Information Distribution over GRASP

(draft-liu-anima-grasp-distribution)

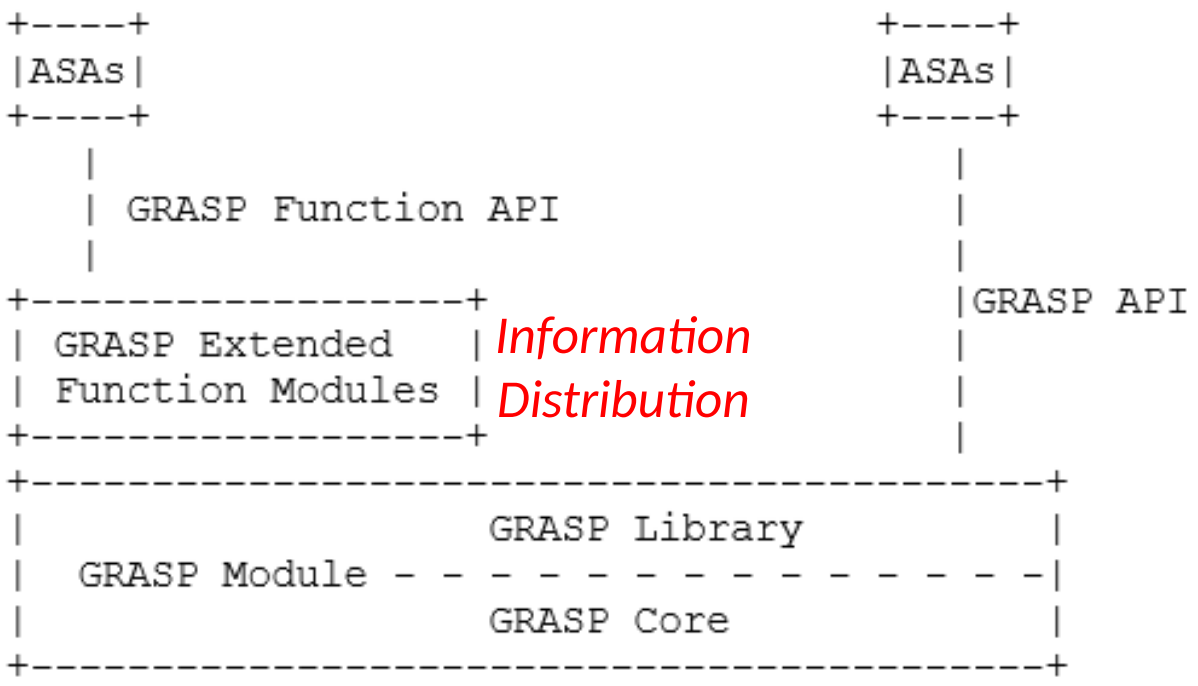
Bing Liu (speaker), Sheng Jiang

@Anima WG, ietf100, November 2017

Information Distribution in Anima

- It is a function to handle different patterns of i nformation exchange between autonomic nod es
- Adopting GRASP as bearing protocol
  - Just focuses on sending/receiving mechanisms, rat her than the function/logics achieved through Neg otiation or Synchronization which is the main targ et of GRASP (service agnostic)
  - Could be seen as some extension/enhancement to GRASP

Information Distribution in Anima



(Figure is from draft-liu-anima-grasp-api)

Patterns of Information Distribution

- Flood
  - One node distributes the information to all neighbors
  - Application scenario: distributing some global parameters/policies, e.g. “Intent”
  - Already covered by GRASP M_Flood
- Selective Flood
  - One node distributes the message to the neighbors matching a/a set of conditions
  - An optimization to Flood, to reduce some unnecessary traffic (several u se cases were discussed in IETF97)
  - Need defining new GRASP Objectives (selection criteria)
- Point to Point
  - Actively distribute the information to the nodes that newly get online
  - This might mostly happen between neighbors
  - Need defining a new type of message, e.g. Unsolicited Synchronization

Is that all?

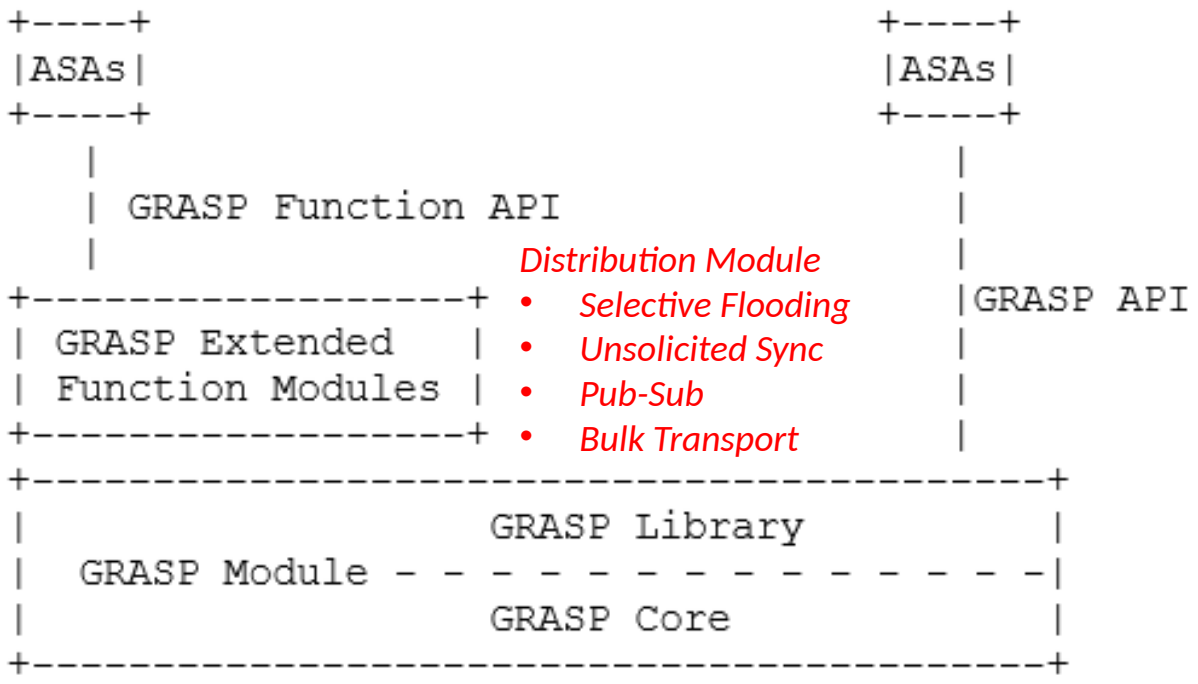
- Current draft only discusses “Pushing” the info rmation to other nodes
- How about “Pulling” information from others?
  - Pub-sub should be in the scope

Is that all?

- It’s even better to have a bit transport capabili ty
  - Transfer some configuration files that cannot be e ncapsulated within several packets
  - Transfer the ASAs for dynamic deployment

  draft-carpenter-anima-grasp-bulk-00 is addressing th is

The more complete picture



- More complete capabilities to support diverse communication requirements from ASAs
- Self-contained mechanisms other than incorporating different pieces of protocols together (which normally means much more code space and extra configuration burden)

Comments? Questions?

Should the WG adopt this draft?

Thank you!

IETF100, Singapore