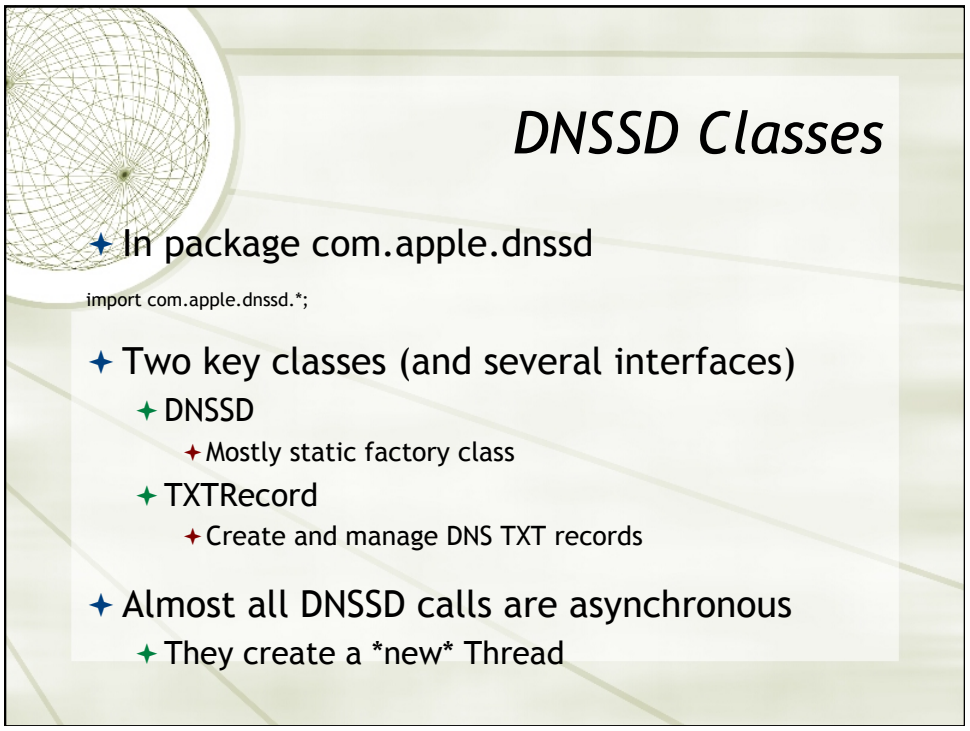
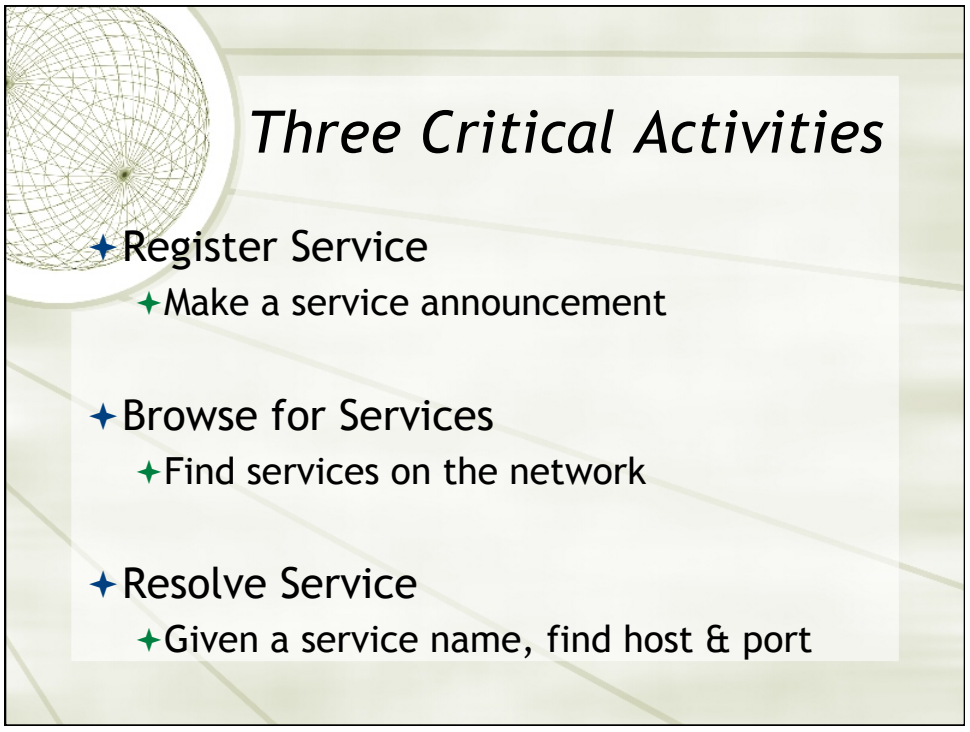
DNS-SD Register and Resolve in Java

Info 341 Networking and Distributed Applications

Project Part 3

- Handout


DNSSD Classes
In package com.apple.dnssd
import com.apple.dnssd.*;
Two key classes (and several interfaces)
DNSSD
Mostly static factory class
TXTRecord
Create and manage DNS TXT records
Almost all DNSSD calls are asynchronous
They create a *new* Thread


Three Critical Activities
Register Service
Make a service announcement
Browse for Services
Find services on the network
Resolve Service
Given a service name, find host & port

Browse Service

- Last time
  - Concrete example of building a browser
- What do you have to do?

Browse Service

- Last time
  - Concrete example of building a browser
- What do you have to do?
  - Implement BrowseListener Interface
  - Create a browser thread with the DNSSD.browse() factory method

Announce/Register a Service

- Registering a service is similar
  - Implement RegisterListener Interface
  - Create a register thread with the DNSSD.register() factory method
  - Let's look …

Service Registration

- Two forms of service registration
  - Simple form

```
DNSSDRegistration reg = null;
reg = DNSSD.register("Foo Service", "_fake._tcp", 10201, myRegisterListener);
```

  - Slightly more complex form

```
DNSSDRegistration reg = null;
reg = DNSSD.register(0, DNSSD.ALL_INTERFACES, name,
    sd_service, null, null, port, txtRec, myRegisterListener);
```

Register Listener Interface

- Know if the service was registered correctly

```
public class MyRegListener implements RegisterListener {
    …
    public void serviceRegistered (DNSSDRegistration reg, int flags, String serviceName,
        String regType, String domain) {
        …
    }
    public void operationFailed (DNSSDService service, int errorCode) {
        …
    }
    <other methods>
}
```

DNSSD - Quick Test

- How would you unregister, unannounce a service?
  - Suppose …

```
DNSSDRegistration reg = null;
reg = DNSSD.register("Foo Service", "_fake._tcp", 10201, myRegisterListener);
```

DNSSD - Quick Test

- How would you unregister, unannounce a service?
  - Suppose …

```
DNSSDRegistration reg = null;
reg = DNSSD.register("Foo Service", "_fake._tcp", 10201, myRegisterListener);
```

  - Because reg is a thread you can just call the stop() method on reg and you're done.

Resolving a Service

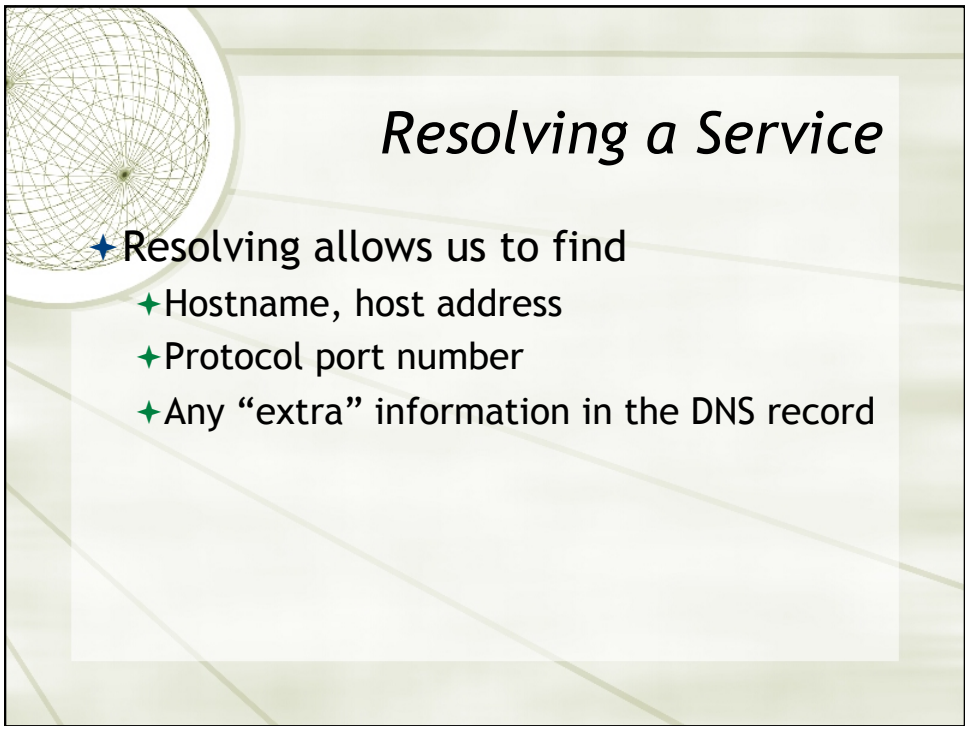
- We can Register/Announce services
- We can Browse/Find services
- Does this mean we can use a service?

Resolving a Service

- We can Register/Announce services
- We can Browse/Find services
- Does this mean we can use a service?
  - The BrowseListener returns what?

Resolving a Service

- We can Register/Announce services
- We can Browse/Find services
- Does this mean we can use a service?
  - The BrowseListener returns what?
    - serviceFound(DNSSDService br, int flags, int ifIndex, String serviceName, String regType, String domain)
  - What do we need before we can use a service?


Resolving a Service
Resolving allows us to find
Hostname, host address
Protocol port number
Any “extra” information in the DNS record


Resolve/Query
Resolve
Initiate a request to get information necessary for contacting the service
DNSSDService resolver = null;
resolver = DNSSD.resolve(0, DNSSD.ALL_INTERFACES, name, sd_service, "local",
myResolveListener);

Resolve Interface

- Resolve Interface
  - Return information necessary for contacting the service

```
public class MyResolveListener implements ResolveListener {
      …
    public void serviceResolved (DNSSDService serv, int flags, int ifIndex, String fullName,
            String hostName, int port, TXTRecord rec) {
            …
    }
    public void operationFailed (DNSSDService serv, int errorCode) {
            …
    }
    <other methods>
}
```

Things to Notice

- Across Register, Browse, & Resolve
  - Initiate asynchronous services (threads)
    - Must eventually call stop()
  - Pass some form of Listener object
  - Listener interfaces implement callback methods
  - All Listeners implement “operationFailed”


How can we explore this?