Freie Universität Berlin

Service Placement in Ad Hoc Networks

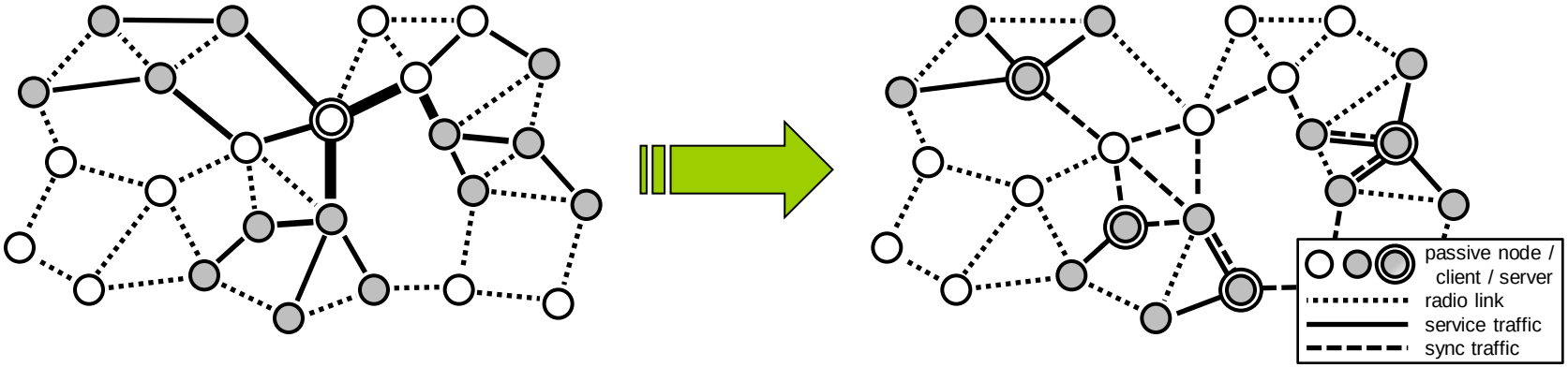
Georg Wittenburg and Jochen Schiller
Freie Universität Berlin

Kick-off Meeting GI/ITG KuVS Fachgespräch
„NGN Service Delivery Platforms & Service Overlay Networks“
Berlin, Germany

Introduction / Motivation

- Cooperation between nodes is a key assumption in ad hoc networking (e.g., medium access, routing, transport, overlays, …)
  - ➢ *Can this paradigm be extended to include the application layer?*

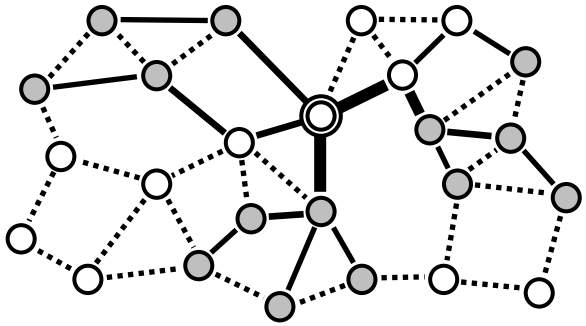
- ***Service placement*** – Process of selecting a set of nodes to host a service in light of a given service demand and network topology



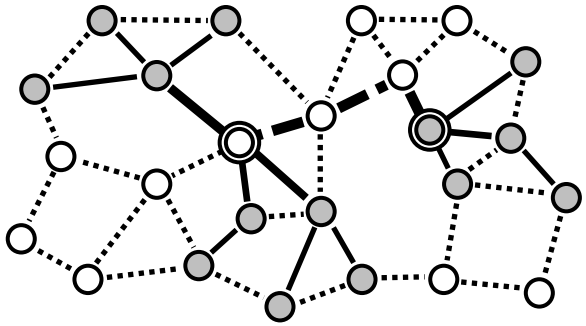
- Benefits:
  - Service configuration, i.e., the set of nodes to host a service, is adapted automatically at run-time
  - ➢ Reduction overall network traffic and latency
  - ➢ Optimize network performance according to service-specific metrics

Intuition

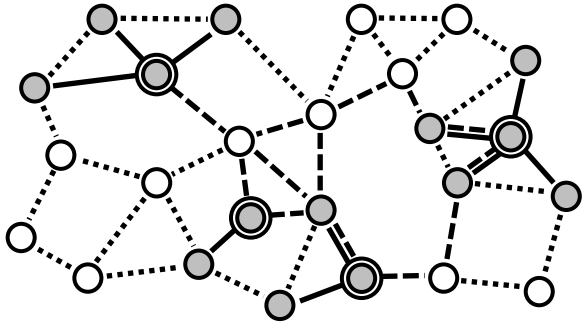
- Service placement largely depends on service-specific demands on synchronizing shared data between service instances.



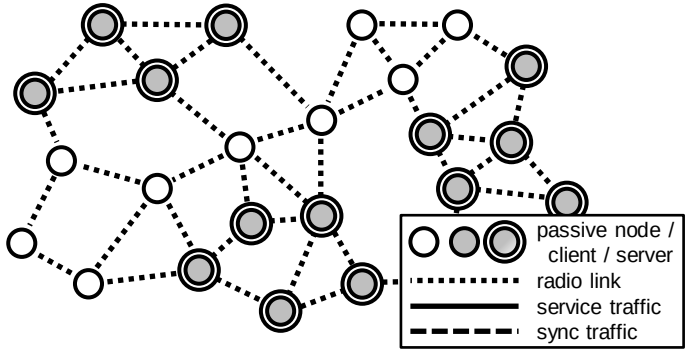
- Example #1: Transactional database
  - One single service instance



- Example #2: Directory service
  - Several service instances



- Example #3: Web server
  - Multiple, per-cluster service instances



- Example #4: Spell checker
  - One service instance per node

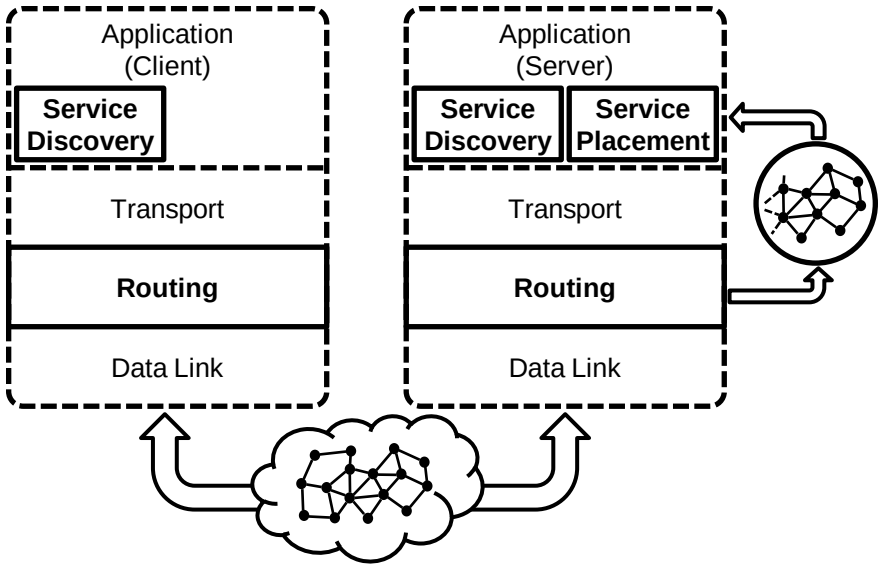
Problem Statement

- Subproblems:
  - *Where* to place service instances?
  - *How many* service instances for cost optimal operation?
  - *When* to adapt current configuration of services?
  - *How* to transfer services between nodes?

- Pitfalls:
  - Excessive control overhead
    - Up-to-date information required on network topology and regional service demand
    - Signaling between service instances
  - Cost of changing a service configuration
    - Transfer of service implementation and state
      - Handling of service requests while transfer is in progress
    - Service and route discovery induced by changing service hosts

SP*i* Service Placement Architecture

- Service placement middleware
  - Collect statistics on service demand, coordinate placement
- Enhanced routing protocol
  - Provide topology information to service placement middleware
- Enhanced service discovery protocol
  - Inform clients about current service configuration



➢ Cross-layering approach to reduce signaling overhead

SP*i* Service Placement Steps

Freie Universität Berlin

1) Aggregate topology and demand information on dynamically assigned service coordinator
   - Network topology provided by routing protocol (possibly incomplete)
   - Statistics on past service demand provided by middleware

2) Calculate optimal configuration by finding minimum of cost function
   - Cost function comprises service and synchronization traffic

3) Decide whether to change service configuration
   a) Compare to cost of current configuration
   b) Compare savings to estimate of the cost of adaptation

4) If adaptation is beneficial, issue commands to current service hosts
   - Commands: Replicate, migrate, shutdown

Evaluation

Freie Universität Berlin

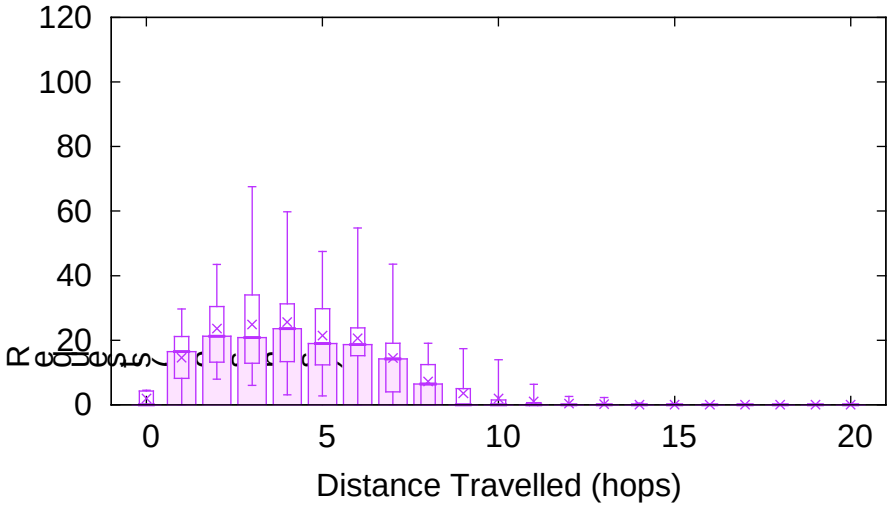
- Cross-platform implementation (Linux, Win32, ns network simulator) of SP*i* architecture
- Focus in this talk:

Comparison of traditional client/server against service placement (with SP*i*)

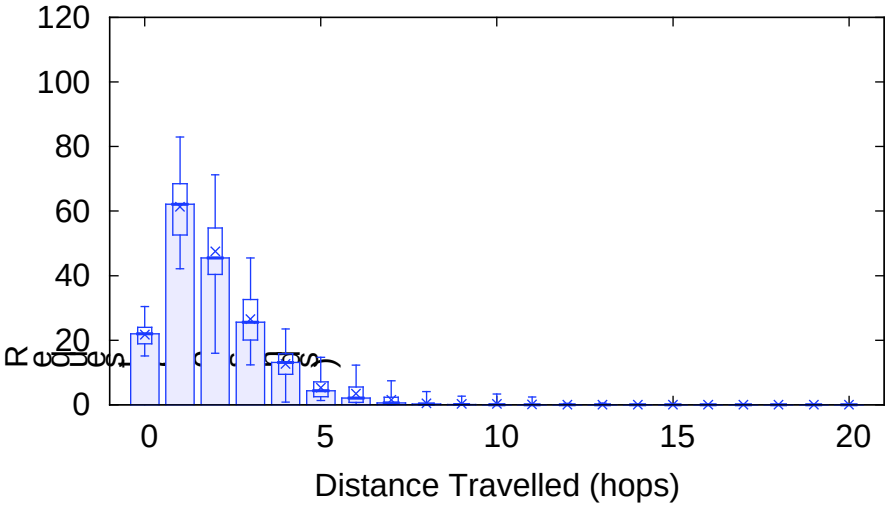
- Setup:
  - ns 2.33 with 802.11b, DYMO routing, 20 runs
  - Parameters:
    - Service size 100 kB
    - Sync traffic 10 %
    - Active nodes 50 %
    - Simulated time 20 minutes
  - Variables:
    - Network size [16,25,... 100] nodes
      - Random node placement
      - Area adapted to achieve near constant node degree
    - Network load [0.1,0.2,1.0,2.0,5.0] requests/s

Results – Hop Count of Requests

- No service placement

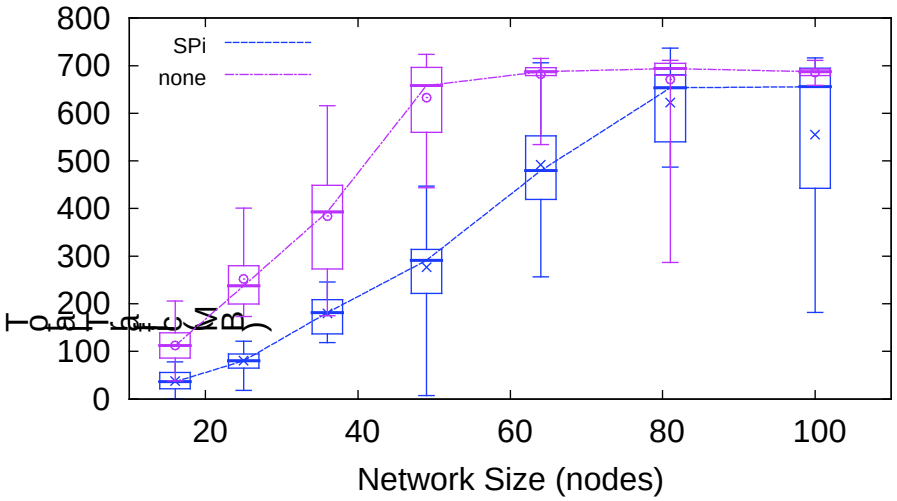


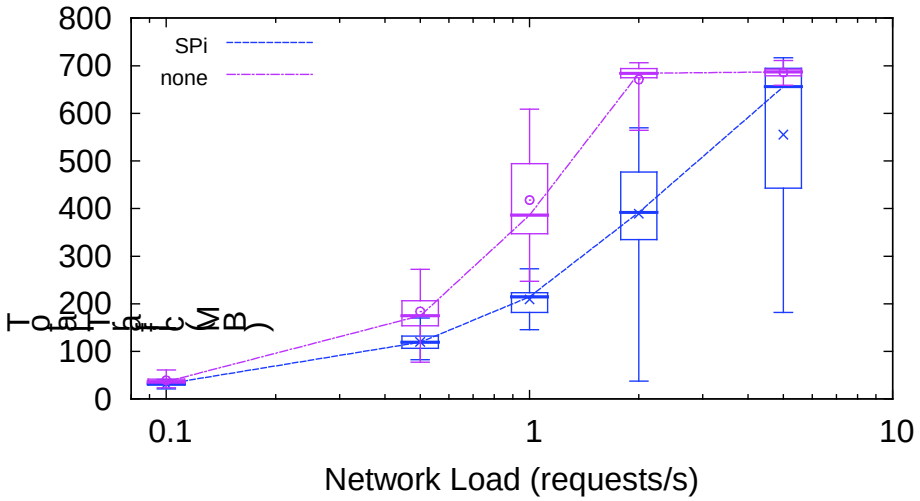
- SP*i* service placement



➢ Service configuration established by SP*i* reduces distance between service hosts and clients

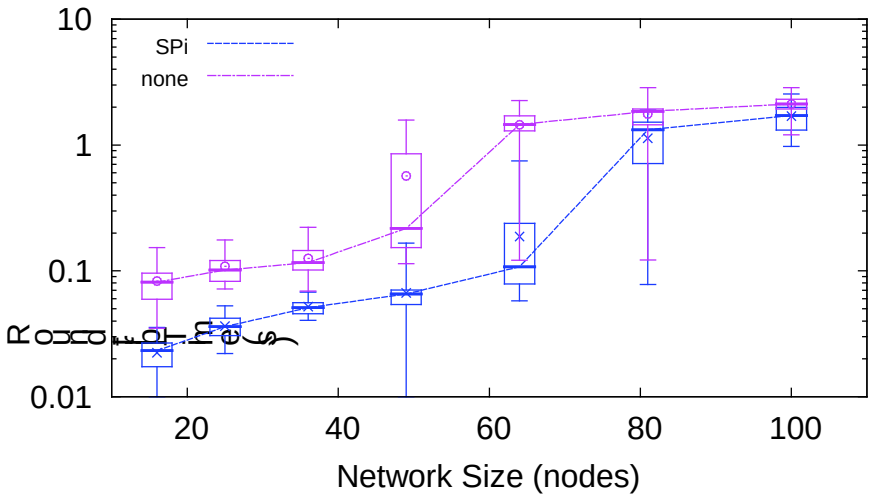
Results – Total Traffic

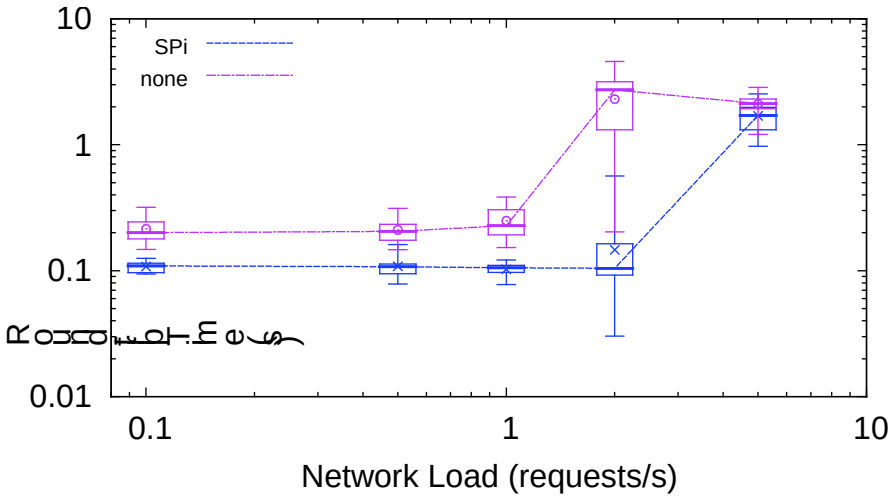




- Reduction of traffic by roughly one half
- No network saturation at medium network sizes and loads

Results – Round-trip Time





- Reduction of round-trip time by roughly one half
- One order of magnitude improvements at medium network sizes and loads

Conclusion

- Advantages of service placement:
  - Reduced distance between service hosts and clients
  - Reduction in both traffic and round-trip time
  - Larger networks remain operational under higher load

- Other benefits not covered in this talk:
  - Dynamically adapt service configuration to changing regional demand
  - Replication of service instances results in higher reliability
    - ➢ Protects against node failures and high levels of churn
  - Naturally supports group mobility

Solution Space

Freie Universität Berlin

Fixed Coordinator

- Central node controls network-wide service configurations

\+ Optimal service configuration
o Medium signaling overhead
– Requires infrastructure, i.e., not applicable to ad hoc networks

Dynamic Coordinator

- Dynamically assigned, per-service node controls service configuration

\+ Optimal service configuration
o Medium signaling overhead

Distributed Consensus

- All service hosts distributively control service configuration

\+ Optimal service configuration
– High signaling overhead

Local Decisions

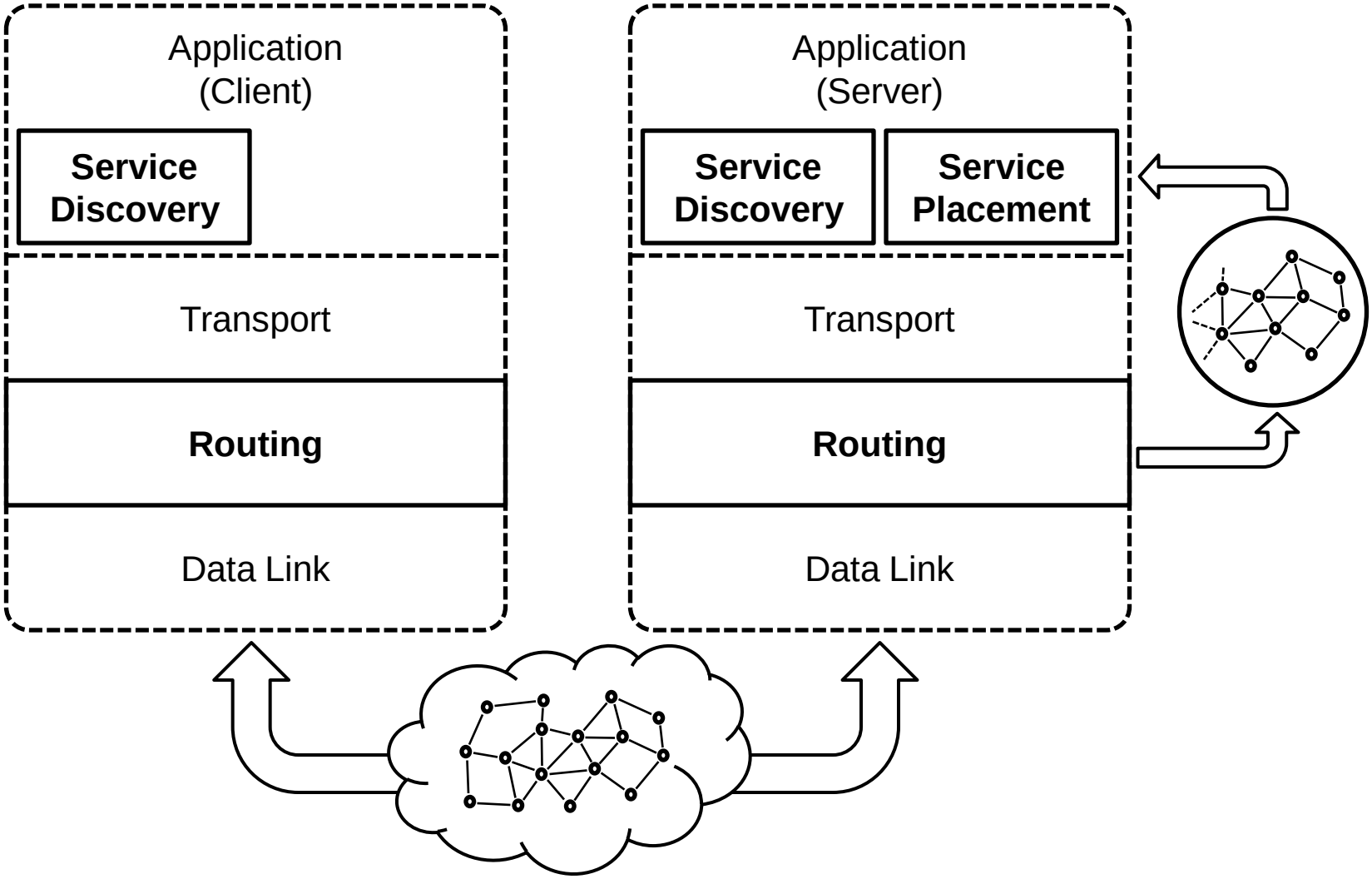
- Each service host controls its placement based on locally available data

\+ Low signaling overhead
– Suboptimal service configuration

centralized ⟷ distributed

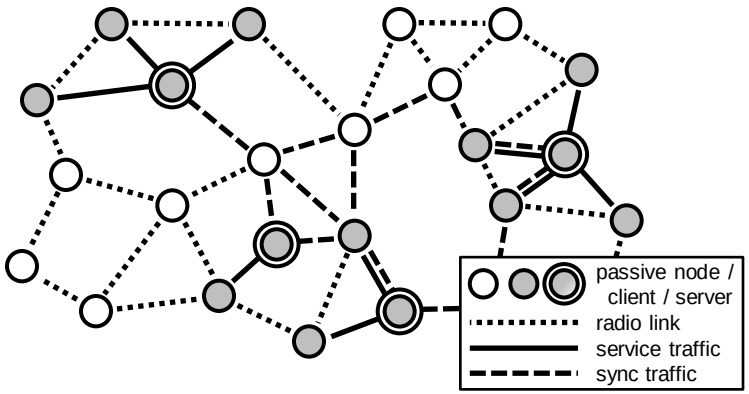
➢ Overview of current approaches in: G. Wittenburg and J. Schiller. A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks. In *Proc. of PerCom '08*, Hong Kong, China, March 2008.

SP*i* Component Overview

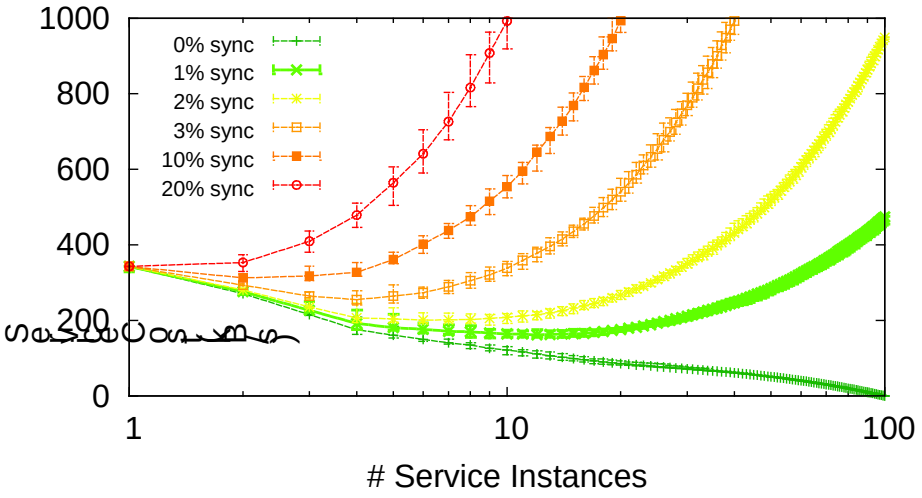


Calculating the Service Configuration

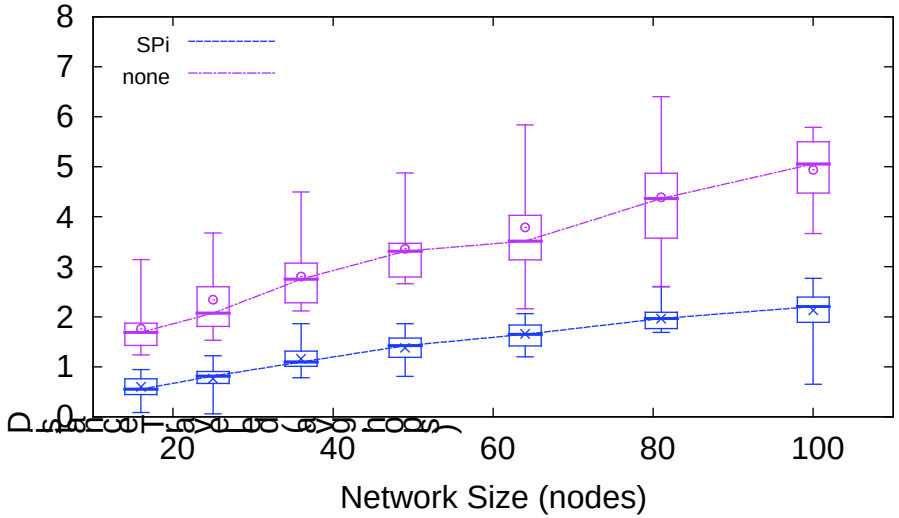
- Problem: Find a set of nodes to host service that minimizes cost function given
  - past service demand of client nodes
  - network topology (possibly incomplete) as extracted from routing information
  - fraction of traffic required for synchronization
- ➢ Cost function comprises service and synchronization traffic



- Algorithm (intuition):
  - Clients use closest service instance
  - ➢ Clients form clusters around hosts
- Algorithm (steps):
  - Initialize service configuration with one service host per client (known good solution)
  1) Calculate cost of current configuration
  2) If cost lower than cost of best known solution, store configuration
  3) Find pair of adjacent clusters with lowest service demand
  4) Merge cluster pair
  5) Repeat until no clusters left to merge
  - ➢ Stored solution minimizes cost function

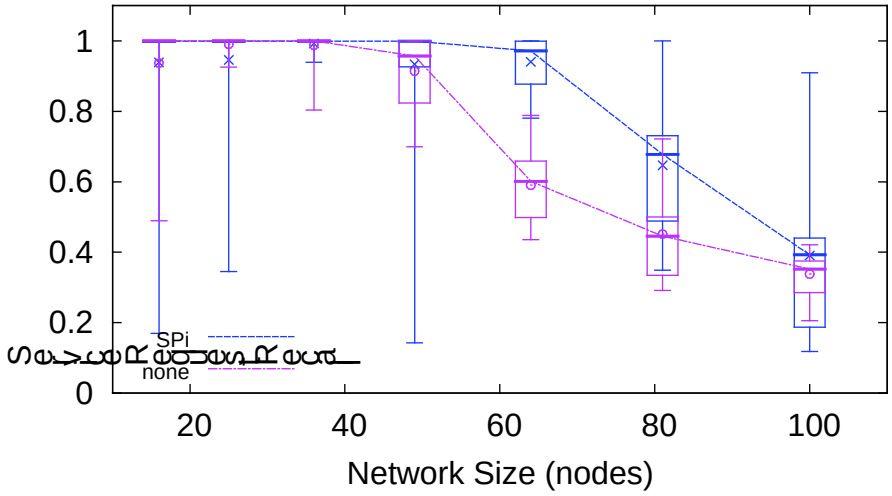


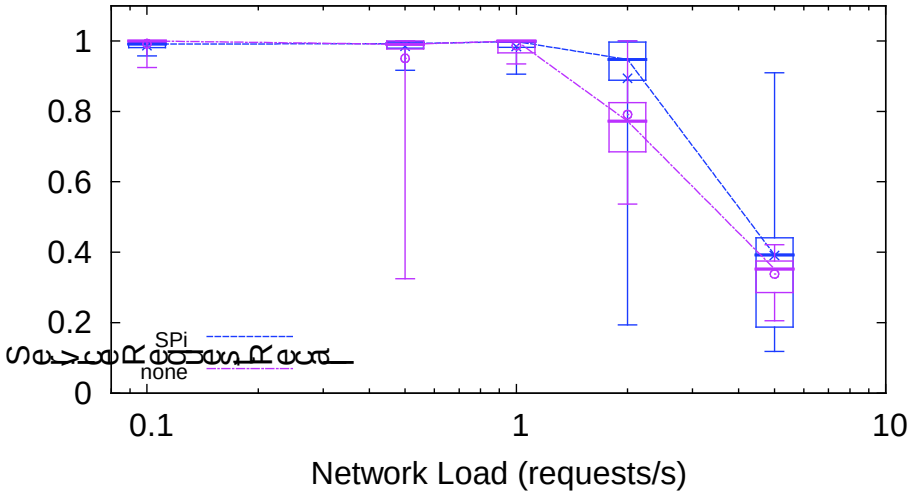
Results – Average Hop Count



- Distance between service hosts and clients reduced by roughly one half for all network sizes
- Generally less random distance less random

Results – Recall





- Higher level of recall at medium network sizes and loads