Topics

- Overview
- Changes
- Open issues
- Request for help
- Discussion, next steps
Implementation model

ASA 1

ASA 2

ASA 3

ASA 4

API

Function library

Unicast sockets

GRASP core

Internal Functions

Discovery logic

Discovery multicast handler

Caches: (Discovery, Objectives, Flooded objectives, Sessions)

Flood multicast handler

Multicast sockets

IPC
Important data structures

- **objective**
  - name
  - syn or neg
  - loop_count
  - value  # any structure you want

- **ASA_locator**
  - locator  # normally IPv6 address
  - protocol # IPPROTO_TCP or IPPROTO_UDP
  - port
  - etc
Simplified summary of calls (1)

- register_asa(asa_name)
- register_objective(objective)
- discover(objective)
- send_invalid()
Simplified summary of calls (2)

- request_negotiate(objective, peer)
- listen_negotiate(objective)
- negotiate_step(objective)
- negotiate_wait(timeout)
- end_negotiate(result, reason)
Simplified summary of calls (3)

- synchronize(objective, peer)
- listen_synchronize(objective)
- flood(objectives)
- get_flood(objective, locators)
Recent changes

- Up to date with approved GRASP
  - Noted that simple nodes might not need the API at all, because a subset of GRASP could be integrated in a simple ASA.
  - Added `send_invalid()`
Missing features

- A few GRASP features lack API support in the current spec:
  - explicit locators for an objective*
  - rapid mode synchronization*
  - rapid mode negotiation

* already added in Python code
Open question: handling asynchronous operations

- GRASP is intrinsically asynchronous.
- Two approaches:
  1. Assume threaded environment. Some calls imply blocking (e.g. `discover()`, `request_negotiate()`).
  2. Assume event-loop environment. These calls will return 'noReply' until the callback occurs.
- Is it OK to describe both, as an implementation choice?
Need help

- Mapping to Python was easy
- Still need help on developing a robust mapping to C
  - Early draft of header file at
    
    https://github.com/becarpenter/graspy/blob/master/graspi.h
Discussion + next steps

- Comments? Questions?
- Should the WG adopt this draft?