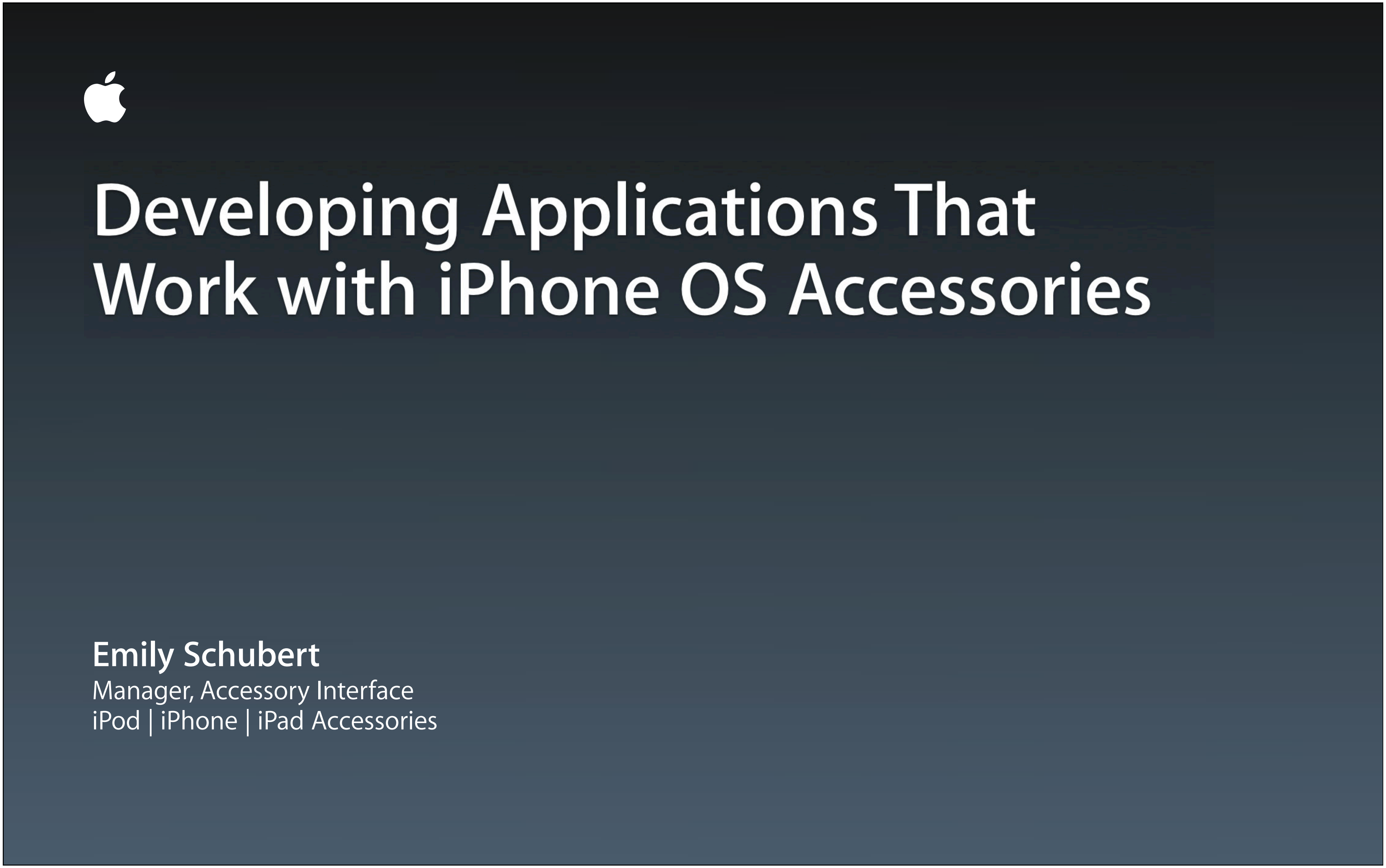


Developing Applications That Work with iPhone OS Accessories

Emily Schubert
Manager, Accessory Interface
iPod | iPhone | iPad Accessories

Introduction

- iPhone OS provides interfaces for applications to interact with external devices
- New opportunities in iOS 4

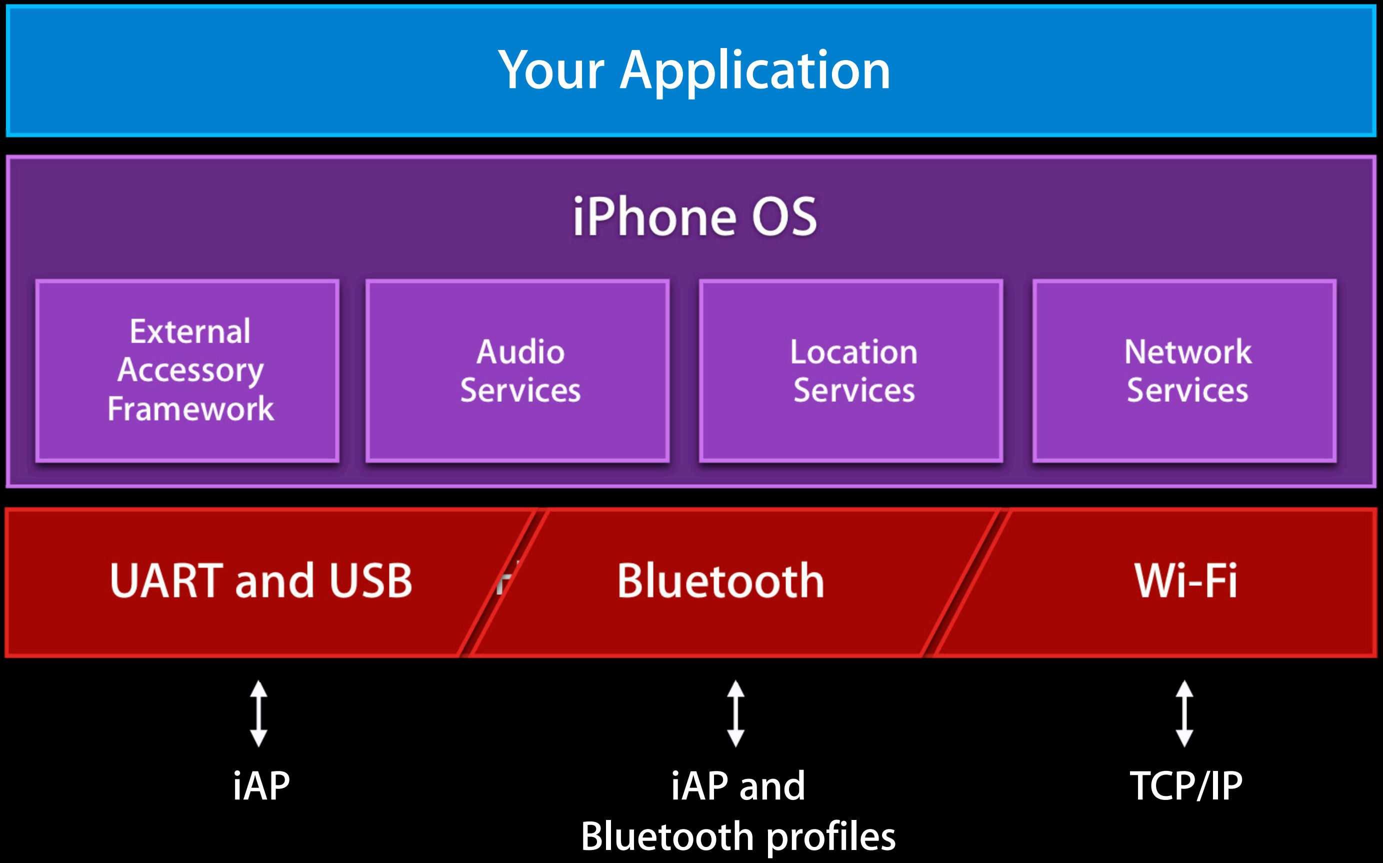
iPhone OS Accessories

30-pin

Bluetooth

Wi-Fi

iPhone OS Accessories



Your Application

iPhone OS

External Accessory Framework

Audio Services

Location Services

Network Services

UART and USB

Bluetooth

Wi-Fi

iAP

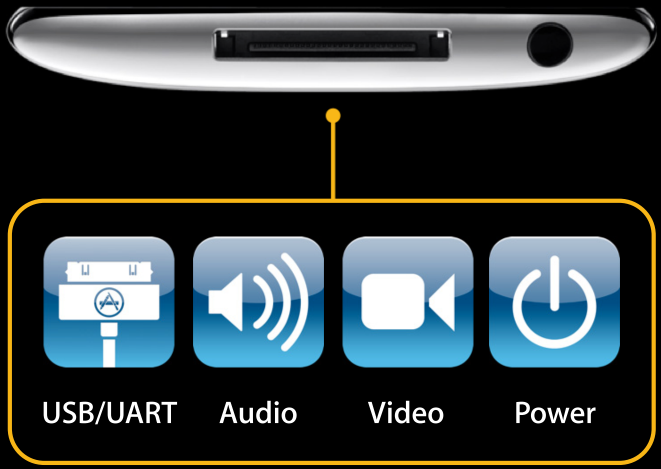
iAP and Bluetooth profiles

TCP/IP

Dock Connector

30-Pin Dock Connector

- Standard since 2003 on iPod, iPhone, and iPad
- Available interfaces
- Reference designs from several developers available



USB/UART

Audio

Video

Power

iPod Accessory Protocol

iPod Accessory Protocol (iAP)

- Allows accessories to communicate with and control iPod, iPhone, and iPad
- Details available through MFi Program
  http://developer.apple.com/mfi/

Made for iPod

Made for iPhone

Made for iPad

Features Available to Apps

- Standard command sets
  - Audio
  - Video
  - Location
  - Multimedia remote control
  - Keyboards
  - Accessibility
- Custom protocols over iAP
  - Defined by app or accessory developer

Audio

- Audio is routed to/from your app using Core Audio
- Supported audio features

Output	analog	line out, headset
	digital	USB, Bluetooth
Input	analog	line in, headset
	digital	USB

Audio

Supported models

	iPod touch	iPhone	iPad
Line out	✓	✓	✓
Headset out	✓	✓	✓
USB audio out	✓	✓	✓
Bluetooth audio out	2nd, 3rd generation	3G, 3GS, 4	✓
Line in	✓	3G, 3GS	
Headset in	2nd, 3rd generation	✓	✓
USB audio in	3rd generation	3GS, 4	✓

Video

- iPhone OS 3.2 and iOS 4
- Access external display using UIScreen

	iPod touch	iPhone	iPad
Component	✓	✓	✓
S Video	✓	✓	✓
Composite	✓	✓	✓
VGA Adapter		iPhone 4	✓

Location

- An accessory can provide location information to the iPhone OS device
  - GPS information
  - Heading
  - Information is then available to your app via Core Location

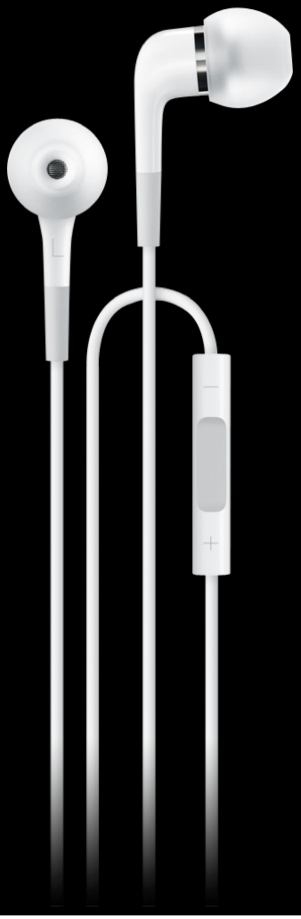
Multimedia Remote Control

iOS 4

- Your app can now receive multimedia remote control commands from an accessory

▶ ⏸ ⏹ ⏮ ⏭ ⏪ ⏩

- Also available via the headphone remote and mic system



Keyboards

- More keyboards can now connect to iPhone OS devices
  - 30-pin keyboards
  - Standard Bluetooth keyboards
- In UIKit: UIKeyInput and UITextInput
  - See Text and Web Programming Guide for details
- Opportunity for interesting form factors

Accessibility

iOS 4

- Accessories can now leverage VoiceOver to control every app
- Gives control to users who cannot manipulate the device and to those who cannot see it
  - Move to <x,y>
  - Touch event <x,y>
  - Move to next, previous, etc.
- Make sure your app is accessible
  - Implement UIAccessibility API

Custom Protocols Over iAP

iOS 4

- With iPhone OS 3, apps were able to communicate with accessories using a custom protocol
- New in iOS 4
  - Ability for multiple accessories to communicate with the same app
  - Remote launch of app
  - Protocols can be declared optional

Summary

- Broad accessory support is available via iAP
- New features for iOS 4
  - Multimedia remote control
  - Keyboard
  - Accessibility
  - Features for custom protocols over iAP

Referenced Sessions

Simplifying Networking Using Bonjour	Nob Hill Wednesday 10:15AM
Accessibility on iPhone OS	Nob Hill Wednesday 4:30PM
Fundamentals of Digital Audio for Mac OS X and iPhone OS	Mission Wednesday 10:15AM

iPhone OS Accessories That Communicate Over Bluetooth

Brian Tucker
Senior Software Engineering Manager
Mobile Bluetooth Technologies

Who, What, Goal?

- Who is this for?
  - Bluetooth accessory manufacturers
  - iPhone application developers
- What do you get?
  - Better understanding of iPhone OS Bluetooth
  - Tips and tricks in making a better Bluetooth accessory
- Goal
  - Create the best possible Bluetooth customer experience!

Bluetooth on iPhone OS

- What’s new in iOS 4
- State of Bluetooth on iPhone OS
- Three areas to improve Bluetooth accessory interaction with iPhone OS

What's New in iOS 4

- Keyboard support
  - HID profile

- MPEG-2 AAC LC audio codec for A2DP
  - Sampling rate: 44.1 kHz
  - Channels: Stereo
  - Bit rate: 128 kbps
  - VBR: Yes

What’s New in iOS 4

- Voice commands over Bluetooth
- In-band ringtones
  - Ringtone on phone is now heard on Bluetooth accessory
- Braille keyboard support
- Support for multiple handsfree and A2DP connections
- iPhone volume control of A2DP audio streams

State of iPhone OS Bluetooth

- Standard protocols
  - Hands-Free Profile (HFP)
  - Phone Book Access Profile (PBAP)
  - Advanced Audio Distribution Profile (A2DP)
  - Human Interface Device Profile (HID)
  - Personal Area Networking (PAN)
  - Device ID Profile (DID)
- Custom protocols
  - iPod Accessory Protocol

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall
Bluetooth

Working
with GameKit

Wi-Fi
Coexistence

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall Bluetooth

Working with GameKit

Wi-Fi Coexistence

Overall Bluetooth

- Implement Bluetooth 2.1 + EDR
- Support Secure Simple Pairing (SSP)
- Use Extended Inquiry Response for Friendly Name
- Implement Device ID Profile (DID)

Overall Bluetooth

- Support role switch without complaint
  - Embrace slave mode
- Support sniff
  - Variant intervals
  - iPhone OS will ask for sniff mode when appropriate
- Disconnect before power off

Profiles

- Advanced Audio Distribution Profile (A2DP)
  - Support high SBC bit rates: ~330 kbps (53 Bit Pool)
  - Implement AAC
  - Support AVDTP 1.3
    - Measured latency command
- Handsfree Profile (HFP)
  - Support Voice Recognition Activation (BVRA)
  - Support eSCO packets

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall Bluetooth

Working with GameKit

Wi-Fi Coexistence

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall Bluetooth

Working with GameKit

Wi-Fi Coexistence

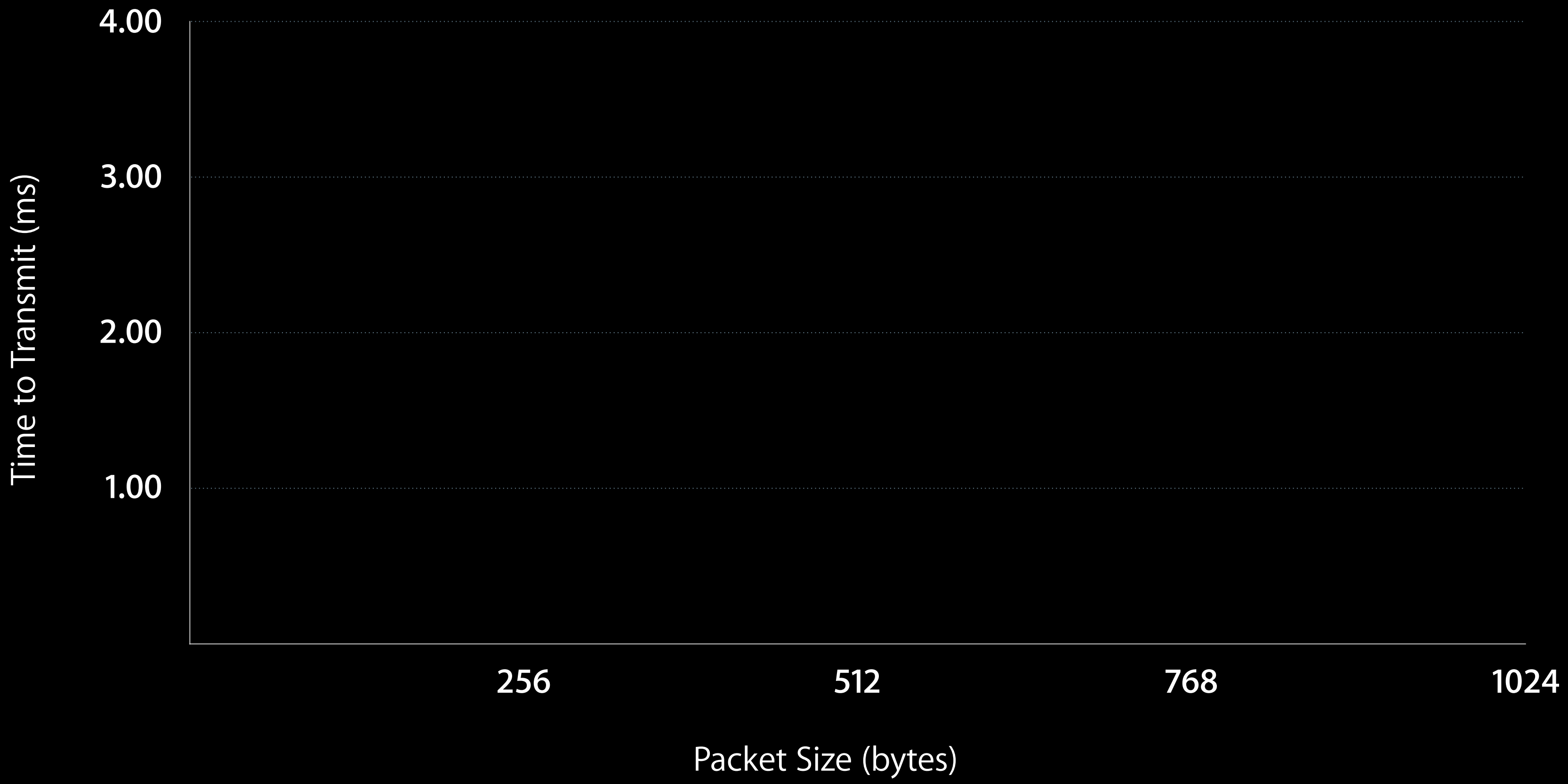
Working with Device to Device

- Use client/server when appropriate
- Only browse when browsing!
- Only advertise when advertising!

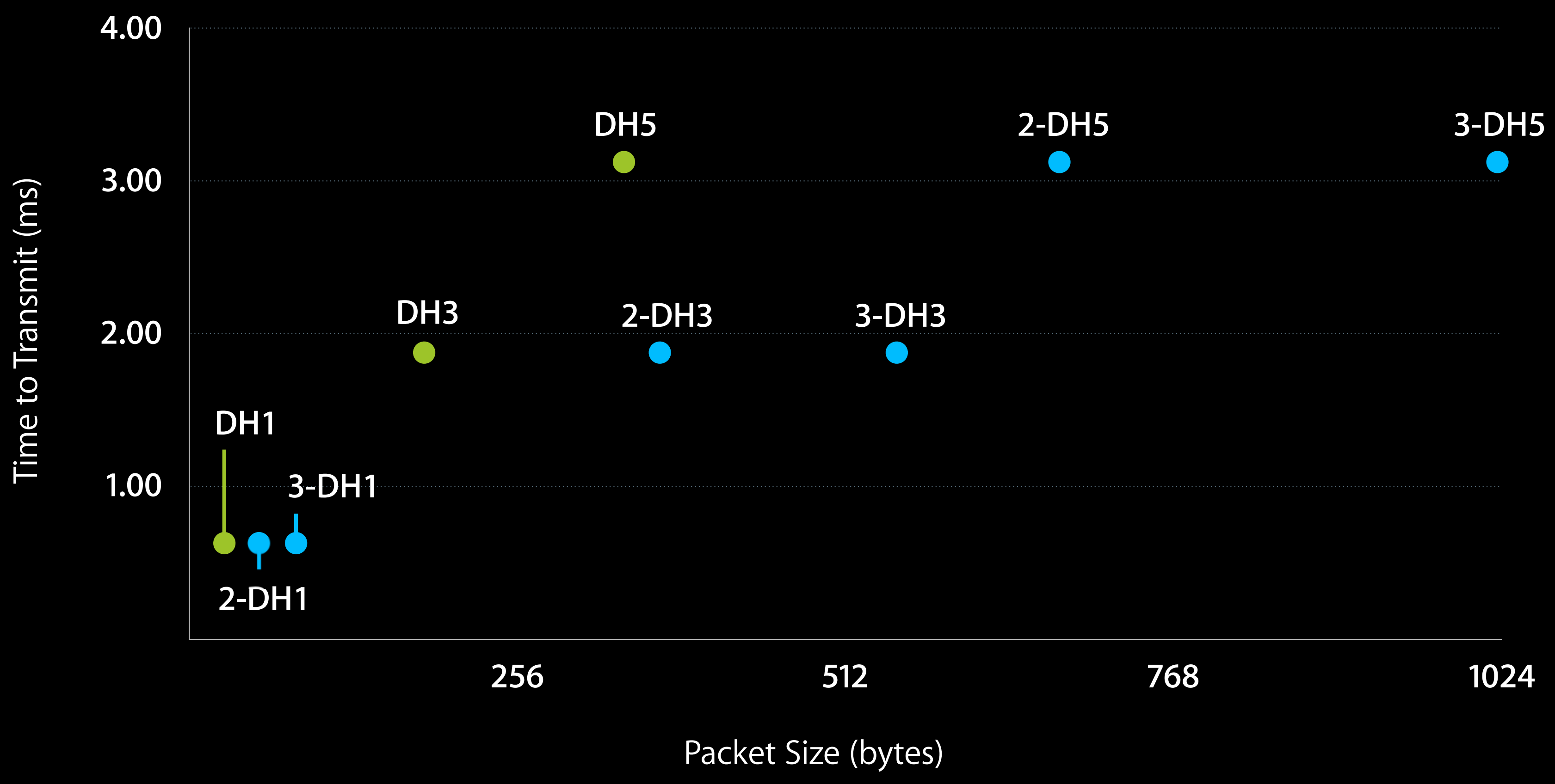
Working with Device to Device

- Maximize the wireless network
  - Understanding Bluetooth packets helps to understand Bluetooth performance
  - Wireless network ≠ Wired network

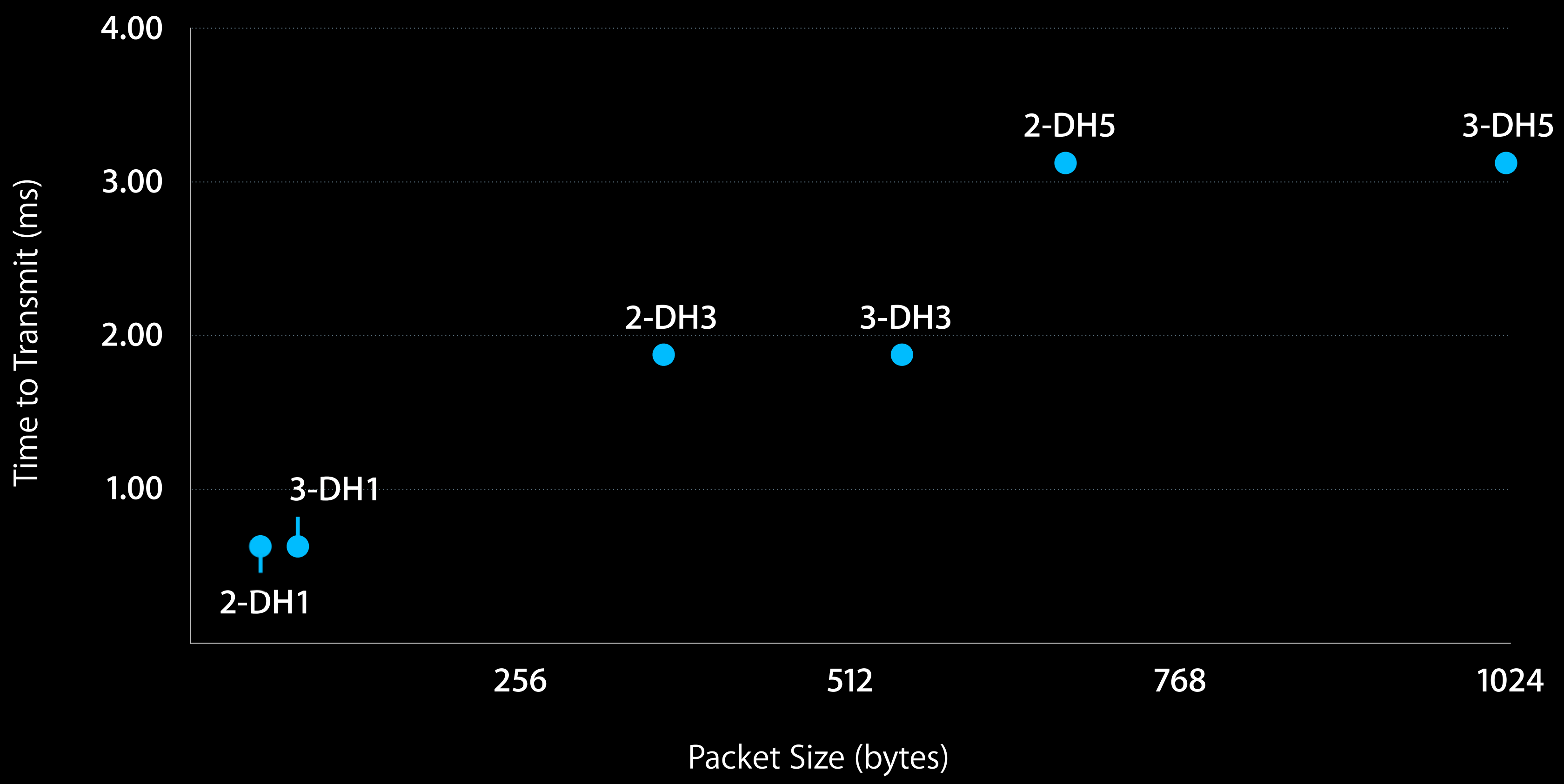
Bluetooth Packets

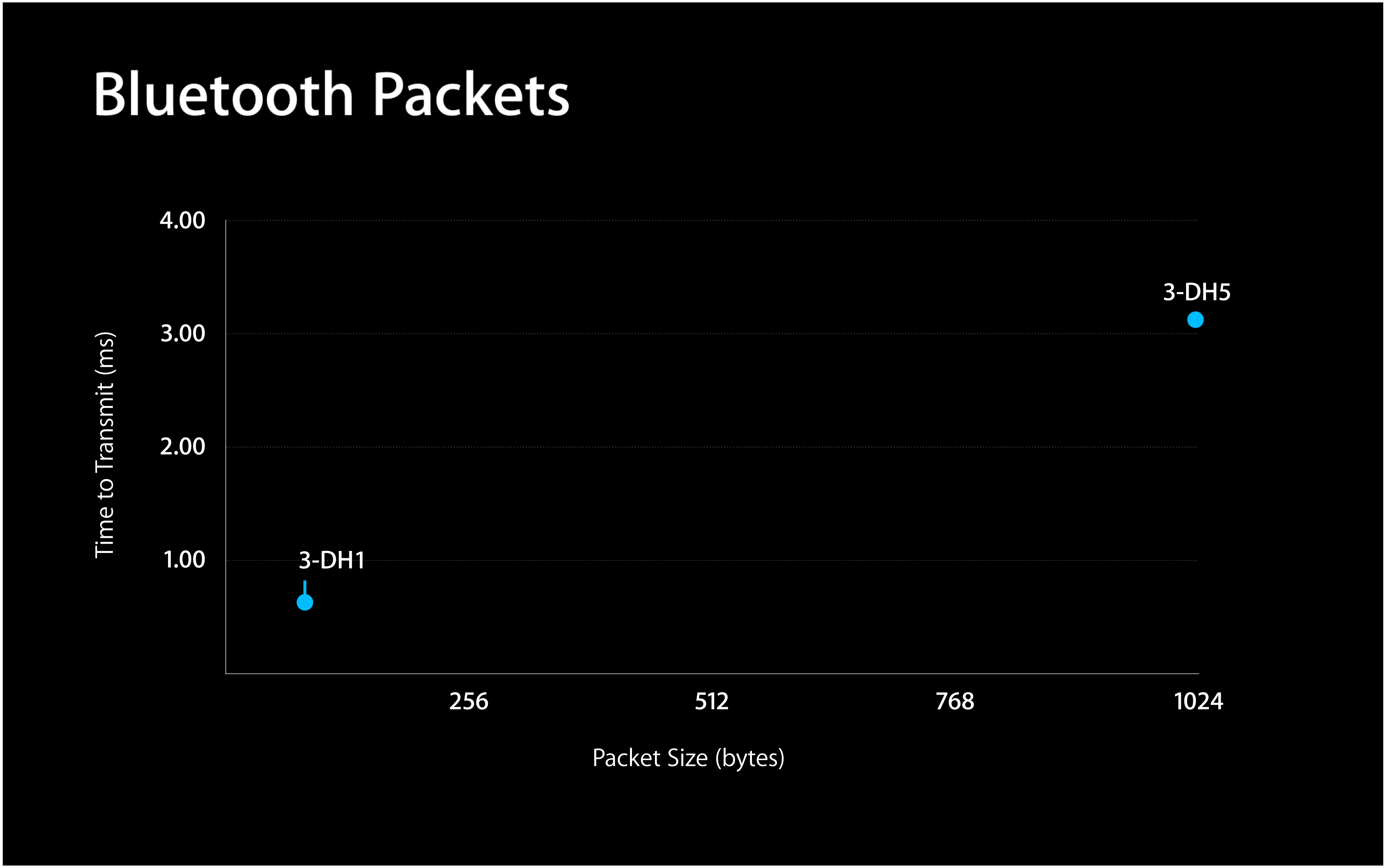


Bluetooth Packets



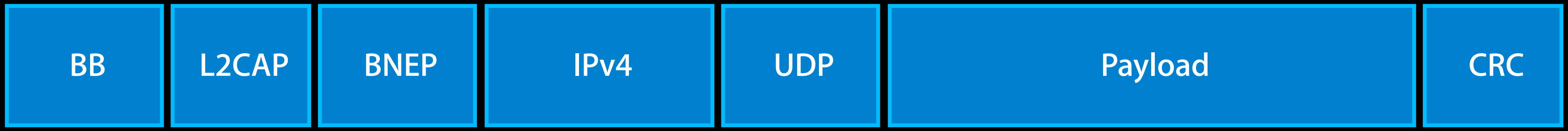
Bluetooth Packets





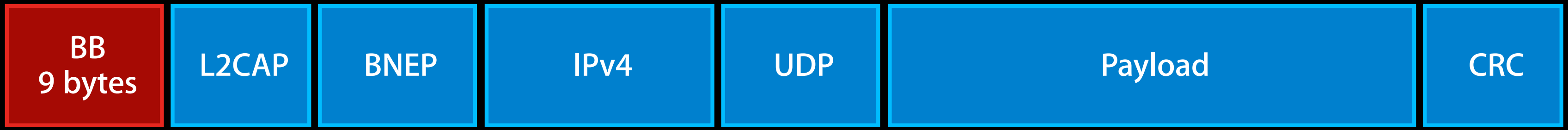
Bluetooth Packets

Packet Anatomy 101-3-DH1



BB | L2CAP | BNEP | IPv4 | UDP | Payload | CRC

Packet Anatomy 101-3-DH1



BB
9 bytes

L2CAP

BNEP

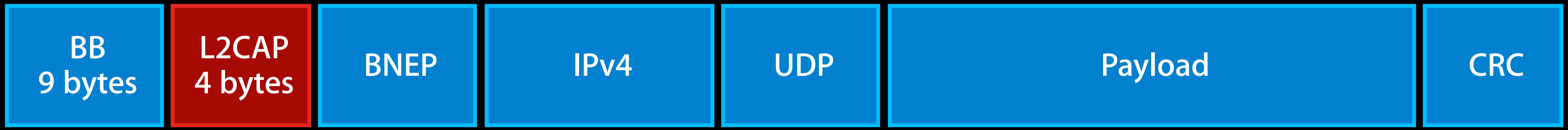
IPv4

UDP

Payload

CRC

Packet Anatomy 101-3-DH1



BB
9 bytes

L2CAP
4 bytes

BNEP

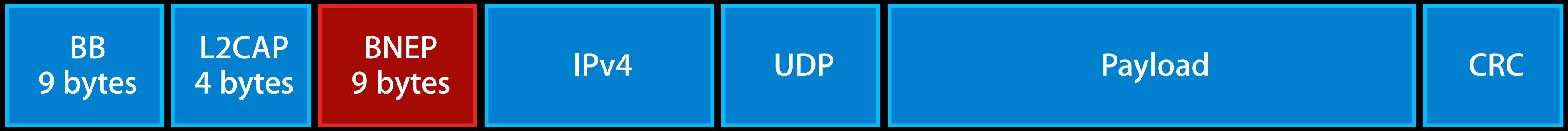
IPv4

UDP

Payload

CRC

Packet Anatomy 101-3-DH1



BB
9 bytes

L2CAP
4 bytes

BNEP
9 bytes

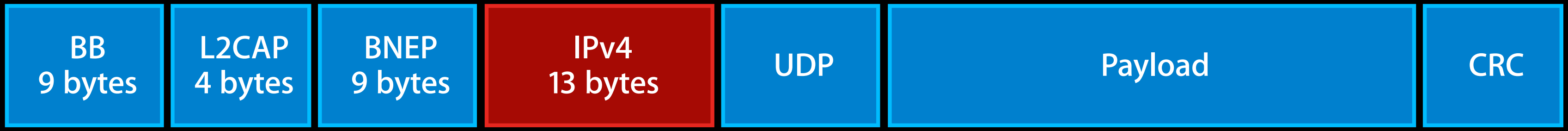
IPv4

UDP

Payload

CRC

Packet Anatomy 101-3-DH1



BB
9 bytes

L2CAP
4 bytes

BNEP
9 bytes

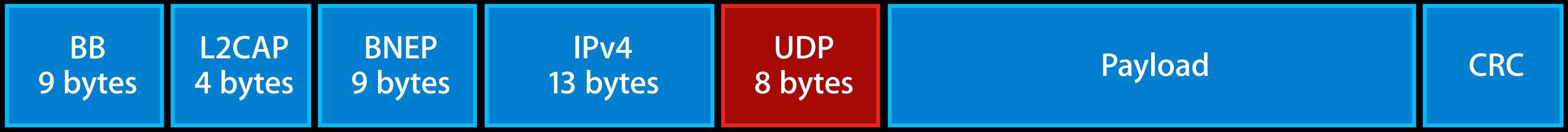
IPv4
13 bytes

UDP

Payload

CRC

Packet Anatomy 101-3-DH1



BB
9 bytes

L2CAP
4 bytes

BNEP
9 bytes

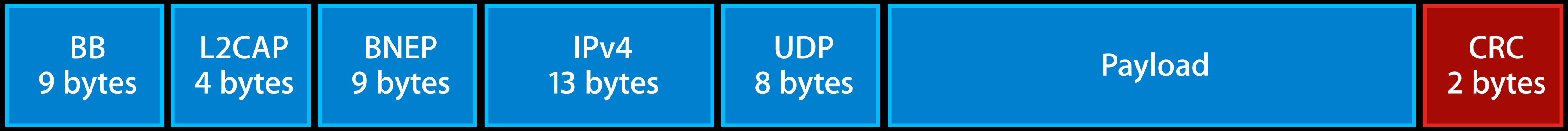
IPv4
13 bytes

UDP
8 bytes

Payload

CRC

Packet Anatomy 101-3-DH1



BB
9 bytes

L2CAP
4 bytes

BNEP
9 bytes

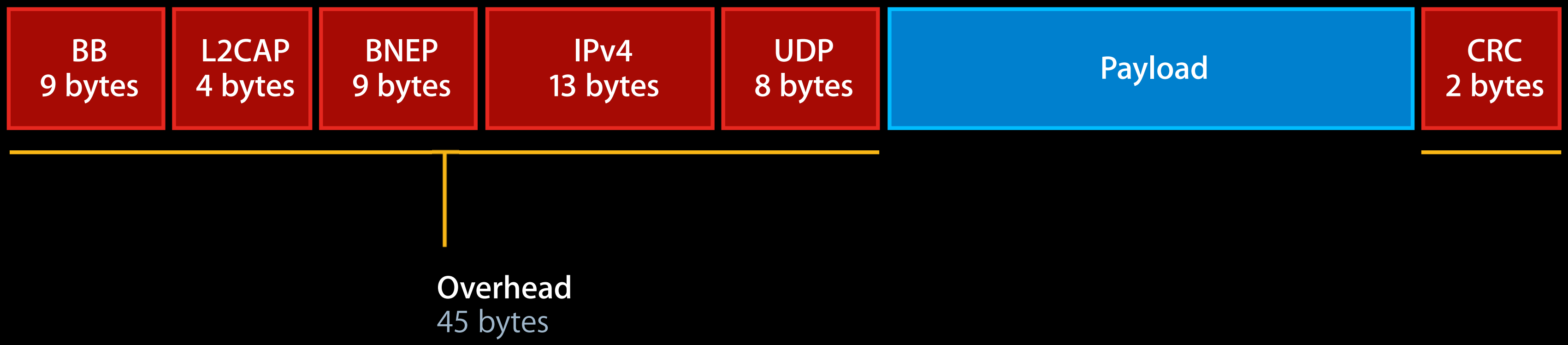
IPv4
13 bytes

UDP
8 bytes

Payload

CRC
2 bytes

Packet Anatomy 101-3-DH1



BB
9 bytes

L2CAP
4 bytes

BNEP
9 bytes

IPv4
13 bytes

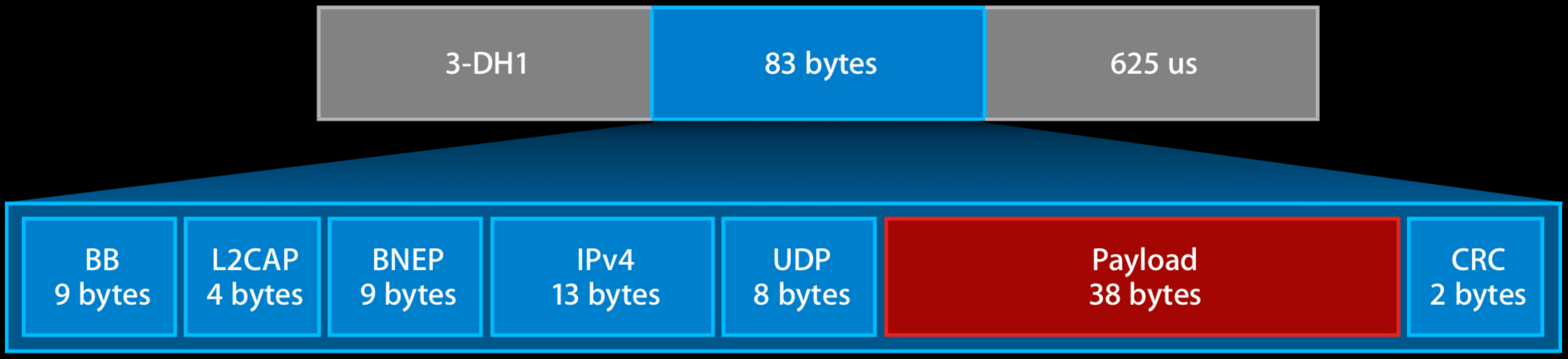
UDP
8 bytes

Payload

CRC
2 bytes

Overhead
45 bytes

Packet Anatomy 101-3-DH1



3-DH1

83 bytes

625 us

BB
9 bytes

L2CAP
4 bytes

BNEP
9 bytes

IPv4
13 bytes

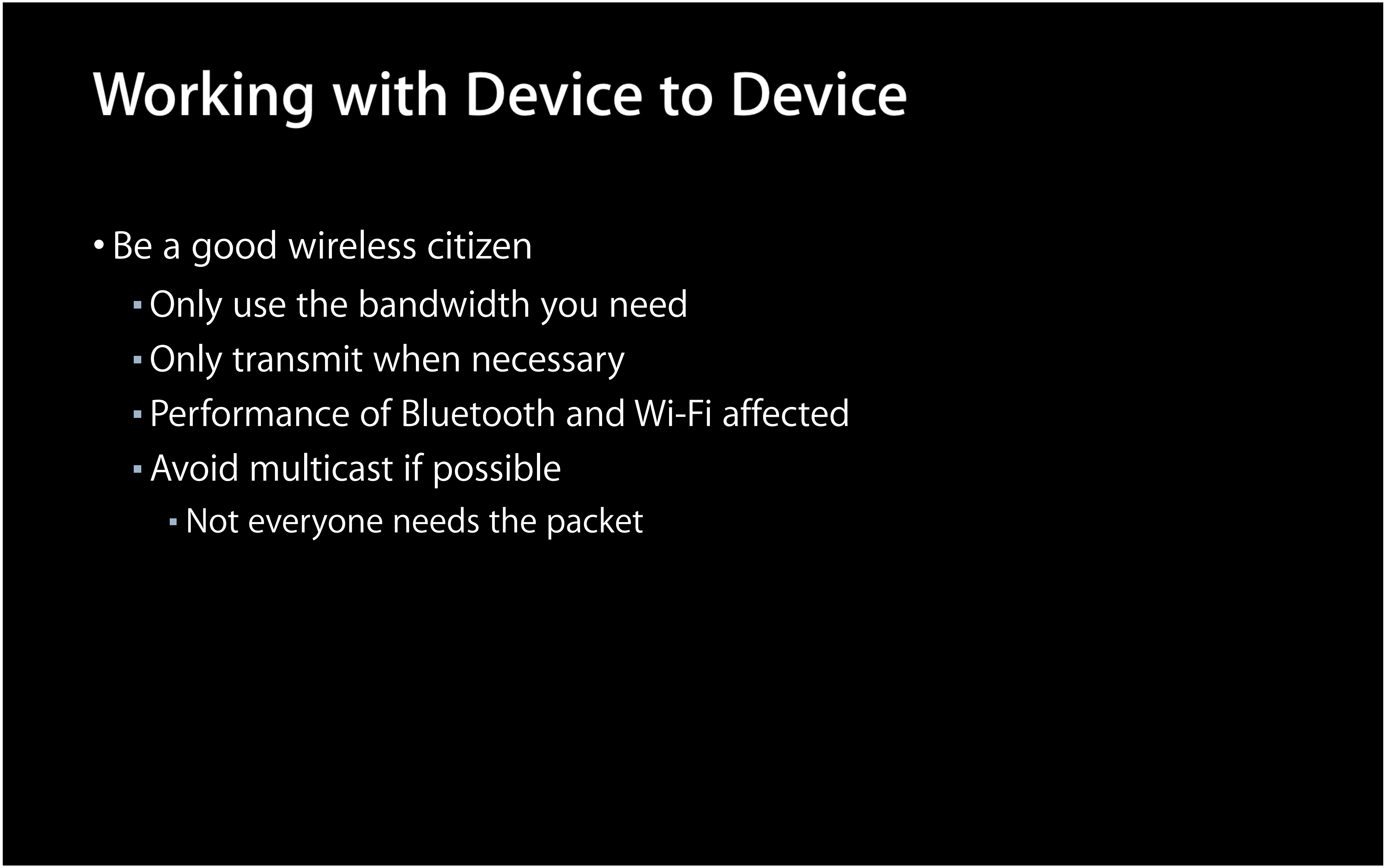
UDP
8 bytes

Payload
38 bytes

CRC
2 bytes

Working with Device to Device

- High bandwidth
  - Maximum payload of ~980 bytes
  - Maximum duty cycle of ~5 ms
- High packet count
  - Maximum payload of ~30 bytes
  - Maximum duty cycle of ~2.5 ms



Working with Device to Device

- Be a good wireless citizen
  - Only use the bandwidth you need
  - Only transmit when necessary
  - Performance of Bluetooth and Wi-Fi affected
  - Avoid multicast if possible
    - Not everyone needs the packet

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall Bluetooth

Working with GameKit

Wi-Fi Coexistence

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall Bluetooth

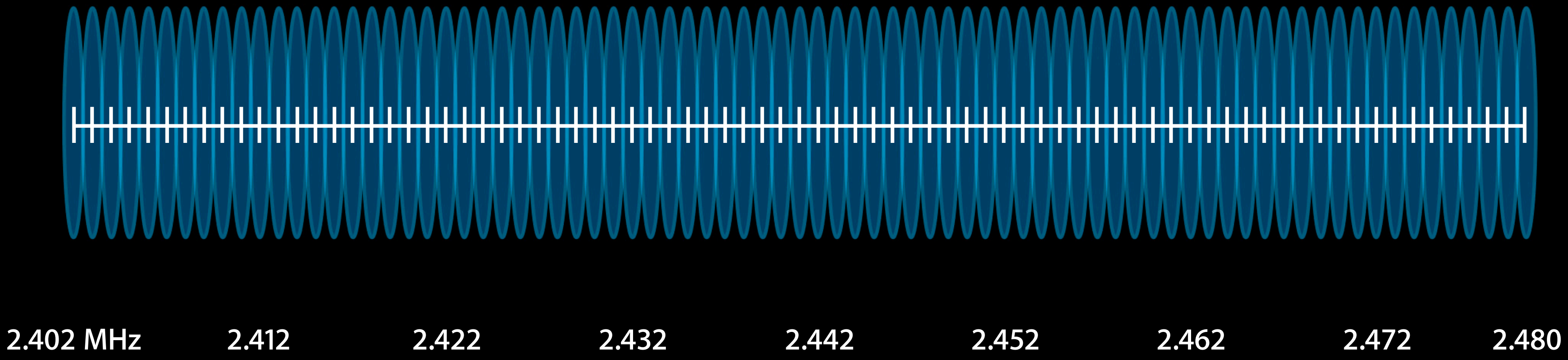
Working with GameKit

Wi-Fi Coexistence

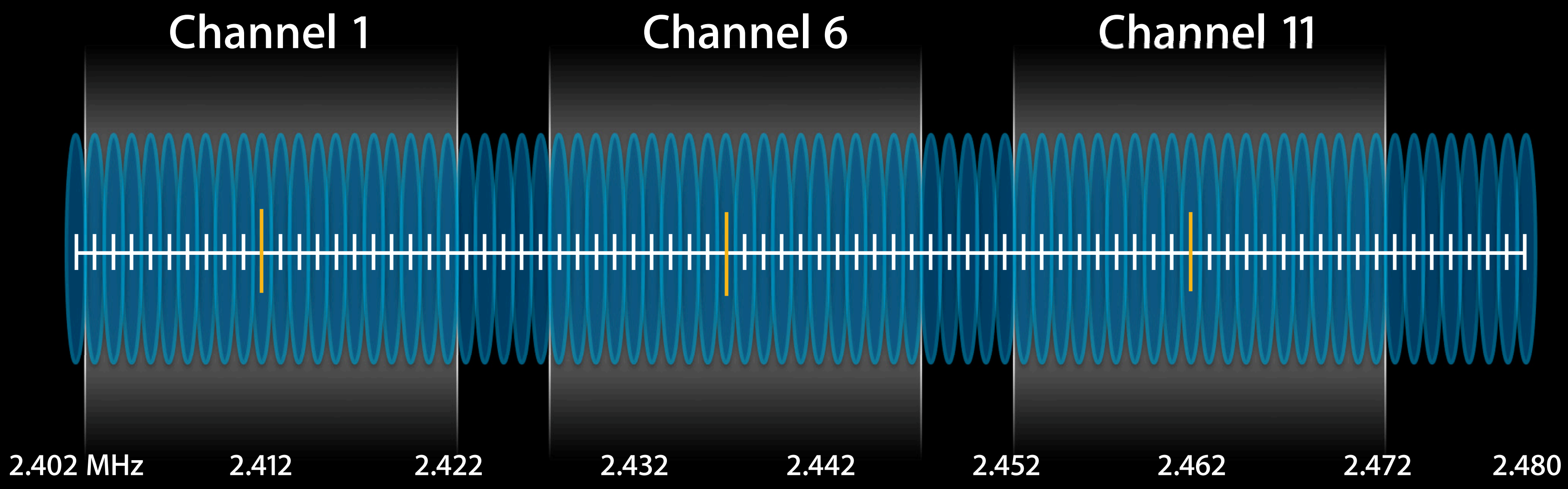
Coexistence

- Wi-Fi and Bluetooth must share the road
  - Industrial, scientific, and medical radio band - ISM
  - 2.4 - 2.5 GHz
  - Wi-Fi uses thirteen 22 MHz channels spaced 5 MHz apart
  - Bluetooth uses seventy-nine 1 MHz channels spaced 1 MHz apart

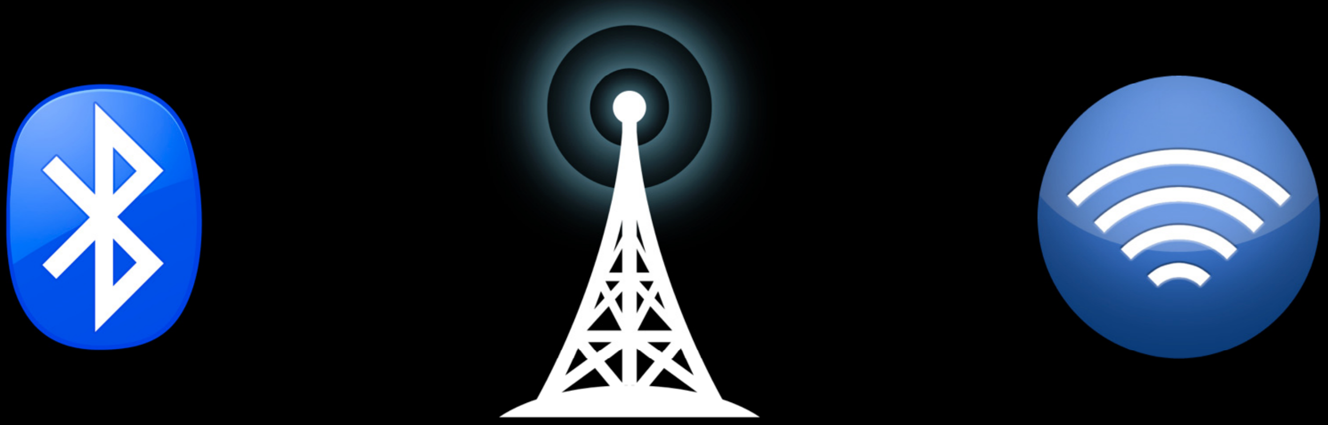
Bluetooth Channels



Popular Wi-Fi Channels

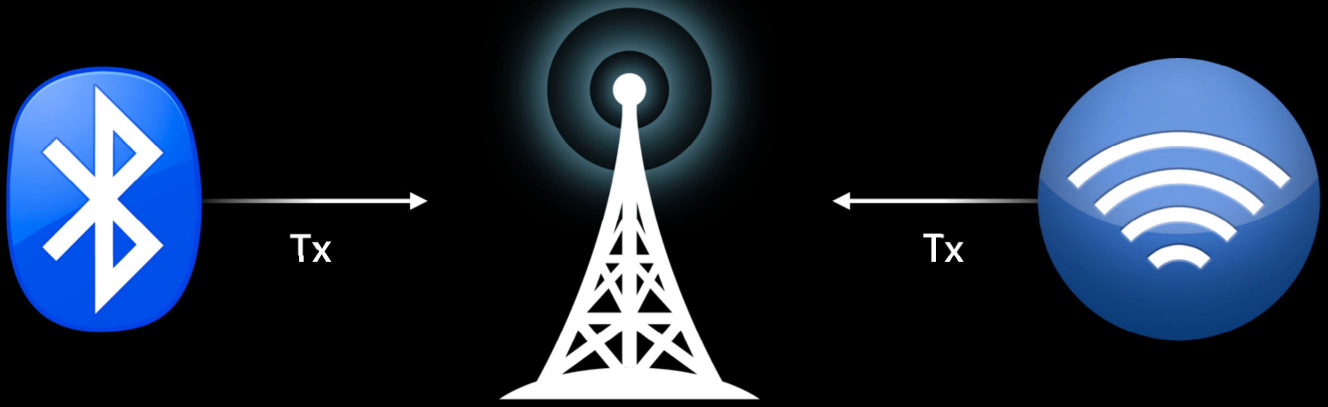


Sharing the Antenna



2.4 GHz Antenna

Sharing the Antenna

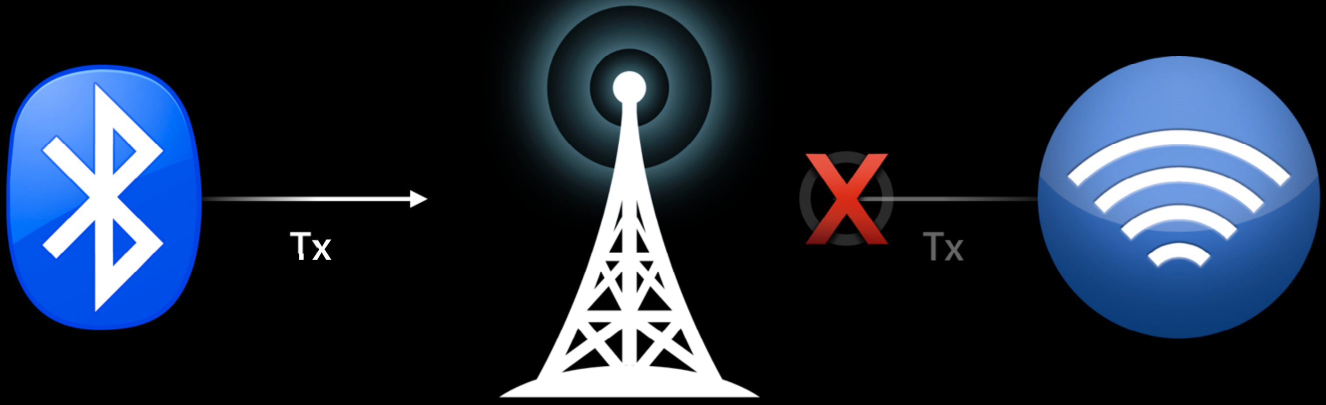


Tx

Tx

2.4 GHz Antenna

Sharing the Antenna

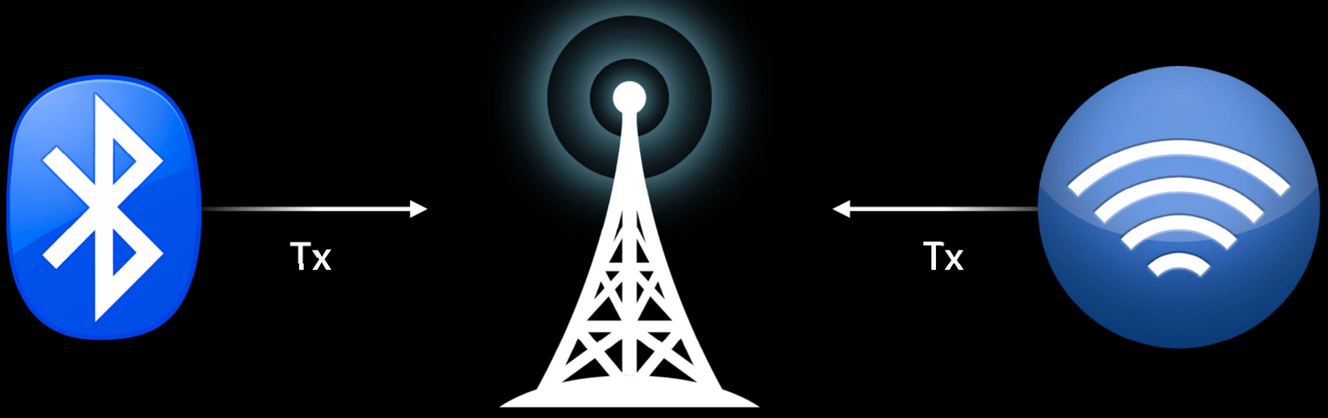


Tx

Tx

2.4 GHz Antenna

Sharing the Antenna

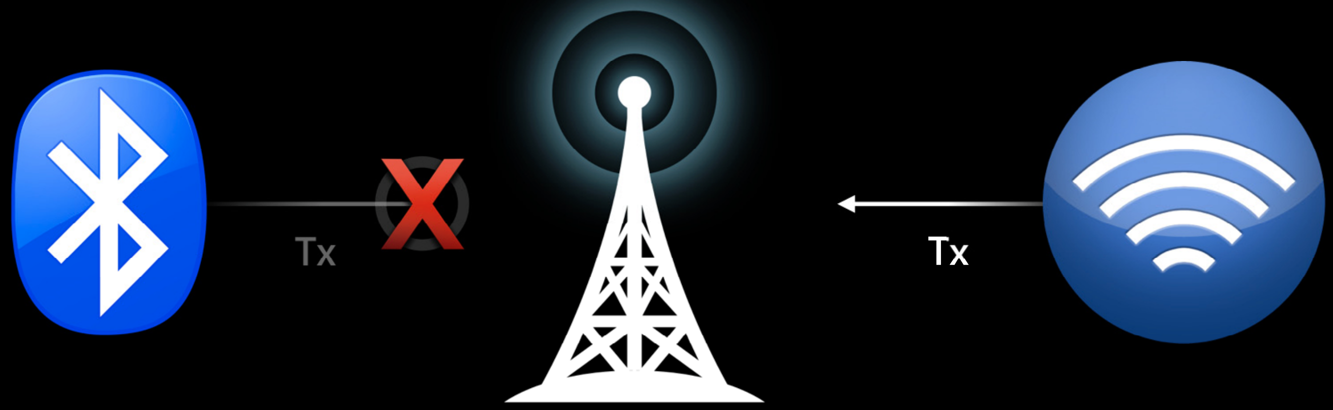


Tx

Tx

2.4 GHz Antenna

Sharing the Antenna

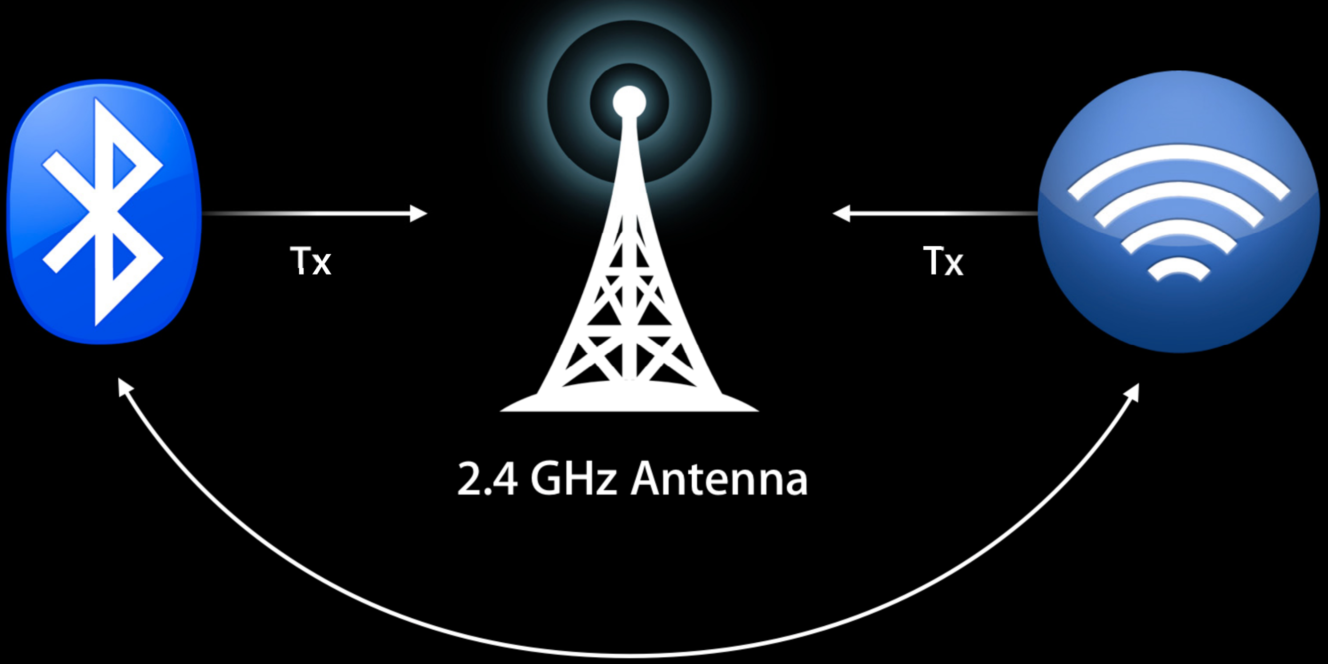


Tx

Tx

2.4 GHz Antenna

Sharing the Antenna



Tx

Tx

2.4 GHz Antenna

Antenna Arbitration

Help Us Help You

- Implement Bluetooth 2.1 + EDR
- Support sniff
- Optimize packet usage
- Support lower bandwidth codecs
  - AAC

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall Bluetooth

Working with GameKit

Wi-Fi Coexistence

Three Areas to Improve Accessory Interaction with iPhone Bluetooth

Overall Bluetooth

Working with GameKit

Wi-Fi Coexistence

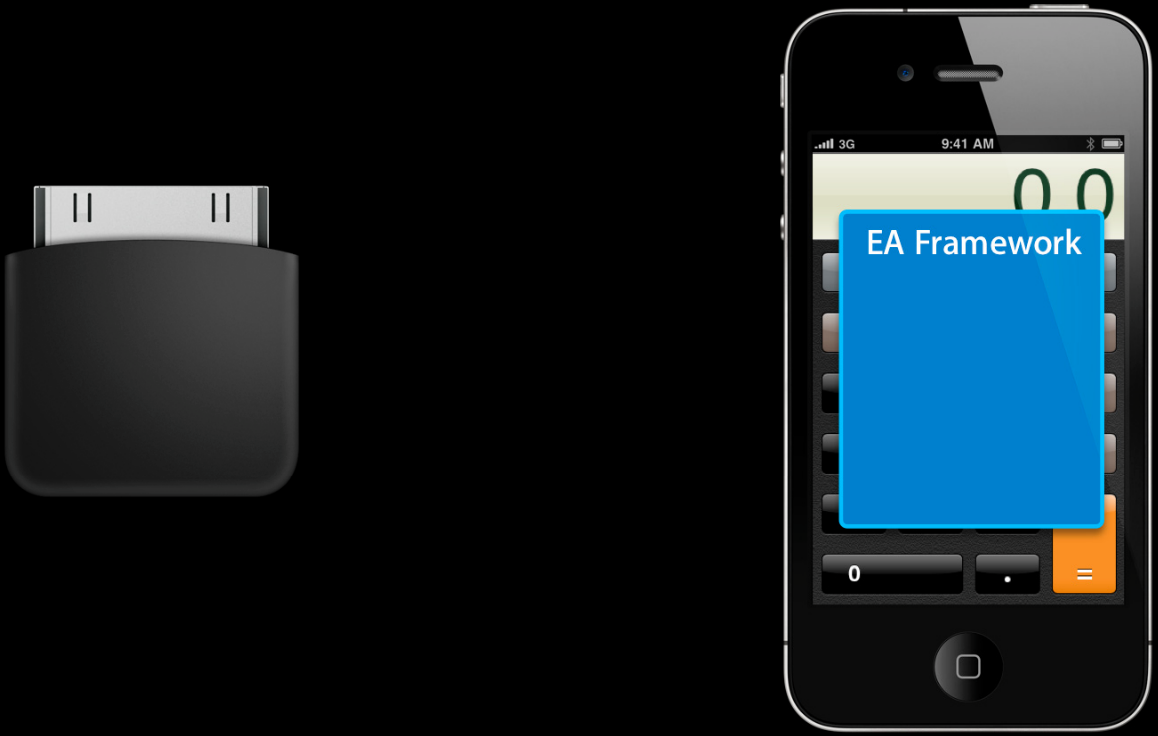
External Accessory Framework

Paul Holden
iPhone Applications Engineer

External Accessory Framework

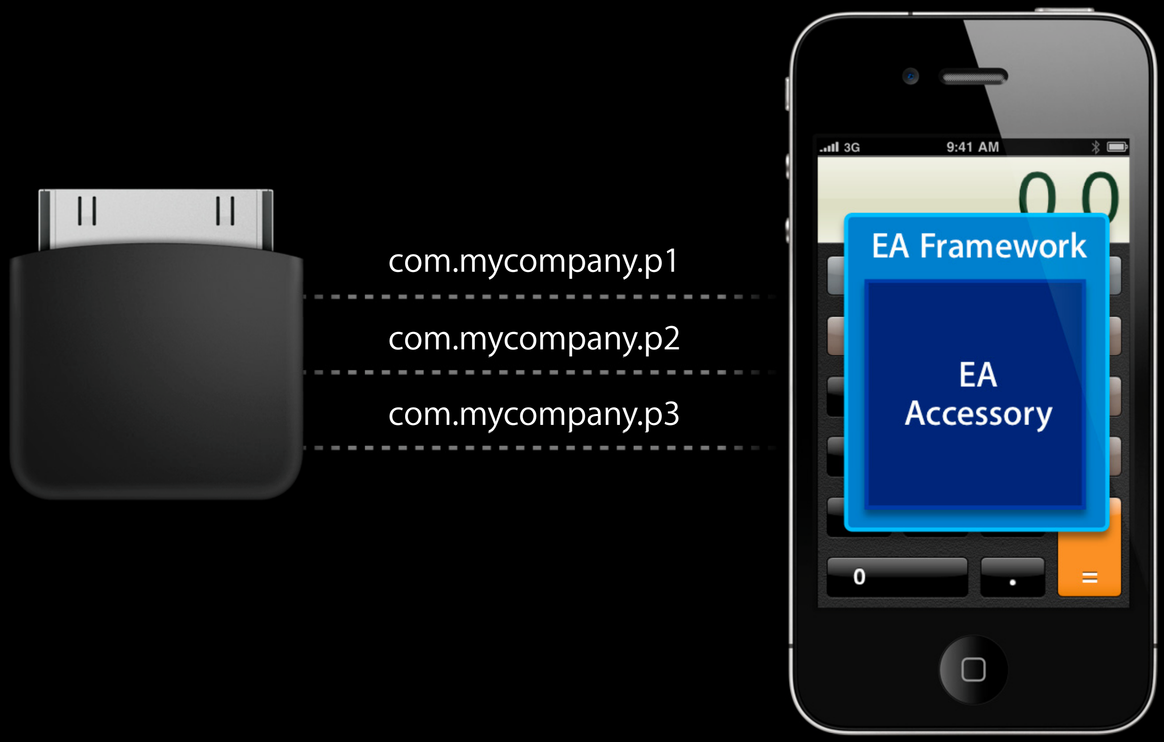
- Architecture
- API review
- Multitasking
- AppStore

The EA Architecture



EA Framework

The EA Architecture—Connect



com.mycompany.p1

com.mycompany.p2

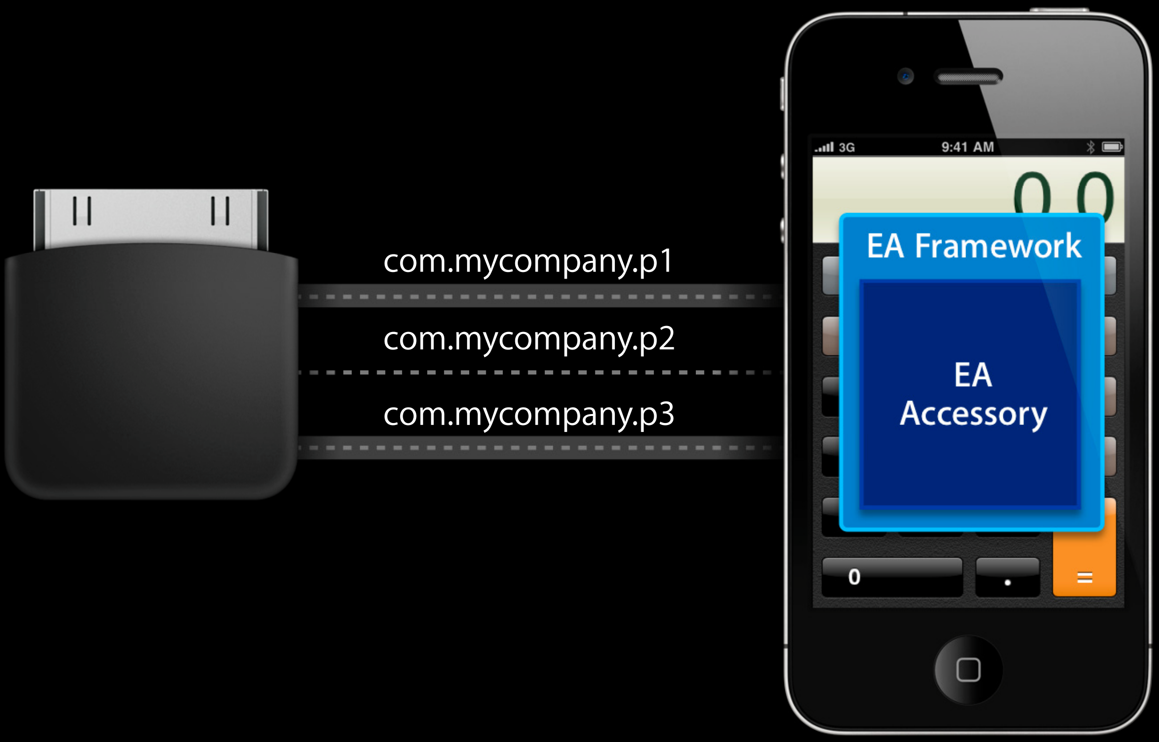
com.mycompany.p3

EA Framework

EA Accessory

----- EA Protocol

The EA Architecture—Data Transfer



com.mycompany.p1

com.mycompany.p2

com.mycompany.p3

EA Framework

EA Accessory

- - - - - EA Protocol

▬ EA Protocol with an associated EA session

External Accessory Framework

- EAAccessoryManager
- EAAccessory
    - EAAccessoryDelegate protocol
- EASession

EAAccessoryManager API

```
NSString *const EAAccessoryDidConnectNotification;
NSString *const EAAccessoryDidDisconnectNotification;

+ (EAAccessoryManager *)sharedAccessoryManager;

- (void)registerForLocalNotifications;
- (void)unregisterForLocalNotifications;

@property (nonatomic, readonly) NSArray *connectedAccessories;
```

EAAccessory API

```
@property(nonatomic, readonly) NSUInteger connectionID;
@property(nonatomic, readonly) NSArray *protocolStrings;
@property(nonatomic, assign) id<EAAccessoryDelegate> delegate;
```

EAAccessoryDelegate API

```
@protocol EAAccessoryDelegate <NSObject>
@optional
- (void)accessoryDidDisconnect:(EAAccessory *)accessory;
@end
```

EASession API

```
- (id)initWithAccessory:(EAAccessory *)accessory forProtocol:
  (NSString *)protocolString;

@property (nonatomic, readonly) NSInputStream *inputStream;
@property (nonatomic, readonly) NSOutputStream *outputStream;
```

One EASession per EA Protocol per EAAccessory

NSInputSteam/NSOutputStream

- Subclasses of NSStream
- “Introduction to Stream Programming Guide for Cocoa“
    - Available on ADC website

- Delegate handles stream events

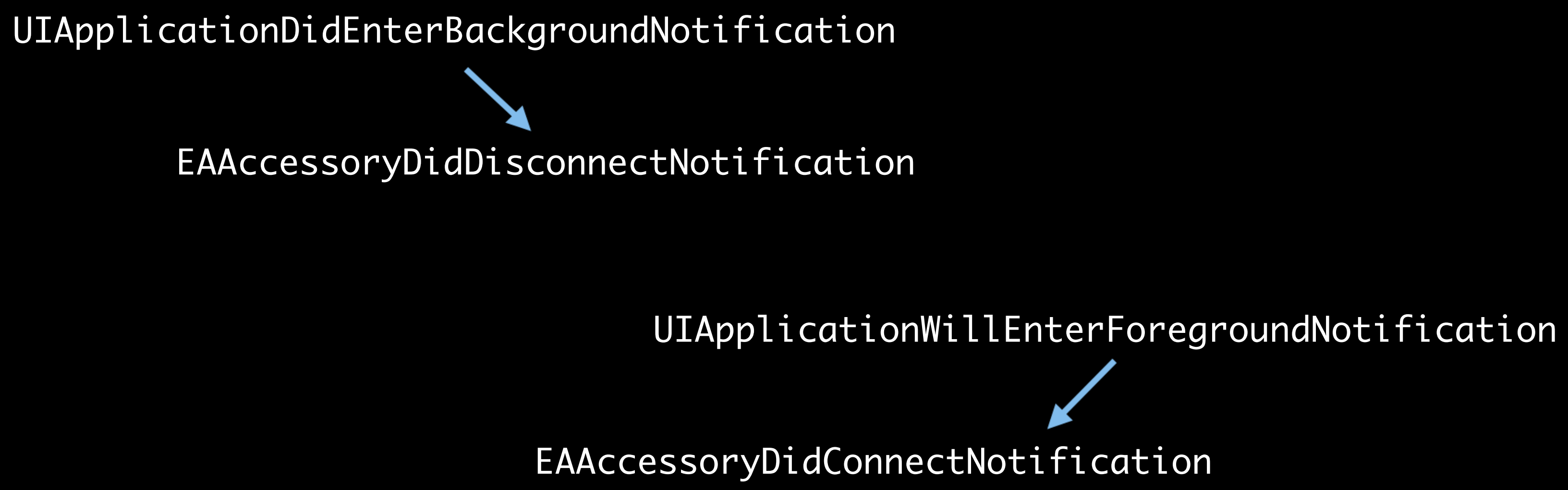
```
 - (void)stream:(NSStream *)theStream handleEvent:
(NSStreamEvent)streamEvent
```

What’s New…

- Multitasking
- AppStore interactions

Multitasking

No EA events in background



UIApplicationDidEnterBackgroundNotification

EAAccessoryDidDisconnectNotification

UIApplicationWillEnterForegroundNotification

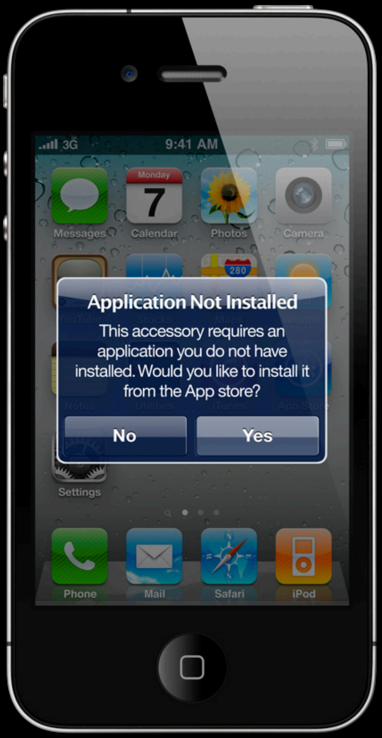
EAAccessoryDidConnectNotification

Multitasking

Release EASession instances on accessory disconnect

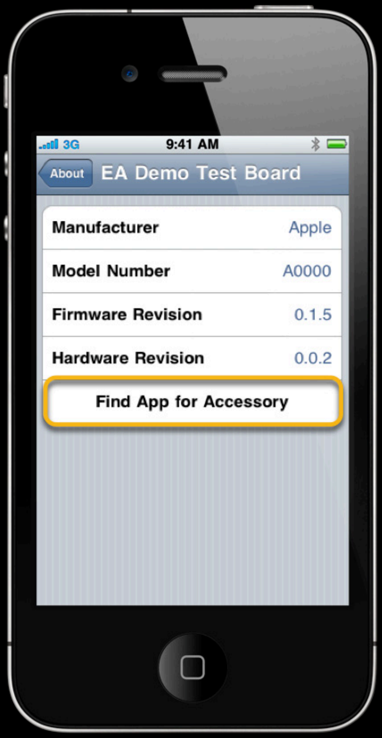
```
#import <ExternalAccessory/ExternalAccessory.h>


- (void)closeSession
{
    [[_session inputStream] close];
    [[_session inputStream] removeFromRunLoop:[NSRunLoop currentRunLoop]
forMode:NSDefaultRunLoopMode];
    [[_session inputStream] setDelegate:nil];

    [[_session outputStream] close];
    [[_session outputStream] removeFromRunLoop:[NSRunLoop currentRunLoop]
forMode:NSDefaultRunLoopMode];
    [[_session outputStream] setDelegate:nil];

    [_session release];
}
```

AppStore Interactions



Application Not Installed

This accessory requires an application you do not have installed. Would you like to install it from the App store?



EA Demo Test Board

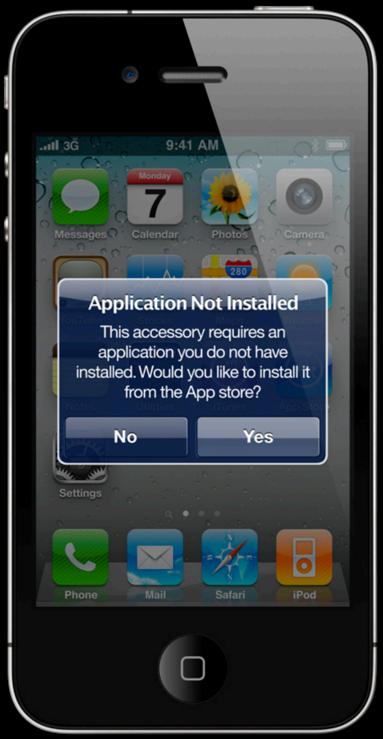
Manufacturer	Apple
Model Number	A0000
Firmware Revision	0.1.5
Hardware Revision	0.0.2

AppStore Interactions

Associating an app with EA Protocols using Info.plist

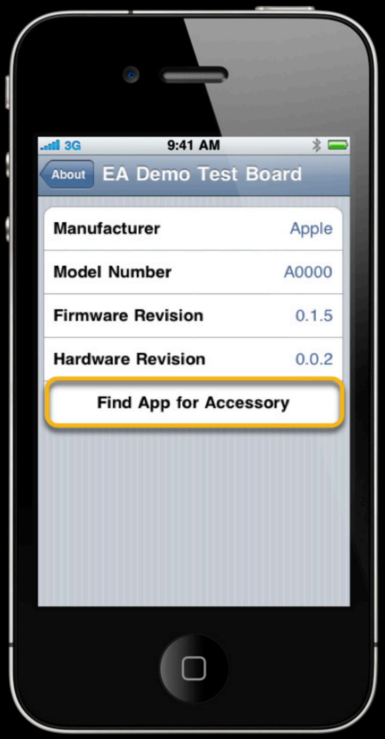
```
<plist version="1.0">
<dict>
    <key>CFBundleDevelopmentRegion</key>
...
    <key>UISupportedExternalAccessoryProtocols</key>
    <array>
        <string>com.yourcompany.p1</string>
        <string>com.yourcompany.p2</string>
    </array>
</dict>
</plist>
```

AppStore Interactions



Application Not Installed

This accessory requires an application you do not have installed. Would you like to install it from the App store?



EA Demo Test Board

Manufacturer	Apple
Model Number	A0000
Firmware Revision	0.1.5
Hardware Revision	0.0.2

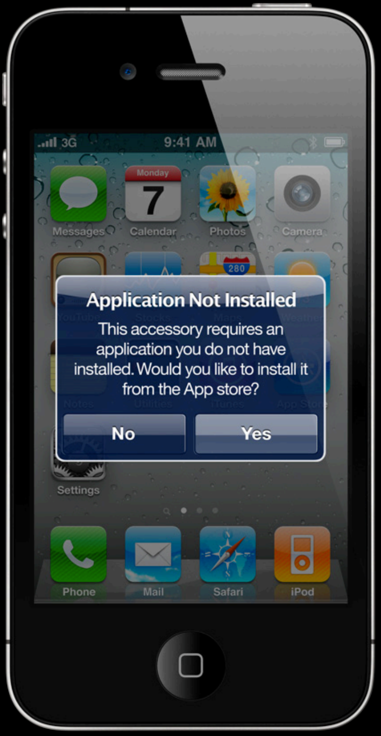
AppStore Interactions

- Properties for each EA Protocol
  - Show Alert if no EA Protocol is supported
  - Show Alert if EA Protocol x is not supported
  - No Alert
  - No Action
- Accessory adds these properties using iAP

iPhone OS 3

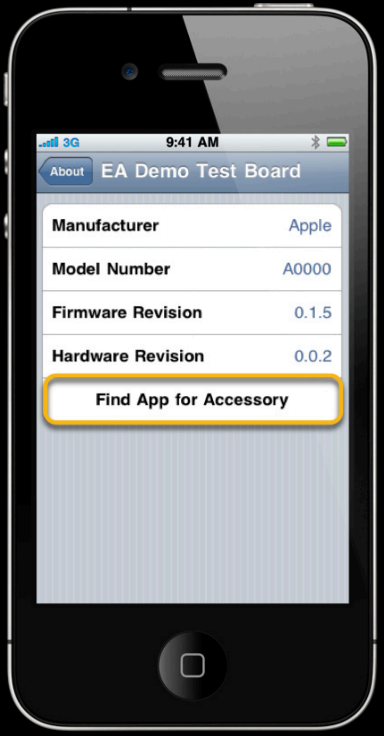
AppStore Interactions

Show Alert if no EA Protocol is supported



Application Not Installed

This accessory requires an application you do not have installed. Would you like to install it from the App store?

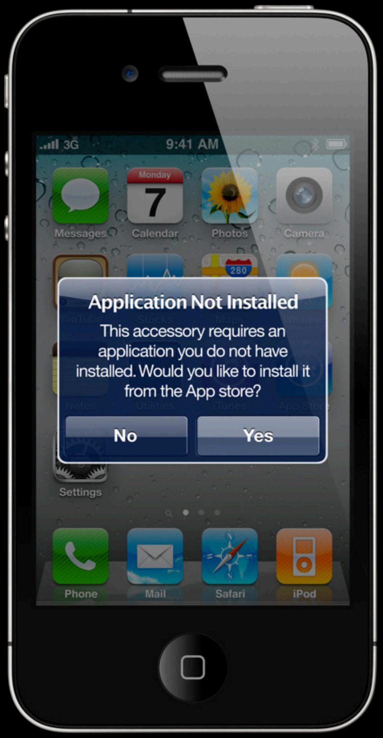


EA Demo Test Board

Manufacturer	Apple
Model Number	A0000
Firmware Revision	0.1.5
Hardware Revision	0.0.2

AppStore Interactions

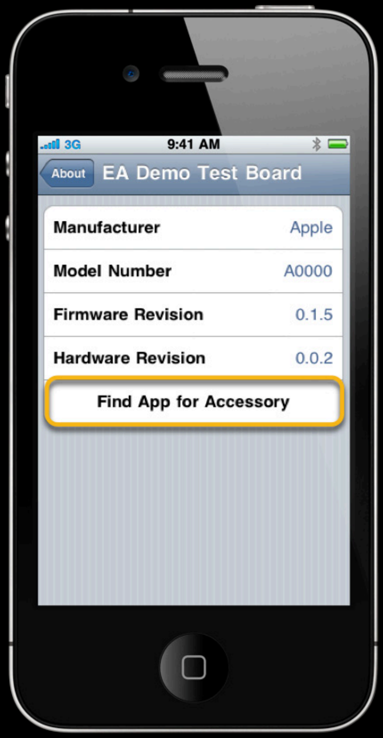
Show Alert if EA Protocol x is not supported



Application Not Installed

This accessory requires an application you do not have installed. Would you like to install it from the App store?

No | Yes



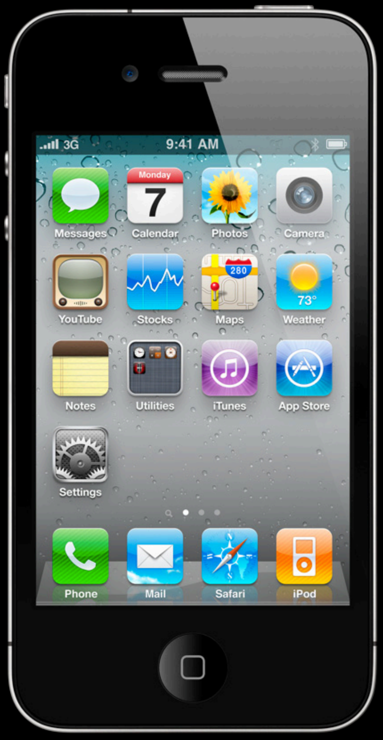
EA Demo Test Board

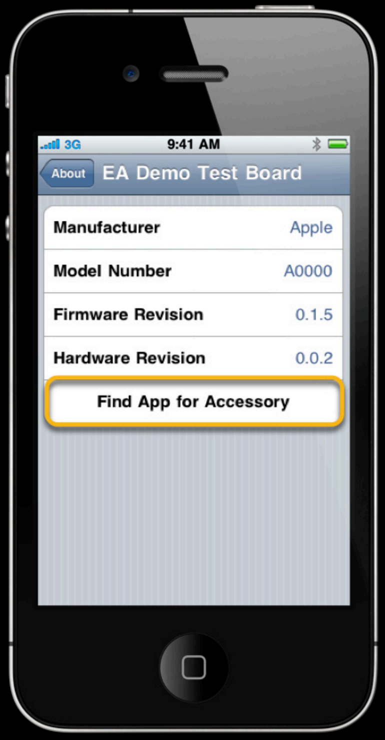
Manufacturer	Apple
Model Number	A0000
Firmware Revision	0.1.5
Hardware Revision	0.0.2

Find App for Accessory

AppStore Interactions

No alert



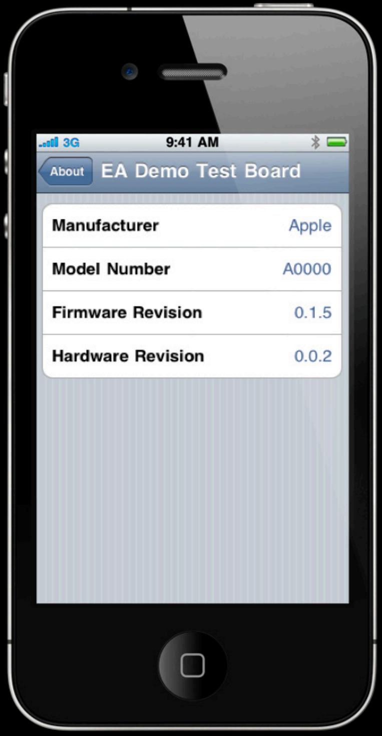


EA Demo Test Board

Manufacturer	Apple
Model Number	A0000
Firmware Revision	0.1.5
Hardware Revision	0.0.2

AppStore Interactions

No action



EA Demo Test Board

Manufacturer	Apple
Model Number	A0000
Firmware Revision	0.1.5
Hardware Revision	0.0.2

More Information

Stephen Chick
iPhone Evangelism
chick@apple.com

Craig Keithley
iPod Technology Evangelist
keithley@apple.com

Developer Programs
MFi Program
http://developer.apple.com/mfi/

Documentation
iPhone OS Accessories
http://developer.apple.com/iphone/program/accessories/

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Simplifying Networking Using Bonjour	Nob Hill Wednesday 10:15AM
Accessibility on iPhone OS	Nob Hill Wednesday 4:30PM
Fundamentals of Digital Audio for Mac OS X and iPhone OS	Mission Wednesday 10:15AM
What’s New in Cocoa Touch (R)	Marina Friday 11:30AM

Labs

iPhone OS Accessories Lab	Core OS Lab B Tuesday 2:00PM
Bluetooth Lab	Core OS Lab B Wednesday 9:00AM

WWDC10

[redacted]