Data Management in Pervasive Computing Environments

by

Manish Gupta

Mtech-1 student

for

CS632 Course Seminar

Department of Computer Science and Engineering,

Indian Institute of Technology,Bombay

Mumbai

Motivation

- Mobile devices – an inseparable part of life
- cellphones, PDAs ,bluetooth devices etc.
- size and cost are reducing
- functionality is increasing
- networking technologies are improving
- wireless communication needs mobile data management

Client-Proxy-Server Model

- Mobiles are clients.
- Servers are on the wired world
- Important issues are disconnection management,low bandwidth, and device resource constraints.
- extremely lightweight client database
- Client has a (partial) replica of the main database on the wired side or selected data is continuously broadcast into the environment.
- A new peer-peer model is needed for spontaneous interaction and to prevent server bottleneck.
- e.g.Consider a car in search of petrol

Peer-Peer model

- Resource-rich, data-intensive environment
- Users and devices, including handhelds, wearables etc are mobile and continuously exchange data
- Semi-autonomous, self-describing, highly interactive and adaptive peers
- Cross-layer interaction between their data management and communication layers
- Infer and express information
- Obtain and store such information pro-actively

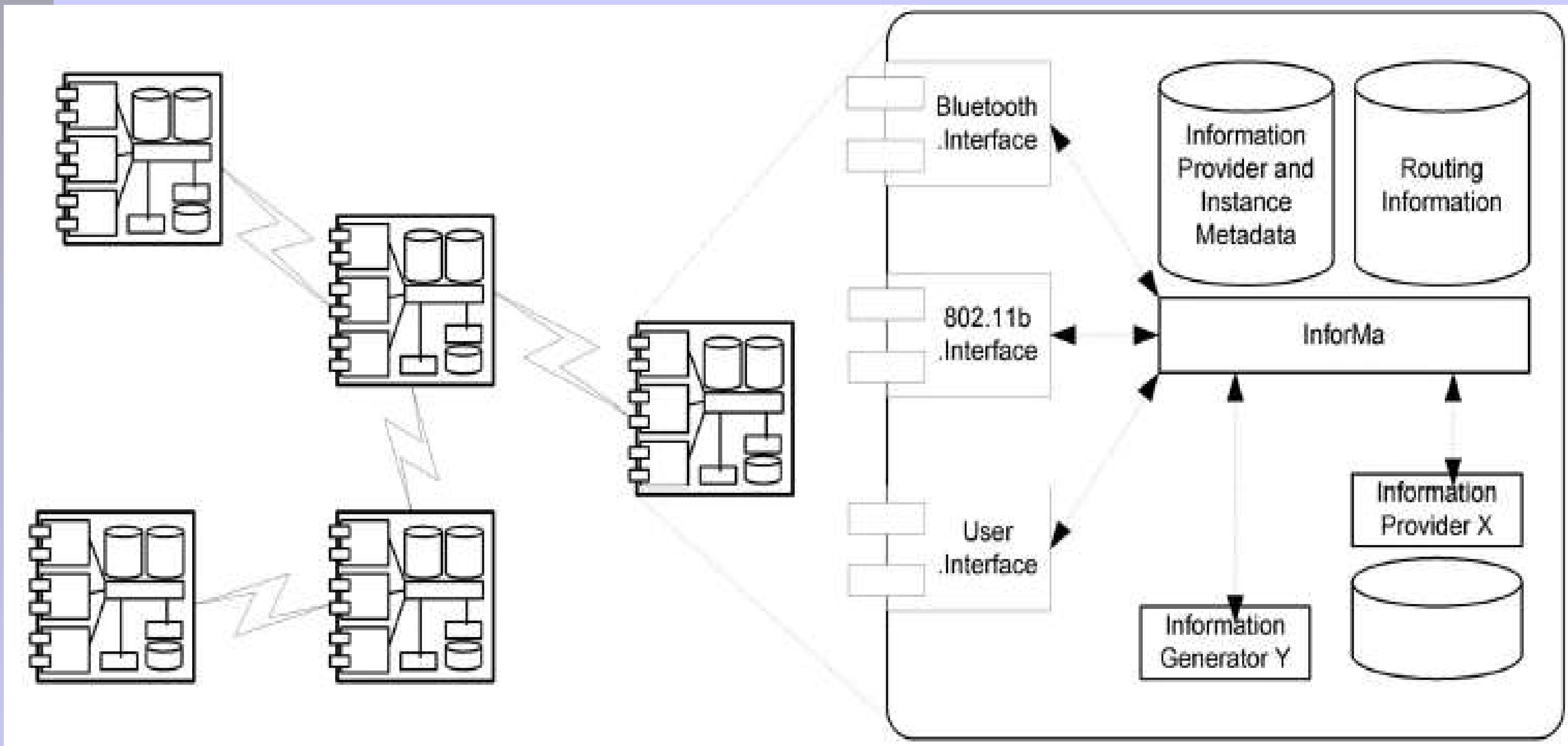
Motivating example

- Person A has a meeting scheduled at a shopping mall with a person B.
- A notifies his device about this new appointment.
- Getting the way to the mall(visual displays)
- Dynamic route decision
- Searching for objects at the mall
- Notifying about nearest restaurant of choice

Information Providers, Consumers, and Managers

- Information Providers - hold partial set of heterogeneous data annotated in a semantic language.
- Information Consumers - query and update data.
- Information manager
  - Underlying network communication and most of the data management functions.
  - info about types of devices in vicinity and info they can provide.
  - data cache
  - user profile reflecting some of the user's beliefs, desires, and intentions(BDI)
    - Caching strategy adaption
    - Initiate a collaboration with peers
    - Describe content of data objects

Entity details and interaction



Features of a pervasive model

- **autonomy, distribution, heterogeneity, and mobility**
- autonomous - no centralized control of the individual client databases.
- heterogeneous - in terms of devices, data resources, and networking technologies. We assume that entities can speak to each other in some neutral format.
- distributed - parts of data objects on different devices and there is replication as entities cache data
- mobility - devices can change their locations and no fixed set of entities is always accessible to a given device.

Data Management Challenges

- Spatio-temporal variation of data and data source availability. So cache the data.
- Lack of a global catalog and schema.
  Each device describes and optionally advertises its capabilities to its neighbors using heterogeneous ontologies, defined via a common vocabulary encoded in a Semantic Web language. This allows every device to dynamically construct a subset of the local catalog.
- No guarantee of reconnection. Best-effort service only can be guaranteed.
- No guarantee of collaboration - privacy and trust.Best-effort service only can be guaranteed.

Consequence of the challenges

- Serendipitous query answering
- Proactive info gathering in background
- Adaptation to the needs and preferences of users and the current context.
- Determine what data to obtain proactively and its relative worth by
  - User preferences and needs
  - Context
  - Battery power and storage space

Information Providers

- Every device may hold one or more Information Providers.
- Interface to a possibly inconsistent and empty subset of the global data repository.
- Describes its capabilities in terms of ontologies defined in a semantically rich language. e.g. DAML+OIL
- Registration message
  reg(service model s, process models p, and input restrictions I, lifetime t it will be available,willingness to process queries for remote devices a).
- Local InforMa routes its messages, cached data and advertises its services

Information Consumers and InforMa

- Consumers register and get queries resolved with local InforMa.
- Information Manager (InforMa)functions
  - From the data management perspective
    - Discover available sources
    - construct dynamic indexes and catalogs
    - support queries
    - provide caching mechanisms for addressing the dynamic nature of the environments.
  - From the networking perspective
    - Discover devices, interact with them, and route messages.
    - Maintain information about Providers

Types of Info Managers

- Type 0 - Maintain only one local Provider. No remote info caching, no reasoning, no parsing mechanisms. It periodically broadcasts data sent to it by the Provider. Good for resource limited devices/ads, such as petrol pumps, bus stations.
- Four types of InforMa based on the collaboration level:
  - 1. InforMa does not cache any remote ads or answers to queries.
  - 2. InforMa caches remote advertisements only for the lifetime specified in the message or until replaced by another entry.
  - 3. InforMa caches both advertisements and answers.
  - 4. InforMa also caches all advertisements/answers and makes them available to other peers.

Info query processing

- Consumer supplies (service model, input parameters)
- InforMa parses the query using DAML+OIL rules
- InforMa checks for cached answers and returns 1 if available.
- Else InforMa asks the local providers that provide the service and can accept the input parameters and returns answer returned by them.
- Else if Consumer is local, InforMa finds an appropriate remote provider and routes the query to it.

Caching and BDI model

- Each InforMa stores query answers together with ads and registrations of local and remote Providers in a cache.
- InforMa uses Beliefs, Desires, and Intentions model.
- Beliefs = query restrictions that are assigned utility and reliability functions which use current time, location, and other info in the profile as inputs to calculate importance value.
- Intention = a standing query with a starting point and time period during which the InforMa attempts to obtain and cache an answer. The standing query also has utility and reliability functions in order to prioritize the InforMa's activity.
- Desires = user's wishes that cannot be directly converted into standing queries but instead require additional user-provided rules.

Using profiles for caching

- For caching, the InforMa uses the profile in two ways:
  1) to preallocate space for specific data type
  2) to assign utility value to each entry
- First heuristic leads to LRU+P and MRU+P algos
- Choose e_i as victim for new entry n of type D if

$$e_i = \emptyset$$

$$\text{cache full} \wedge \mid E_D \mid \geq max_D \wedge e_i = \text{LRU}(\forall e \mid e \in E_D)$$

$$\text{cache full} \wedge \mid E_D \mid < max_D \wedge e_i =$$

$$\text{LRU}(\forall e \mid e \in E_C \wedge C \neq D \wedge \mid E_C \mid > max_C).$$

Using both hueristics

- S+P employing both heuristics
- Choose v to make space for page n of type D if

$$v = \begin{cases} e_i & e_i = \emptyset \\ e_i & \text{full} \wedge \mid E_D \mid \geq max_D \wedge e_i \in E_D \wedge U(e_i) = \\ & U_{min}(\forall e \mid e \in E_D) \wedge U(n) > U(e_i) \\ e_i & \text{full} \wedge \mid E_D \mid < max_D \wedge e_i \notin E_D \wedge \\ & U(e_i) = U_{min}(\forall e \mid e \in E_C \wedge C \neq D \wedge \mid E_C \mid > max_C) \\ & \wedge\, U(n) > U(e_i) \\ \emptyset & \text{otherwise.} \end{cases}$$

Discovery and Routing

- Both push and pull-based approaches-- each InforMa can advertise its capabilities or solicit capabilities of other peers limited to 1 hop neighbors
- Each InforMa intercepts all messages it receives or routes in order to provide shortcuts for cached routes. It also maintains entries for 1-hop away peers.
- InforMa_route_query(src; dst; query; route; hopcount; intercept?)

```
  if(dst=local Provider) find the cached answer or ask local provider.
  else if(dst=remote Provider)
      if(src=local consumer) route<- calculate the route to dst
      if(intercept=yes) find answer in cache and return if valid
      increment hopcount
      if(willing to forward and hopcount<=maxhops)
            n<- next hop(dst,route)
            if( n=null and src is local consumer)  n=broadcast
            if(n!=null) forwardto(n,src,dst,query,route,hopcount) and
                 return
  return error.
```

Metrics for performance experiments

- For experiments, we either vary or measure these parameters:
  - Query Frequency - rate at which a user asks the device an identical question. Two frequencies
    - 1) unlimited(Single Queries)
    - 2) 5 minutes(Repeating Queries)
  - Cache Size.
  - Cache Replacement Algorithm. - LRU, MRU, profile-based LRU+P and MRU+P, and the semantic S+P
  - Cache Allocation - eight information types. Compare the results obtained for each algorithm a to an optimal case o using a cosine similarity A.
  - Cache Hit Success Rate.
  - Cache Update Period related to battery life.
  - Peer Query Response Time.

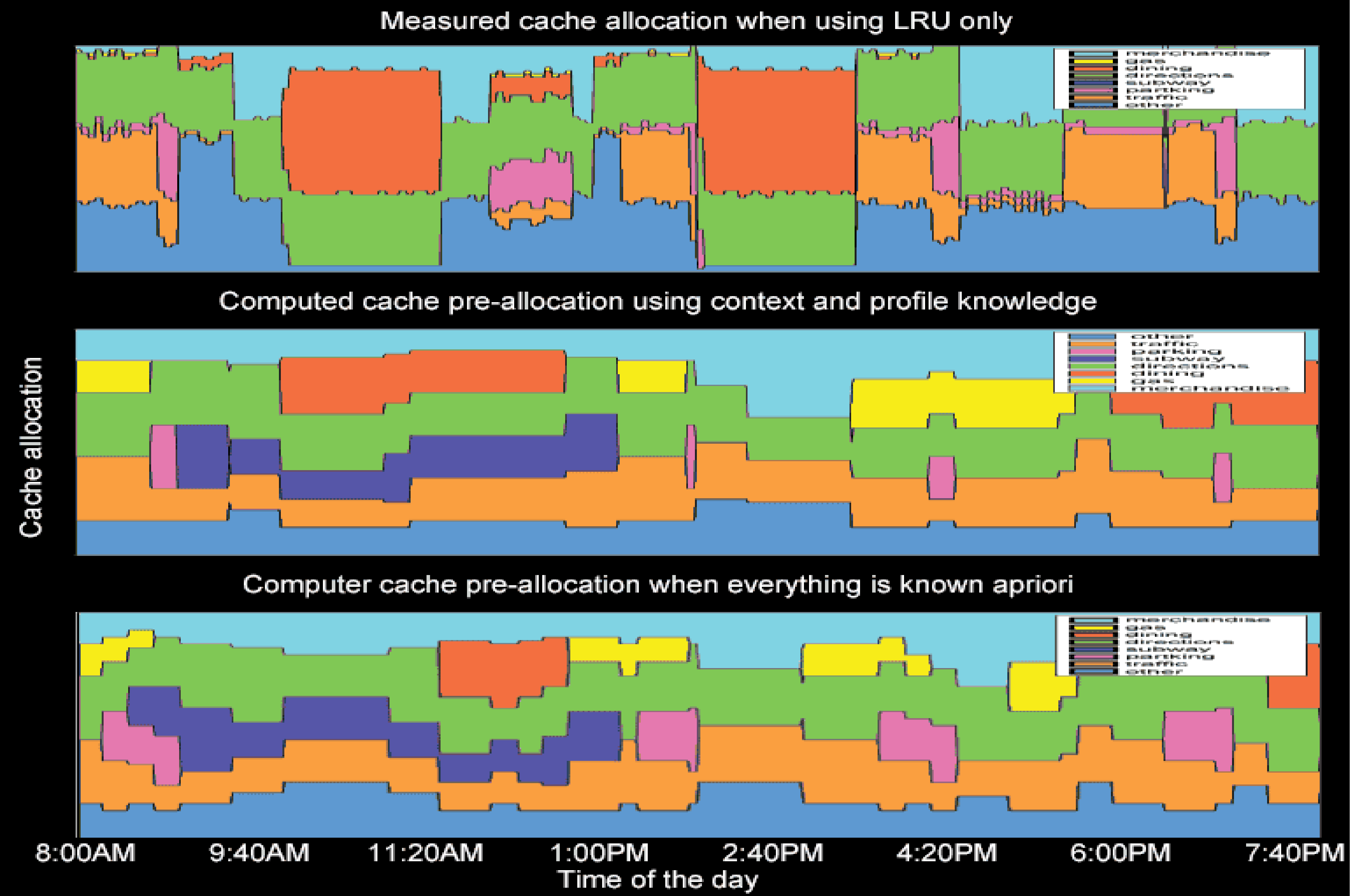
Experimental Setup

- 12-hour period in which person A travels between three distinct cities to attend meetings.
- Meetings are scheduled a priori or scheduled as the day progresses.
- Device has A's profile that can be converted into query restrictions and standing queries for the eight different types of information.
- Information providers are distributed with different types of information along the travelled path. These providers simulate cars, electronic lights on office buildings, petrol pumps, subway infrastructure, and other people's mobile devices.

Studies

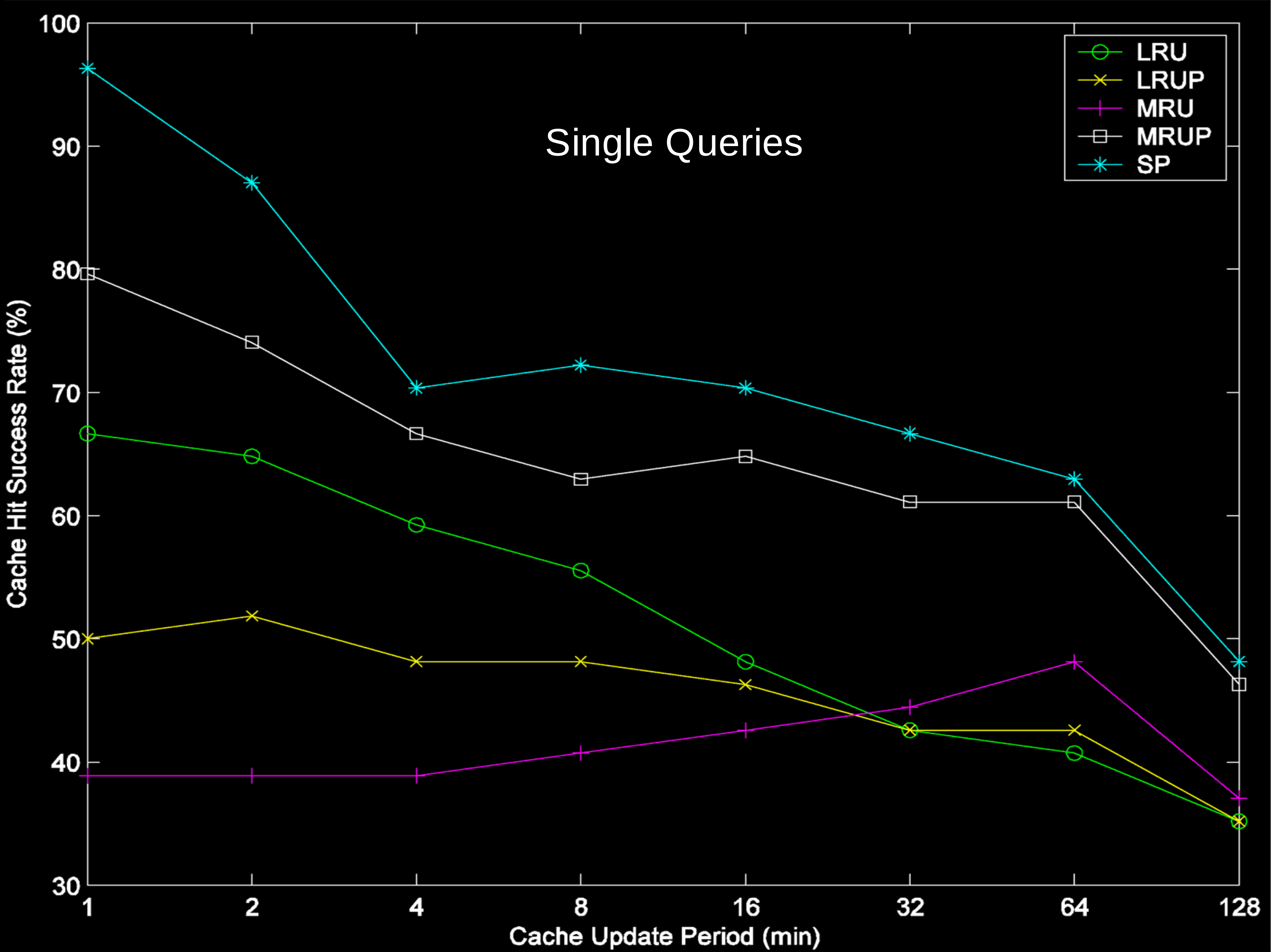
- Effects of:
  - Cache Replacement Algorithms versus Cache Allocation
  - Cache Update Rate versus Cache Hit Success Rate for Single Queries
  - Cache Update Rate versus Cache Hit Success Rate for Repeating Queries
  - Networking Technology versus System Performance

Cache Replacement Algos vs Cache Allocation

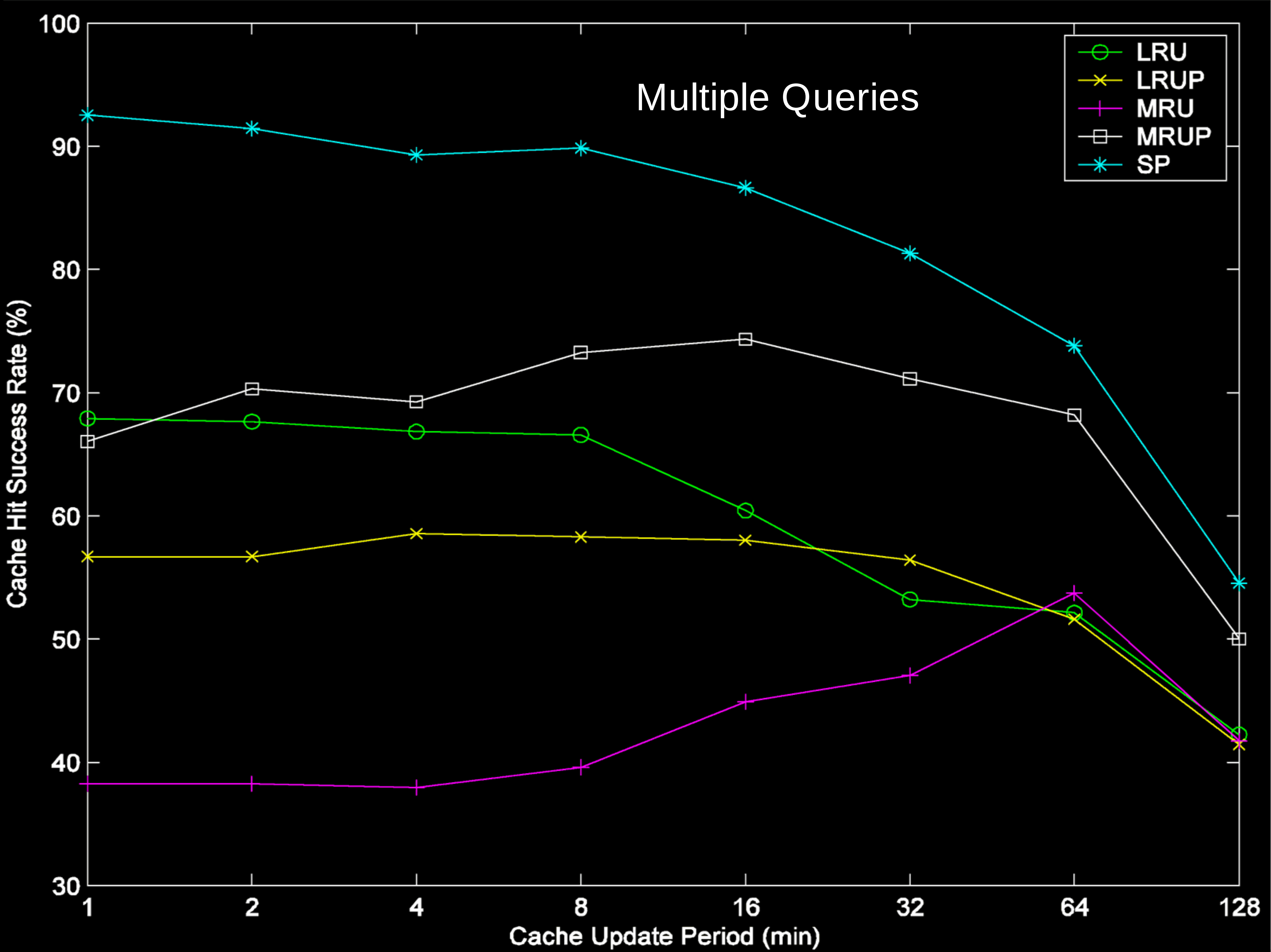


Cache Update Rate vs Cache Hit Success Rate for Single and Multiple Queries

- How does each cache replacement algorithm perform given varying update periods for single and repeated queries?
- Update period(1-128 minutes)
  - Device's preferred refresh rate to prolong battery life
  - A rate at which info providers appear/disappear
  - Most data has 10-minute lifetime
- S+P rules but drops the most
- Stable Region of operation-- 4 to 16 minutes
- Every algo has a chance of performing better for repeating queries



Single Queries
LRU
LRUP
MRU
MRUP
SP
Cache Hit Success Rate (%)
Cache Update Period (min)
100
90
80
70
60
50
40
30
1
2
4
8
16
32
64
128



Multiple Queries
LRU
LRUP
MRU
MRUP
SP
Cache Hit Success Rate (%)
Cache Update Period (min)
100
90
80
70
60
50
40
30
1
2
4
8
16
32
64
128

Networking Technology versus System Performance

- Half-duplex interaction with connect-send-disconnect mode.
- Each exchanged message has an average size of 1.0KB.
- Four laptops interacting over a period of 100 minutes.
- Evaluate the system by randomly selecting a query and assigning it to one of the four devices and monitoring information present at each cache and their routing capabilities.
- Study the impact on performance for reasoning over cache entries by each InforMa compared to transmission time.
  - For a 30KB cache, the processing time was, on average, 5ms per query after 100 runs.
  - In Bluetooth environments, it takes 4.56s to transfer a 1.0KB query and to send a response.
- High mobility?? - Bluetooth is slow.

Conclusions

- Mobile devices, which are getting constantly enhanced, will become both sources and consumers of information and will be able to cooperate with other devices in their vicinity in order to pursue their individual and collective tasks.
- Data management challenges that need to be dealt with e.g.data and data source availability, lack of a global catalog and schema, no guarantee of reconnection among peers, no guarantee of collaboration among peers, and the issues of commits and aborts due to the serendipitous nature of the environment.
- We discussed a framework that is peer based, operates on a best-effort basis, is network agnostic and uses both static and dynamic information to allow devices to behave proactively.

References

- On Data Management in Pervasive Computing Environments --Filip Perich, Anupam Joshi, Senior Member, IEEE, Timothy Finin, and Yelena Yesha, Senior Member, IEEE
- On Peer-to-Peer Data Management in Pervasive Computing Environments--Yelena Yesha,University of Maryland Baltimore County

Thank you