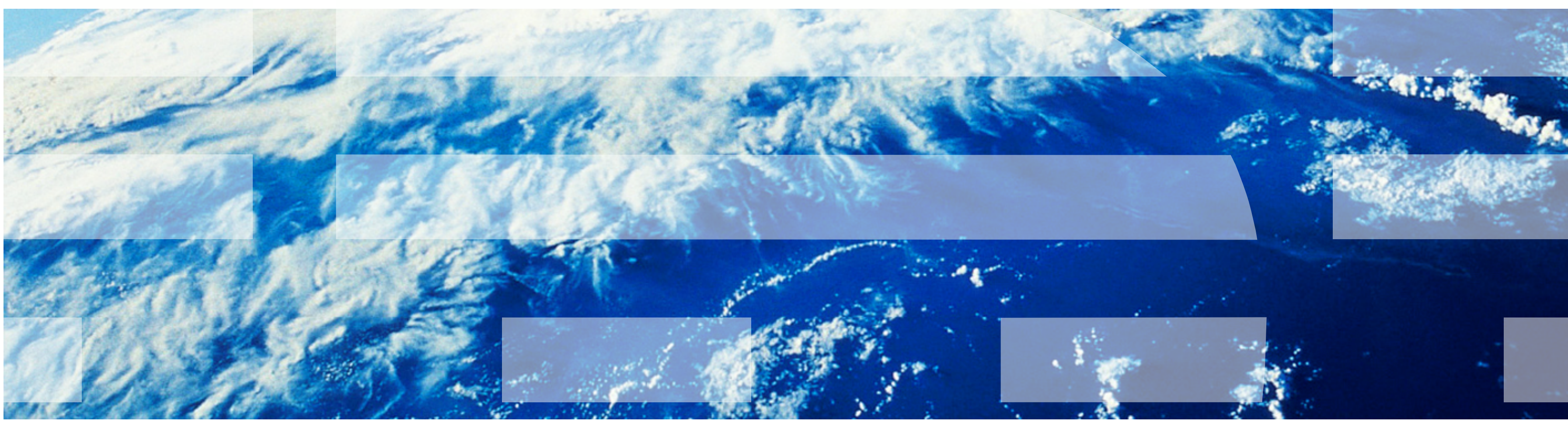
Design and Optimization of Mobile Aggregation Architectures

Ioannis Papapanagiotou, PhD
(Yanni)

Computer & Information Technology
Purdue University



http://people.engr.ncsu.edu/ipapapa/Purdue/PurdueCIT.pdf

Who am I?

- Dual Major PhD in North Carolina State University
  – Computer Engineering
  – Operations Research
- IBM (3 years)
  – Emerging Technology Institute:
    • In house incubator team reporting directly to the CTO and VP
  – TJ Watson Research Center:
    • IT & Wireless convergence group
- NCSU (3 years RA & TA)
  – Taught Engineering, Networking and Simulation courses
  – Research funded by IBM, Cisco Systems, Time Warner Cable
- Appointments with
  – University of Patras
  – Technical Chamber of Greece
  – VTT Technical Institute in Finland
- Fellowships:
  – IBM PhD Fellow
  – Academy of Athens PhD Fellow

What I do

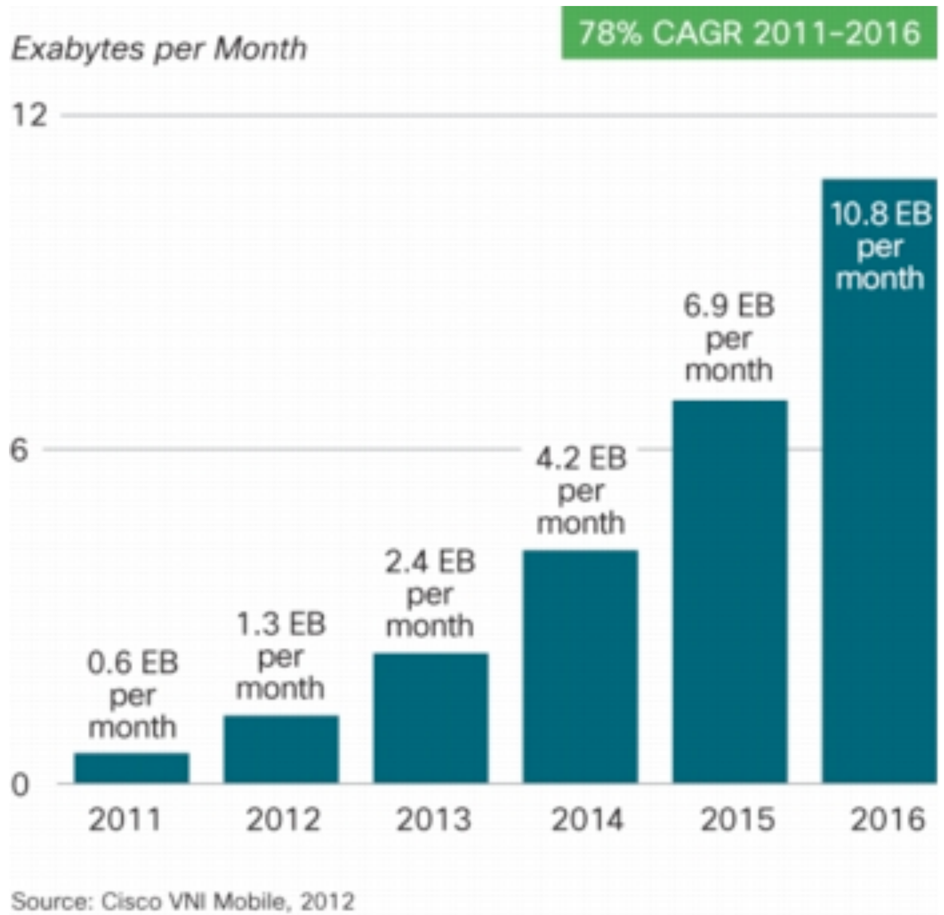
- Current work
  - Server side NoSQL/JSON solution for end-stores
  - Working with front-end caches and databases
- Research Interests:
  - Data analytics in wireless networks
  - Byte level deduplication
  - Performance analysis & simulations
  - Elasticity in Cloud Computing
- Experience with E-Learning technologies:
  - Synchronous (e.g. virtual meeting rooms) and Asynchronous (e.g. video recordings) types of assisting.
  - Emulation of computer networking labs through virtualization.
- Publication Record
  - 5 Peer-Reviewed journals
  - 12 Conference proceedings
  - 8 Submitted patents
  - 3 Book chapters
  - 121 Citations
- More info: http://people.engr.ncsu.edu/ipapapa

Motivation

Smartphones

+ =

Need For Speed

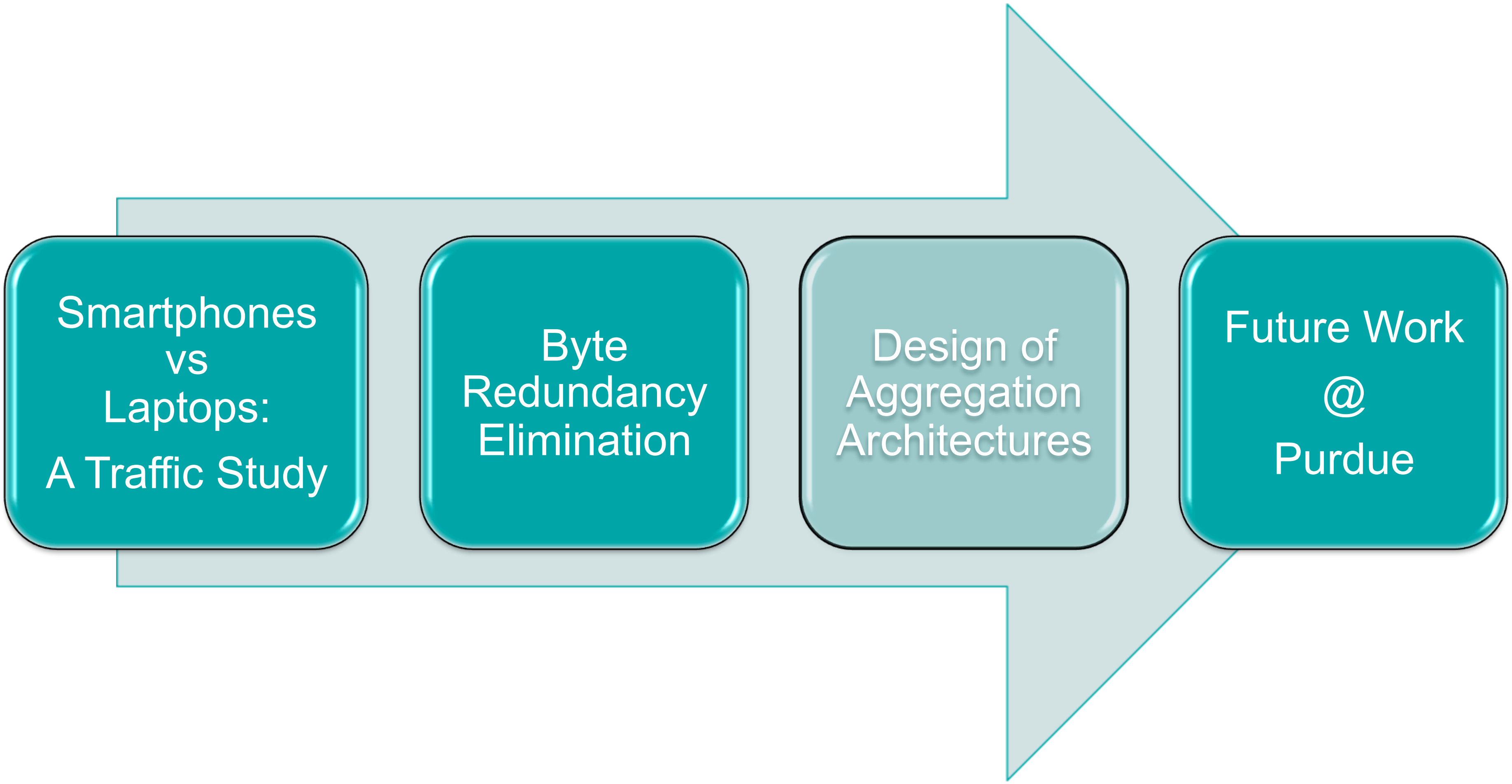


Wireless Service Providers have to transfer this traffic to the Internet!

Introduction

- Operating System fingerprinting algorithm based on data mining principles
  - Compared Smartphones vs Laptops.
- Analyzed IP lease traffic behavior
  - Proposed novel IP lease policy that improves address space utilization six-fold and without increasing the overhead.
- Analyzed Web traffic behavior and object caching
  - Demonstrated that a 10MB browser cache can save on average ~5%.
  - Showed the importance of storing policies in proxy caches.
- Designed a “Hybrid” Byte Redundancy Elimination
  - Provides more savings, with less memory space and is x3 faster.
  - Added QoS in redundancy elimination.

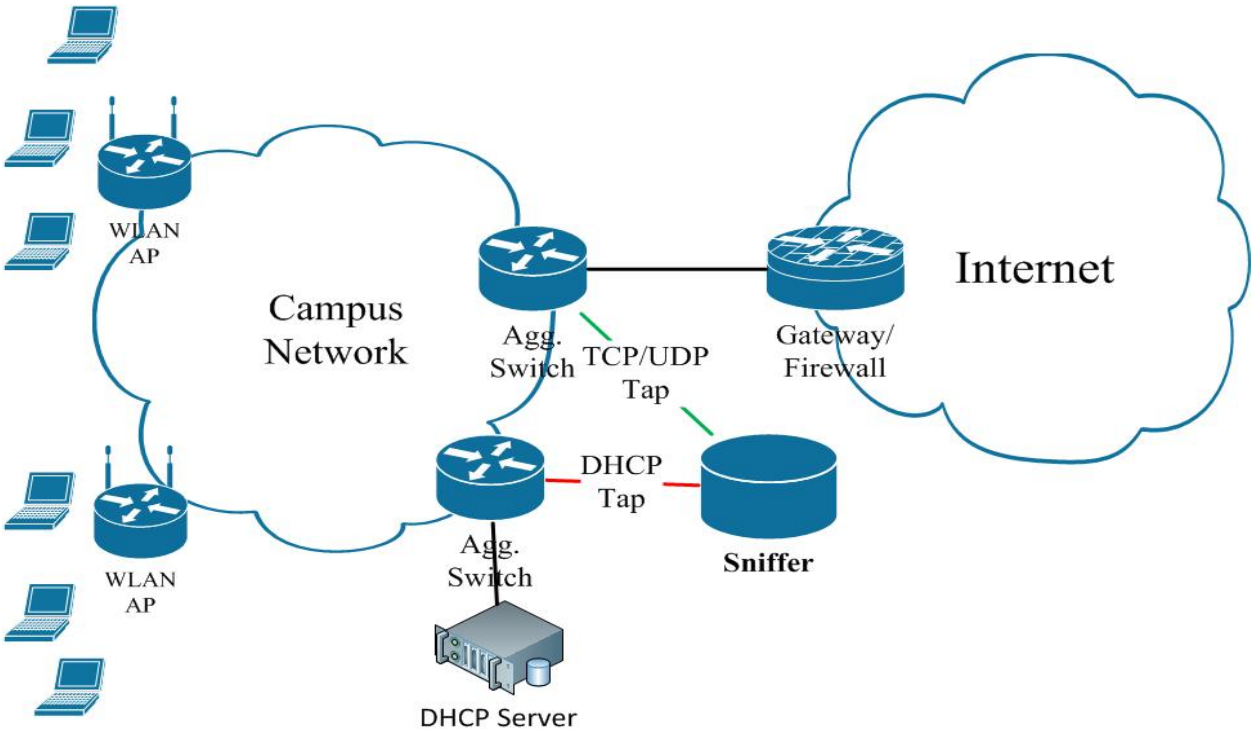
Presentation Outline



Section 1

Smartphones vs Laptops: A traffic study

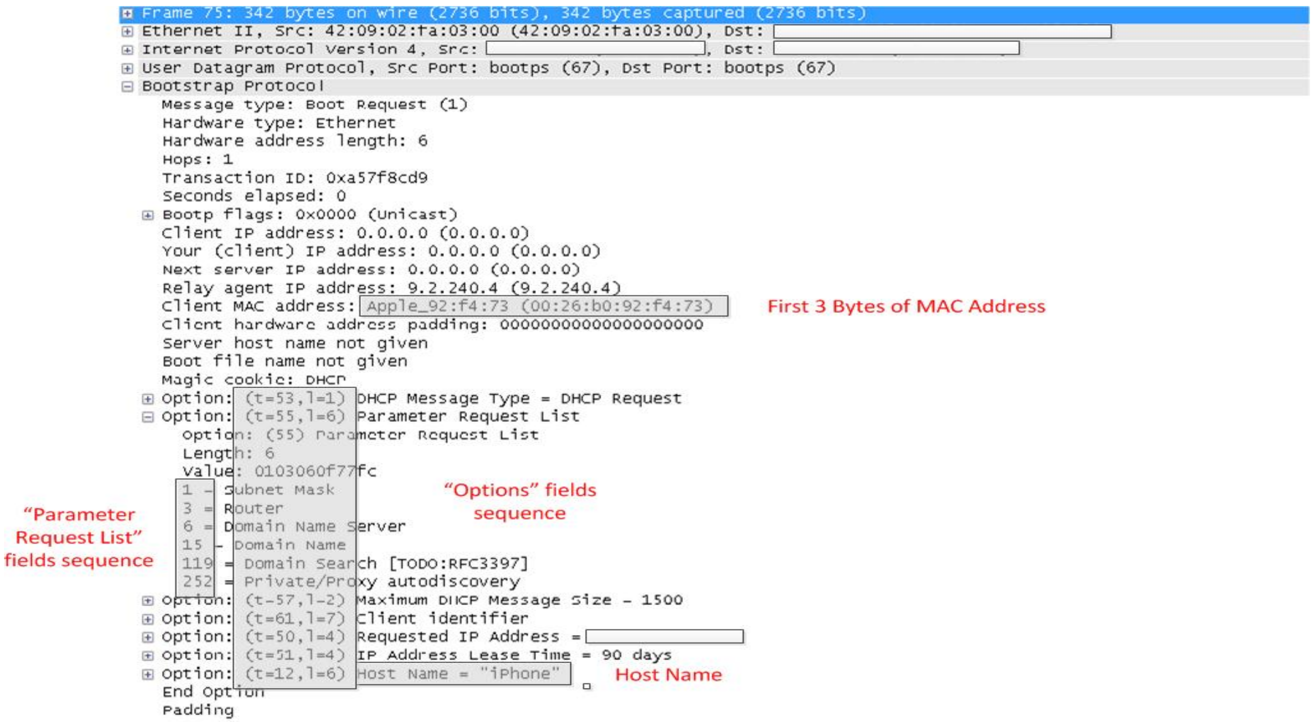
Trace Collection



Trace Type	IBM Research	NCSU Centennial
Dates (2012)	Feb 29 – Mar 25	Jan 15 – Feb 15
Client MAC address	2980	8726
Available IP address	4096 (8 * /23)	2048 (/21)
DHCP Leases	12h	15 min
TCP/UDP Bytes	2.5TB	4.9TB
Software Used	Hacked Bro IDS 2.0	

Device Identification

- We found correlation between the fields of the DHCP Request header and the OS
  1. *Host-Name*
  2. *Vendor-Name*
  3. *Parameter-Request-List*
  4. Organization Unique Identifier (OUI)
  5. *Options* parameter sequence
- Developed a **data mining algorithm** that quantifies the correlation



Classification Outcome

		IBM Research		NCSU Centennial	
Device	OS	#	%	#	%
Laptop	All	**2176**	**73.02**	**3970**	**45.50**
	Windows	1787	59.92	2819	32.31
	Mac OS X	385	12.92	1131	12.96
	Linux	4	0.13	20	0.23
Smartphone	All	**735**	**23.66**	**4489**	**51.44**
	iPhone/iPad/iPod	577	19.36	3069	35.17
	Android	126	4.24	1336	15.29
	BlackBerry	31	1.04	84	0.96
	Windows Mobile	1	0.03	2	0.02
Other	All	**69**	**2.32**	**267**	**3.06**
	Cisco VoIP	9	0.32	-	-
	Unidentified	60	2.01	267	3.06
All		2980	100	8726	100

Simulation: Differentiated Lease Policy

Wrote a trace-driven simulator to evaluate different lease policies:

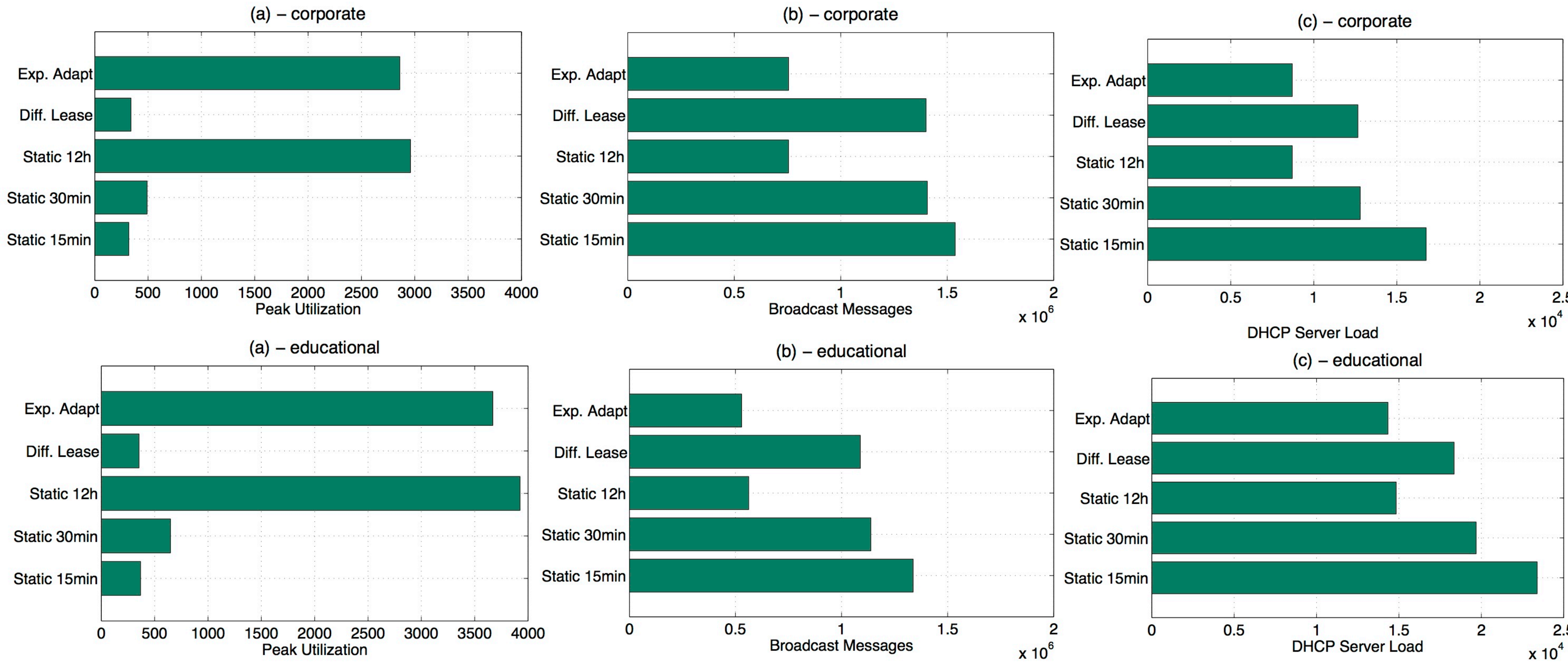
a) **Static Policies**: Fixed lease of 15 min, 30 min or 12 hours (most common case).

b) **Exponential Adaptation**: Allocates a short lease to client once it arrives, and doubles the lease time every time the client renews the lease [GaTech IMC 2007]

c) **Differential Lease:** Allocates different lease values based on device:

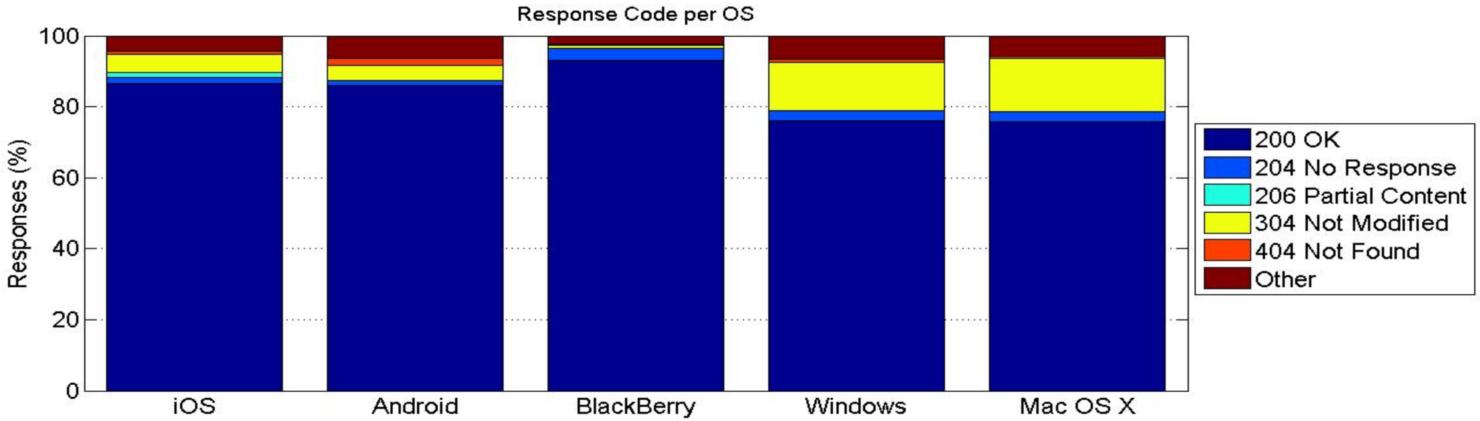
Trace	iOS	Android	RIM	Windows	MAC
Corporate	1000	2000	2000	4000	4000
Educational	500	1000	1000	2000	2000

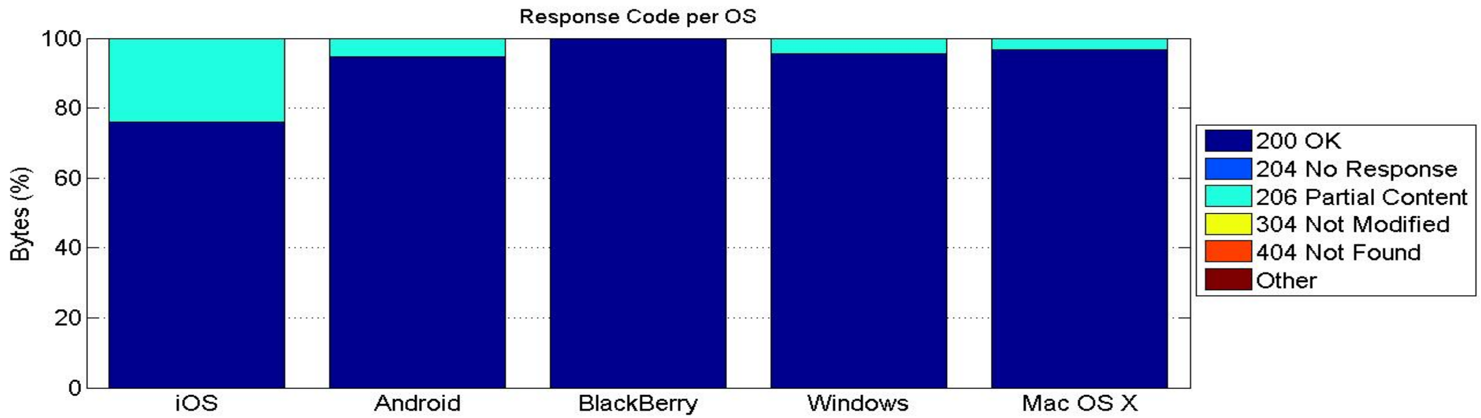
Simulation Results



- Differential lease policy performs well:
  - Achieves low address space utilization (comparable to short lease time)
  - Reasonable number of broadcasts (compared to short lease times)
  - Reasonable server load

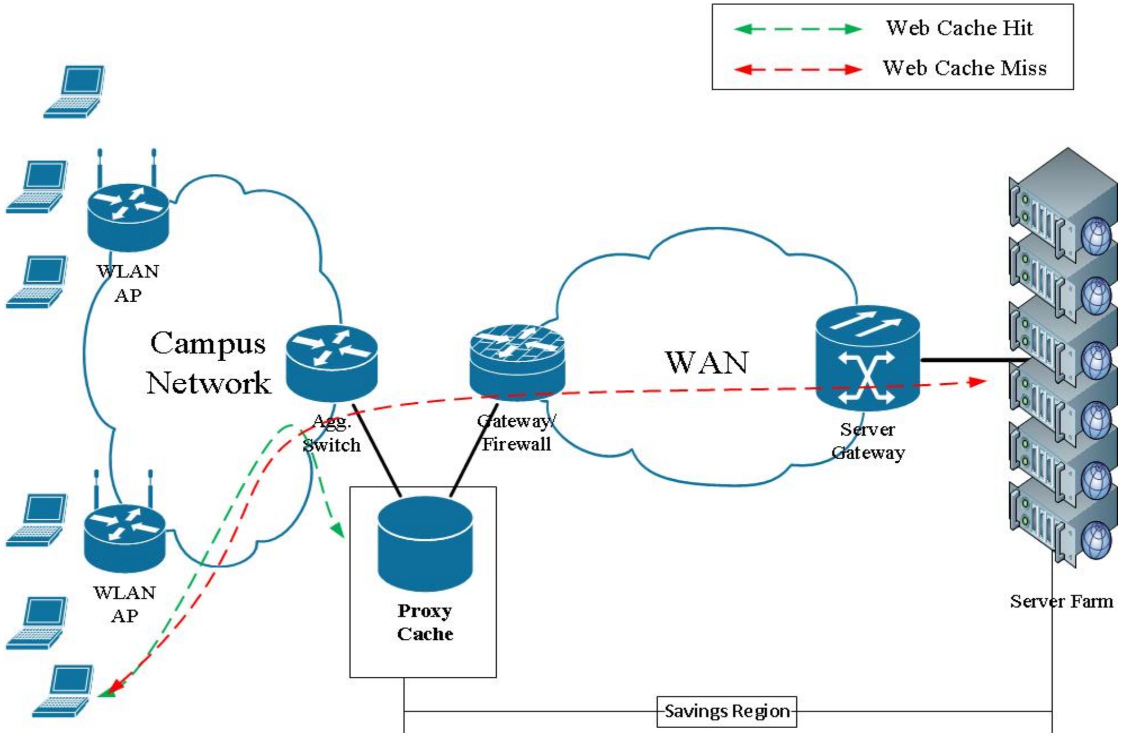
Web Traffic: Response Code





- An HTTP Response code indicates the "status" of the object in the server.
- **304 Not Modified:** validate the client's cached copy on the browser cache.
  - Windows and Mac OS X generate more than smartphones.
- **206 Partial Content:** used for large objects that must be chunked.
  - iOS generate way more than any other OS.

Web Traffic: Proxy Cache Advantages & Metrics



- **Web proxy caches** are used to improve the performance of the Web.
  – Reduce network bandwidth
  – Reduce web server load
  – Improve response time

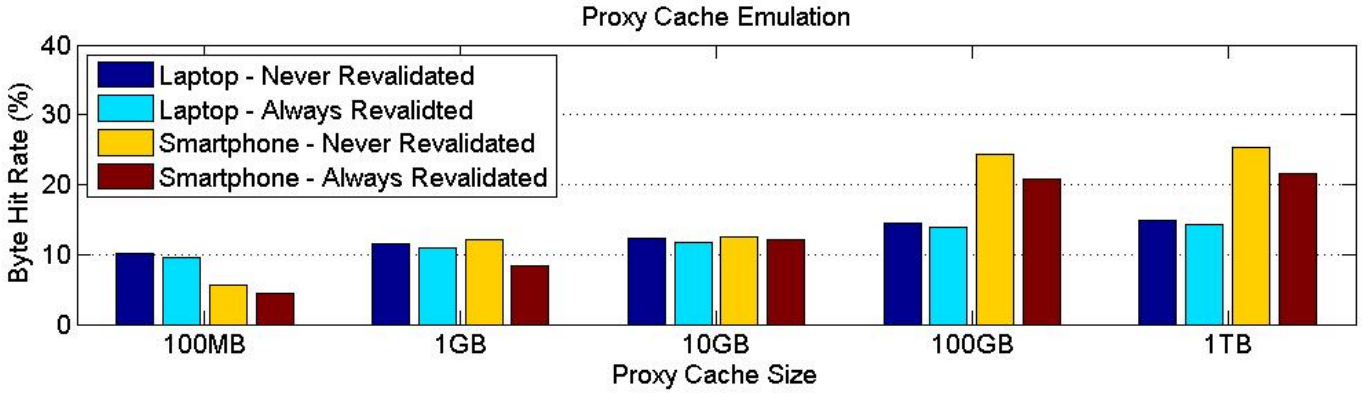
- **Request Hit Rate:** The percentage of all requests that can be satisfied by searching the cache for a copy of the requested object.

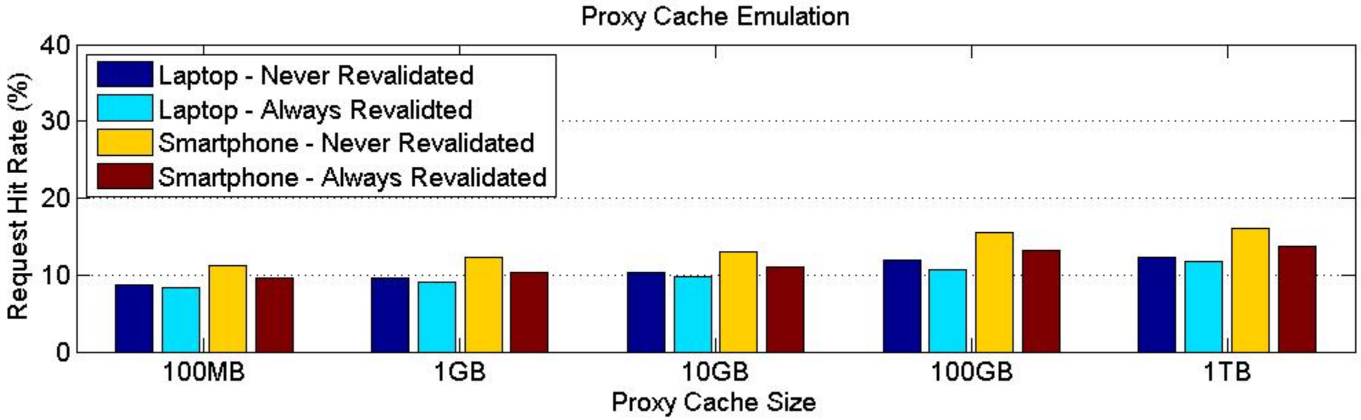
- **Byte Hit Rate:** The percentage of all data that is directly transferred from the cache, rather than the origin server.

Web Traffic: Proxy Cache Size & Freshness

We evaluate the effectiveness based on two configurations for objects that are not fresh in cache:

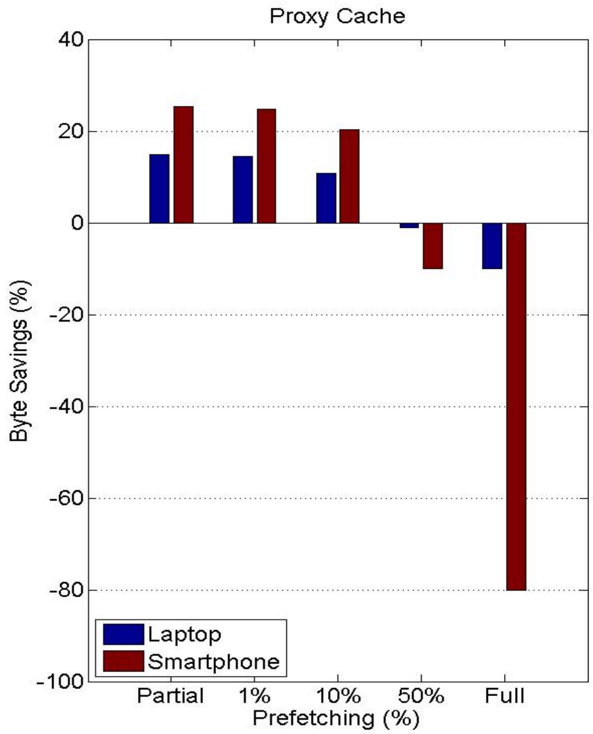
1. *Never Revalidated*: The cached object is the same as the one in the server.
2. *Always Revalidated*: The cached object is different than the one in the server.

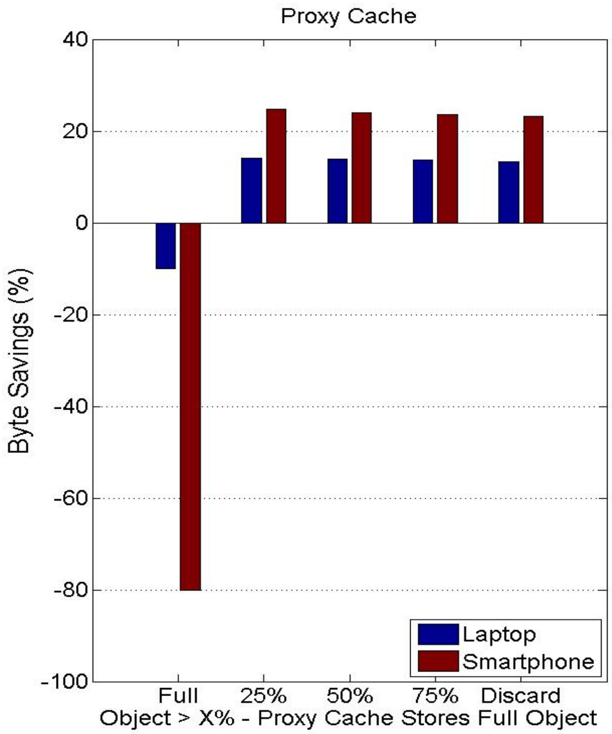




- Diminishing returns ~100GB, hence replacement policies are not relevant any more.
- Savings range from 10-25%, and Request hit rates 10-15%.

Web Traffic: Proxy Cache Storing Policies

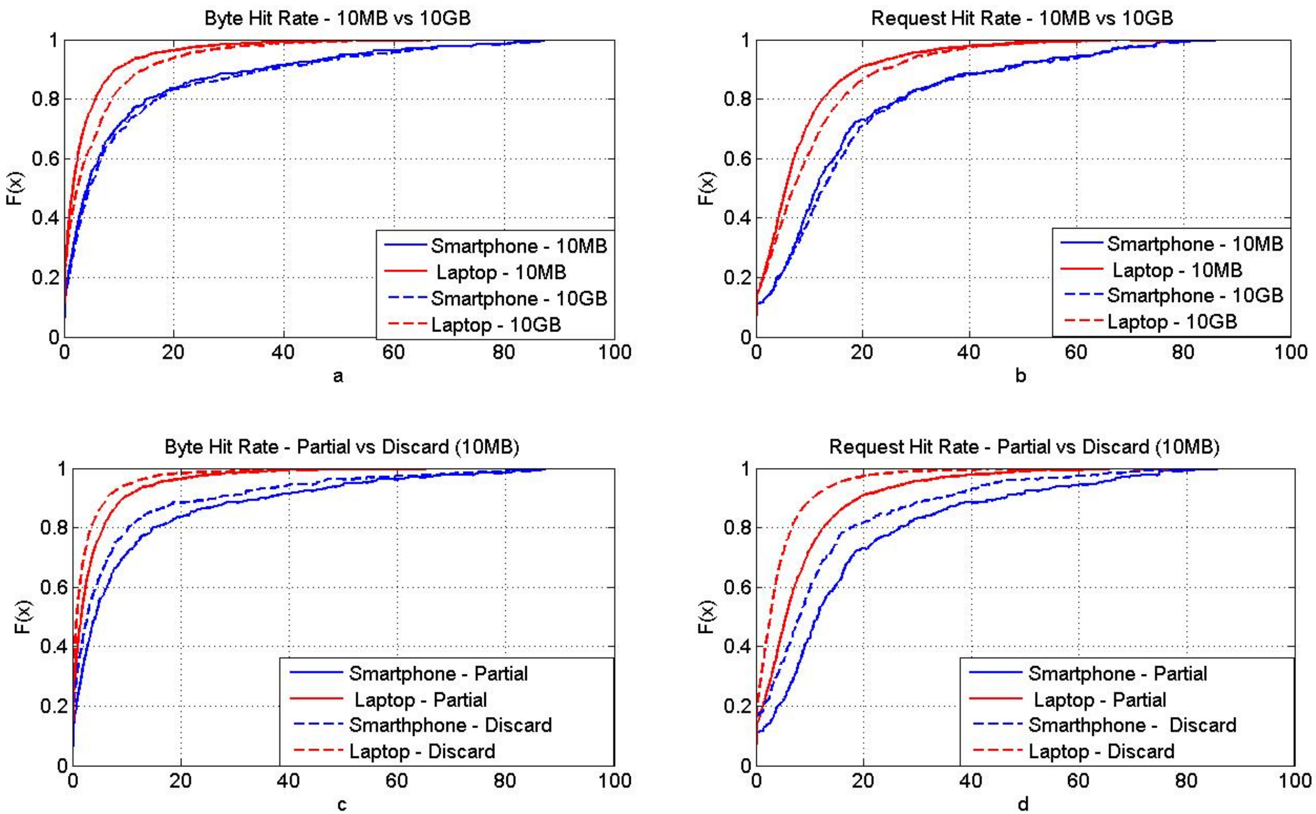




- We evaluate the case where the cache is configured to download X% more data that the data requested by the client.
- **Full download** (cache pre-fetches the whole object) policy results in abusive BW demands in smartphones.
- **Partial storing** (cache stores only what user requested) is the optimum policy.
- A partially downloaded object is cached in its entirety if more than X% of its size has been downloaded by the client.
- 0% corresponds to the **Full download** policy.
- **Discarding** (any partially downloaded object is not cached) is more beneficial than other conditional download policies.

Web Traffic: Browser Cache Efficiency

We replay the IBM trace for each device through an additional simulated cache dedicated solely to that device:



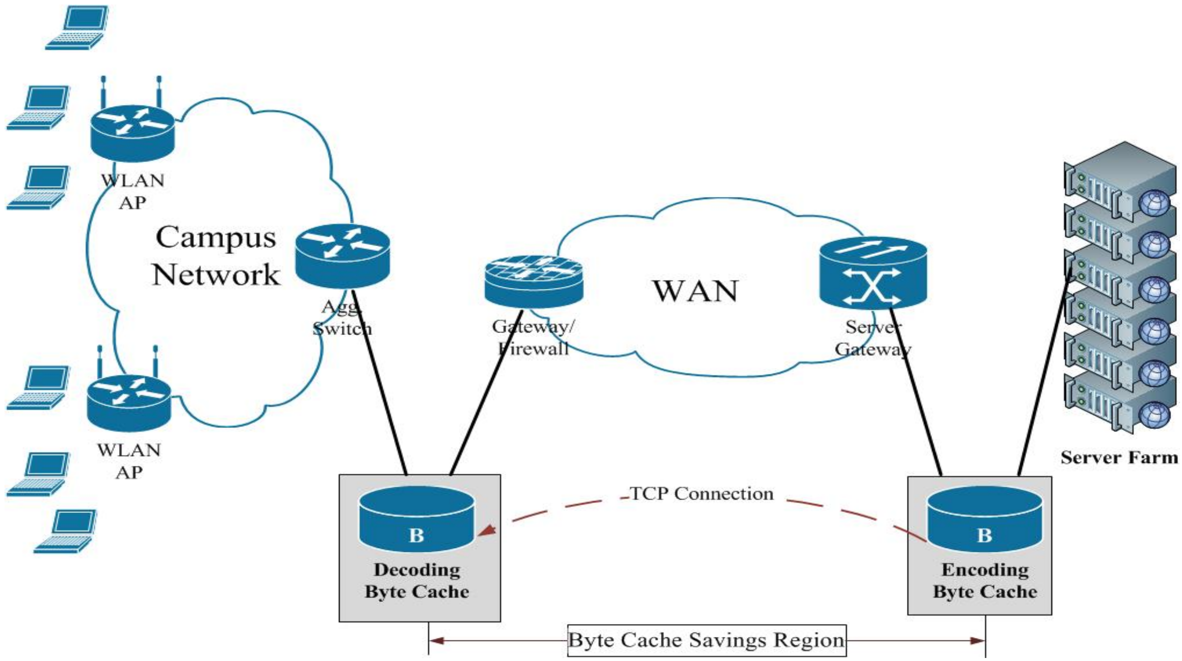
- Current laptop browser caches are more efficient than the smartphone ones.
- A small browser cache (10MB) is sufficient to capture most of the savings in smartphones.
- A browser cache should be able to handle partially downloaded objects.

Section 2

Byte Redundancy Elimination in Wireless Networks

Proxy Caches:
- Limited savings: Partially modified objects are treated as different ones.
- Almost 60% of the Web data are uncacheable (based on RFC2616).
- From the cacheable objects, almost 70% may be referenced only once.
- Object revalidation may result in fetching the object from the server.

Byte Cache: Introduction



A Byte Cache is a system of devices that can identify redundancy at the byte level.

- Requires two middleware boxes (encoder & decoder).
- Uses Rabin fingerprinting to generate the hashes.
- Hashes are then used as indices to the content, or to break the content in chunks.

Fingerprint size = 4

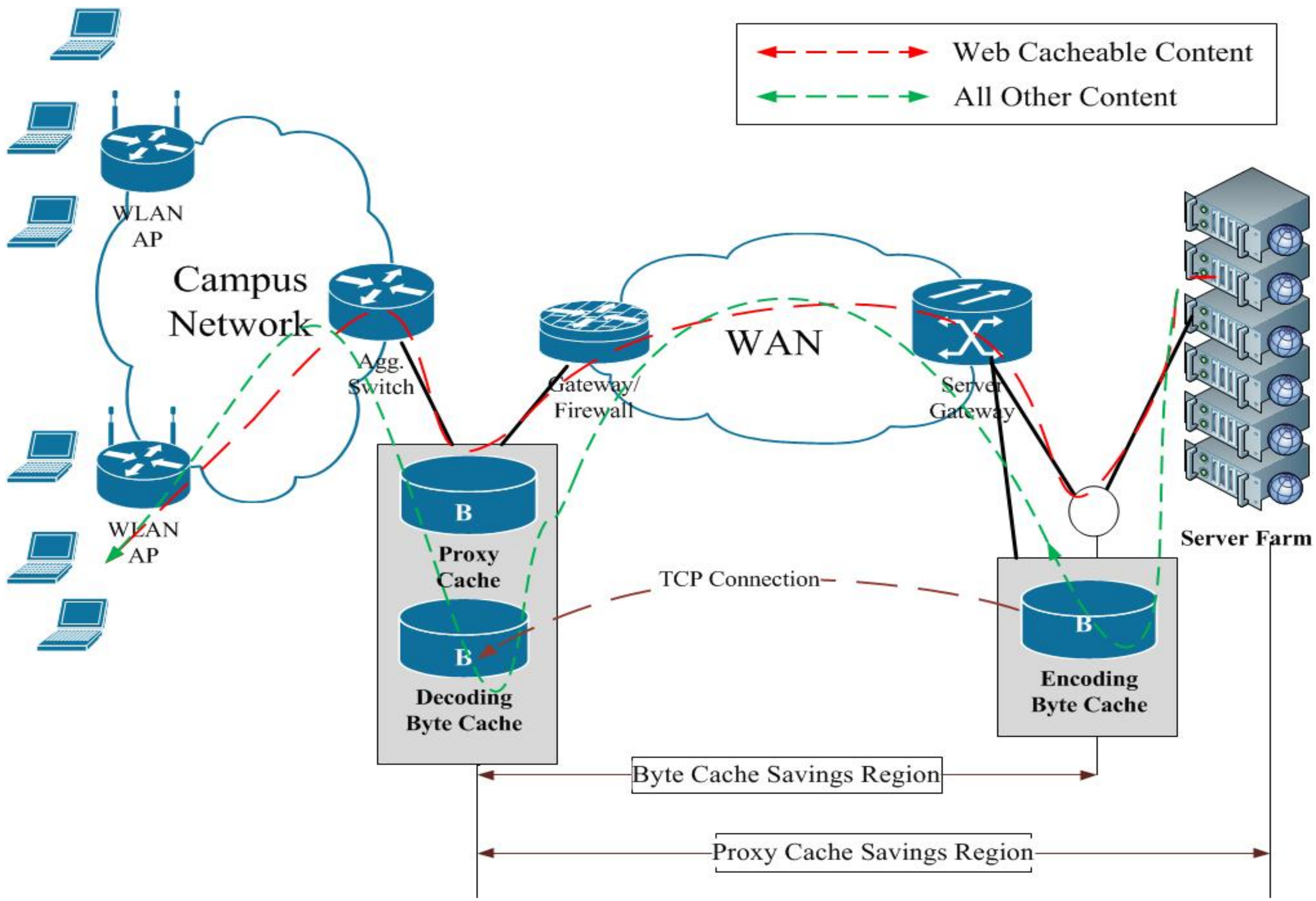
HDD	RAM
Content	Hash
ABCDFEFGHIJ	H(ABCD) H(BCDE) H(CDEF) H(DEFG) H(EFGH) H(FGHI) H(GHIJ)

Packet 1
srcIP: 15.2.1.1
dstIP: 150.150.1.1
Payload: ABCDEFGHIJ

Byte Caches: Issues

- Fingerprint generation is CPU intensive.
  - Can be a bottleneck in high-bandwidth WAN links.
- Requires a lot of memory space.
  - A hash index is required per object in proxy caches vs. per 32B chunks in DRE systems
- Fingerprint generation and chunk storage may be useless for some protocols.
  - E.g. Encrypted traffic.

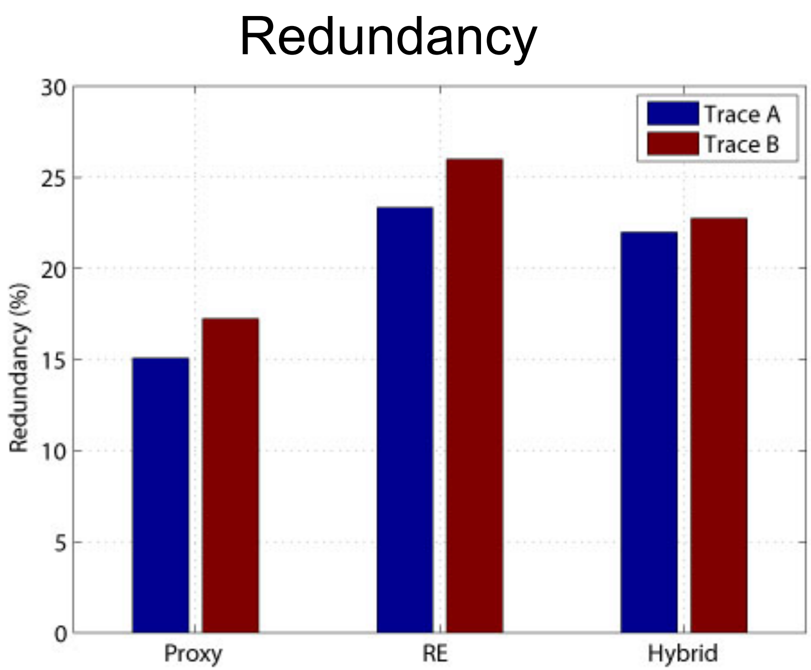
Hybrid Byte Cache: Static Scheduler

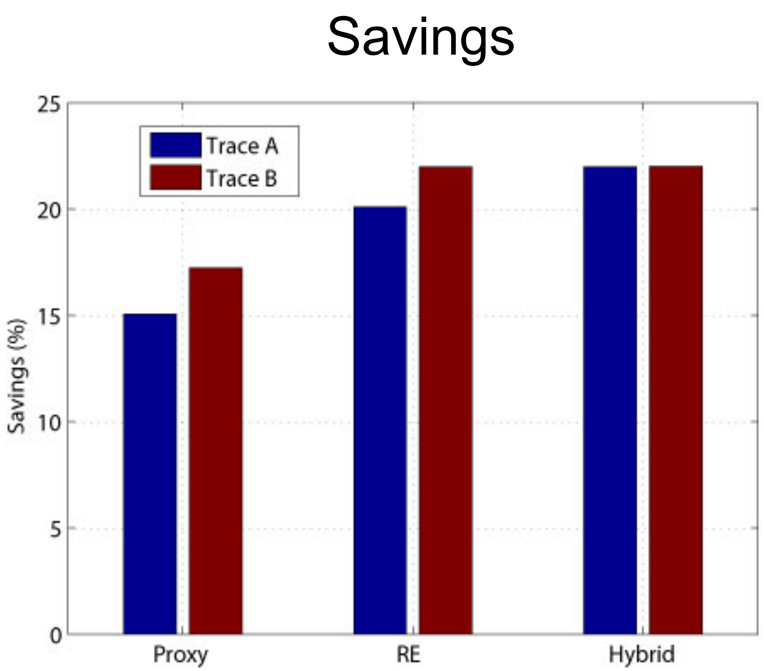


- Scheduler:
  - Web cacheable content is forwarded to a **Proxy Cache** Module
  - Web uncacheable content is forwarded to an **Encoding Byte Cache** Module
  - Other data are sent unprocessed

Hybrid Byte Cache: Redundancy & Savings

- Full packet traces captured in IBM (Trace A: 19GB, Trace B: 64GB)
  - Both traces consist of 80% TCP and 75% of Web traffic.



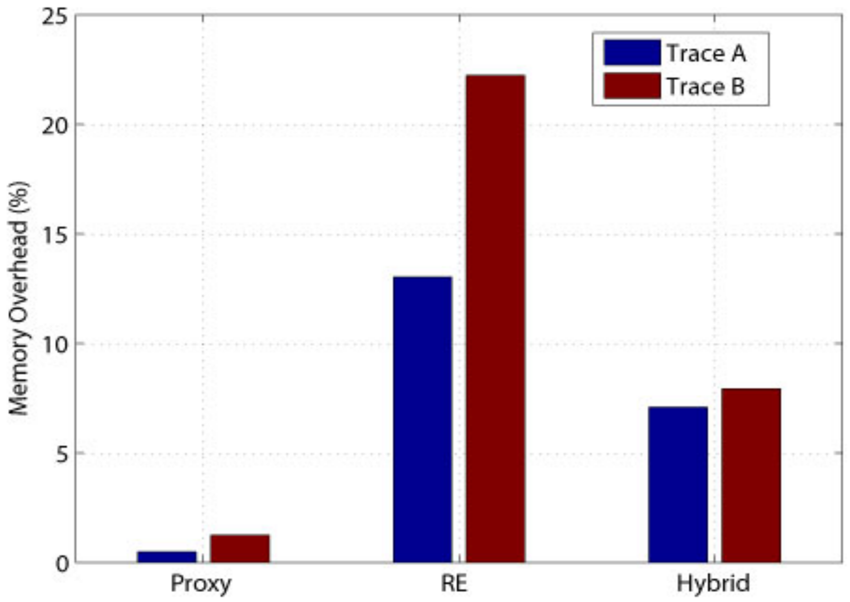


- Redundancy in DRE is better than in Proxy or Hybrid DRE
  - Note that redundancy includes the hash overhead.
  - Providers are mostly interested for Savings.

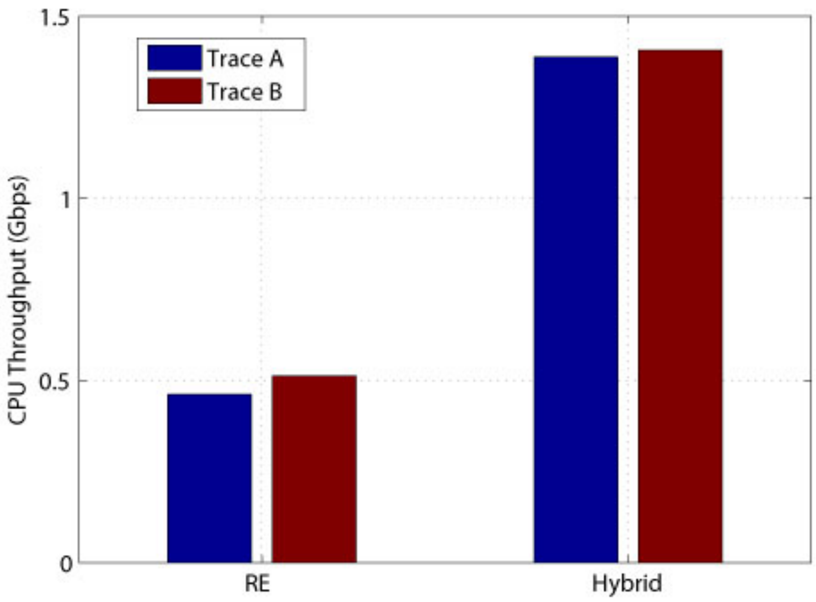
- Hybrid DRE provides more savings than Proxy or DRE.

Hybrid Byte Cache: System Performance

Memory requirement



CPU Throughput



- "Memory Overhead" indicates the amount of RAM needed to store the hashes that correspond to the content.
- Hybrid DRE requires more than half the memory compared to DRE.
- Hybrid DRE increases throughput almost 3 times compared to DRE.
  - Therefore, Hybrid DRE can be installed in faster links.

Conclusion

Summary

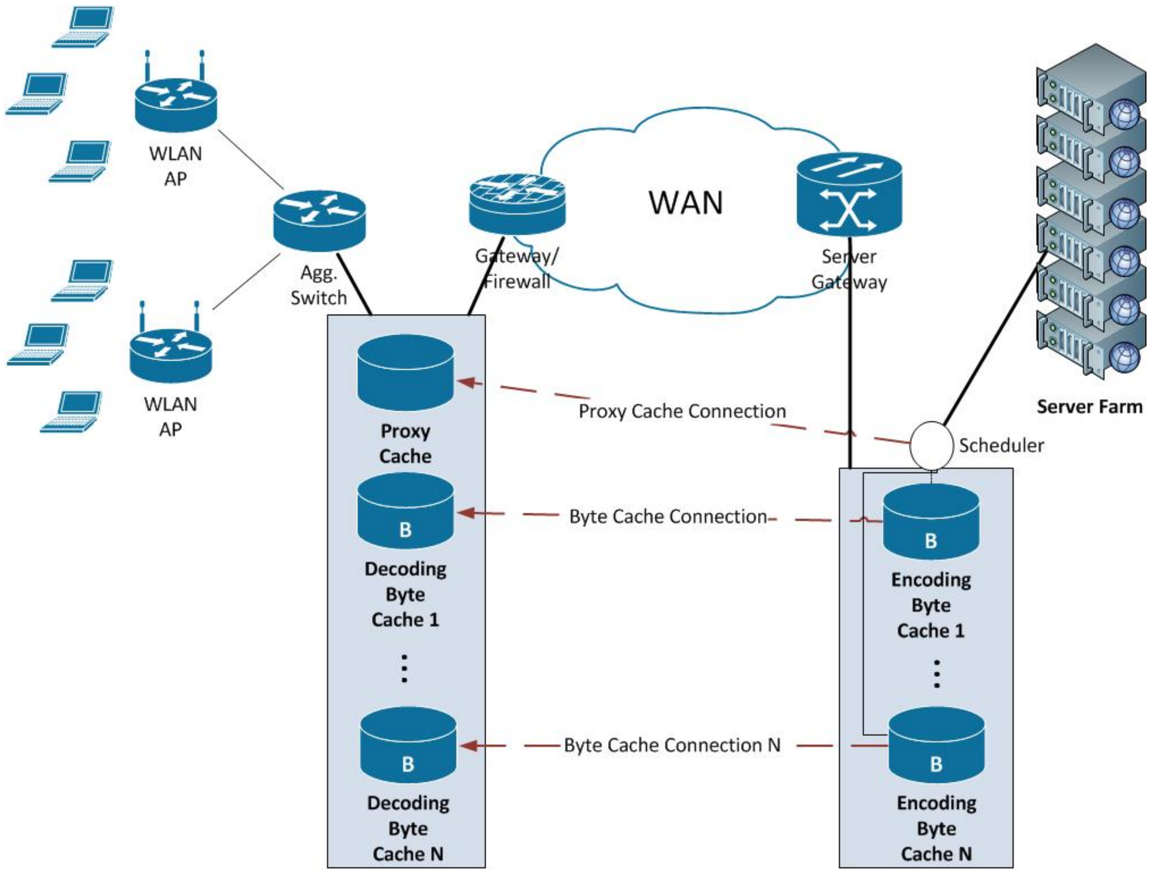
- Differentiated DHCP Lease policy should be implemented in wireless networks.
- A Smartphone vendor should consider adding an extra **10MB** of Browser Cache:
  - ~5% of savings.
- A **100GB,** properly configured proxy cache can provide:
  - 25% of savings for smartphones.
  - 15% of savings in a wireless network.
- A Byte Cache system may provide more savings than a proxy.
- The proposed Hybrid DRE leverages the advantages of both the Proxy Cache and the Byte Cache:
  - Better savings
  - Less CPU cycles
  - Less memory

Future Work at Purdue University

Research Plan

- Collaborations
  - IBM Research, Cisco Systems and Time Warner
  - North Carolina State University and University of Michigan
    - NSF proposals
  - European Universities
- Research Areas
  - Data analytics in mobile networks
  - Byte level deduplication
  - Performance analysis & simulations
  - Elasticity in Cloud Computing
  - Applications to the Smart Grid
- Extend Intellectual Property

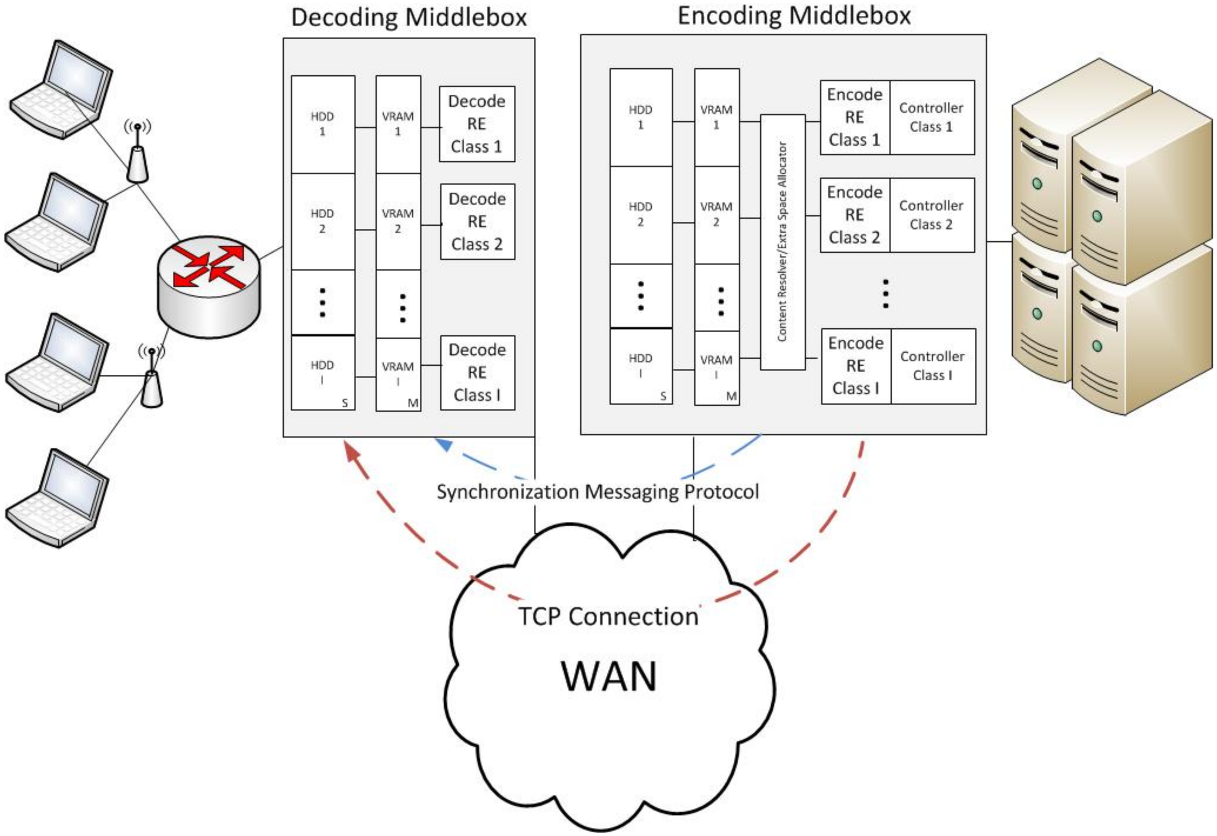
Future Work #1: Dynamic Hybrid DRE



Dynamic Scheduler:

- Based on machine learning, predicts the best caching module for each traffic flow.
- Optimized to take into account the savings and system performance.
- Decreases CPU usage, benefits from parallel processing, increases savings and system failover.

Future Work #2: Quality of Service in Data Deduplication



- Users are differentiated in class categories:
  - Each class processes the flows with a different set of parameters.
- We introduce a controlling mechanism that dynamically allocates memory and space:
  - Per-class controllers monitor the performance and send messages to the central controller such that the SLAs are met.

Thank you!