Data Caching: Data Invalidation

Instructor: Dr. Sunho Lim (Ph.D., Assistant Professor)

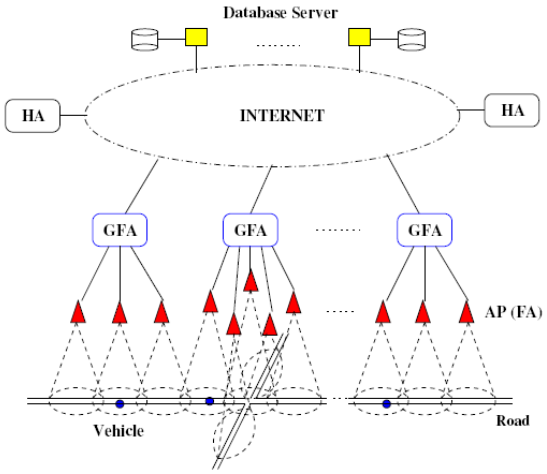
Lecture 06

sunho.lim@ttu.edu

Introduction: Internet-based VANETs

- Vehicular Ad hoc NETworks (VANETs)
  - Vehicle-to-vehicle or vehicle-to-roadside communications
  - Drivers can access the Internet service and information on wheel through a roadside unit, e.g., an access point (AP), an info-station, or a message relay box
- Internet-based Vehicular Ad hoc NETworks (IVANETs)
  - In order to provide a flexible connectivity, accessibility, and a rich set of services
  - Integration of a VANET with a wireless infrastructure

System Model



- A hierarchical network model
  - APs, GFAs, HAs, Data servers
  - Facilitate cache invalidation and location management
- Having the HAs/GFAs to take care of,
  - Micro-mobility
  - Micro-invalidation

Introduction: Cache Invalidation

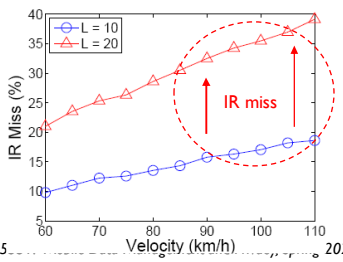
- A key optimization technique in improving the communication performance, caching
  - Caching frequently accessed data items in a local storage
  - Less of a problem to determine which data item to cache
    - The storage space in a vehicle is not critical compared to mobile devices
  - Energy conservation is not an issue
    - Vehicles are supported by own built-in battery
- A critical design issue, cache invalidation
  - Applications require data consistency with the server to avoid using any stale data
  - Upon update, a server invalidates the cached copies of data item

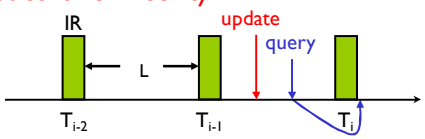
Motivation: Observations

- High-speed mobility,
  - Vehicles reside in a coverage area for a short period of time (e.g. 5 to 24 seconds)
  - Multiple coverage areas are involved in broadcasting an invalidation report (IR), so the cost of broadcasting becomes non-negligible
- Unlikely adjacent vehicles have common data items in real,
  - Wasteful to broadcast the same contents of IR to different vehicles
- Support a scalable invalidation operation with the minimized IR traffics,
  - Coordinate with network agents of location management

Revisit Cache Invalidation: Single-cell based Approach

- Most of the prior IR-based schemes are variation of the Timestamp (TS) scheme,
  - Server periodically (L) broadcasts IR (includes a update history of the past broadcast intervals)
  - Tradeoff between the communication overhead and query latency
  - The TS scheme does not consider mobility





Query delay of half of the broadcast interval (L/2)

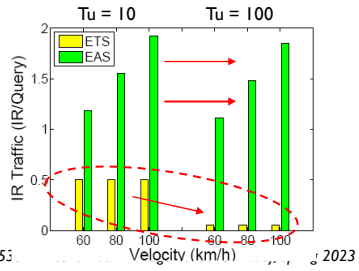
As the velocity increases, more vehicles handoff before receiving the IR, and thus IR miss ratio increases. → Sensitive to L

Revisit Cache Invalidation: Multi-cell based Approach

- To reduce the IR miss ratio due to mobility, multi-cell based designs extended from the Asynchronous Stateful (AS) scheme are proposed,
  - A network component (e.g. mobile switching center or server) located higher network hierarchy than a BS executes cache invalidation operations
  - Upon update, a stateful sever proactively unicats IRs into multiple cells, where the updated data item is cached.
- Server's proactive IR transmission is neither a scalable solution nor efficient, if the updated data item is not queried.

Revisit Cache Invalidation: Multi-cell based Approach (cont.)

- Compare two extended schemes:
  - Extended TS (ETS) – a stateless server broadcasts an IR to every cells whenever vehicles cache an updated data item.
  - Extended AS (EAS) – a stateful server broadcasts an IR to selective cell(s). Upon handoff, send all the valid cached data items' ids for validity check, counted as an IR traffic.



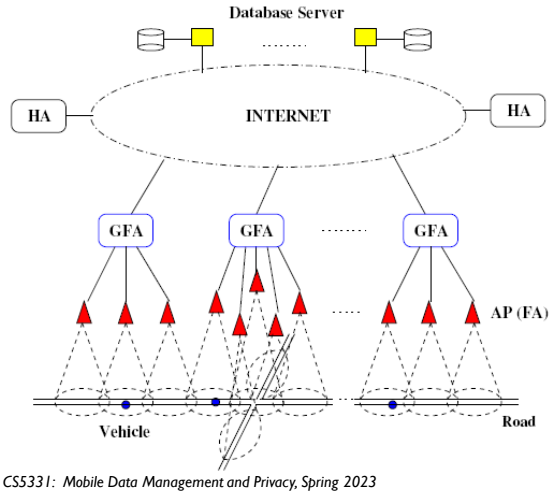
In ETS, same amount of IR traffic and minimized the impact of velocity

In EAS, high IR traffic regardless of cache update interval because of additional invalidation check upon every handoff

Revisit Cache Invalidation: Summary

- Extended TS (ETS),
  - Reduce the impact of mobility but exist a non-negligible query delay
  - The stateless server blindly broadcasts an IR to entire cells – scalability issue
- Extended AS (EAS),
  - No query delay but incur heavy IR traffic mainly due to mobility
- Reduction of the mobility impact and IR traffic are conflicting requirements and optimization of both is extremely complex.

System Model



- A hierarchical network model
  - APs, GFAs, HAs, Data servers
  - Facilitate cache invalidation and location management
- Having the HAs/GFAs to take care of,
  - Micro-mobility
  - Micro-invalidation

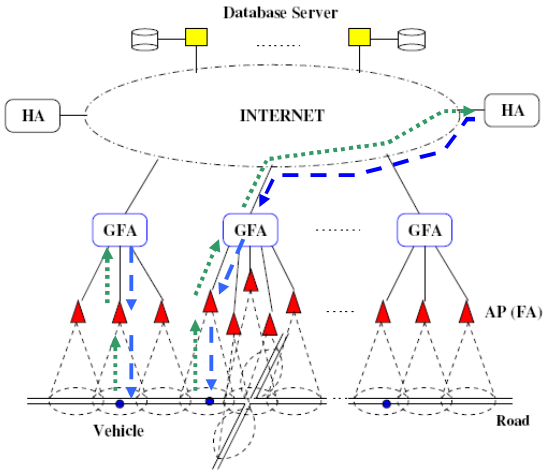
A Cooperative Cache Invalidation Approach

- A server and network agents of location management coordinate together for cache invalidation operations.
- Unlike most of the prior stateful server approach,
  - The server does not keep track of a vehicle's current location for IR broadcast, but maintains a list of which data item is accessed by which vehicle.
  - Network agents maintain vehicles' current location.
- Upon update, the server sends an IR to an HA rather than blindly broadcasting the IR to multiple cells.
- Then, the HA judiciously refines and re-distribute the IR to appropriate GFA(s)
  - GFA(s) do not pro-actively send the IR but reply on-demand basis

A Cooperative Cache Invalidation Approach (cont.)

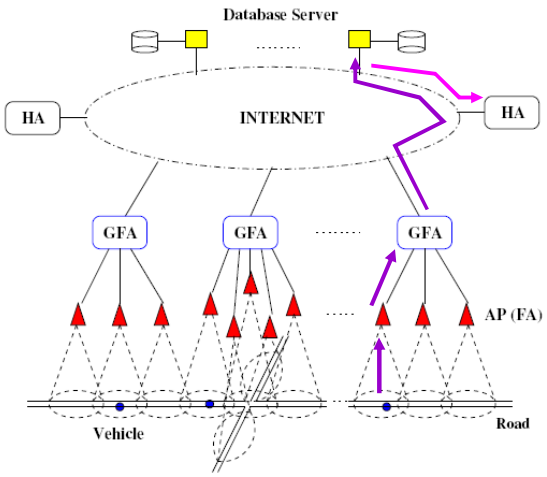
- The rationale behind this approach,
  - A vehicle's location is implicitly maintained by both a GFA and an HA
    - → no duplicating operation
  - Broadcasting the same IR to different vehicles is inefficient
    - → low probability of finding common data items in adjacent vehicles
  - The query rate of vehicles is independent of the cache update rate of the server
    - → no blind IR broadcasting

Cooperative Cache Invalidation (CCI): Location Update



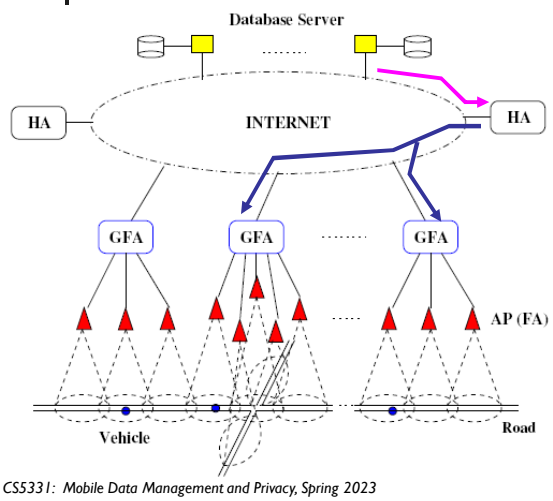
- Keep track of vehicles' whereabout
- Use a 2-way handshake to implement location update
  - Regional or home request/reply packet
- An AP (FA)
  - Role as a foreign agent

Cooperative Cache Invalidation (CCI): IR Broadcast from Sever



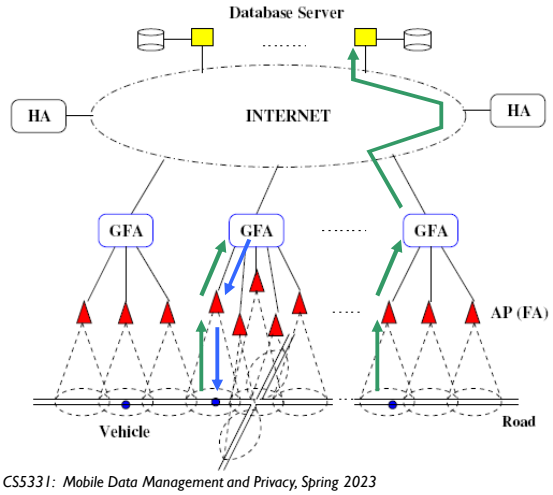
- A server and network agents coordinate together
  - A **cooperative stateful** server
- Maintain a list of which data item is accessed by which vehicle
  - $[id(d_x), [vid(v_{x,y}), t_{x,y}]]$
- Send an IR, **IR(s)**, to a HA
  - $<id(dx), vid(v_{x,y}), t_{cur}>$

Cooperative Cache Invalidation (CCI): IR Broadcast from HA



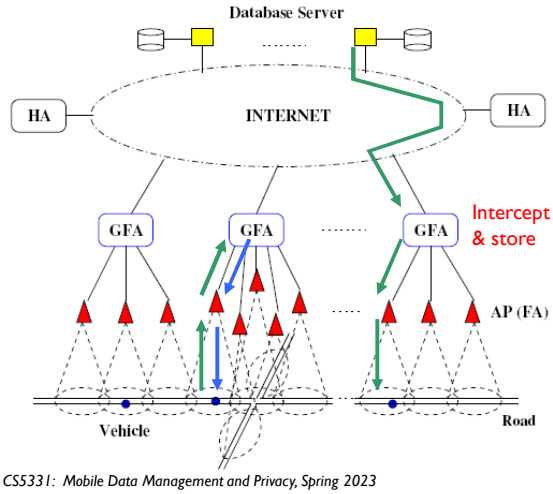
- An HA,
  - Maintain which vehicle is currently roaming under which GFA
  - [vid(y_v), g_i]
- Upon receive IR(s),
  - Create an IR, **IR(h),** based on matched GFAs
  - May have more than one IRs
  - May have different size
- Send IR(h), to GFA(s)
  - <id(d_x), t_{cur}>

Cooperative Cache Invalidation (CCI): Query Request & Invalidation



- A **pull-based** approach
- Send a request packet
  - [CoA, id(d_x), f_x]
- The GFA may reply directly or forward the packet

Enhanced Cooperative Cache Invalidation (ECCI)



- A GFA intercepts the ACK packet and caches the attached date items
  - Maintain a set of valid data items based on the IR
- Reply ACK directly without forwarding to the server