Zero configuration networking

Outline

- What is Zero Configuration Networking
- Example applications
- Description of Protocols
- Available Implementations

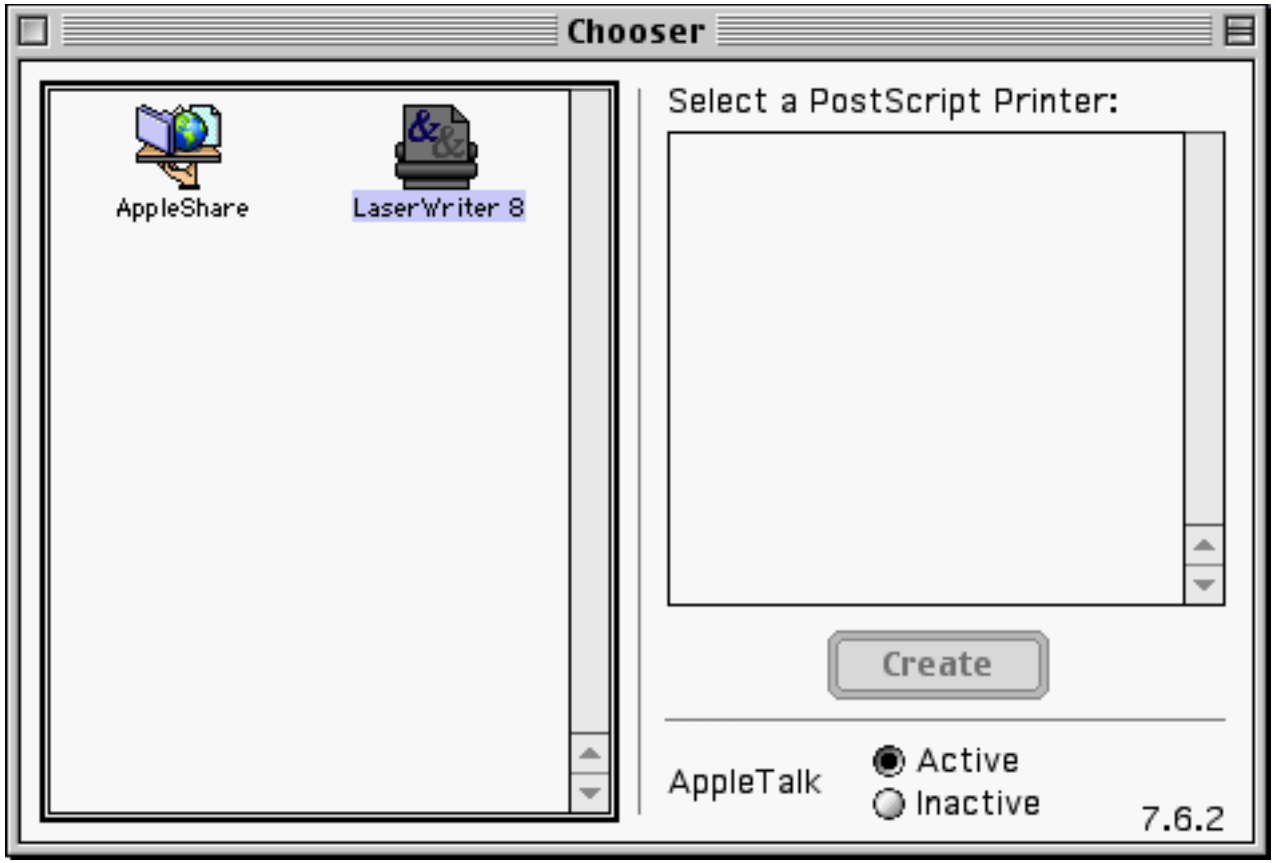
Outline

- What is Zero Configuration Networking
- Example applications
- Description of Protocols
- Available Implementations

Objective

- What is Zero Configuration Networking
  - Origins
  - What Problems does it Solve?
  - What Problems does it Not Solve?

Origins: AppleTalk



Origins: Internet Engineering Task Force (IETF)

- Zero Configuration Working Group
- Established September 1999
- Charter fragment:

> The goal of the Zero Configuration Networking (ZEROCONF) Working Group is to enable networking in the absence of configuration and administration.

What is required to be connected?

- To be phisically connected (wired or wireless)
- An IP address
- A name (to find the address)
- A way to found network services (shared FS, printer, ...)
- Typically we have DHCP, DNS, etc …
- … but in ad HOC network?

Problems Solved

- Automatic IP Address Configuration
- Server-free Hostname Resolution
- Directory-less Service Discovery

Problems Not Solved

- Security
  - Authenticity
  - Access Control

Outline

- What is Zero Configuration Networking
- Example applications
- Description of Protocols
- Available Implementations

Outline

- What is Zero Configuration Networking
- Example applications
- Description of Protocols
- Available Implementations

Example Applications

- Printer configuration
- Music sharing
- Peer to peer chat
- Gaming
- Apple TV with AirPlay

Existing Initiatives

- Apple’s Bonjour
- Jmdns
- Avahi
- zeroconf.sourceforge.net

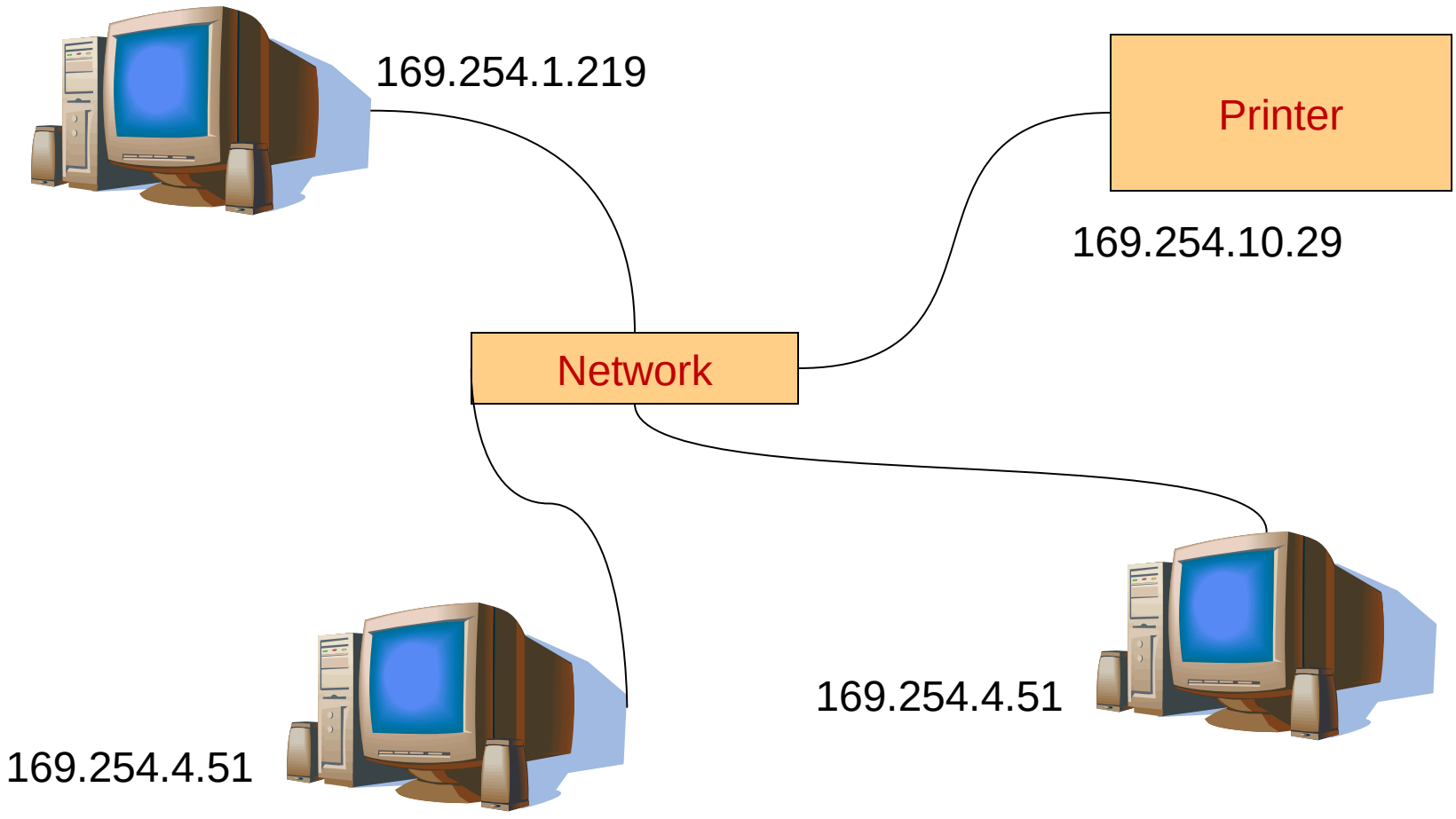
Agenda

- What is Zero Configuration Networking
- Example applications
- Description of Protocols
- Available Implementations

IP Address Autoconfiguraton

- In Absence of DHCP Server
- Link-local Addressing
- IPv4: pick random in 169.254/16
  - Address Defense
- Supported on Mac, Linux and Windows

Link-local Addressing



IP Address Autoconfiguraton

- Generally happens after DHCP times out
- Host picks Random Address
  - MAC used as seed
- Sends out ARP Request for Address
- If no Answer, Assumes Address
- If Answer, Host Starts Over

ARP probes(1)

Time	Source	Destination	Protocol	Info
3.703964	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 0.0.0.0
3.983703	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.004198	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 0.0.0.0
4.283867	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.304479	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 0.0.0.0
4.584088	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.884300	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.905167	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 169.254.187.245
5.184522	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 169.254.186.86
5.205780	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 169.254.187.245
5.485642	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 169.254.186.86
26.260885	dimsumthinking.local	Broadcast	ARP	Who has 169.254.186.86? Tell 169.254.187.245
26.260929	foo.local	Broadcast	ARP	169.254.186.86 is at 00:03:93:ef:c4:8c

```
Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    Sender MAC address: 00:03:93:ef:c4:8c (foo.local)
    Sender IP address: 0.0.0.0 (0.0.0.0)
    Target MAC address: 00:00:00:00:00:00 (00:00:00_00:00:00)
    Target IP address: 169.254.186.86 (169.254.186.86)
```

ARP probes(2)

- Claiming a link-local IP address
- Probing for address availability
  - At startup, reboot, wakeup, interface activation
- Announcing your address
- Defending your address
- Conflicts
  - Pick a new IP address

Hostname Resolution(1)

- In Absence of DNS Server
- Multicast DNS
  - DNS-like protocol
  - Every host has responder
  - Pseudo top level domain *.local*
- Hostnames picked by user
  - Hostname defense

MulticastDNS (mDNS)

- Send a nearly standard DNS request
- Request destination is not a server but :
  - Multicast group 224.0.0.251 for IPv4

- In detail:
  - Standard DNS client send UDP packet port 53
  - mDNS client send UDP packet port 5353

MulticastDNS (mDNS)

- After a multicast request ...
- One or more group members answer:
  - Single answer for name-address resolution
  - Multiple answer for service request
- All response are in multicast to allow information sharing for all clients

Hostname Resolution(2)

- Claiming a name
- Probing for name uniqueness
- Announcing your name
- Defending
- Conflicts
  - Solved by a lexicographic rule

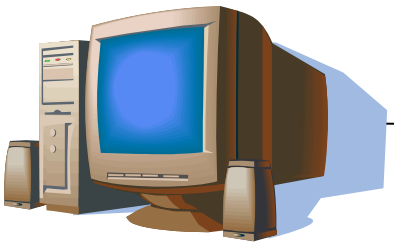
Service Discovery (1)

- Human-readable Service Names
- Runs on mDNS
- Query for Service Type returns list
  - Long-lived queries
- Service Name resolves to
  - hostname
  - port, add. info
  - IP address

Service Discovery (2)

- A way to found all instances of a target service
- The OS proposes to a user (or process) a list of instances implementing a service
- Example: printers, shared FS or directories
- Like in “Network neighborhood” of Windows

DNS-SD



PC_Bill
169.254.1.219

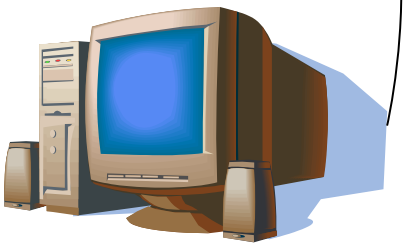
Bill's Files

Laserjet, Closet Under the Stairs

Printer

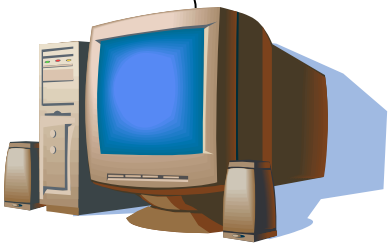
169.254.10.29
lj21569478

Network



PC_Larry
169.254.4.51

Larry's Tunes



Mac_Steve
169.254.4.51

Steve's Movies

Service Discovery with DNS (1)

- The DNS record set was extended to add such function [RFC 2782]:
  - A record with label “_http._tcp.example.com” lists all pair port:address for server http on tcp in example.com domain
- The DNS-SD adds an indirection level to provide a list of named services to the client

Service Discovery with DNS (2)

- Query with for a named service
  - Ex: “_ipp._tcp.example.com”
- Answer with a list of tuples: <instance>.<service>.<domain>
  - Ex: “ColorPrinter._ipp._tcp.example.com”, “BNPrinter._ipp._tcp.example.com”
- Results showed to user/process that picks one
- Solve name for chosen service instance

DNS SD general extension of DNS

- Implemented both in mDNS and standard DNS server
- Used in Internet to publish WAN services
- Require authentication mechanisms to restrict access to authorized users

Outline

- What is Zero Configuration Networking
- Example applications
- Description of Protocols
- Available Implementations

Implementations

- Bonjour: first implementation on market
  - Apple Trademark
  - Introduced in Mac OS X 10.2
  - Incorporated in iTunes for Windows
  - Released For Windows, POSIX platforms, Java
- Linux: “avahi” o “mDNSResponder ”

Conclusions

- Easy to use
- Cross-platform
- Dynamic
- Configuration-free

More Information

- http://developer.apple.com/networking/bonjour/
  - Developer Info
- http://www.apple.com/macosx/features/bonjour/
  - Marketing
- http://www.zeroconf.org/ (IETF site)
- http://www.lists.apple.com/mailman/listinfo/bonjour-dev