Laboratoire LSR
Logiciels Systèmes Réseaux
Software, Systems, Networks

Ambient networking
Les réseaux ambiants

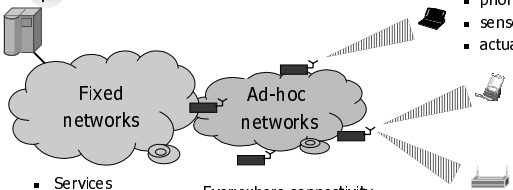
Andrzej Duda
LSR-IMAG Laboratory
CNRS and Institut National Polytechnique de Grenoble
Andrzej.Duda@imag.fr

Overview

- Introduction
- Issues in ambient networking
- Mobility management
- Service discovery
- Ad hoc routing
- Our current work
  - QoS and mobility in WLAN
  - ProxyScript
  - OmniSphere: a service infrastructure

Ambient networking

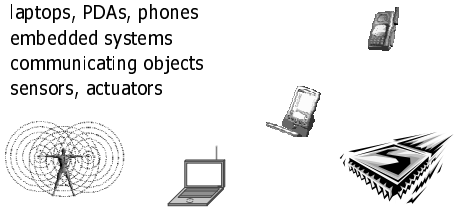
- Terminals
  - laptops
  - PDAs
  - phones
  - sensors
  - actuators



- Services
  - storage
  - computing
  - contents
- Everywhere connectivity
  - ad-hoc routing
  - discovery
  - configuration
- Wireless communication
  - WLAN
  - PAN
  - UMTS

Ambient networking

- Different wireless networks
  - 802.15, Bluetooth, 802.11, 802.16, LMDS, UMTS
- Humans, machines, objects
  - laptops, PDAs, phones
  - embedded systems
  - communicating objects
  - sensors, actuators



Ambient networking

- Billions of entities
- General public
  - need for auto-configuration
- Reactive to the environment
  - QoS, localization, energy, cost
- Multi-networks
- Services
  - storage, replication, transcoding, adaptation, collaboration
- New paradigms for supervision
  - from interactive to proactive

Wireless appliances

- Small battery powered devices
  - wireless connectivity
  - limited resources (CPU, energy)
  - some may be wearable
  - one function - one device (speakers, screen etc)
  - sensors/actuators (MEMS - Micro Electro Mechanical Systems)
  - multiple devices per person

Wireless communication

- Different types of wireless networks
  - RF
    - 802.16, LMDS
    - GSM/GPRS/EDGE/UMTS, DECT
    - WLAN (802.11, Hiperlan, HomeRF)
    - PAN (Bluetooth, 802.15)
  - RFID
    - small, battery less identification devices
  - Magnetic, electric
    - low-bandwidth PANs
  - Infrared
    - IrDA, AIR

Wireless communication

- Spectrum is scarce
  - small range, better efficiency, better frequency reuse
- Spontaneous systems
  - limit communication to short distance
  - communicate only with close neighbors

Ad-hoc network

- Provide everywhere connectivity
  - deploy network infrastructure where needed
  - ad hoc routing
    - self-organized network
    - self-maintained
  - forwarding on behalf of other terminals
- Provide some services as in the fixed infrastructure

Fixed network

- Provide resources
  - bandwidth, storage, computing, contents
- Upper layers
  - wireless middleware (wCorba, J2ME, JXTA)
  - services
    - abstractions of devices or software components
  - useful services
    - discovery
    - multimedia streams
    - asynchronous notification
    - mobile agents

Ambient networking

- Ambient means "surrounding on all sides"
  - provide universal access to the network
  - dynamic network infrastructure
    - ad-hoc mobile networking
- Ambient means "friendly helpful environment"
  - provide services to wireless appliances
  - dynamic service infrastructure
    - service discovery
    - active networks

Related terms

- Ubiquitous (constantly encountered)
  - everything has a CPU
  - net-aware embedded devices everywhere
- Pervasive (diffusing throughout)
  - interconnectedness
  - integrate sensors and actuators, home devices
- Proactive
  - many embedded devices
  - need another supervision model - opposition to the traditional interactive mode of operation

Requirements

- Scalability
  - billions of devices
- Energy efficient
  - battery powered appliances
- Dynamic behavior
  - appliances are mobile
  - context changes
- Adaptation
  - to location and context
  - to user's preferences and intentions

Requirements

- Autonomous operation
  - sensors and actuators should work unattended
- Auto-configurable
  - no need for a network expert
  - application scenarios: home networking
- Customizable
  - customize standard configuration
  - different roles at different places

Main ambient networking issues

- Mobility
- Service discovery
- Ad hoc routing
- Some others
  - auto-configuration
  - location
  - security
  - awareness
  - sensor networking
- Active networking may be applied in many areas

Part 1: Mobility

- Problem statement
- Global IP mobility
  - Mobile IP
- Local IP mobility
  - Cellular IP, HAWAII
- Session mobility
  - MIGRATE
- Application mobility
  - SIP Mobile

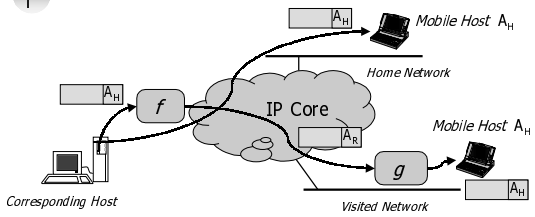
Issues in mobility

- Terminal mobility
  - horizontal hand-off (same network)
  - vertical hand-off (different networks)
- Personal mobility
  - same person at different places/terminals
- Network mobility
  - whole network moves (car, plane, train)
  - constrained movement (set of cars)
- Session mobility
  - move an X session seamlessly

Mobility in the Internet

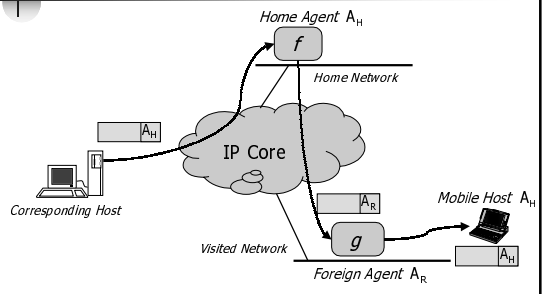
- IP address
  - used as a routing directive
  - used as a end-point identifier
- TCP transport session
  - 4-tuple: <src addr, src port, dest addr, dest port>
- Host moves
  - get a new address -> translate addresses
  - keep its address -> change routing

Global mobility

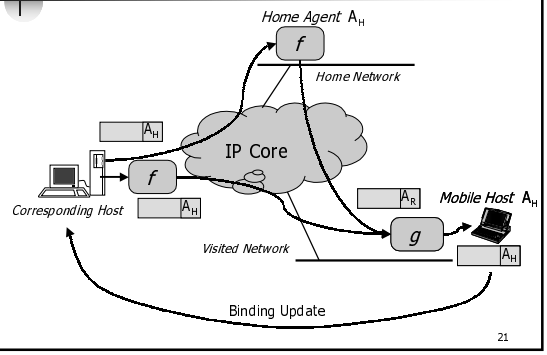


f: home address $A_H \rightarrow$ forwarding address A_R
g: forwarding address $A_R \rightarrow$ home address A_H

Mobile IPv4



Mobile IPv6



Mobile IP characteristics

- Routing
  - triangular in IPv4
- Address translation and tunneling
  - tunnel overhead in IPv4
    - adds 20 bytes to a 60 byte RTP packet
  - address translation in IPv6
- Global solution
  - keep open TCP connections when moving
  - treats all forms of mobility uniformly
    - short move - high overhead
    - large move - don't need to keep open connections

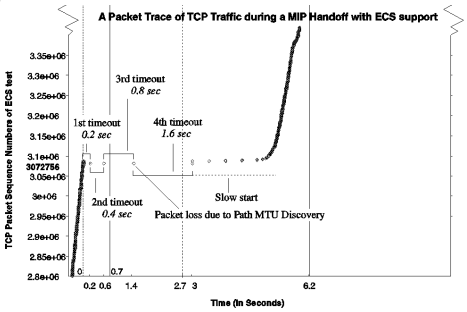
Mobile IP characteristics

- If QoS support required
  - establish new QoS bindings after handoff
- No paging support
  - passive connectivity: when in idle state, host does not transmit
  - how to find the idle host: *paging*
  - keeping approximate location information
    - less signaling
    - lower load over the air interface
  - saves energy

Performance

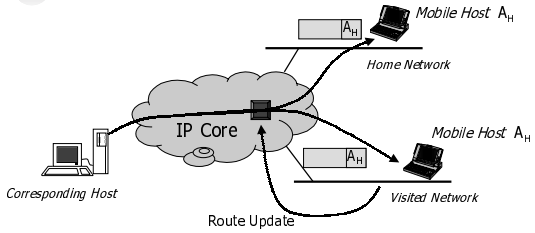
- High signaling overhead
- Tunnel overhead
- Hand-off delay
  - discover movement (periodic Agent advertisments)
  - get new address (DHCP)
  - register at Home Agent
  - send Binding Update

Mobile IPv4
Agent advertisment each 1 s

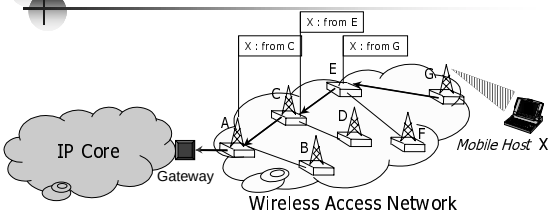


Fikouras *et al.* 2001

Local mobility - micro-mobility

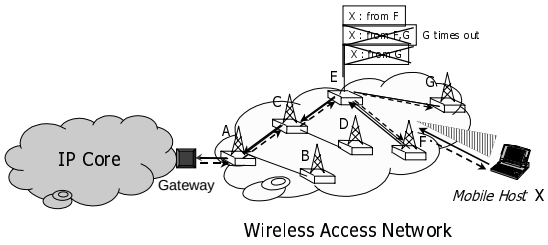


Cellular IP - create a route



- Beacons sent by gateway and flooded
- Cache at Cellular IP Routers
  - routing information (route update packets)
  - paging information (paging update packets)

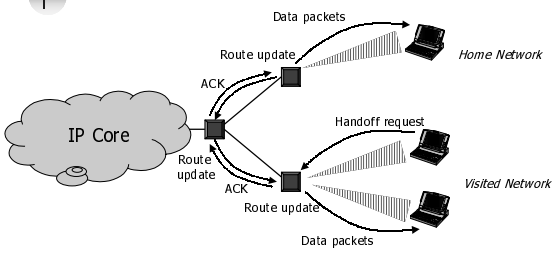
Cellular IP - handoff



Cellular IP characteristics

- Cellular IP routers in the wireless acces network
  - paging and routing update messages
  - mobile host that actively receive data must send route-update packets periodically
- Fast handoff
  - update route dynamically
- Local solution
  - routing entry per mobile host
  - gateway processes all updates for all hosts
- Efficient QoS support possible

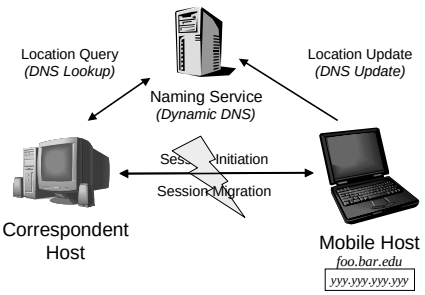
HAWAII

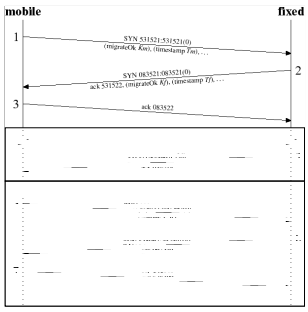


HAWAII characteristics

- Standard IP infrastructure in the wireless acces network
  - dynamic route update - host routes
  - supports paging
- Fast handoff
  - RTT of the handoff request
- Local solution
  - routing entry per mobile host
- Efficient QoS support possible

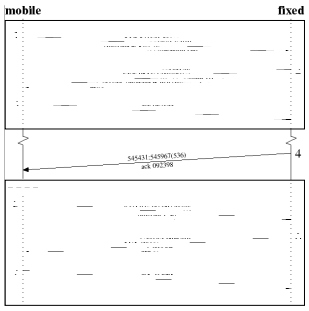
Session mobility: MIGRATE





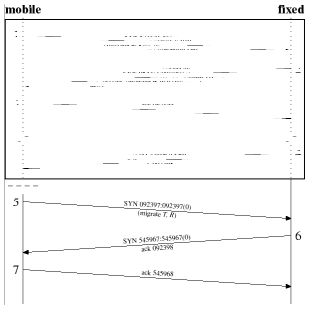
TCP Connection Migration

1. **Initial SYN**
2. **SYN/ACK**
3. **ACK (with data)**
4. Normal data transfer
5. Migrate SYN
6. Migrate SYN/ACK
7. ACK (with data)



TCP Connection Migration

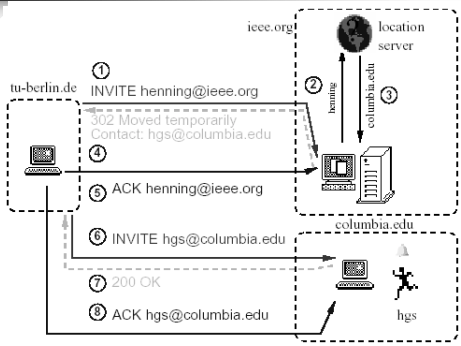
1. Initial SYN
2. SYN/ACK
3. ACK (with data)
4. **Normal data transfer**
5. Migrate SYN
6. Migrate SYN/ACK
7. ACK (with data)



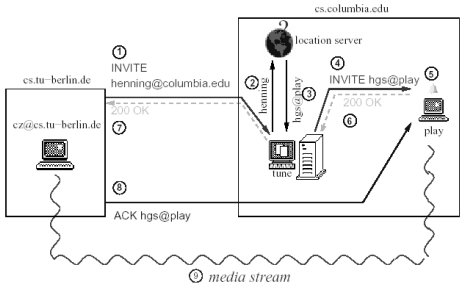
TCP Connection Migration

1. Initial SYN
2. SYN/ACK
3. ACK (with data)
4. Normal data transfer
5. **Migrate SYN**
6. **Migrate SYN/ACK**
7. **ACK (with data)**

Application mobility: SIP



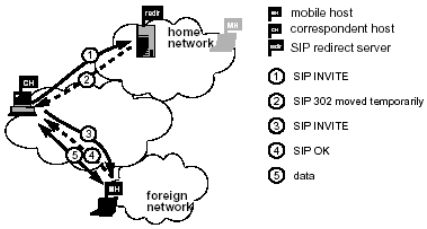
SIP



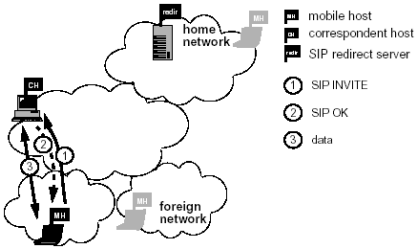
Mobile SIP

- Pre-call mobility
  - SIP proxy, redirect
- Mid-call mobility
  - SIP re-INVITE, RTP
- Recovery from disconnection

Mobile SIP: pre-call



Mobile SIP: mid-call

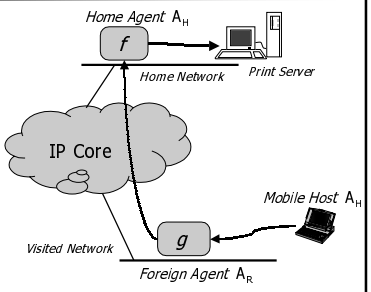


Conclusion

- Mobile IP
  - global solution, performance problems
- Micro-mobility
  - fast local handoffs
- Upper layers
  - combine the best features
- Our experience
  - extending *DiffServ* to wireless LANs
  - efficient micro-mobility protocol

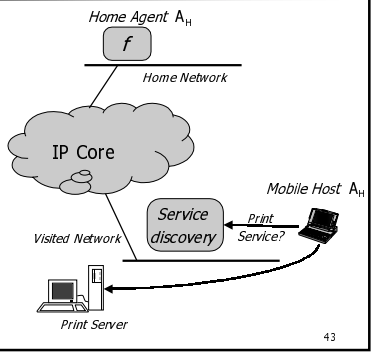
Service discovery

- Mobile host communicates with local servers
- Mobility management takes care of forwarding



Service discovery

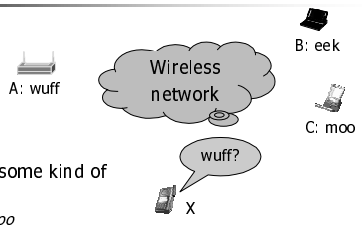
- Discover local services
- Use them locally



Part 2: Service discovery

- Problem statement
- Pull model
- Push model
- Service model and service description
- Case studies
  - SLP
  - Jini
  - UPnP

Service discovery

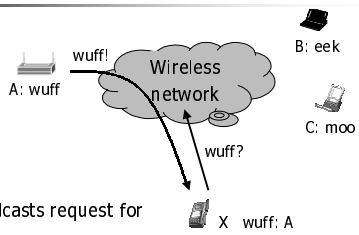


- Devices offer some kind of service
  - *wuff, eek, moo*
- X joins the group, needs *wuff* service
  - which device provides a required service?

Basic two approaches

- Pull model
  - query the environment when service required
  - reactive (on demand) solution
- Push model
  - push service information into the environment
  - proactive (prepared) solution
- Details of
  - distributed pull
  - centralized pull
  - distributed push

Distributed pull

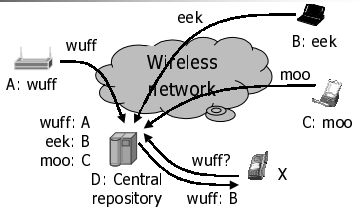


- Node X broadcasts request for *wuff*
- Node A responds with service information
- X uses *wuff*

Distributed pull

- Problems
  - RF broadcast rather unreliable
  - what if B in sleep mode?
  - broadcast may be slow (wake up sleeping nodes)
  - reachability
- Examples of protocols
  - SLP (hybrid)
    - either distributed pull or centralized pull
  - UPnP (hybrid)
    - distributed pull and distributed push

Centralized pull

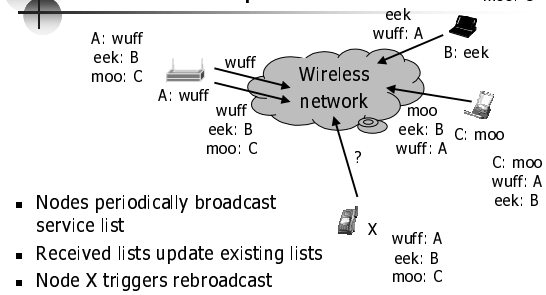


- Nodes A, B, C register with D
- Node X queries D for *wuff*
- Node D responds with service information

Centralized pull

- Problems
  - finding central repository
  - length of lease
  - single point of failure
- Examples of protocols
  - Jini
    - service registry
  - SLP (hybrid)
    - either distributed pull or centralized pull

Distributed push



- Nodes periodically broadcast service list
- Received lists update existing lists
- Node X triggers rebroadcast
- X list updated

Distributed push

- Problems
  - length of lease
  - broadcast interval
- Advantages
  - always updated view of environment
- Examples of protocols
  - DEAPSpace (IBM)
    - project result: push model more efficient than pull
  - UPnP (hybrid)
    - distributed pull and distributed push

Efficiency

- Tradeoff between bandwidth and timeliness
  - if we want short response time, we need pro-active actions, e.g. periodic broadcast, which in turn limits bandwidth
  - if we want to conserve bandwidth, we need reactive actions, that may be slow

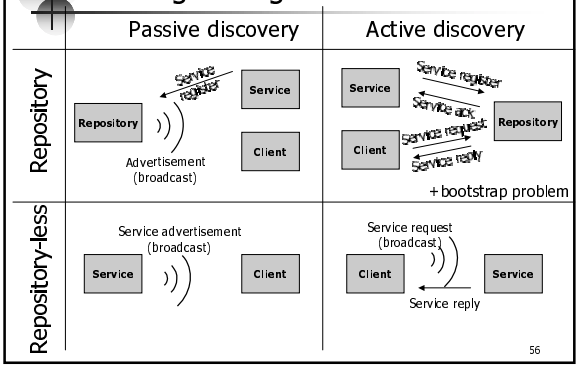
Another view: repository or not

- Using a repository
  - Central data base
  - Requires some administration
  - Complex queries (eventually on several services attributes)
  - Scales up by replication and distribution
  - Allows federated or hierarchical structures
  - More unicast communications
- Repository-less
  - No servers needed
  - Zero-conf
  - Simple queries
  - Requires a very simple protocol
  - More multicast communications

Another view: passive vs. active

- Passive discovery
  - Services advertise their availability, address, and other necessary information
  - Clients that receive advertisements may contact services as they wish
- Active discovery
  - Clients request service(s) of some kind and receive information about services in response
  - Services or directories listen for requests and respond appropriately

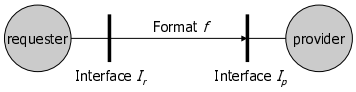
Putting all together



Service

- What is a service?
  - processor concept
    - input | processing | output
  - API concept
    - set of interface descriptions
  - Component concept
    - piece of software implementing specific intefaces
- Service description
  - simple to generate and parse
  - compact

Service model



- Input and output interfaces
  - data in particular format
  - provider *satisfies* a requester if the provider interface *matches* the requester interface
  - two interfaces match if their data format sets are not disjoint

Service description

- Wireless environment imposes restrictions
  - compact: conserves network bandwidth and device memory
  - easy to parse: limited CPU and memory on devices
  - precise: for matchmaking
- Textual descriptions
  - ASN.1, XML, SOAP, Java interfaces
- General form

name, service-attribute, input-format, output-format, details

Case study: Service Location Protocol (SLP)

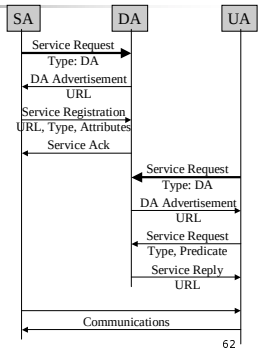
- Actors
  - User, Service and Directory Agents
- Operational Modes
  - either distributed pull or centralized pull
- Bootstrapping (locating DAs)
  - Broadcast
    - DA advertises its presence on a periodic basis (passive)
    - UA and SA multicast requests (active)
  - DHCP option 78

Case study: Service Location Protocol (SLP)

- Semi-reliable broadcast: Multicast Convergence Algorithm
  - Requests multicasted by UA and SA
  - Responses issued by DA (at bootstrap) or SA (directory-less)
  - First request is normal
  - Next requests includes the list of agents that have already responded

SLP: protocol

- Bandwidth usage without DA is higher than in the situation with a DA
- Limiting the number of retransmissions by the MCA, bandwidth usage may become equal
- bandwidth usage is more sensitive to the number of UAs than the number of SAs
- SLP Header ≈ 128 bytes



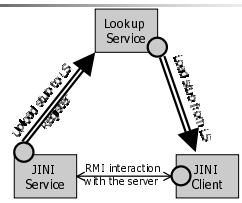
Case study: Jini

- Actors
  - Services, Clients, and Lookup Services
- Operational Modes
  - Active Discovery with Directory
  - Centralized pull
- Bootstrapping (finding LSs)
  - Unicast, when the location of LS is known
  - Local multicast (passive or active)

Case study: Jini

- Leases
  - at registration, to every service is granted a lease for a determined period
  - leases expiration determines automatic deregistration
  - recovery from crashes and system consistency
- Distributed Events
  - LS may provide notifications whenever something interesting happens

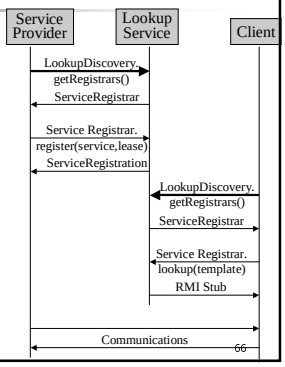
Jini: Code mobility



- Powerful features
  - due to single language environment
  - allows services to push their interfaces to clients they have never met

Jini: protocol

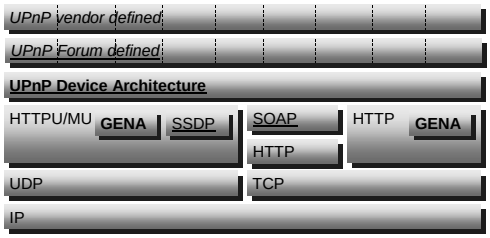
- Multicast is UDP
- Unicast is RMI
- Architecture similar to SLP with DA
- Without code downloading, bandwidth is 50% higher than for SLP
  - due to the overhead generated by TCP and RMI
- With code downloading (60K), bandwidth usage is 17.5 times that of SLP



Case study: Universal Plug and Play

- Actors
  - services embedded in devices
  - Control Points
- Operational Modes
  - directory-less in a hybrid mode
  - distributed pull and distributed push

UPnP: Protocol Stack



UPnP: operation steps

0. Control point and device get addresses
1. Control point finds interesting device
2. Control point learns about device capabilities
3. Control point invokes actions on device
4. Control point listens to state changes of device
5. Control point controls device and/or views device status using HTML UI

UPnP: Addressing (step 0)

- Control point and device get address
  - use a DHCP server
  - else use Auto IP
- Steps
  - pick an address in 169.254/16 range
  - check if it is used (gratuitous ARP)
  - periodically check for DHCP server

UPnP: Discovery (step 1)

- Control point find interesting device using SSDP
- Passive Discovery - Devices
  - advertise their services when added
  - refresh advertisements
  - cancel advertisements when removed
- Active Discovery – Control Points
  - find interesting devices/services when needed
- Very simple messages
  - service/device identifier
  - service/device type
  - URL of the XML root device description

UPnP: Description (step 2)

- Control point learns about device capabilities
- Expressed in XML
- Device description
  - type
  - physical container
  - logical container
    - type
    - URL for description
    - URL for control
    - URL for eventing
  - UI
    - icons
    - URL for presentation

- Services
  - functional units within devices
- Service description
  - actions
  - state variables

UPnP: Control (step 3)

- Control point invokes actions on device using SOAP
- SOAP
  - Simple Object Access Protocol
  - lightweight XML-based protocol for exchange of information in a decentralized, distributed environment
- Remote procedure calls and responses coded in XML

UPnP: Eventing (step 4)

- Control point listens to state changes of device
- Events are send using GENA
  - General Event Notification Architecture
  - send and receive notifications using HTTP over TCP/IP and administratively-scoped multicast UDP
  - SUBSCRIBE to notifications
  - UNSUBSCRIBE
  - NOTIFY
    - of device availability
    - of state variable changes
- Very simple event model

UPnP: Presentation (5)

- Control point controls device and/or views device status using HTML UI
- Presentation may be:
  - page for viewing the device stats
  - means for human user to control the device

Others

- Salutation
  - repository or repository-less, active discovery
  - centralized or distributed pull
- DEAPspace
  - passive discovery without a repository
  - distributed push
- Secure Service Discovery Service (SSDS)
  - repository, active discovery
  - centralized pull
- Intentional Naming System
  - repository, active discovery
  - centralized pull

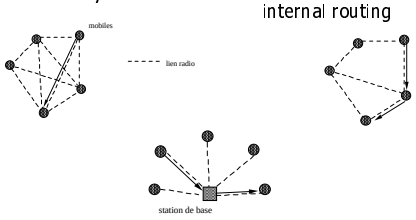
Part 3: Ad-hoc routing

- Wireless architectures
- Reactive (on-demand) protocols
  - AODV
  - LUNAR
- Pro-active (pre-calculated) protocols
  - OLSR
- Performance
- Wireless infrastructure for ambient networks
  - IRULAN

Wireless architectures

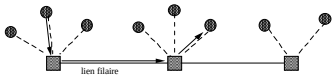
full connectivity

internal routing

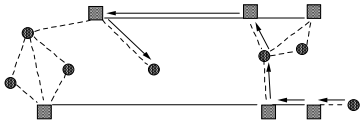


base station

Hybrid wireless architectures



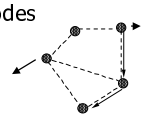
Interconnected base stations (GSM, 802.11 BSS)



Hybrid routing wireless-wired

Ad-hoc architecture

- Create multi-hop connectivity among set of wireless, possibly moving, nodes
- Mobile, wireless hosts act as forwarding nodes as well as end systems
- Need routing protocol to find multi-hop paths
  - needs to be dynamic to adapt to new routes, movement
  - interesting challenges related to interference and power limitations



Ad-hoc routing requirements

- Distribution paths
  - multi-hop paths
  - loop-free
  - minimal data transmission overhead
  - multicast? efficient flooding?
- Self-starting and adaptive to dynamic topology
- Low consumption of memory, bandwidth, power
  - scalable with numbers of nodes
  - localized effects of link failure

Problems using DV or LS

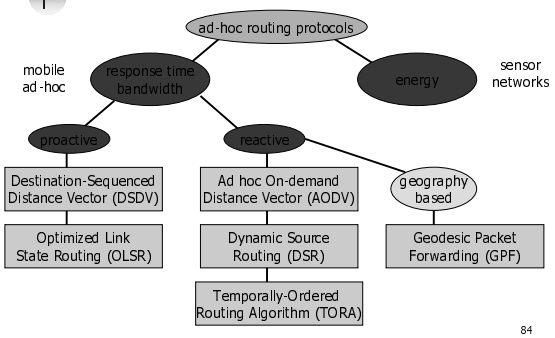
- DV protocols may form loops
  - very wasteful in wireless: bandwidth, power
  - loop avoidance sometimes complex
- LS protocols
  - high storage and communication overhead

- DV - Distance Vector (e.g. RIP)
- LS - Link State (e.g. OSPF)

Other problems

- Periodic updates waste power
  - transmitter sends portion of battery power into air
  - reception requires less power, but periodic updates prevent mobile from "sleeping"
- Convergence may be slower in conventional networks but must be fast in ad-hoc networks and be done without frequent updates

Classification



AODV: Ad-hoc On-demand Distance Vector

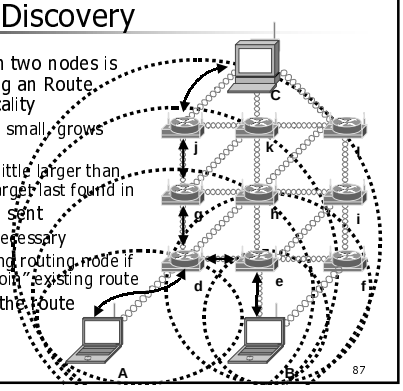
- Targets networks in which
  - routing churn is high enough that maintaining routes is unproductive, and
  - can absorb a network wide broadcast rate
- Modified on-demand Distance Vector routing
- C. Perkins (Nokia)

Messages

- Route Request: "I need a route"
- Route Response: "Route advertisement"
- Route Error: "Withdraw route"
- Periodic route response to neighbors acts as "hello", installing and refreshing route

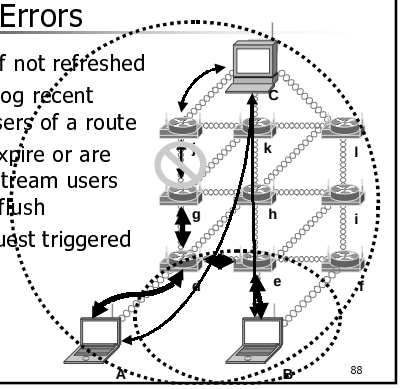
Route Discovery

- A route between two nodes is found by sending an Route Request to a locality
  - Initial locality small, grows with failure
  - After that, a little larger than the locality target last found in
- Route Response sent
  - By target if necessary
  - By neighboring routing node if possible to "join" existing route
- Network stores the route



Route Errors

- Routes expire if not refreshed
- routing nodes log recent downstream users of a route
- When routes expire or are flushed, downstream users are notified to flush
- New route request triggered

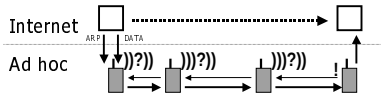


LUNAR Overview

Lightweight Underlay Network Ad hoc Routing (Uppsala University and Fraunhofer/FOKUS)

- Less ambitious than MANET protocols:
  - a dozen nodes, up to 3 hops *(ad hoc horizon!)*
- Simpler than AODV, OLSR, DSR ...
  - 30 to 70% less code, but
  - similar or better performance
  - supports IP unicast *and* broadcast
  - no parameters: includes wireless card and IP *auto-config*

LUNAR - repeated ARP rewriting



- Strategy I: *simplify IP integration*
  (a) fool IP, do L2.5 routing, create subnet illusion
  (b) catch ARP request: translate then rebroadcast to perform discovery and to build delivery tunnels
- Strategy II: *keep ad hoc logic simple and robust*
  (a) do nothing for next 3 seconds,
  (b) do everything again, if needed

LUNAR - automatic IP integration

- IP subnet illusion:
  - LUNAR private subnet number 192.168.42.0/24
  - dynamic assignment of host number à la AppleTalk and Windows APIPA: pick and probe for conflicts
- Beyond spontaneous nets: Default Gatewaying
  - use DHCP for GW assignment, not node addresses
  - (multiple gateway choice also possible)
  - run NAT on the default gateway(s)
- DNS information: also gathered via DHCP
- Roaming & handover: not LUNAR business (mIP)

State of LUNAR

- LUNAR user space daemon for Linux 2.4
  - sources available (GPL), demo-on-a-floppy
  - http://www.docs.uu.se/selnet/lunar/
- Activities (spring 2002):
  - Internet Draft in preparation
  - Porting to Lego Mindstorms (μLunar)
  - Simulation/comparison in ns-2

OLSR: Optimized Link State Routing

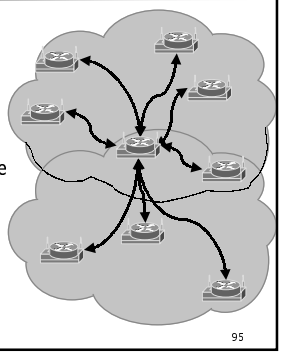
- Targets networks in which
  - routing churn is consistent with pre-calculated routes, and
  - capable of supporting fairly intelligent devices
- modified Shortest Path First routing
- P. Jacquet and HIPERCOM project (INRIA)

Messages

- "Hello"
  - neighbor discovery and relationship maintenance
- Topology Update
  - distributes subset of interconnectivity information

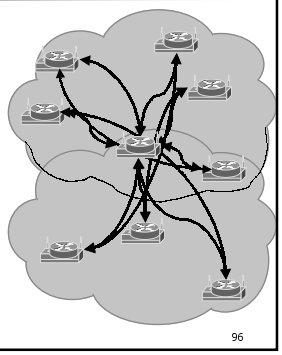
Neighbor relationships

- Each device emits a periodic "Hello"
  - advertise itself to its neighbors
  - determine who else is there
  - select some systems to act as Multi-Point Relays



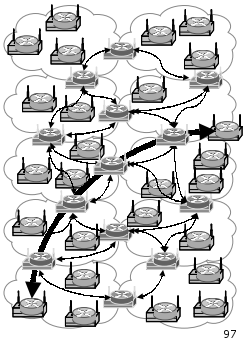
Multi-Point Relays

- Passes Topology Information
  - acts as router between hosts, in wired parlance
  - minimizes information retransmission
  - forms a routing backbone

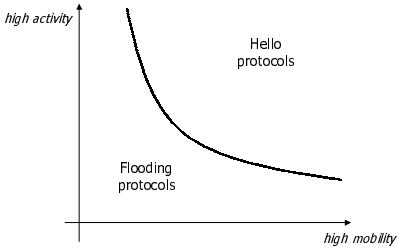


Structure of an OLSR Network

- MPRs form routing backbone
  - other nodes act as "hosts"
- As devices move
  - topological relationships change
  - routes change
  - backbone shape and composition changes

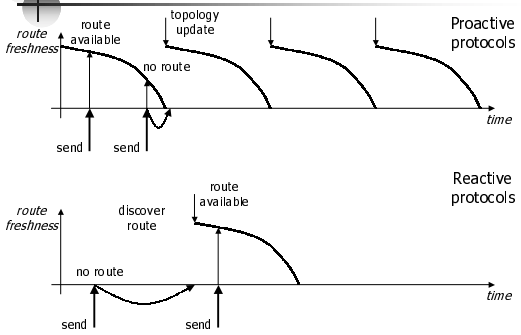


Comparisons



- Protocols based on flooding (AODV) vs. protocols based on hello (OLSR)

Comparisons



Performances

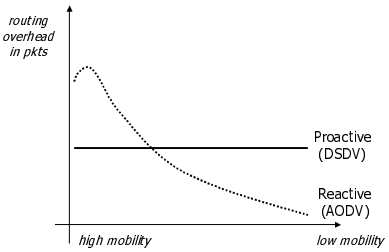
- Routing overhead high for AODV when small pause times (high mobility)
- AODV delivers 95-100% of packets
- AODV does well for low mobility

Comparisons

- Reactive protocols are based on flooding (AODV, DSR, TORA)
  - no guarantee of the optimal route
- Proactive protocols discover topology (OLSR)
  - hello protocol for neighbor discovery
  - guarantee of the optimal route
  - possible multiple routes

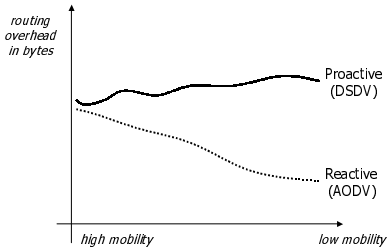
Performances

- Routing overhead high for AODV when small pause times (high mobility)



Performances

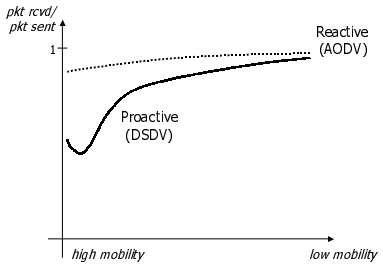
- Routing overhead high for AODV when small pause times (high mobility)



103

Performances

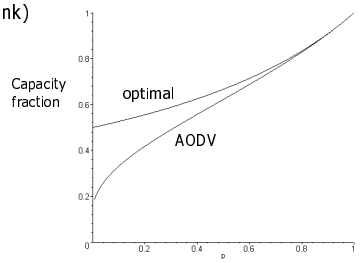
- AODV delivers 95-100% of packets



104

Performances of AODV

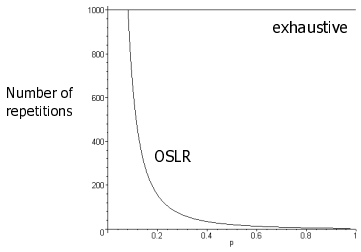
- Useful bandwidth as a fraction of the network capacity (random graph, large number of nodes, p - availability of a link)



105

Performances of OSLR

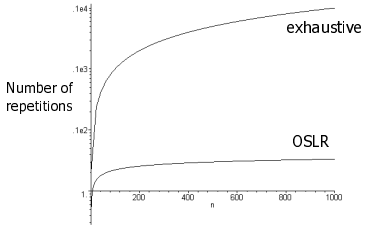
- Cost of flooding (random graph, 1000 nodes, p - availability of a link)



106

Performances of OSLR

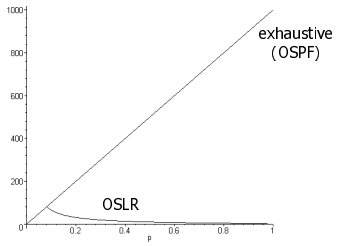
- Cost of flooding (random graph, variable number of nodes, p = 0.9)



107

Performances of OSLR

- Size of control packets (random graph, 1000 nodes, variable p)

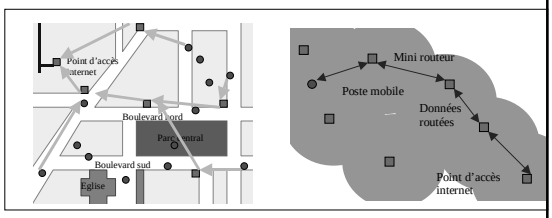


108

IRULAN project

Local network at the city scale

- Wireless fast access to the Internet with ad-hoc routing over partial and evolutive infrastructure

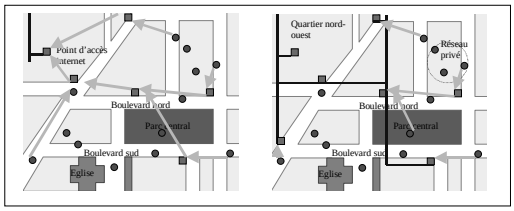


IRULAN

- High speed internet at 11 Mbps or more
  - IEEE 802.11b, 802.11a, HIPERLAN, Bluetooth (1Mbps)
- Mini-routers deployed in the area to cover
- Mini-routers control data delivery to mobile nodes

IRULAN

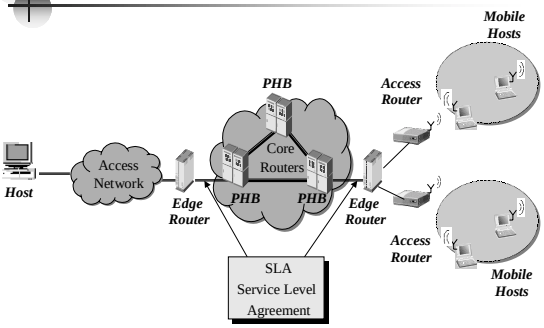
- Densification and private « islands »
- Turning around not covered areas



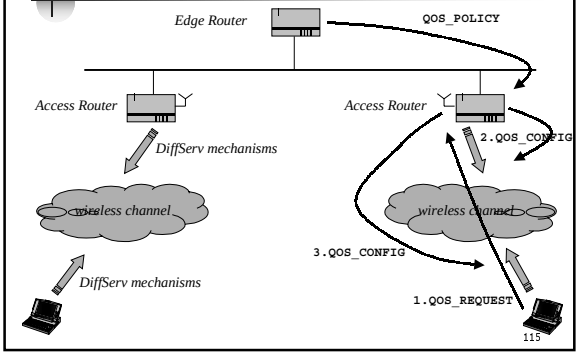
Our current work

- Quality of service and mobility for the Wireless Internet
  - *DiffServ* over 802.11b WLAN
- ProxyScript
  - generic active node for active services
- OmniSphere: the Information Universe
  - flexible infrastructure for ambient services

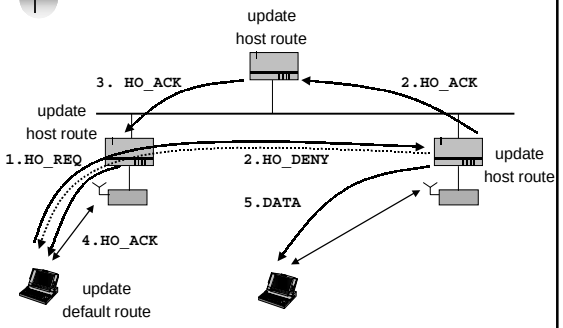
DiffServ over 802.11b WLAN



Hierarchial QoS management



Hand-off



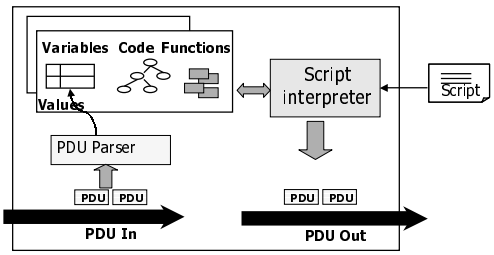
DiffServ over 802.11b WLAN

- *DiffServ* QoS mechanisms in mobile hosts and access routers, IPv6 stack
- Sources are constrained according to level of available resources
- Class isolation, better performance measures for priority EF class
- Integrated micro-mobility management

ProxyScript

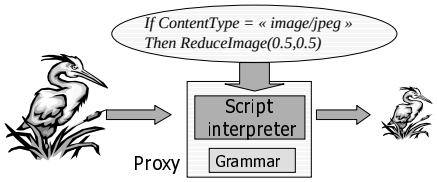
- Generic active node
  - packet analyzer
    - parse packets for patterns or properties
  - programmable actions
    - invoked when some patterns found
- Programmable gateway
  - specify the grammar of a protocol, values or properties
  - associate actions that work on the values
  - use a library of functions related to a given protocol

ProxyScript architecture



ProxyScript

- Script language for programming active services
- Supports some predefined variables:
  - *ContentType, ContentLength etc….*
- Used to program actions on PDUs

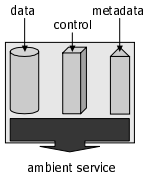


Programmable proxies

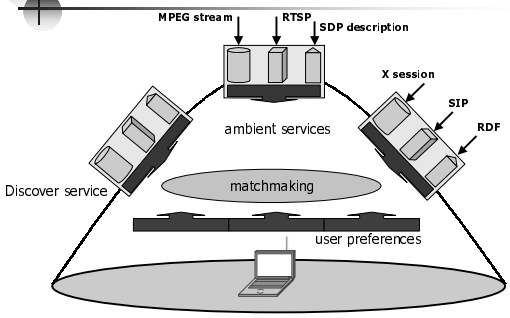
- SIP
  - personalized actions upon INVITE
  - set up transcoding
- X window
  - multiplexing streams
  - dynamic control for cooperative environments

OmniSphere

- OmniSphere: the Information Universe
  - *ambient service* encapsulates *typed data*, *channels*, *metadata*, and *control*
- *typed data* and content attributes
- *metadata* associated with contents
- *channels* for transporting typed data
- *control* for performing some operations on typed data or channels



OmniSphere



Conclusion

- Ambient networking
  - new exciting research problems
- Active networking may help solving many problems
  - dynamic customized behavior
- Many other issues
  - auto-configuration
  - location
  - security
  - context awareness
  - sensor networking

Acknowledgments

- Help from many persons gratefully acknowledged:
  - Justinian Oprescu
  - Laurentiu-Sorin Paun
  - Franck Rousseau
  - Philippe Jacquet et Kiki
  - Fred Baker
  - Christian Tschudin
  - Ramesh Govindan