Petrozavodsk State University
Department of Computer Science

cs.karelia.ru

Olga Bogoiavlenskaia, Dmitry Korzun, Kirill Kulakov

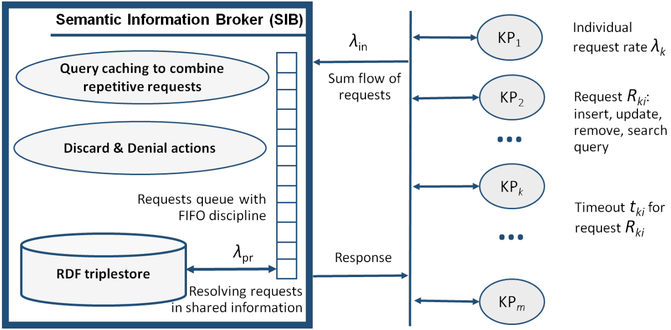
Random Backoff for Active Control of Information Updates in Smart Spaces

The reported study is supported by Russian Fund for Basic Research according to research project # 19-07-01027.

24th FRUCT Conference

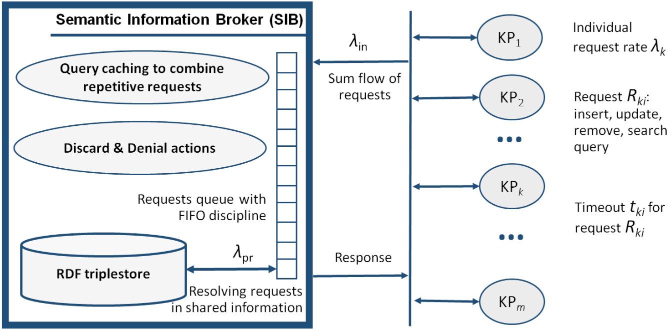
8–12 April 2019, Moscow, Russia

Motivation



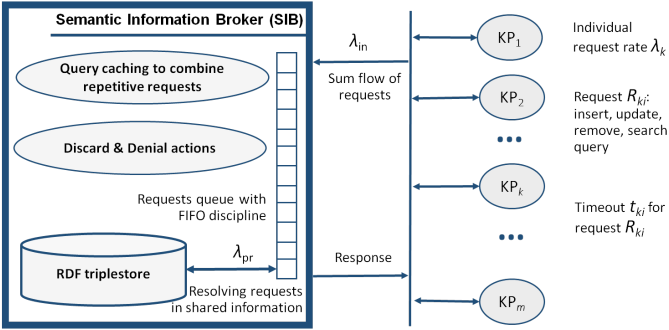
- Semantic information broker (SIB):
  - A “server” for smart space deployed in IoT environment
  - Serving requests of many heterogeneous devices (running a knowledge processor, KP)
  - Persistence queries: information updates (“inverse requests”)
- SIB performance
  - Request queue length: the number of agents *m* and request rate λ
  - Performance fluctuation: mobile and heterogeneous devices, cooperative activity and requests bursts

Basic Mechanisms for SIB



- SIB serves requests on information updates with the intensity λ_{pr}.
- Detection of the same requests from different KPs
- Several queues, for different request types
- Discard of some requests when the load is high

Delegation



- Delegating some load from SIB to KPs:
  load balancing and congestion control
- Active control of information updates:
  KP expects the information is updated and sends check request
- Formally: KP defines timeout t_i between requests R_{i-1} and R_i, $i = 1, 2, \dots$

Our Approach to Control

- The active control timeout is defined as

$$t_i = f\left(t_{i-1}, k_i, w_i\right)$$

where k_i is the number of observed losses,
w_i describes additional workload measures

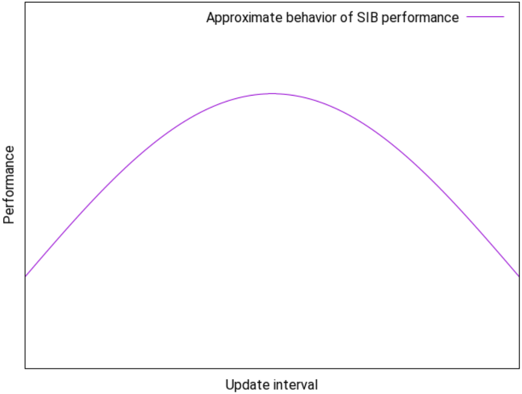
- The two components: Adaptive Strategy Timeout (AST) and Random Backoff Timeout (RBT)

$$t_i = t_i^{\mathrm{ast}} + t_i^{\mathrm{rbt}}$$

- AST implements AIMD-like scheme (Additive Increase and Multiplicative Decrease, see also the known method for TCP congestion control)
- RBT implements additional delay and randomization.

Intuition behind “AST + RBT”: No Aggressive Control

- AST *adopts* to the information update process in the networked environment:
  - less losses ⇝ longer t_i^{ast}
  - many losses ⇝ shorter t_i^{ast}
- When the SIB workload is high then many losses occur
  - ! AST requires short timeout t_i^{ast}
  - ⇝ many requests degrade the SIB performance more
- **RBT solves this small timeout problem** $t_i = t_i^{\mathrm{ast}} + t_i^{\mathrm{rbt}}$
- For short t_i^{ast} higher t_i^{rbt} makes t_i reasonable



Intuition behind “AST + RBT”: Randomization

- Cooperative activity can lead to a requests burst by many KPs
  - request synchronization due to regular patterns in smart space participants activity
- Randomized backoff for desynchronization

$$t_i = t_i^{\mathrm{ast}} + t_i^{\mathrm{rbt}}$$

- RBT provides the random component t_i^{rbt} that smoothes the burst from a time point to a time interval
- See also the classical collision problem in Ethernet

Adaptive Strategy Timeout: Model

- Let $i = 1, 2 \ldots$ enumerate the sequence of requests for update checks from a KP
- Let t_i^{ast} be the time period between consecutive checks $i - 1$ and i
- Let k_i be the number of losses during t_i^{ast}.
- At the end of t_{i-1}^{ast} the decision is made about the next

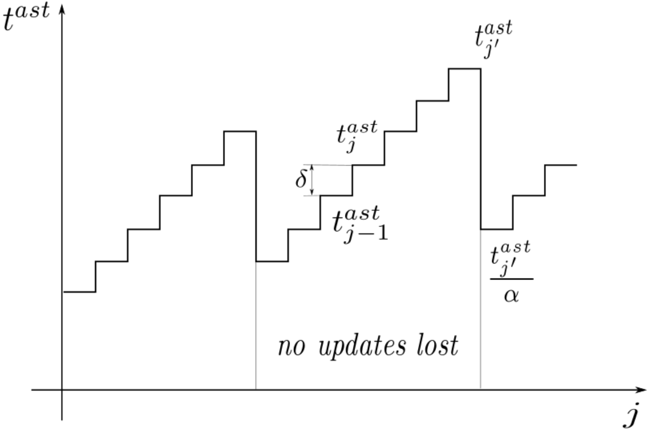
$$t_i^{\text{ast}} = f(t_{i-1}^{\text{ast}}, k_{i-1})$$

Simple case is as follows,

$$t_i^{\text{ast}} = \begin{cases} t_{j-1}/\alpha, & k_{j-1} > 0 \\ t_{i-1}^{\text{ast}} + \delta & k_{i-1} = 0, \end{cases}$$

(E.g., in the TCP congestion control, $\alpha = 2$)

Adaptive Strategy Timeout: Illustration



Random Backoff Timeout

- The backoff method is widely used and studied extensively
- Random exponential backoff algorithm with collision detection:

  $$t_i^{\mathrm{rbt}} = \min\{t_{i-1}^{\mathrm{rbt}}\beta, t_{\max}\}$$

  where $\beta > 1$ is the backoff exponent
- E.g., $t_{\max} < nt_{\mathrm{avg}}$, where t_{avg} is the measured average request resolving time, i.e., stopping the growth value after n rounds
- More randomization,

  $$t_i^{\mathrm{rbt}} = \min\{t_{i-1}^{\mathrm{rbt}}\beta, t_{\max}\} + \epsilon$$

  where ϵ is a small random value, e.g., using the normal (Gaussian) distribution with mean $\mu = 0$ and variance t_{i-1}^{rbt}

Collision Detection for RBT

- Collision: the situation when more waiting is needed
- The active control timeout

$$t_i = t_i^{\mathrm{ast}} + t_i^{\mathrm{rbt}}$$

  - becomes small
  - SIB overload is detected
- Requests burst occurs (due to synchronized activity of many KPs)
  - local observation of the SIB performance
  - SIB collects and provides the context information
  - KPs constructs the same service and know “the pattern”

Active Control Levels

1. *No active control*: The case is subject to losses when the information update rate is high compared with the SIB device capacity.
2. *Basic active control*: Any KP can occasionally request the SIB to check for updates (deterministic or random timeouts)
3. *Adaptive strategy*: KP adjusts its information check activity to the observed losses
4. *Active timeout control*: Adaptive strategy + Random backoff. Smoothing aggressive forms of the Adaptive strategy and burst desynchronization

Analytic Confirmation for RBT

- Let τ be average timeout
  - single KP generates check requests with the rate $1/\tau$
  - m clients generate with m/τ
  - Simplified assumption: the total requests flow follows the Poisson distribution (no bursts).
- Let N_t be the SIB capacity limit, i.e., the maximum number of the requests arrived during the interval t which SIB can process without unacceptable delay or failure
- The probability that the N_t limit is achieved in the system is

$$\Pr\{N_t\} = \frac{[\frac{m}{\tau}t]^{N_t}}{N_t!} e^{-[\frac{m}{\tau}t]}.$$

RBT prevents τ to become unacceptably small, reducing the probability of achieving/exceeding N_t

Conclusion

- The “AST **+ RBT**” model

- Two reasons for “**+ RBT**” are studied
    - Individual Adaptive strategies for AST can be aggressive to the SIB performance
    - Requests bursts are a typical situation, and SIB device can be easily loaded

- Further study:
  more experiments are needed to tune the “AST + RBT” model and evaluate the RBT role