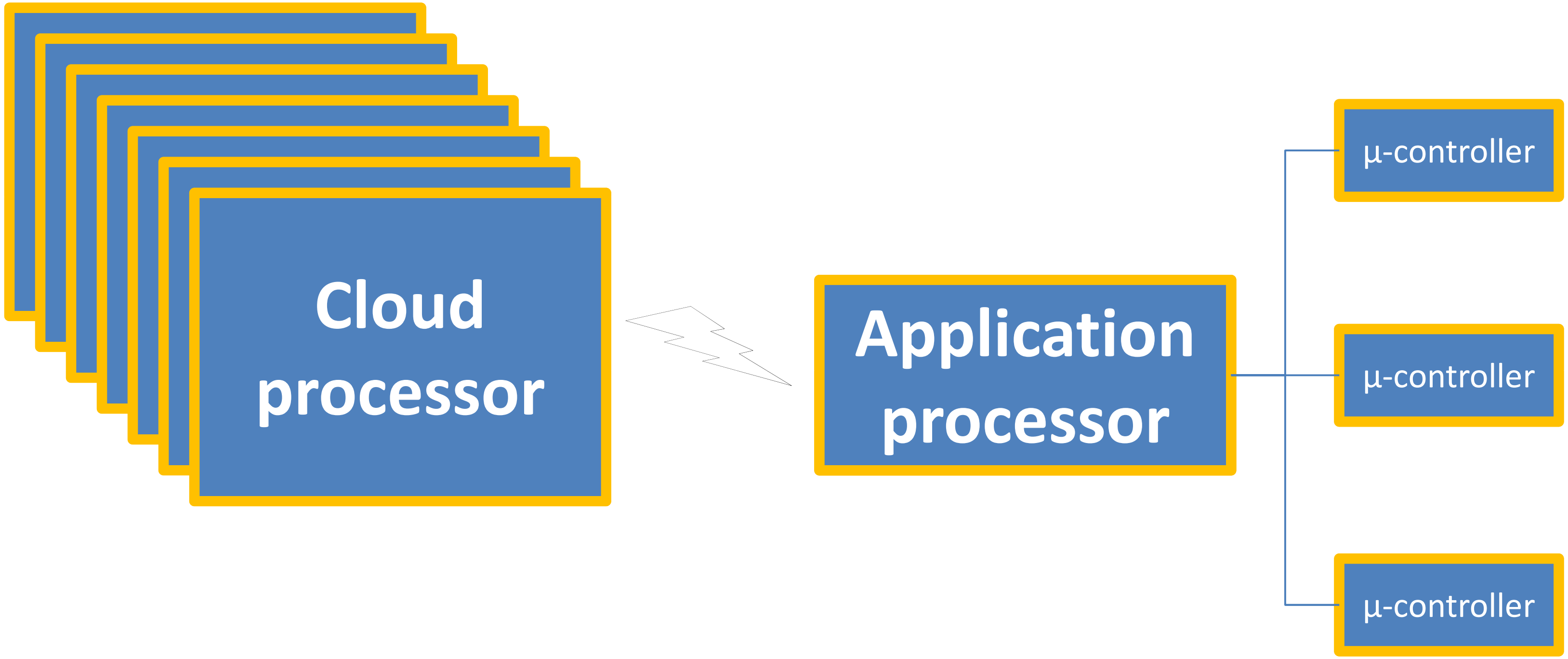
Turducken-like systems

Heterogeneous body-area network

Smartphone 2020

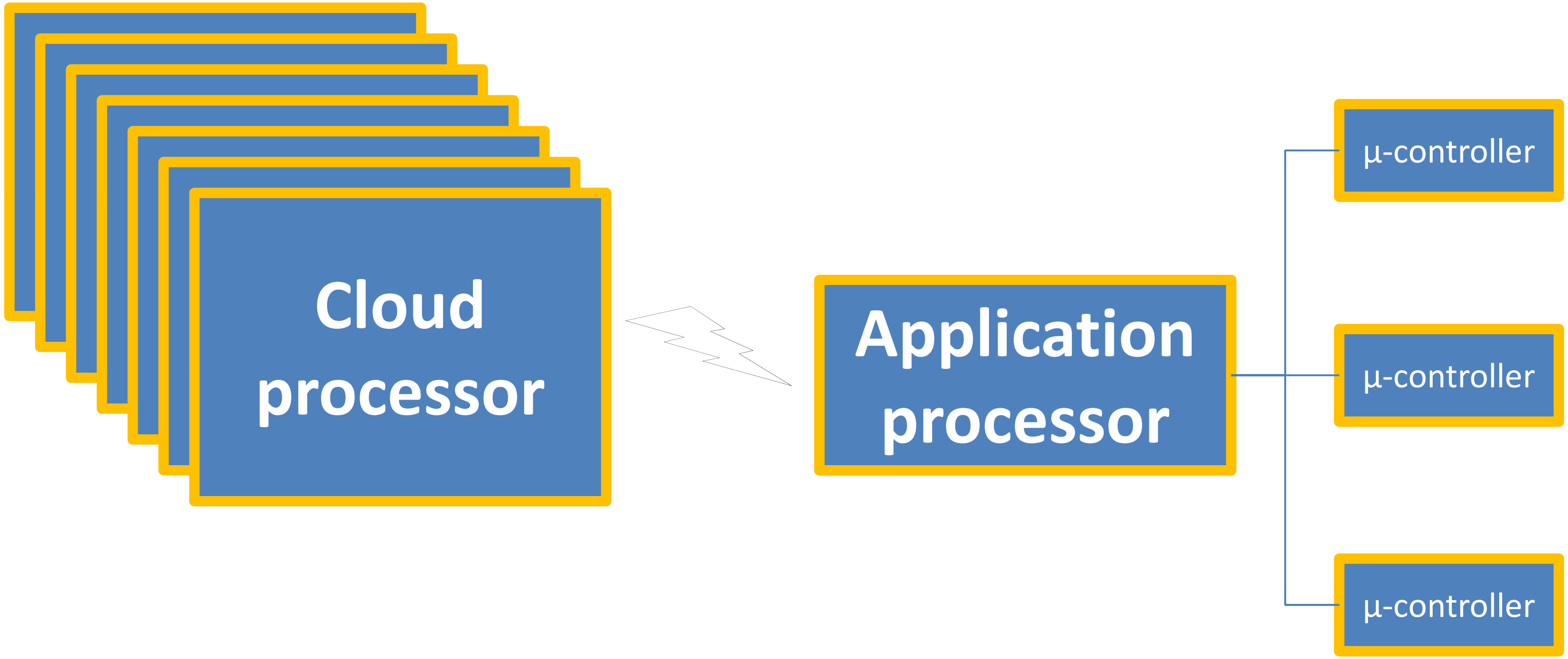


Challenges to programming



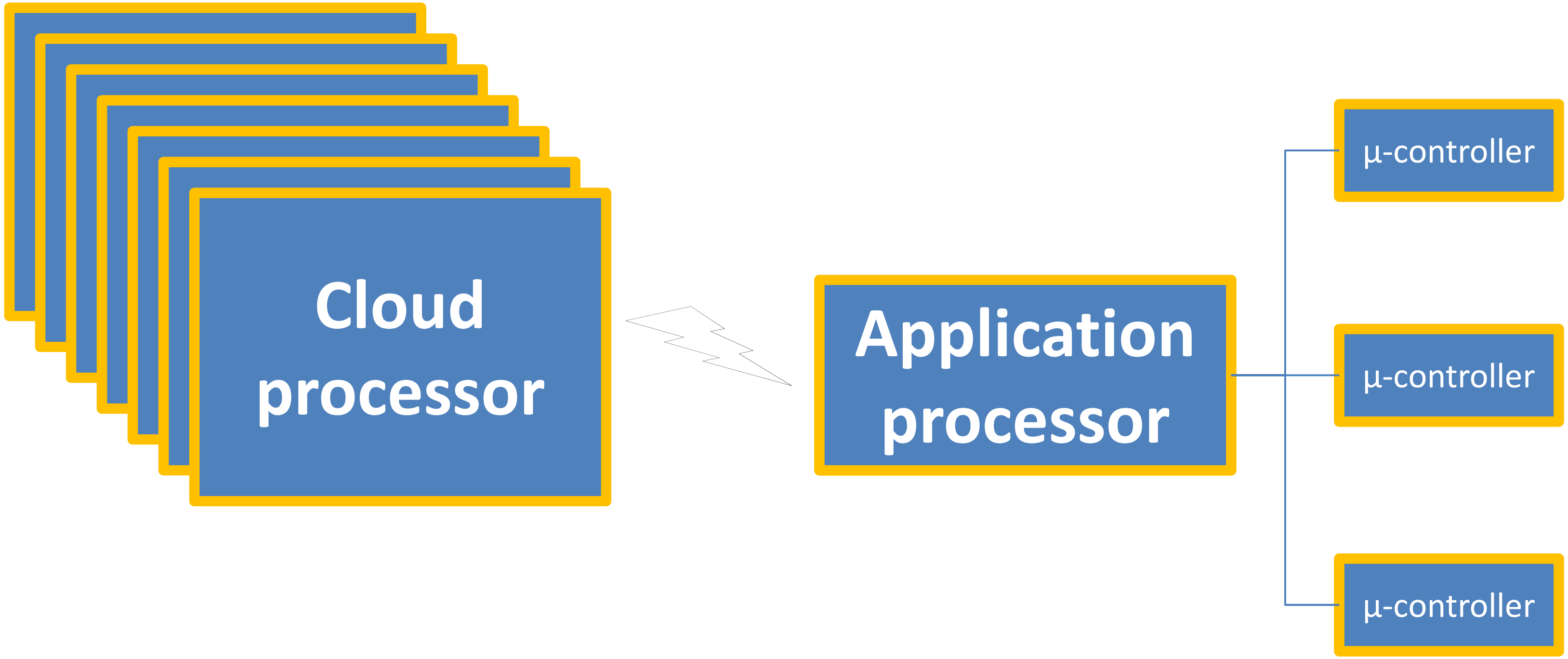
- Resource disparity
  - ISA disparity

Challenges to programming



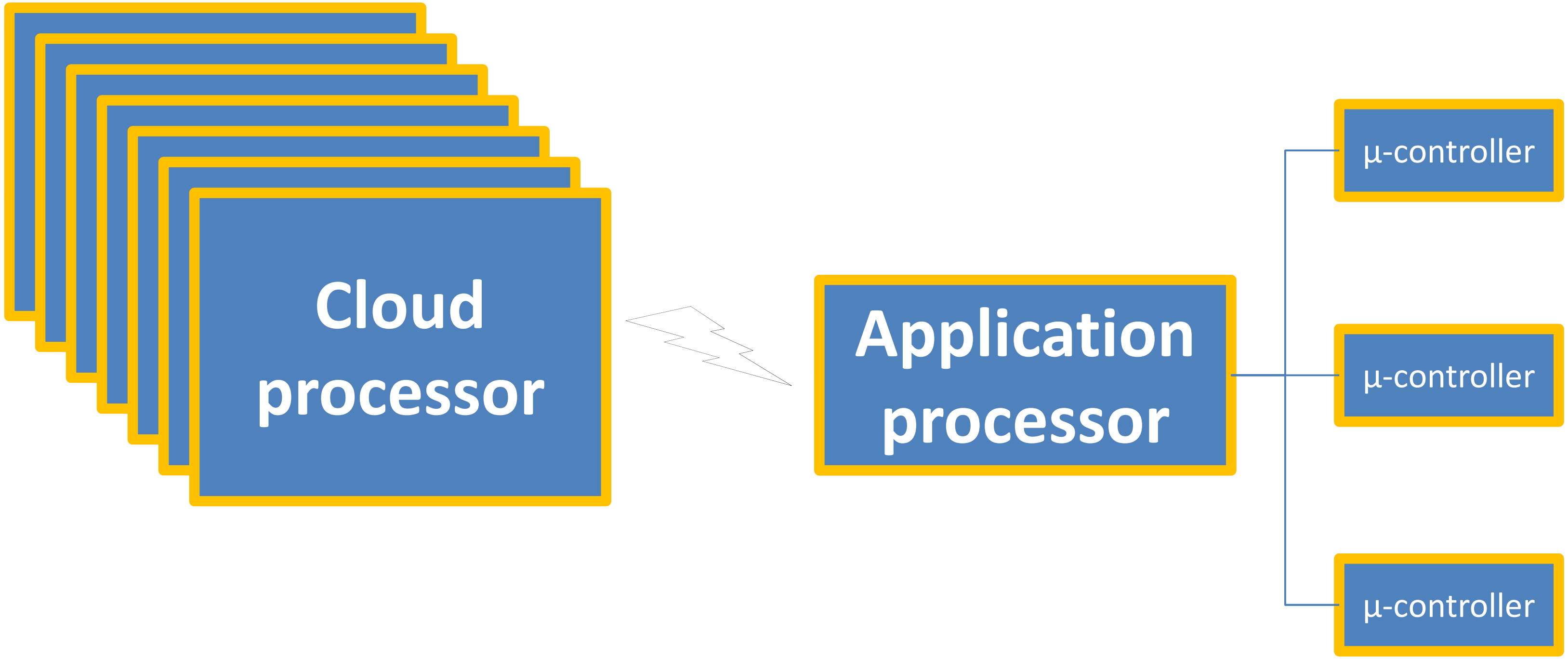
- Resource limitation on “small” processors
  - Virtual machine and coherent memory difficult

Challenges to programming



- Separation of hardware vendors, application developers, and users
  - Developer blind of external computing resources and runtime context

Challenges to programming



- Established programming model and OS

Existing solutions

mPlatform etc.

CPU+GPU systems

Virtual machine

Offloading systems
(active disk, Hydra etc.)

Single ISA

Turducken-like
cohort systems

Complete transparency

No transparency

Prohibitively expensive

High burden on
application developers

Reflex: Transparent programming of heterogeneous mobile systems

http://reflex.recg.rice.edu/

Inspired by the heterogeneous distributed nervous system

Enough transparency

mPlatform etc.

CPU+GPU systems

Offloading systems
(active disk, Hydra etc.)

Turducken-like
cohort systems

Virtual machine

Single ISA

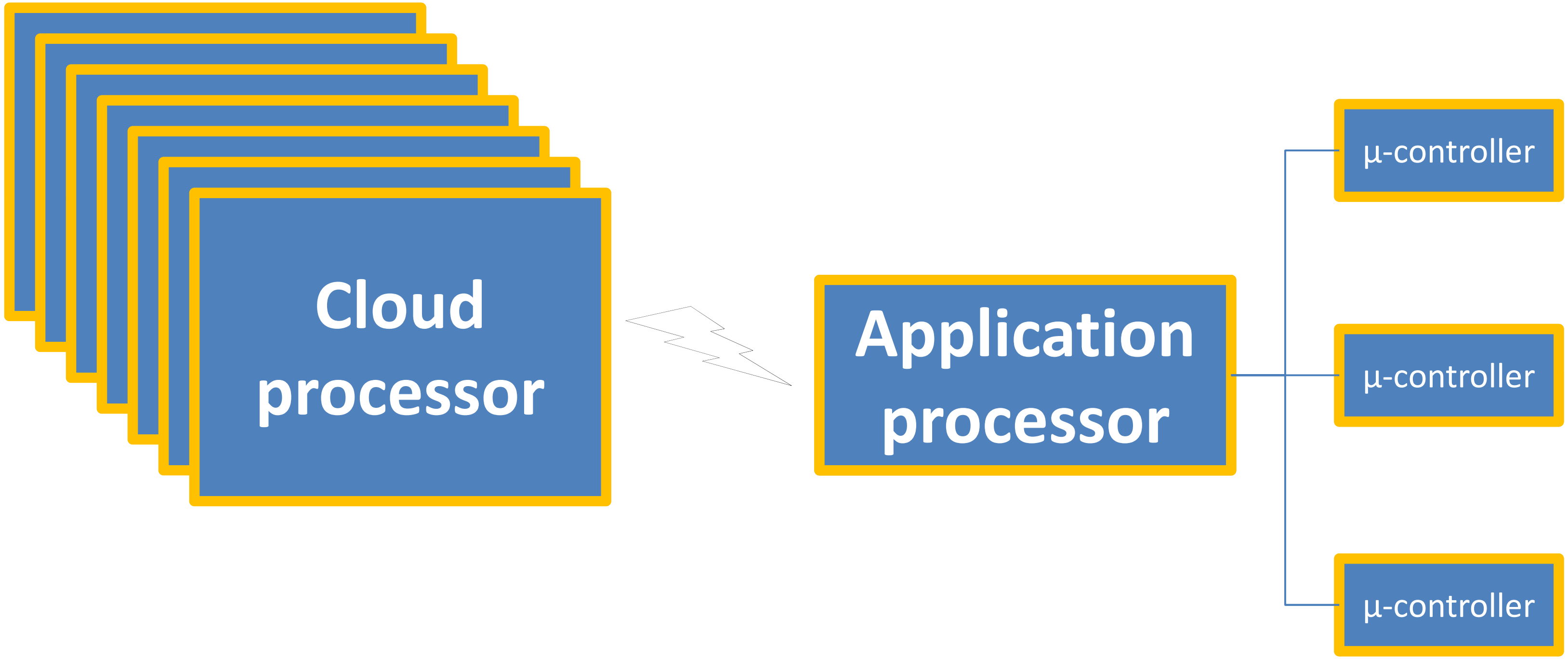
Reflex

Complete transparency

No transparency

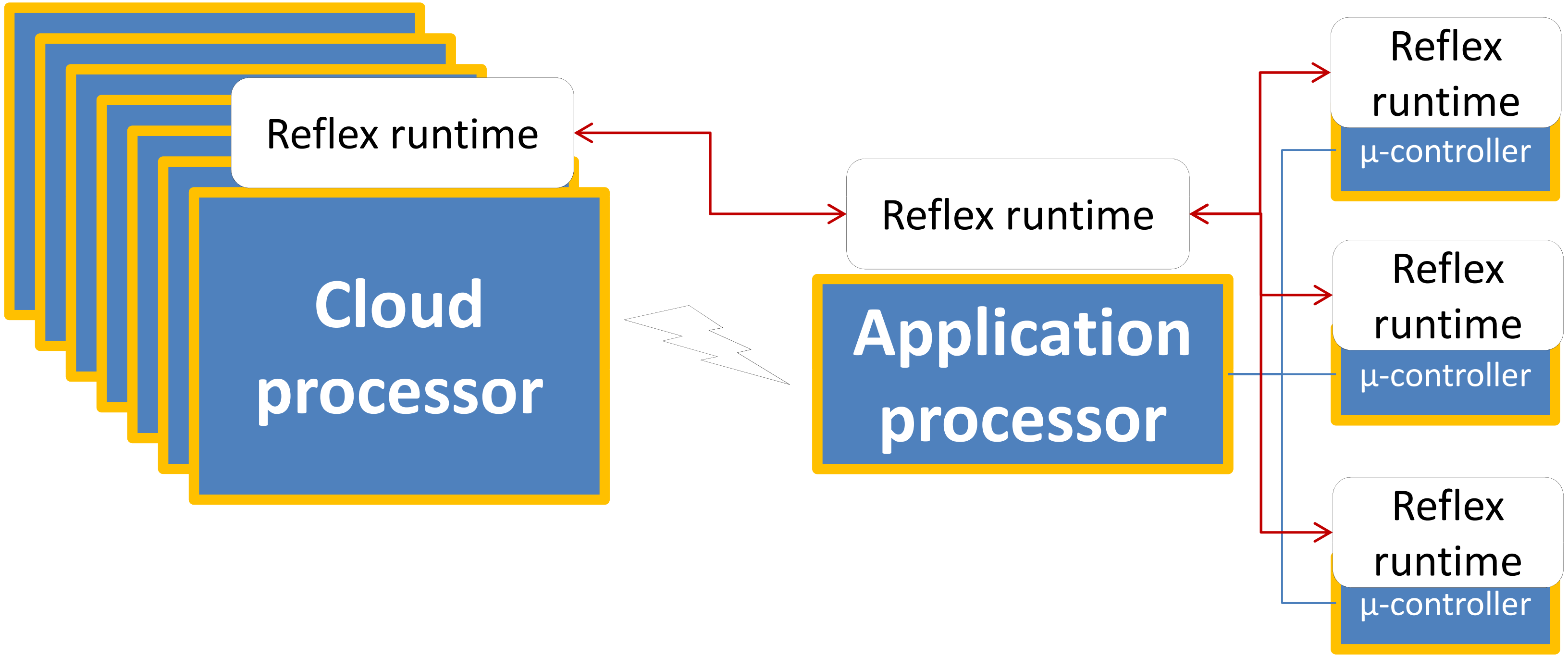
- Ease of programming
- Execution efficiency

Key ideas



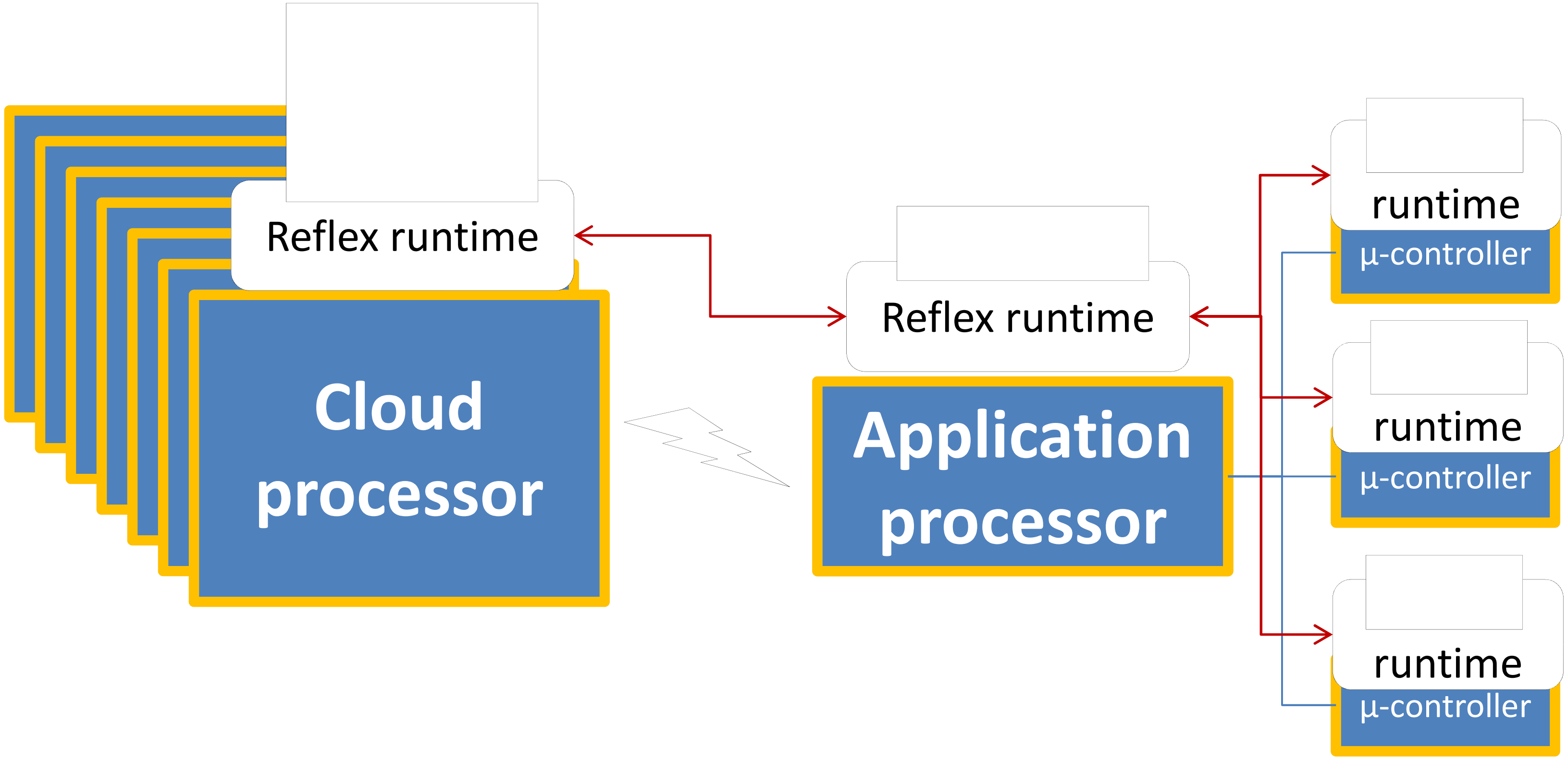
- Light weight virtualization of sensor data acquisition, timer, and memory management

Key ideas



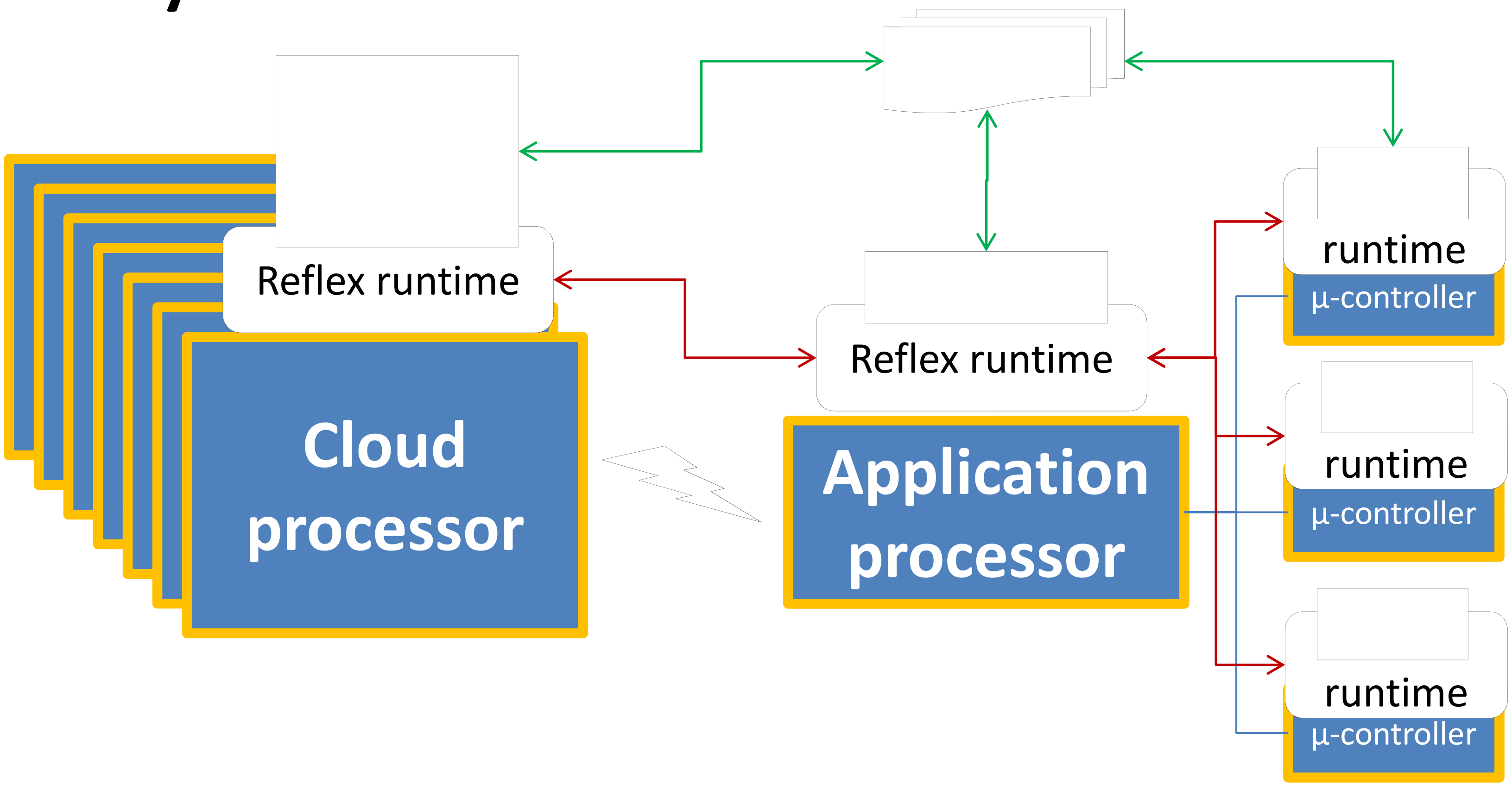
- Distributed runtime for transparent message passing

Key ideas



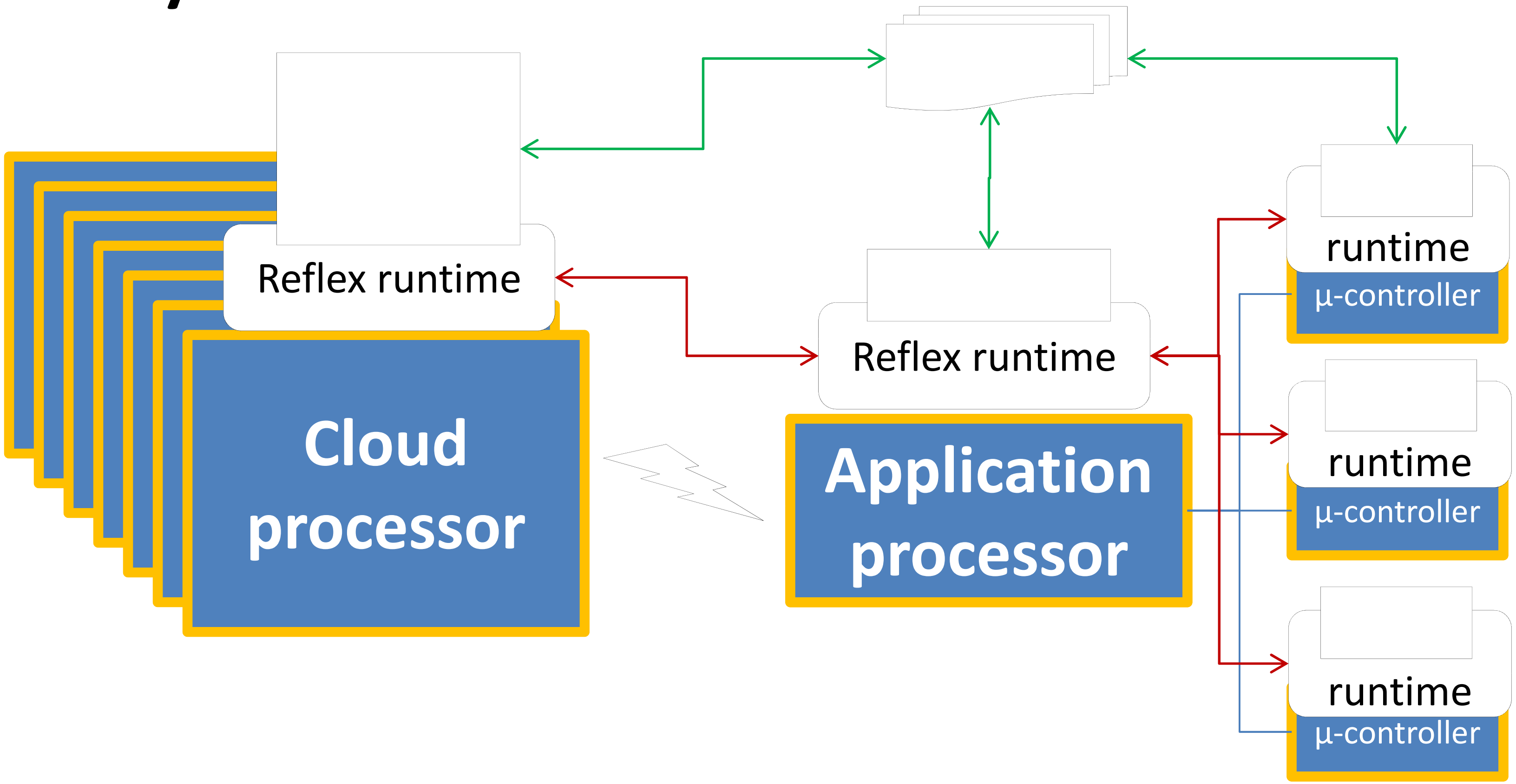
- Automatic code partition through a collaboration between runtime and compiler

Key ideas



- Identify a small coherent memory segment
  - Maintain by message passing through the runtime

Key ideas

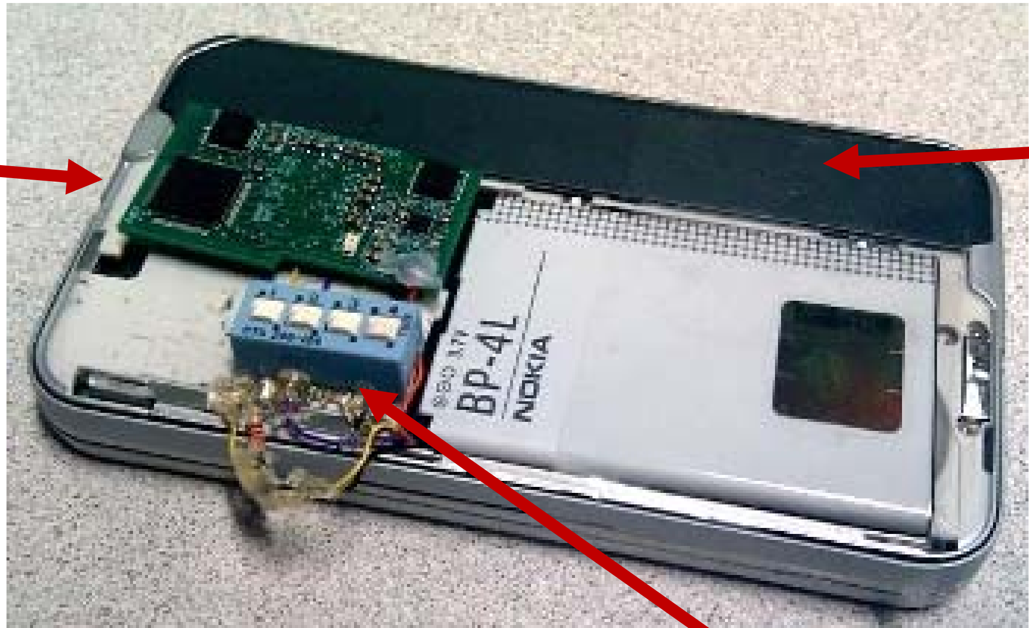


- Type safety for dynamic process migration

Reflex Prototype (board integration)

- Programmable accelerometer (TI MSP430)
- Wired sensor through UART port

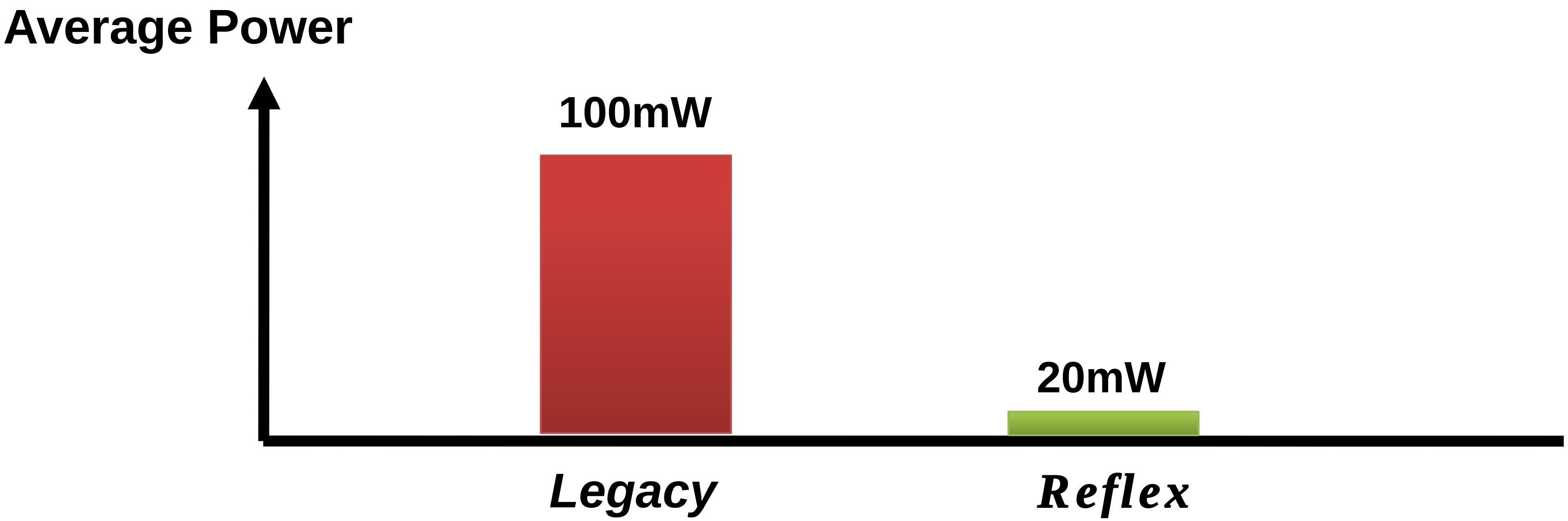
Rice Orbit Sensor



Nokia N810

Serial connection

Fall detection with N810



The secret: we do not fall very often

Coded as part of Smartphone program

```
class SenseletFall : public SenseletBase {
public:
  SenseletFall () { _avg_energy = 0; };

  void OnCreate() { RegisterSensorData(ACCEL, 50); };

  void OnData(uint8_t *readings, uint16_t len) {
    uint16_t energy = readings[0]*readings[0] + \
                      readings[1]*readings[1] + \
                      readings[2]*readings[2];
    //do a simple low-pass filtering
    _avg_energy = _avg_energy / 2 + energy / 2;

    // detect fall accident with the filtered energy
    if (_avg_energy > THRESHOLD) {
      theMainBody.FallAlert();  //RMI
    }
  }

  void OnDestroy() { UnRegisterSensorData(ACCEL); };

private:
  uint16_t _avg_energy;
};
```

Thanks!

http://www.recg.org

NSF

MOTOROLA

Microsoft Research

Texas Instruments