# Wireless Networks of Device Resource Discovery

WilliamAdjieWinoto, Elliot Schwartz Anit ChakraborțyTeremy Lilley Hari Balakrishnāphn Guttag

MIT Lab for Computer Science http://windcsmit.edu/



November 1999

# Application: Locationdependent wireless service



- Spontaneous networking
- Automatically obtain map region & discover device services and people ther
  Access, control services communicate with them
- Handle mobility & group
  - communication
- Locate other usefulisesv

(e.g., nearest café)

App should be able to convenient specify a resource and access



# Challenges

- Configuration
- Routing
- Discovery
- Adaptation
- Security & privacy

Dynamic, mobile environment with no pre-config support for internetworking or service lo

# Today



- Mostly static topol
   & services
- Deploying new servi cumbersome
- Applications cannot learn about network
- Failures are common
- High management cos



# Resource discovery

- Why is this hard?
  - Dynamic environment (mobility, performanc changes, etc.)
  - No pre-configured support, no centralized servers
  - Must be easy to deploy ("ZERO" manual configuration)
  - Heterogeneous seirces & devices
- Approach: a new naming systemo Lutiesn architecture using intentional naming



# Design goals

Names must be descriptive signifying applicationt

Name resolvermust track rapid changes

Robustness

System must overcome resolveænd service failu

Easy configuration

Expressivenes

Responsivenes.

Name resolvensust self-configure





# Intentional Naming System (IN principles

- Names areintentionabased on attributes
  - Apps know WHAT theywant, not WHERE
- INS integrates resolution and diogwar
  - Late binding names to nodes
- INS resolverseplicate and cooperate
  - Soft-state name exchange protocol withcperiod refreshes
- INS resolverself-configure
  - Form an application-level overlay network



# INS service model



# What's in a name?

- Expressive name language (like XML)
- Resolve architecture decoupfice language
- Names are descriptive
  - Providers annouen names
- Names arequeries
  - Attribute-vælunatches
  - Range querse
  - Wildcard matches

[vspace #etgroup [department = arch-lab [stateo\*egon [city #illsboro]]]

data

```
[vspace= camera]
[buildingne-43
```

[room = 504]]

[resolution=800x600]]

```
[access public]
```

[status = ready]

```
[vspace = thermometer]
[buildingne-43
  [room = *]]
[tomporature (200 El
  data
```

# Responsiveness: Late bindir

- Mapping from name to location(s) can change rapidly
- Integrate resolution and messaging to track change
  - INR resolves name by upoaknd-forward, not by returning address
  - lookup(namels a route
  - Forward along route
- A name can map to one locationas(") or to many ("multicast")



# Late binding services

- Intentionahycast
  - INR picksne of several possible locations
  - Choice based **service-controlled metric** [contrast withangcast
  - Overlay used to exchange nameesro
- Intentional multicast
  - INR pickalloverlay neighbors that "sexpres interest" in name
  - Message flows along spanning tree
  - Overlay used to transfatear too



### Robustness: Names as soft-sta

- Resolution via networkpbfcarted resolvers
- Names are weakly conststelike network-layer routes
  - Routing protocol to exchange names
- Fate sharing with issues, not NRs
  - Name unresolved only ificseabsent
- Soft-state with expiration against service/client failure
  - No need for explicit de-accipatr



# Self-configuringsolvers

- INRs configure using a distributed topology formation protocol
- DSR (DNS++) maintains list of candidat and active Rs
- INR-to-INR "ping" experiments" fonk weights"
- Current implementation forms (evolving spanning tree
- INRs self-terminate if load is low



# Efficient name lookups

#### • Data structure



- Lookup
  - AND operationsmang orthogonal attributes
  - For values pick the va) transformed the lookup
- Polynomial-time worst case



# Scaling issues

- Two potential problems
  - Lookup overhead
  - Routing protocol overhead
- Load-balancing by spawning new INR handles lookup problem
- Virtual space partitioning handles routing p problem
  - Just spawning new INR is insufficient







# WIND Applications

- Location-dependent mobile cappilons
- FloorplanA navigation & discovery
- Camera: An image/video service
- Printer: A smart print spooler
- TV & jukebox
- Location-support system barsed o intelligent beacons



#### WIND











LCS

























## Status & performance

- Java implmentation of SIN& applications
- PC-basedresolveperformance
  - 1 resolver: several thobusames @100-1000 lookups/s
  - Discovery time ainen hops
- Scalability
  - Virtual spacertpiations for load-shedding
  - Wide-areadesign in frateu
- Deployment
  - Hook in wide-armachitecture to DNS
  - Standardize virtupalces name (like MIME)
- Paper at SOSP 17



# Related work

- Domain NameSystem
  - Differences expressivess and architecture
- Service Location Protocol
  - More centralized,s lapsontaneous
- Jini
  - INS can be udefor self-organizationult-tolerant discovery
- Universal Plangd-Play & SSDP
  - XML-based descriptions; INS fields w
- Intentionalmes in othecontexts
  - Semantic filæstems, adapive web cacgin DistributedDirector





# Conclusion

- Achieving self-organizing netexprises a flexible naming system for researchiscovery
  - INS works in dynamic, between networks
  - Expressiveness: names convey intent
  - Responsiveness: late binding
  - Robustness: soft-state names
  - Configuration: Resolated for configure
- Application-level overlay networkgeoad way to build flexible, self-organizing network appli

