CSE 535 Mobile Computing  2/15/05

Maintain Strong Cache Consistency in Mobile (Disconnected) Environment.

Broadcasting Timestamp.
- push based
- sending periodic invalidation (reports)
  - synchronous.
- answering queries (access to cached data item) at a client were delayed till the arrival next Invalid Report.
  - this increases the latency of queries.
What was the mechanism used?

Cost of increasing $\omega$ is increase in size of Invalidation Reports. $\Rightarrow$ downtime B/W increased.
As Scheme

- Asynchronous: Invalidation are "pushed" as soon as they are available.
  Queries need not be delayed.
- Stateful: "State" of the Client's cache is maintained at "Home Agent."

  - Home Location Cache
    \[(x, T, Flag)\]
    \[
    \text{Valid / Invalid}
    \]

- Mobile Client
  - Maintains a "cache timestamp" which is the timestamp of last invalidation it received.
  - Assumes that the client can determine when it is disconnected from the R/G.
Are we losing adaptability since the client now needs to be aware of disconnections?
No, it is enabling instead of disabling.
See 1.4 How to Develop or Incorporate Adaptations in Applications.

State-Based Approach.
Environmental States $\rightarrow$ state transition diagram

Fig 1.3 Codai's state transition dig.
What is the cost?

- HA - processing + memory < fixed infrastructure

- not exploiting broadcast nature of wireless channel
  may lead to flooding for "hot" items' invalidations.
Web Caching

HTTP: Cache consistency support.

"if-modified-since" \( \rightarrow \) "200" object

\( \rightarrow \) status code

if not "304" \( \leftarrow \) "document unmodified"

Consistency Maintenance Schemes:
- TTL (Time-to-Live)
  - Adaptive TTL

\[ \text{Active} \quad \text{Not-Active} \]

Document Modification History
- Polling Scheme
- Invalidation $\rightarrow$ with Lease $\rightarrow$ multi-tier Lease

Performance of these schemes can be evaluated in terms:
1) Network load generated
2) Consistency in presence of failures