Lecture #11
Thread

Android Things 2023

nest

We're here to help. Talk to a Google Nest expert at (855) 888-8209. Learn more >

Nest.com has moved to the Google Store.

Nest and Google Home have joined together as Google Nest. Moving forward, you can learn about and buy all Nest products in one place: the Google Store.

Visit the store

Find the Nest products you're looking for on the Google Store.

Thermostats

Cameras

Doorbell

Alarm system

Lock

Smoke + CO alarm


nest
65
HEATING
68


nest

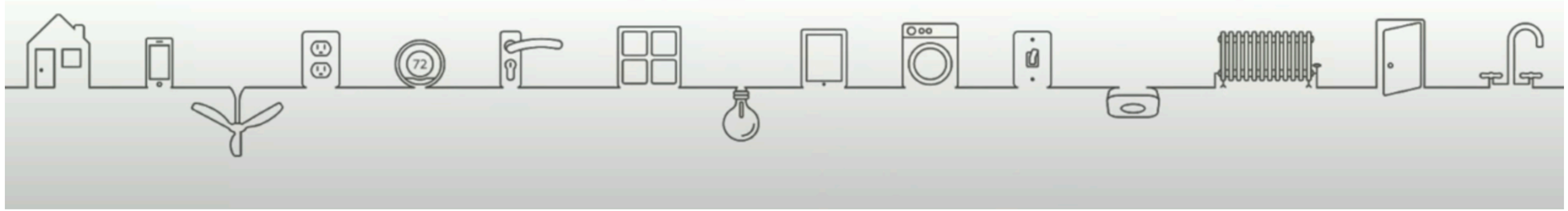

Google
nest
IN 26 MIN
72
casaplex

Nest

Create a home that’s thoughtful – one that takes care of the people inside it and the world around it.

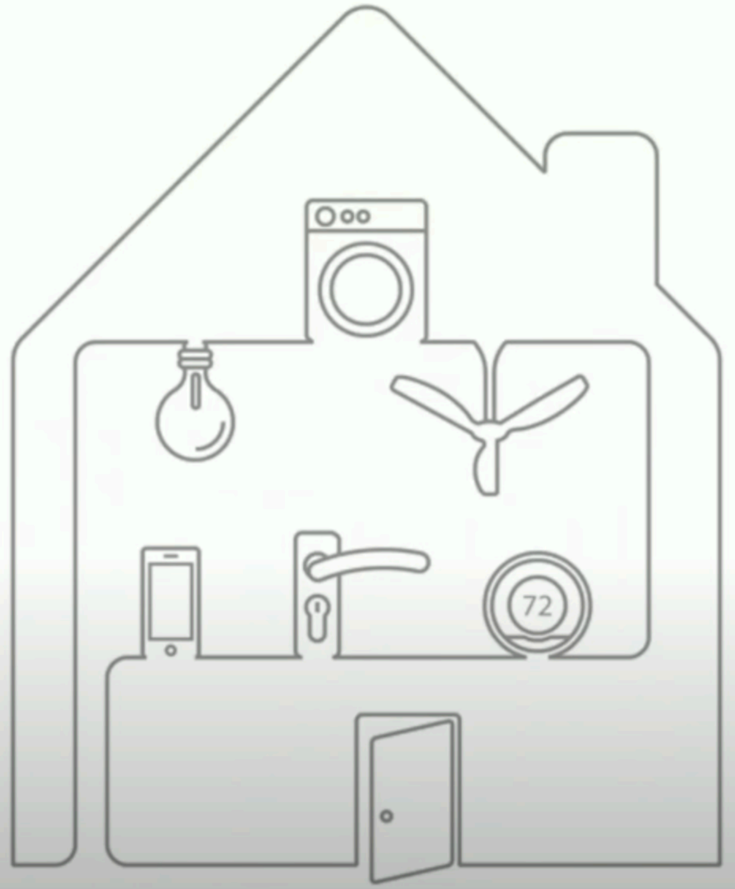
Problem

How to **securely** and **scalably** connect an ecosystem of **low-power** products end-to-end, to cloud services, and to consumers?



Requirements

IP-based
Scalable
Resilient
Low power
Secure



None of the 2010 technologies were satisfying the requirements.

Others agreed


THREAD
GROUP
DoB: July 2014

Thread Protocol

An **open, IPv6-based, low-power, and secure mesh** networking technology for IoT products.

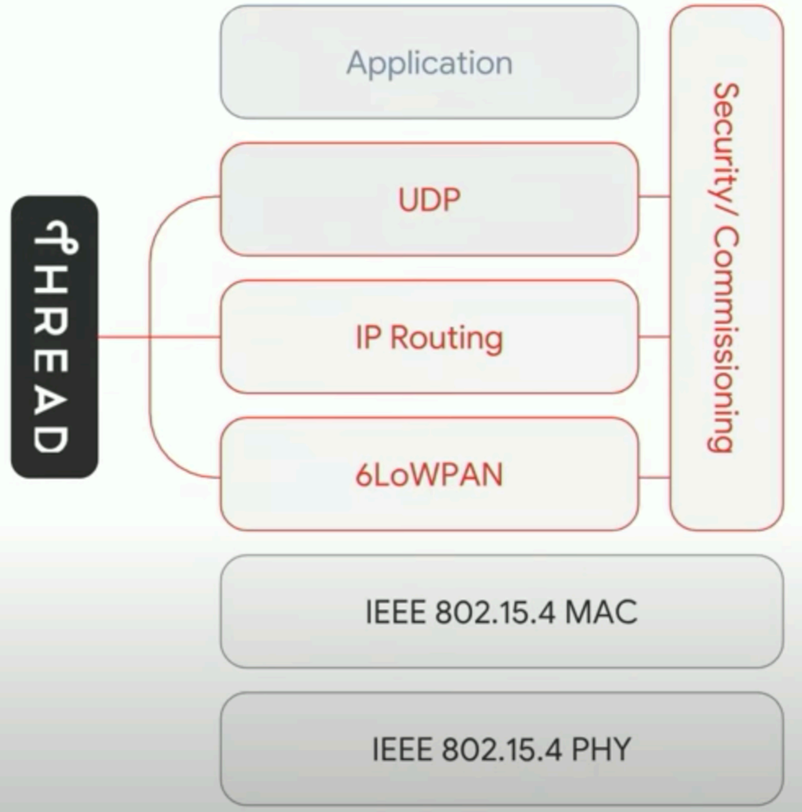
Built on the **same IP technology** that drives every Internet-connected device, but designed specifically for the needs of IoT devices.

Builds on Existing Technologies

Same radio used by ZigBee.

Fast time to market.

Low-cost implementations.



Why IP?

A converged network layer

Build **end-to-end applications** that utilize **multiple link technologies**

Why IP?

A multi-service network

Host **multiple applications** using a **common network infrastructure**

Scalable Mesh

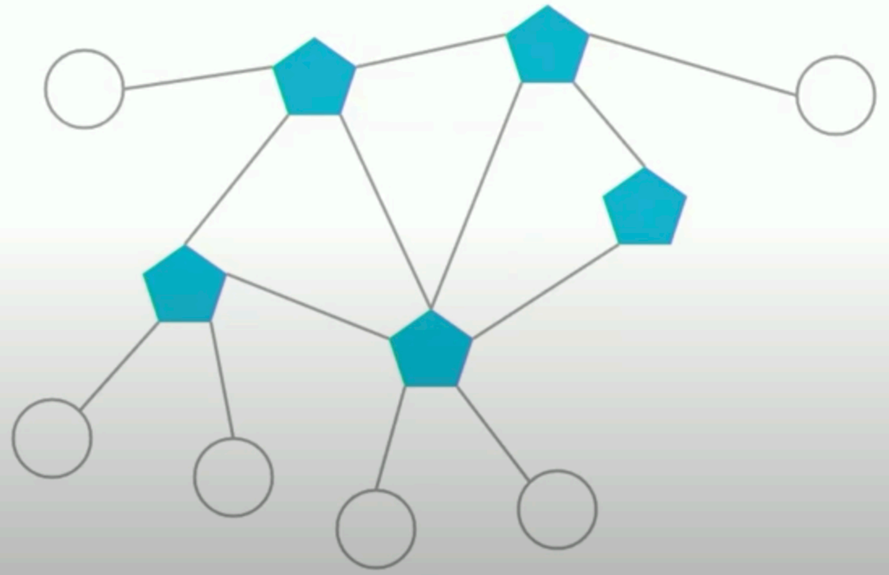
Routers forward messages between neighbors

Up to 32 per network

End devices communicate via a router

Up to 511 per router

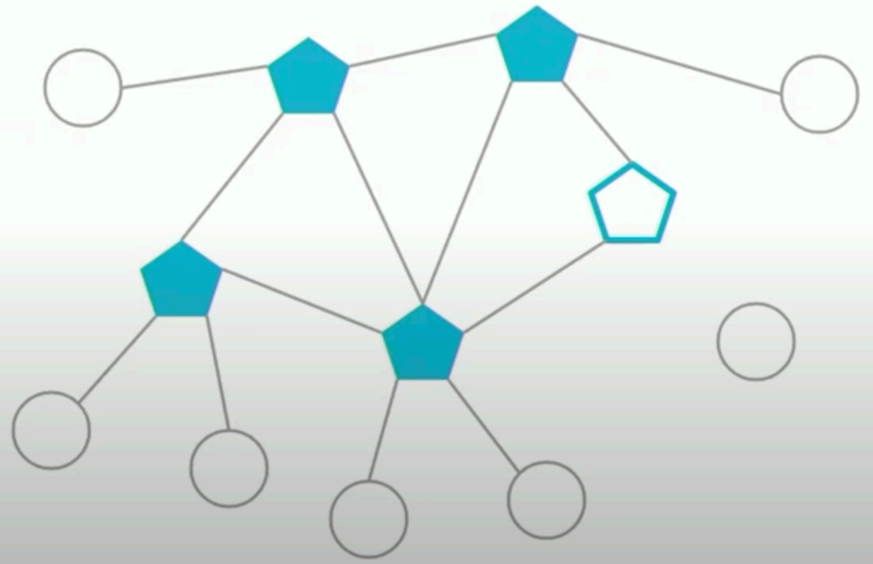
Hundreds of devices per network



Self-Configuring Routers

Add to **increase connectivity and range**

Remove to **reduce redundant connectivity**



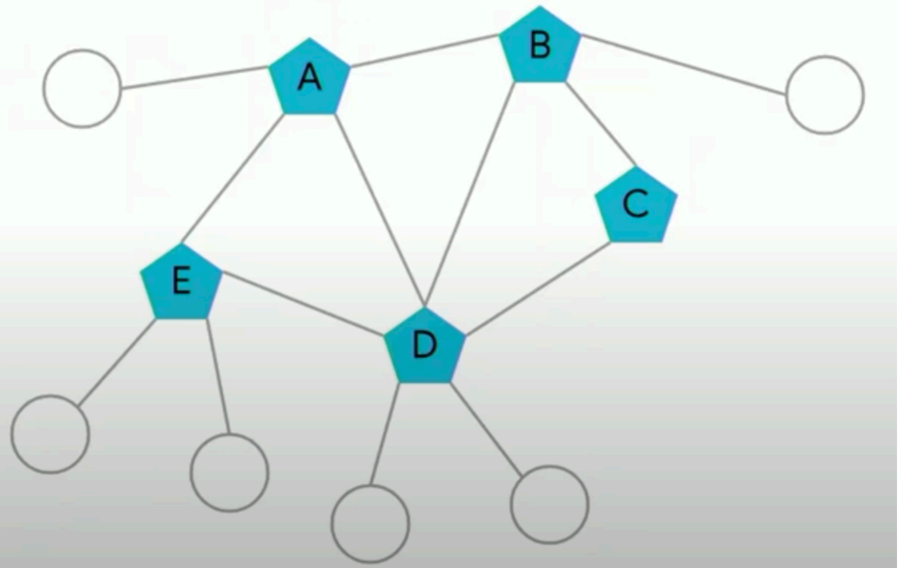
Resilient Mesh Routing

Shortest-path any-to-any routing

Distance-vector routing protocol
Similar to RIP

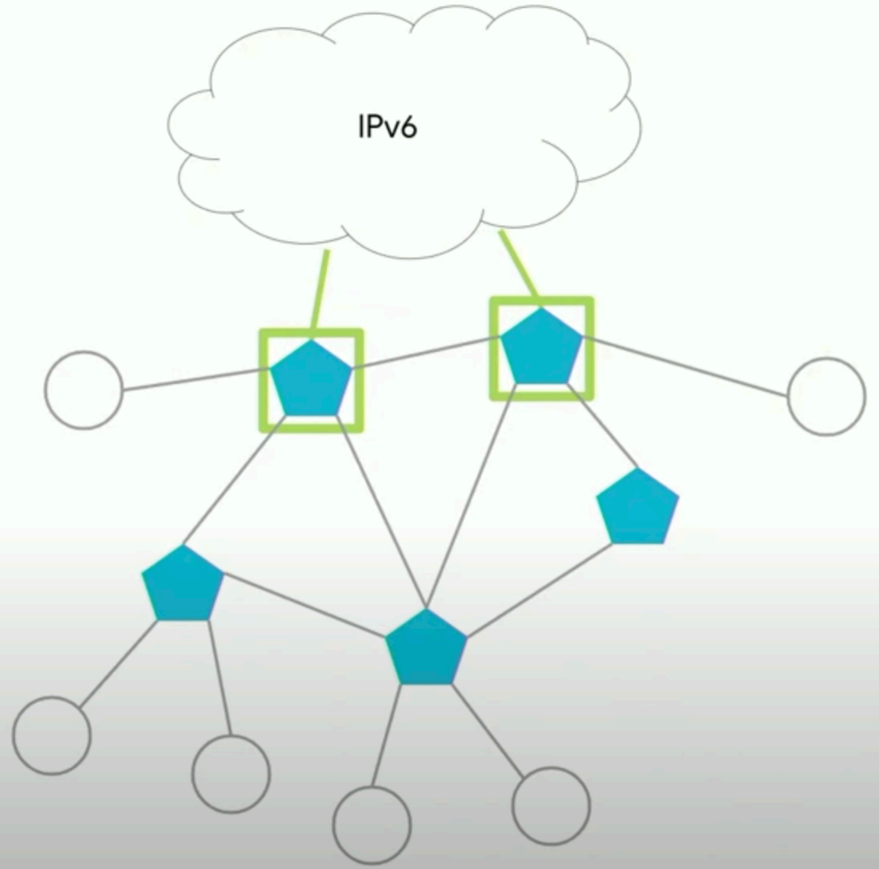
Maintain and advertise best next hop towards each Thread Router.

Dest	Next	Cost
B	B	1
C	B	2
D	D	1
E	E	1



Resilient Border Routing

Multiple border routers to connect Thread to non-Thread networks.

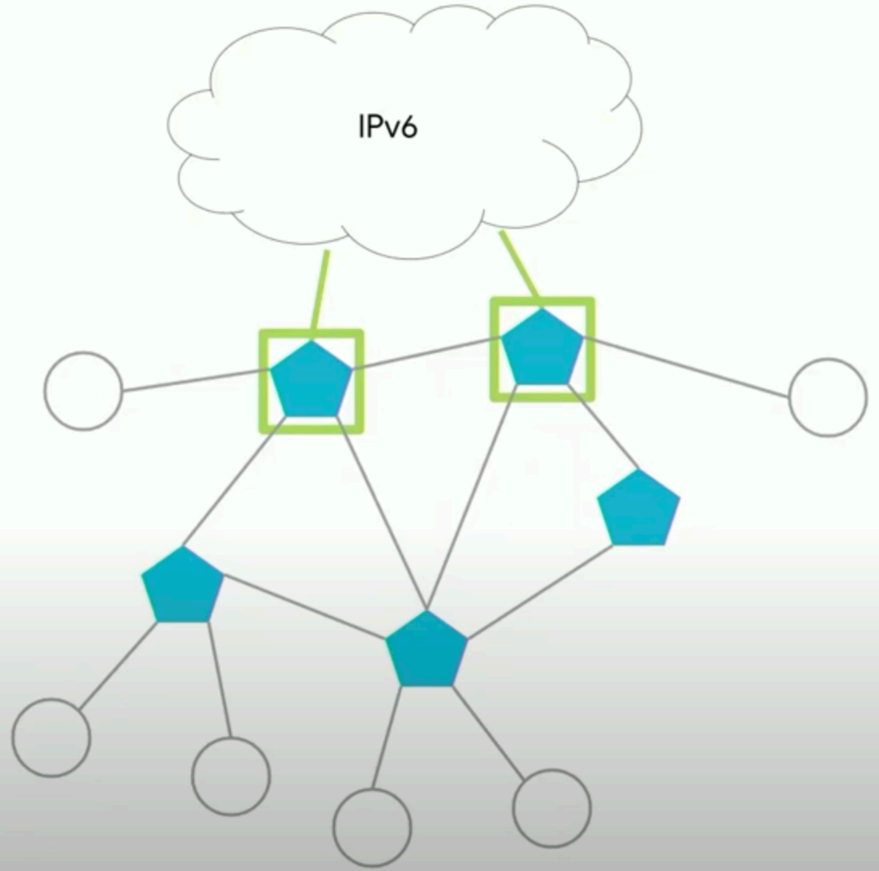


Low Power

Receive power **one-tenth of WiFi**

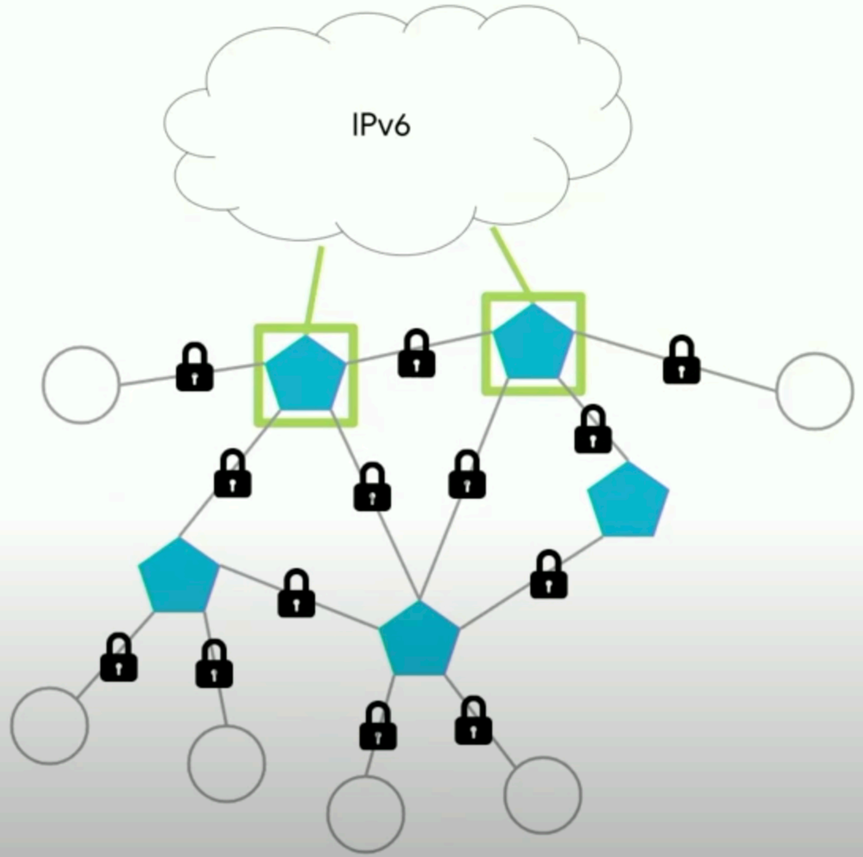
Sleepy end devices **duty-cycles receiver**

Lifetime in **years** with coin-cells



Mandatory Security

All link frames protected using AES-128 **encryption** and **authentication** with **replay protection**.

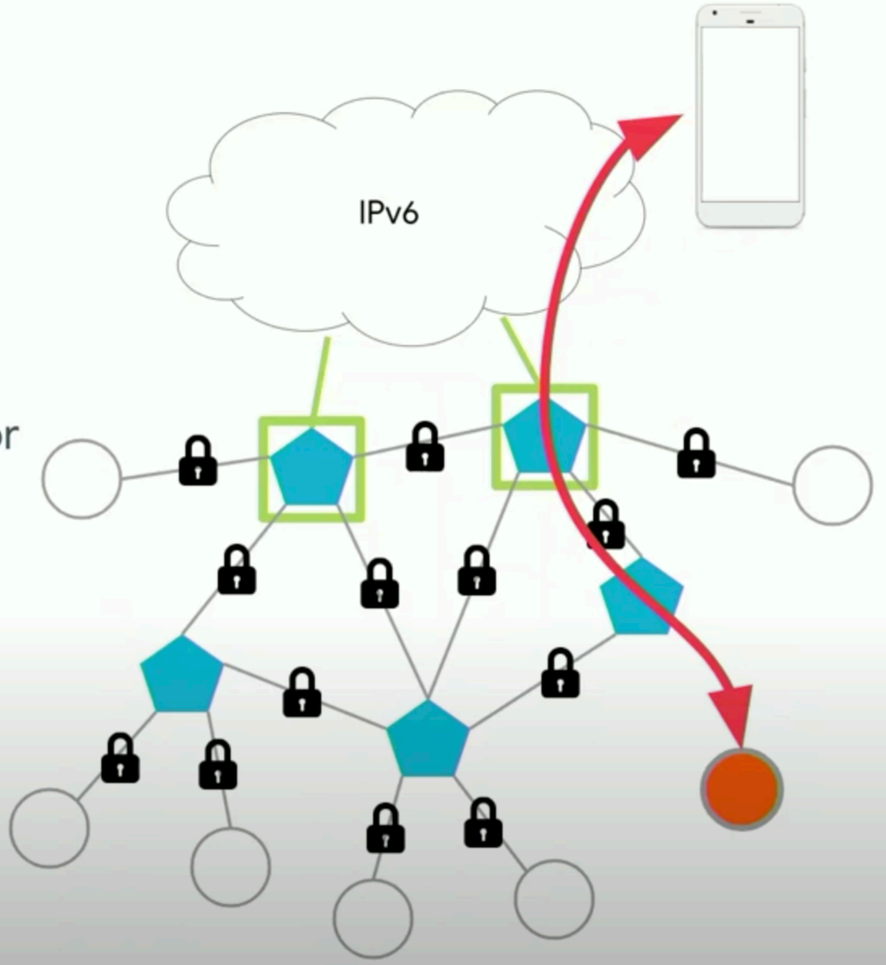


Device Commissioning

User-initiated process involving a physical factor

End-to-end DTLS session via border router

EC J-PAKE ciphersuite for **short pairing codes**



Thread is...

IPv6
Scalable
Resilient
Low power
Secure

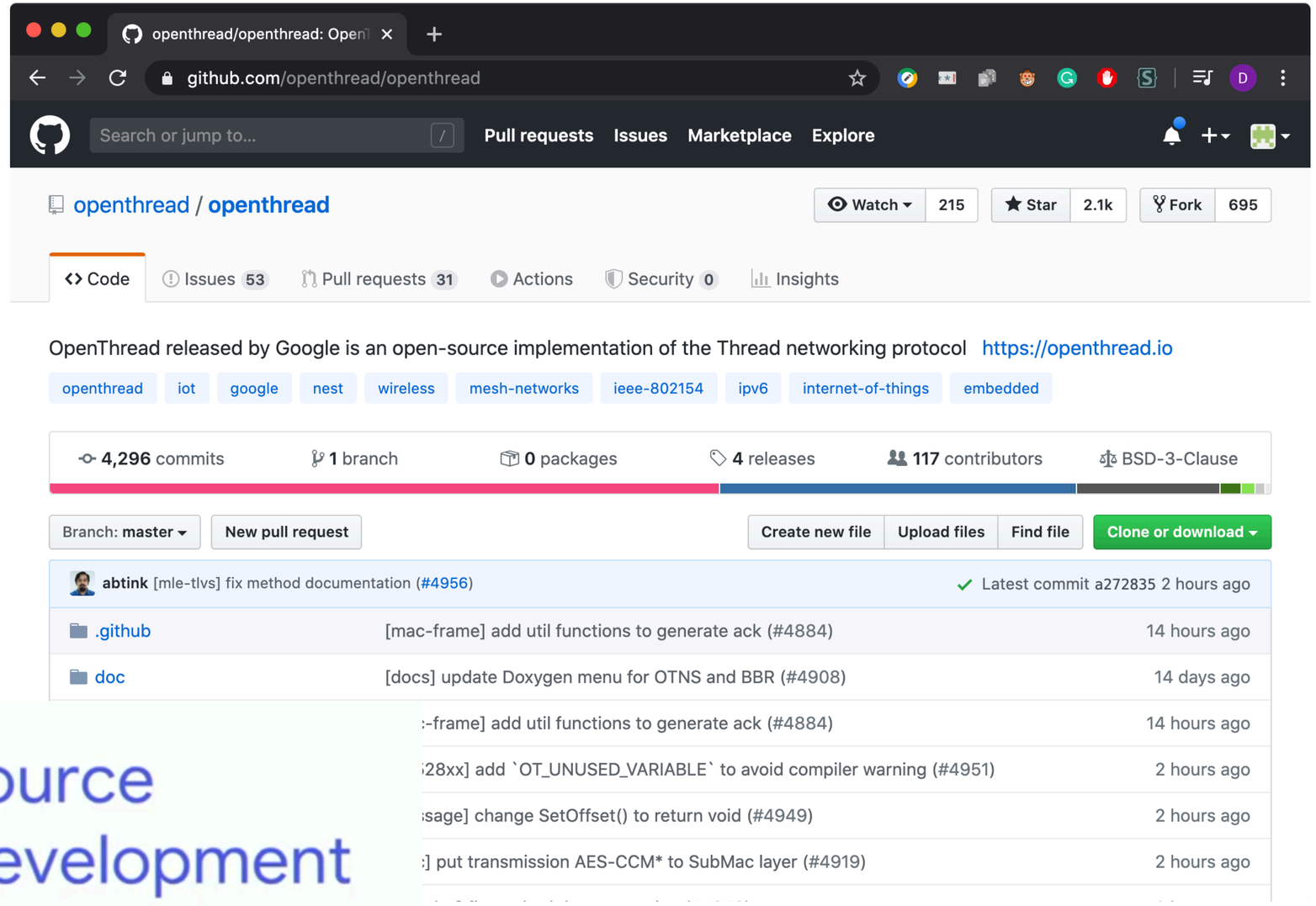
No other technology satisfies these requirements.

OPENTHREAD

released by Nest

An **open source** implementation of Thread released by Nest.

To make the networking **technology used in Nest products** more broadly **available to developers**.



Open Source
Open Development

Hosted on **GitHub**

BSD-3 license

Development Kits

IoT Platforms

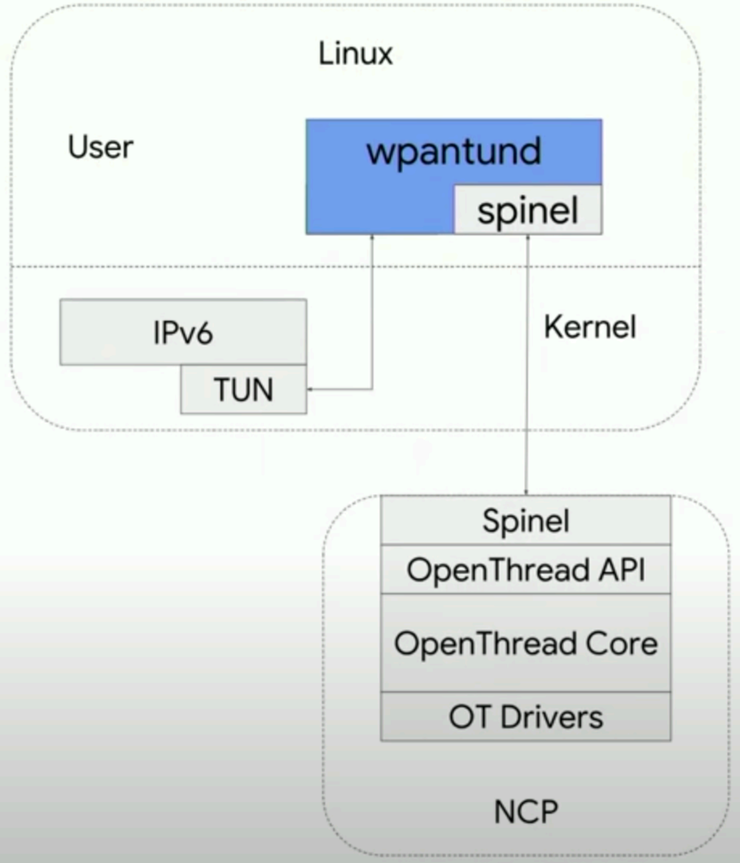
wpantund

User-space NCP network interface driver

Standard Linux network interface

Convenient command-line interface

DBus-based management API



Border Router

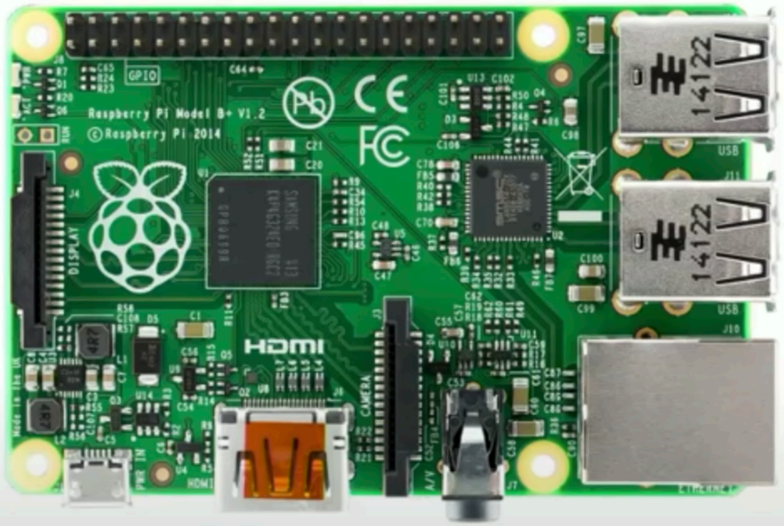
Connect Thread network to WiFi/Ethernet

Raspberry Pi 3B or BeagleBone Black

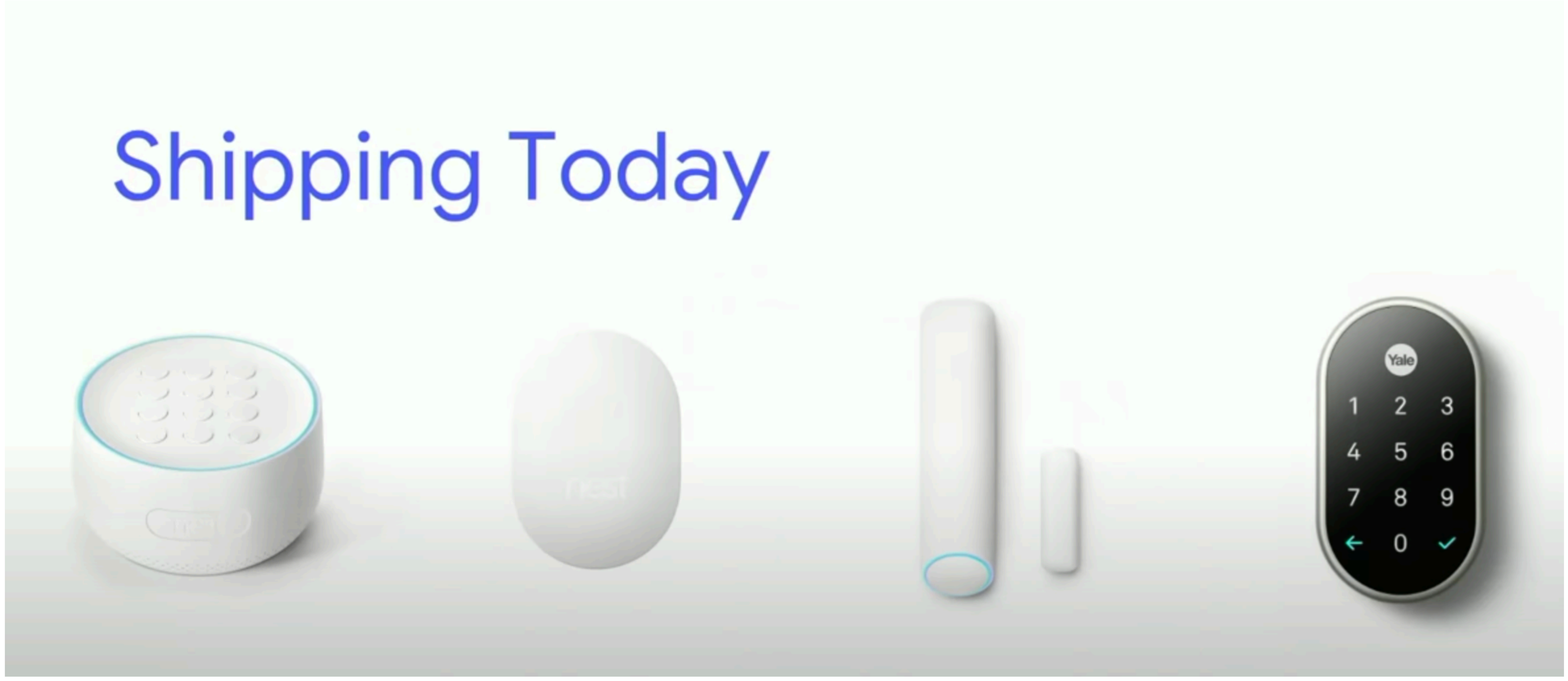
IPv6 autoconfiguration with DHCPv6-PD

IPv4 with NAT64 and DNS64

External Thread Commissioning



Shipping Today



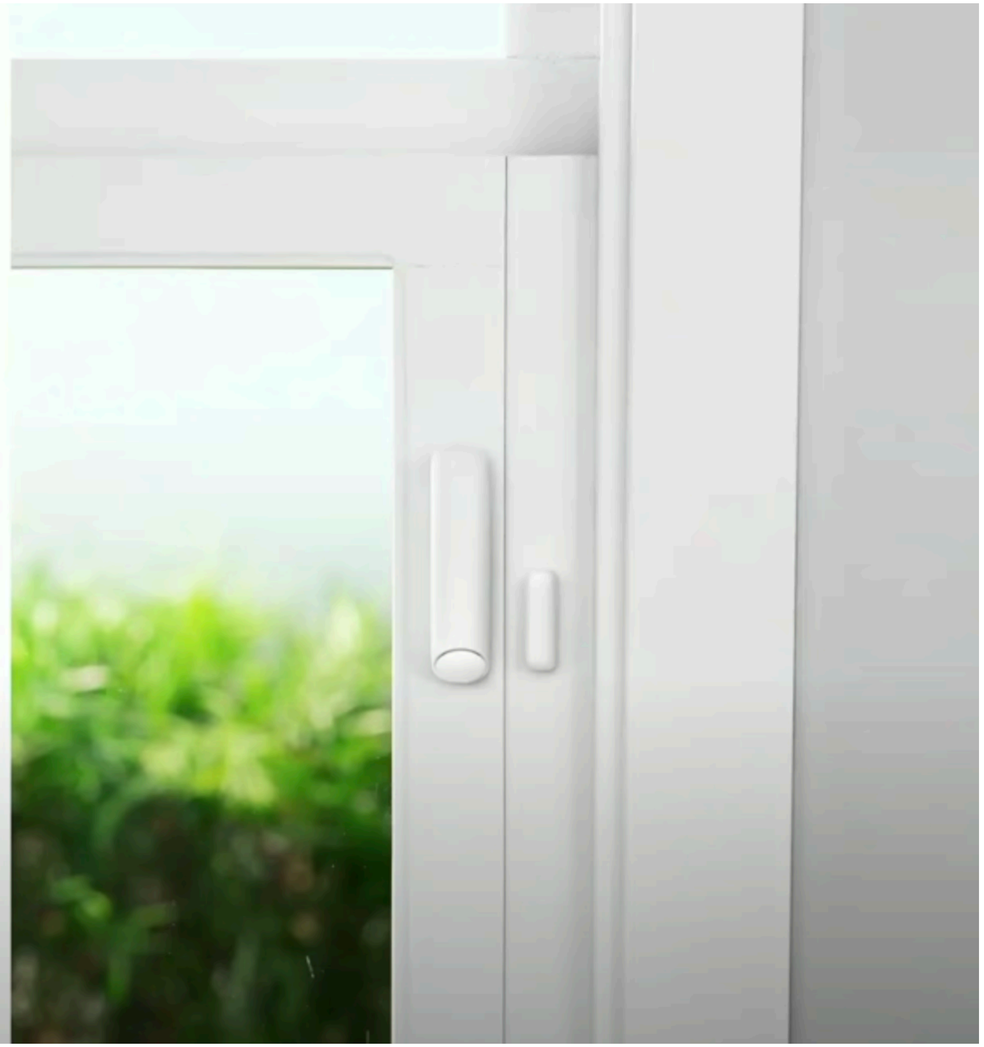
Nest Detect

Open/close and motion

Thread Sleepy End Device

2-year battery life

CR123 lithium battery



Nest x Yale Lock

Remote lock/unlock in seconds

Secure end-to-end communication

Thread Sleepy End Device

1 year battery life

AA batteries

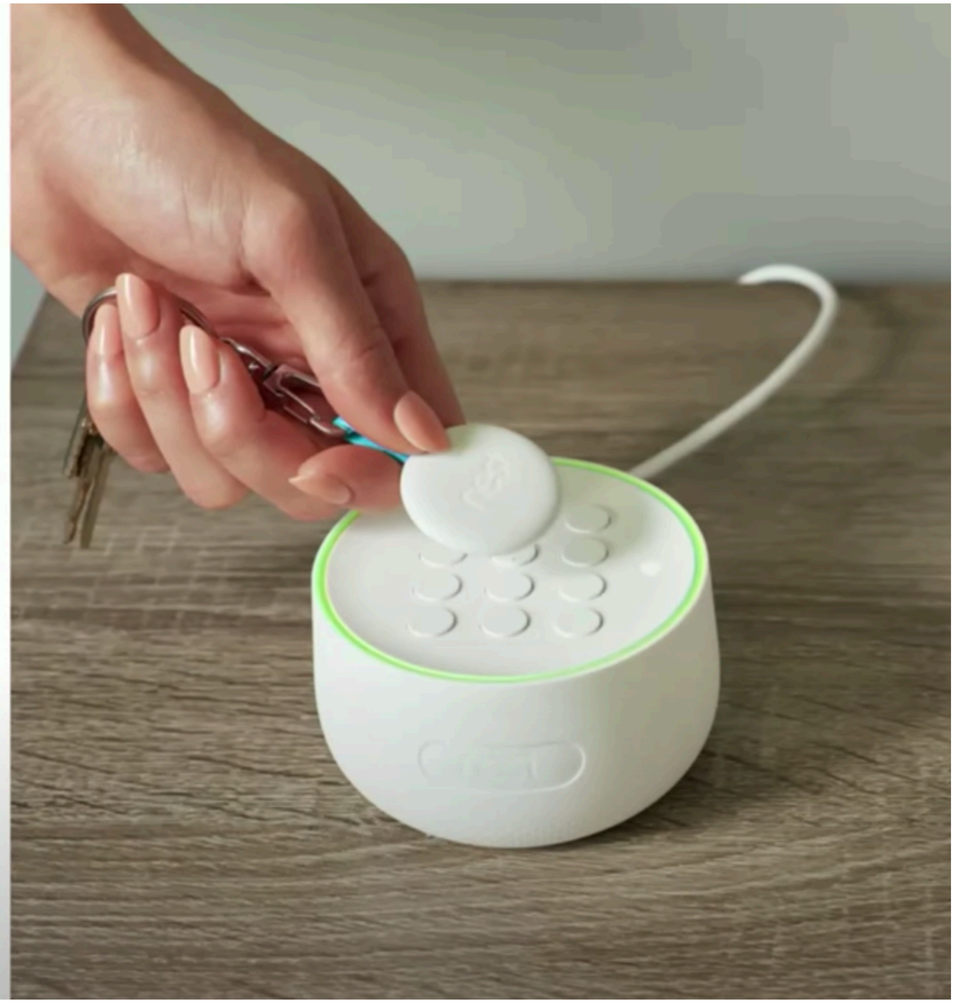
Nest Guard

Security base

Thread Router and Border Router

Wi-Fi, Cellular, and Thread

12-hour battery backup



Nest Connect

Connectivity for Detect and Lock

Thread Router and Border Router

Wi-Fi and Thread

24-hour battery backup

How can we use Thread in our products?

So many options

Several System Architectures

Several Operating Systems

Several Toolchains

Several Certified Radios

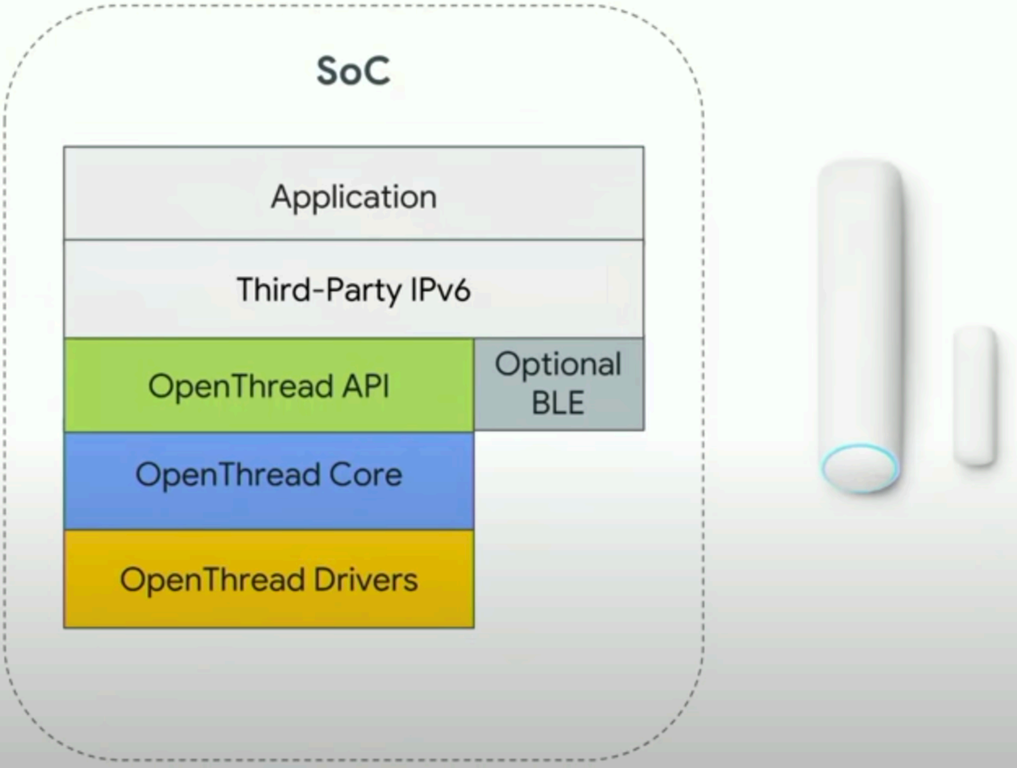
System on Chip

Thread only

Battery powered

Highly integrated

Low cost

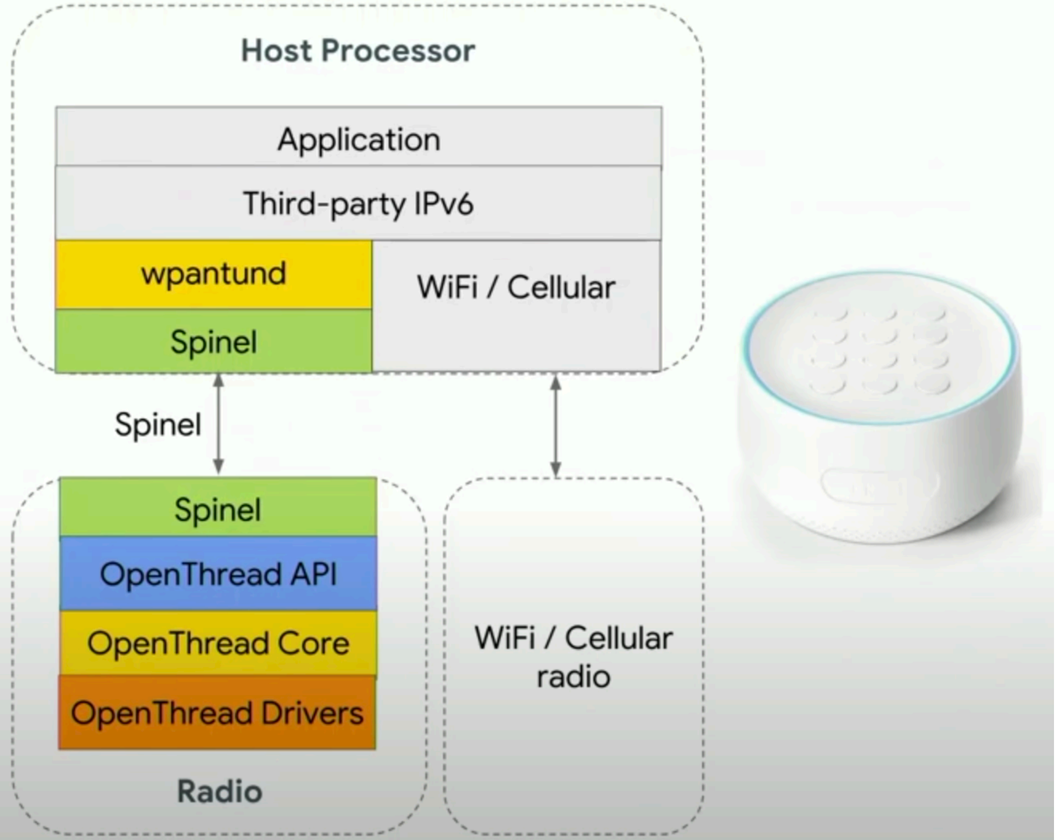


NCP

Thread runs on separate radio

More-capable host processor

Host processor can sleep while maintaining Thread network

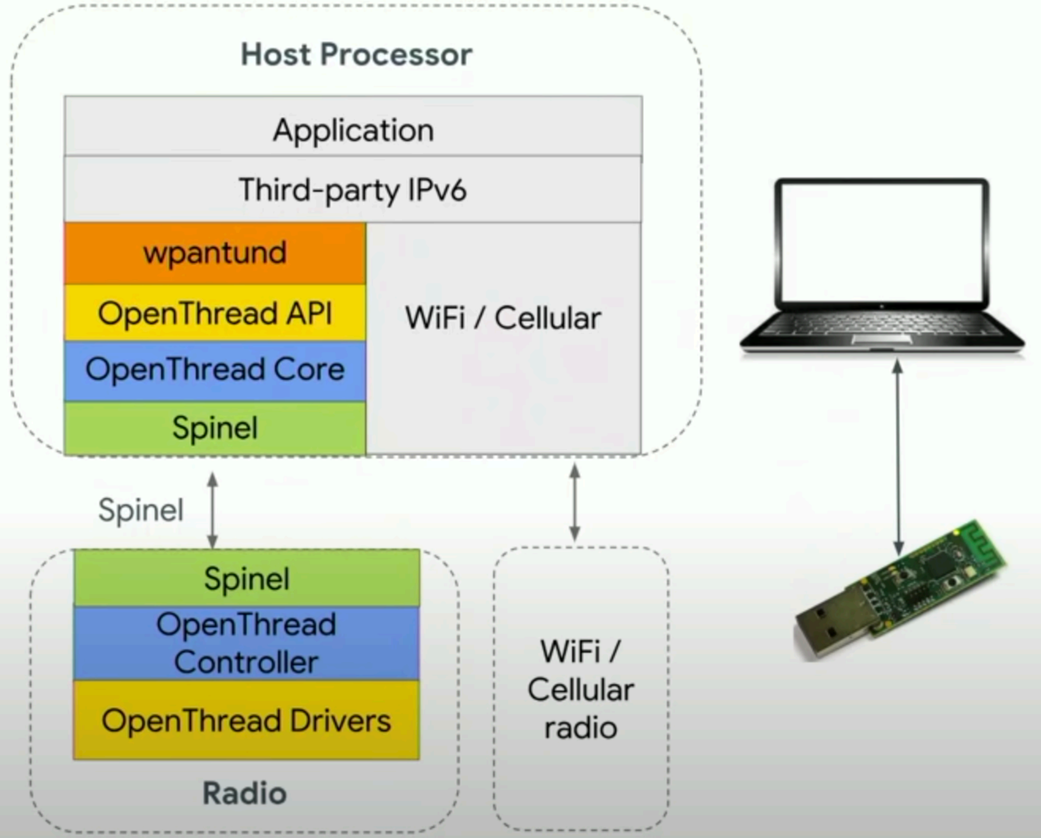


MAC/PHY

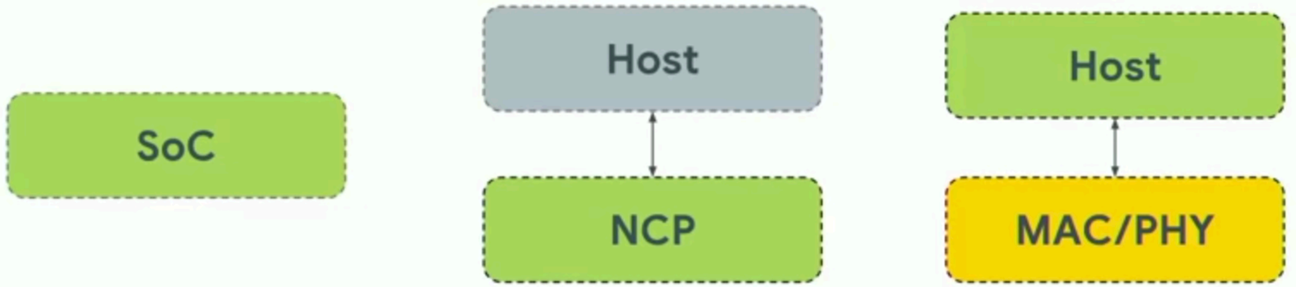
OpenThread runs on the host

Addition to existing devices

Router or Sleepy End device



What is the application?



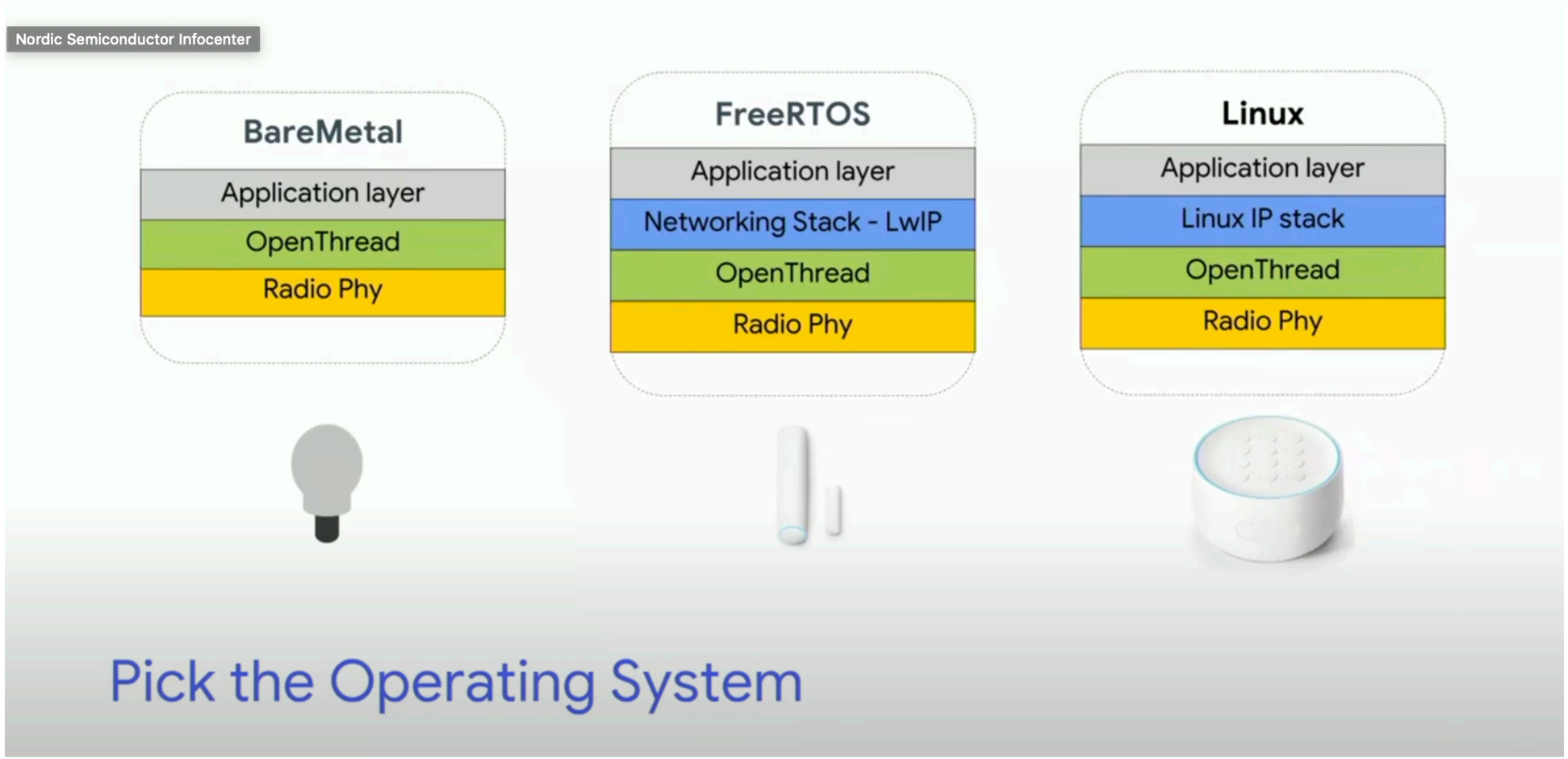
Are there power constraints?

What does the UI look like?

Are there other radios?

Pick the architecture

Nordic Semiconductor Infocenter



Pick the Operating System

Toolchains

Toolchain agnostic

Narrow platform abstraction

Highly portable

Quick system integration

GNU

IAR SYSTEMS

KEIL™ Tools by ARM

Visual Studio

Non-volatile storage ***Alarm***

Random ***Radio***

TI CC2652

TI CC2538

Dialog DA15000

SiLabs EFR32

Nordic nRF52840

NXP KW41Z

Qorvo GP712

Microchip ATSAMR21G18A

Pick the radio

Network and memory requirements

Try out the sample app

Samples with CLI

Host tools - wpantund

Codelabs

```
> child table
| ID  | RLOC16 | Timeout    | Age        | LQ In | C_VN |R|S|D|N| Extended MAC     |
+-----+--------+------------+------------+-------+------+-+-+-+-+------------------+
|   1 | 0xe001 |        240 |         44 |     3 |  237 |1|1|1|1| d28d7f875888fccb |
|   2 | 0xe002 |        240 |         27 |     3 |  237 |0|1|0|1| e2b3540590b0fd87 |
Done
> child 1
Child ID: 1
Rloc: 9c01
Ext Addr: e2b3540590b0fd87
Mode: rsn
Net Data: 184
Timeout: 100
Age: 0
Link Quality In: 3
RSSI: -20
Done
> childmax 5
Done
> childmax
5
Done
```

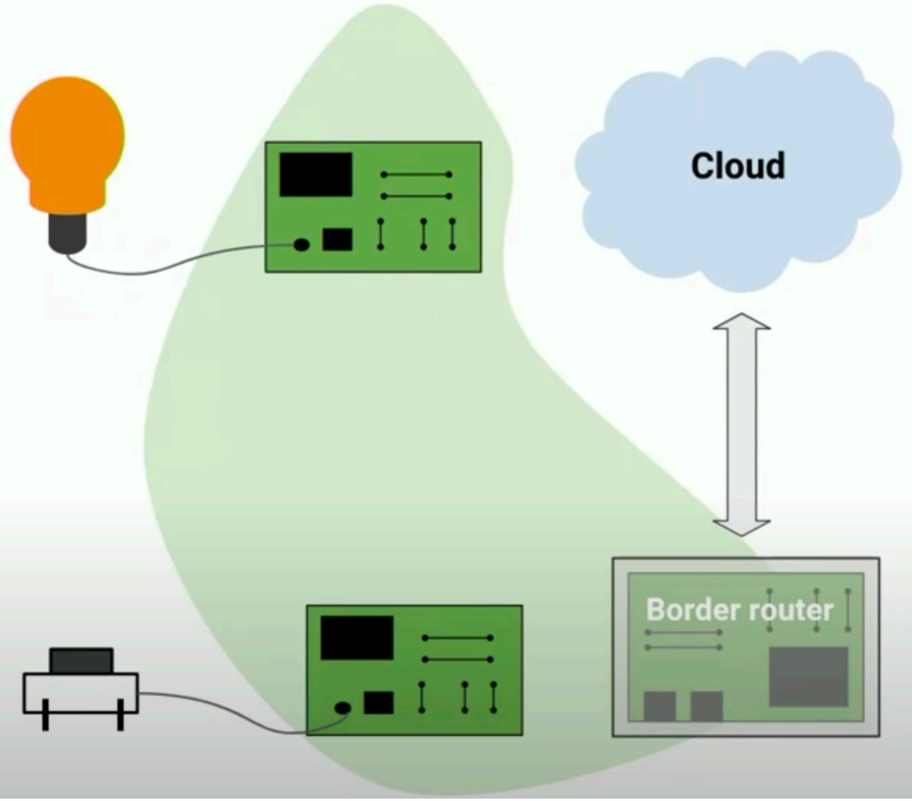
Build your IoT product

Add application support

Add driver support for peripherals

Radio driver support is available from radio/platform vendors

Add optional cloud support



Addressing

Local: Self-organizing (e.g. ULA), centrally managed (e.g. DHCPv6)

Wide-area: Native IPv6 (DHCPv6-PD), IPv4 translations (DNS64, NAT64), custom tunnel, VPNs

Designate border routers

Nest's Application Framework

Selects a naming scheme, addressing, schemas for different applications, application-layer security

Picks a cloud connectivity model

Codelab: Happy and OpenWeave

https://github.com/openweave

Happy - tool created by Nest for lightweight orchestration of simulated network topologies

Optimize and Debug

Fine tune network parameters

Logging, configurable run-time

Counters for detailed debugging

```
#define OPENTHREAD_CONFIG_NUM_MESSAGE_BUFFERS 40
#define OPENTHREAD_CONFIG_MESSAGE_BUFFER_SIZE 128
#define OPENTHREAD_CONFIG_MAX_TX_ATTEMPTS_DIRECT 4
#define OPENTHREAD_CONFIG_MAX_TX_ATTEMPTS_INDIRECT_PER_POLL 1
#define OPENTHREAD_CONFIG_ATTACH_DATA_POLL_PERIOD 100
#define OPENTHREAD_CONFIG_ADDRESS_CACHE_ENTRIES 10
#define OPENTHREAD_CONFIG_CLI_UART_RX_BUFFER_SIZE 512
#define OPENTHREAD_CONFIG_CLI_UART_TX_BUFFER_SIZE 1024
#define OPENTHREAD_CONFIG_MAX_CHILDREN 10
#define OPENTHREAD_CONFIG_DEFAULT_CHILD_TIMEOUT 240
#define OPENTHREAD_CONFIG_6LOWPAN_REASSEMBLY_TIMEOUT 5
#define OPENTHREAD_CONFIG_MAX_ENERGY_RESULTS 64
#define OPENTHREAD_CONFIG_COAP_ACK_RANDOM_FACTOR_NUMERATOR 3
#define OPENTHREAD_CONFIG_MAC_FILTER_SIZE 32
#define OPENTHREAD_CONFIG_STORE_FRAME_COUNTER_AHEAD 1000
#define OPENTHREAD_CONFIG_LOG_LEVEL OT_LOG_LEVEL_CRIT
#define OPENTHREAD_CONFIG_NUM_DHCP_PREFIXES 4
#define OPENTHREAD_CONFIG_NUM_SLAAC_ADDRESSES 4
#define OPENTHREAD_CONFIG_NCP_TX_BUFFER_SIZE 512
#define OPENTHREAD_CONFIG_NCP_SPINEL_ENCRYPTER_EXTRA_DATA_SIZE 0
#define OPENTHREAD_CONFIG_PLATFORM_ASSERT_MANAGEMENT 0
#define OPENTHREAD_CONFIG_ENABLE_SOFTWARE_ACK_TIMEOUT 0
#define OPENTHREAD_CONFIG_ENABLE_SOFTWARE_RETRANSMIT 0
#define OPENTHREAD_CONFIG_ENABLE_SOFTWARE_ENERGY_SCAN 0
#define OPENTHREAD_CONFIG_ENABLE_PLATFORM_USEC_TIMER 0
#define OPENTHREAD_CONFIG_ENABLE_AUTO_START_SUPPORT 1
#define OPENTHREAD_CONFIG_ENABLE_BEACON_RSP_WHEN_JOINABLE 0
#define OPENTHREAD_CONFIG_MBEDTLS_HEAP_SIZE_NO_DTLS 384
#define OPENTHREAD_CONFIG_ENABLE_STEERING_DATA_SET_OOB 0
#define OPENTHREAD_CONFIG_CCA_FAILURE_RATE_AVERAGING_WINDOW 512
#define OPENTHREAD_CONFIG_ENABLE_TX_ERROR_RATE_TRACKING 1
#define OPENTHREAD_CONFIG_FRAME_TX_ERR_RATE_AVERAGING_WINDOW 128
#define OPENTHREAD_CONFIG_CHANNEL_MONITOR_SAMPLE_INTERVAL 41000
#define OPENTHREAD_CONFIG_CHANNEL_MONITOR_RSSI_THRESHOLD -75
#define OPENTHREAD_CONFIG_CHILD_SUPERVISION_INTERVAL 129
#define OPENTHREAD_CONFIG_INFORM_PREVIOUS_PARENT_ON_REATTACH 0
#define OPENTHREAD_CONFIG_PARENT_SEARCH_RSS_THRESHOLD -65
```

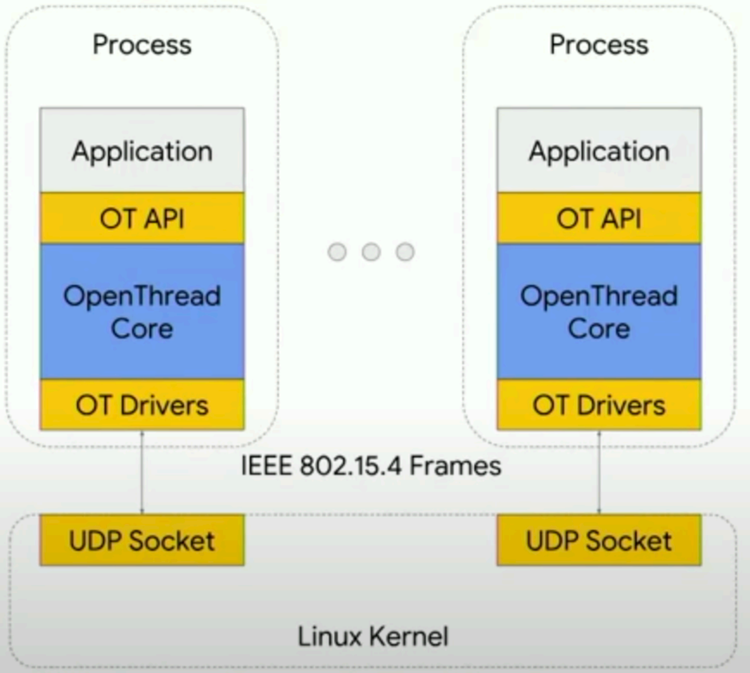
POSIX Emulation

Simulate and test OpenThread networks without hardware.

Faster than real-time.

Codelab: openthread-simulation

Github: OpenThread-Simulator-Tutorial



Toranj - Testing framework

Python based test framework

Simulate complex network topologies

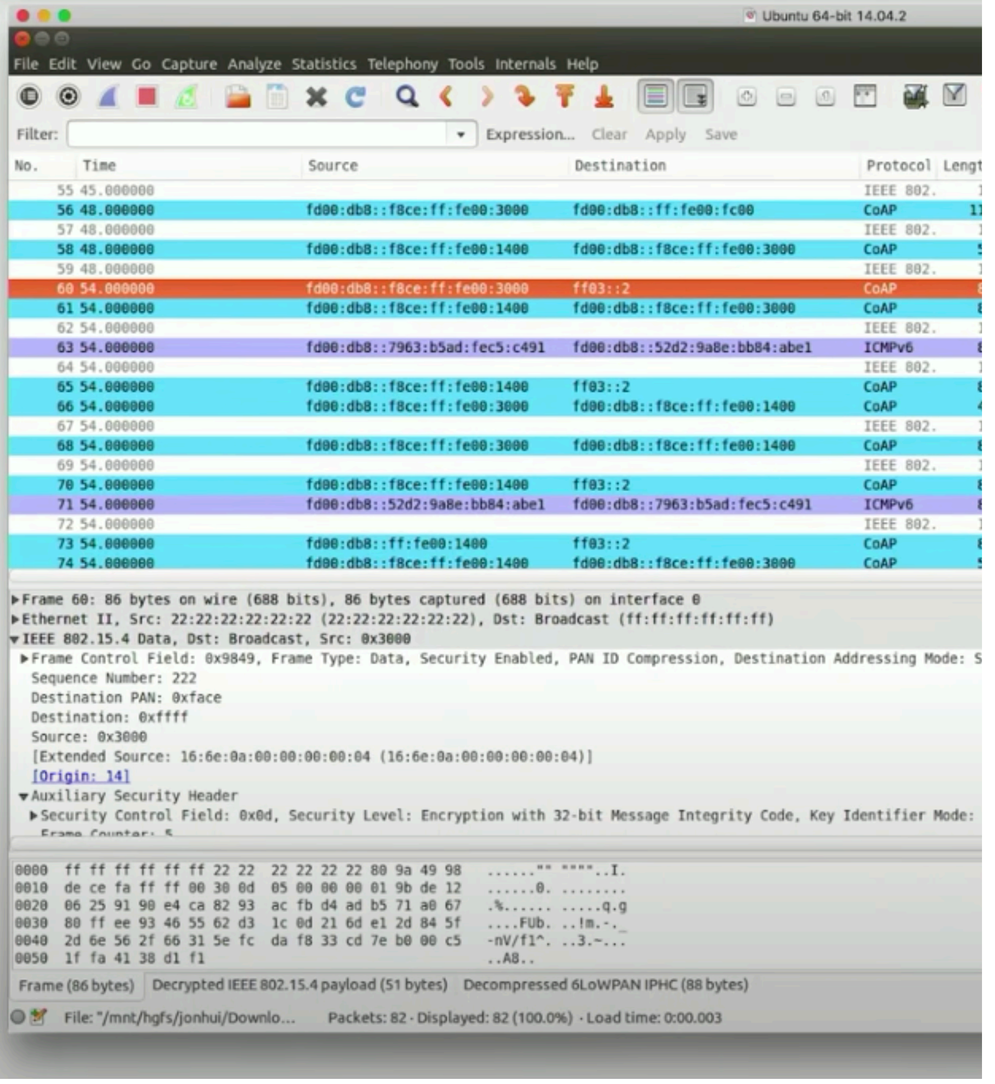
Test IPv6 traffic across several nodes

```
>>> import wpan
>>> node1 = wpan.Node()
>>> node2 = wpan.Node()
>>> node3 = wpan.Node()

>>> wpan.Node.init_all_nodes()

>>> node1.form("test-PAN")
'Forming WPAN "test-PAN" as node type "router"\nSuccessfully formed!'

>>> node1.whitelist_node(node2)
>>> node2.whitelist_node(node1)

>>> node2.join_node(node1, wpan.JOIN_TYPE_ROUTER)
'Joining "test-PAN" C474513CB487778D as node type "router"\nSuccessfully Joined!'

>>> node3.whitelist_node(node2)
>>> node2.whitelist_node(node3)

>>> node3.join_node(node2, wpan.JOIN_TYPE_END_DEVICE)
'Joining "test-PAN" C474513CB487778D as node type "end-device"\nSuccessfully Joined!'

>>> print node2.get(wpan.WPAN_THREAD_NEIGHBOR_TABLE)
[
  "EAC1672C3EAB30A4, RLOC16:9401, LQIn:3, AveRssi:-20, LastRssi:-20, Age:30, LinkFC:6,
MleFC:0, IsChild:yes, RxOnIdle:yes, FFD:yes, SecDataReq:yes, FullNetData:yes"
  "A2042C8762576FD5, RLOC16:dc00, LQIn:3, AveRssi:-20, LastRssi:-20, Age:5, LinkFC:21,
MleFC:18, IsChild:no, RxOnIdle:yes, FFD:yes, SecDataReq:no, FullNetData:yes"
]
>>> print node1.get(wpan.WPAN_THREAD_NEIGHBOR_TABLE)
[
  "960947C53415DAA1, RLOC16:9400, LQIn:3, AveRssi:-20, LastRssi:-20, Age:18, LinkFC:15,
MleFC:11, IsChild:no, RxOnIdle:yes, FFD:yes, SecDataReq:no, FullNetData:yes"
]
```

Traffic Sniffer

Wireshark includes dissectors for Thread.

NCP supports packet capture in both monitor and promiscuous modes.



https://openthread.io

Thread 1.1 + Tools and features	github.com/openthread
Platforms	https://github.com/openthread/wpantund
Guides, samples and codelabs	https://developer.android.com/things/sdk/apis/lowpan
Developer community	codelabs.developers.google.com/codelabs/openthread-simulation
Open source	

Lecture outcomes

- Learned the key concept that form the OpenThread framework.
- Created a demo that is building a thread network.