

# Language Support for Dynamic Allocation in WSNs

# Why?

- Network is dynamic
  - variable number of neighbors
  - adaptive/probabilistic protocols
- WSNs are not hard real-time systems
  - can cope w/ allocation failures (e.g. a packet drop)
  - not sensitive to small run-time overheads
- Examples
  - Packet Forwarders: *SRP*, *basestation*
  - ? ? ?

# Challenges

- Hard to program
  - *malloc / free*
  - pointers
  - no garbage collector
- Little memory
  - fragmentation
  - leaks / dangling ptr
  - corruption heap vs stack

# How? Language support.

- malloc/pointers
  - partially hidden in O.O abstractions
- free (also leak avoidance)
  - automatic in most cases
    - *scope* or *completion*
    - GC is not required
- fragmentation
  - custom allocator w/o fragmentation
- corruption
  - bounded memory guarantees

# Fragmentation

## 1) Memory Pools (*PoolC*, *memb*)

- fixed sizes, low overhead, no fragmentation

## 2) Custom *malloc* (*mmem*)

- var sizes, some overhead, no fragmentation

## 3) Out-of-the-box *malloc*

- var sizes, portable, fragmentation

```
// (1): Memory Pools  
class T (10) with ... end
```

```
// (2,3): malloc  
class T with ... end
```

# Evaluation

- Lines of code
- Memory usage
  - number of failed calls to `malloc/Pool.get()`
- Run-time overhead

# Limitations

- Dynamic long-lived structures
  - graphs