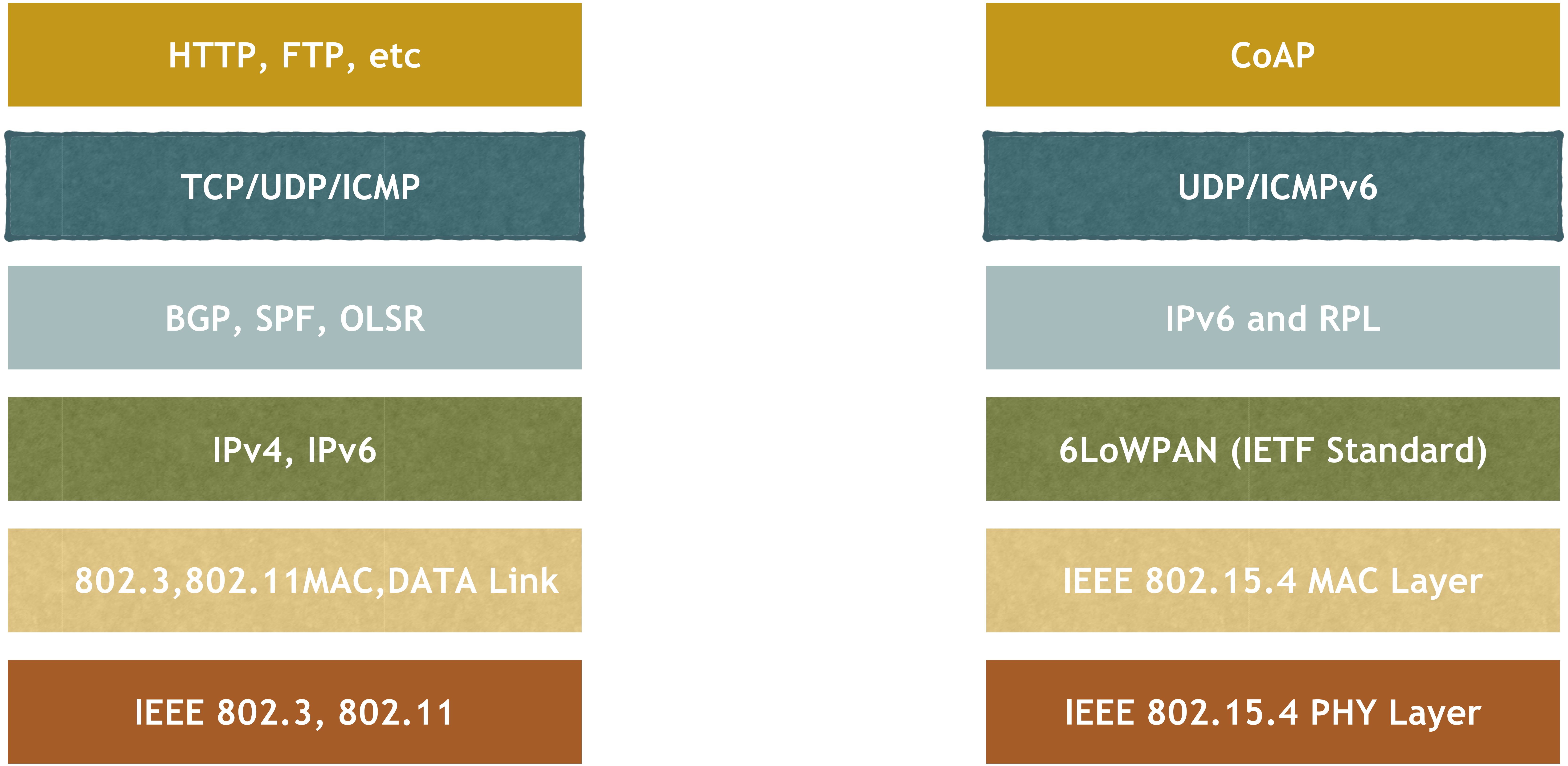
OVERVIEW

- Internet of Things (IOT)
- Open Standard for IOT
- IOT demands
- OS Design for constrained devices
- Tiny OS
- Contiki OS
- RIOT OS
- NuttX OS
- OS comparisons
- Demo

INTERNET OF THINGS (IOT)

- Internet Connected Objects like RFIDs, Sensors, Actuators, Instruments and smart appliances.
- Context aware communication and computing
- Mainly works with IPv6 rather than IPv4
- Powered mainly by Sensor Nodes (Motes) which are low cost, small size and power efficient
- Real Time Guarantee

OPEN STANDARD OF IOT



HTTP, FTP, etc	CoAP
TCP/UDP/ICMP	UDP/ICMPv6
BGP, SPF, OLSR	IPv6 and RPL
IPv4, IPv6	6LoWPAN (IETF Standard)
802.3,802.11MAC,DATA Link	IEEE 802.15.4 MAC Layer
IEEE 802.3, 802.11	IEEE 802.15.4 PHY Layer

WHAT DOES IOT DEMAND?

- Low Power, Low Cost and Low memory footprint (RAM and ROM)
- Should have provision for IPv6, with 6LoWPAN Adaptation Layer
- Separate Routing Protocol for Low Power and Lossy networks
- New light Weight Application Layer Protocol unlike http but should have a support for http also
- Header compression for IPv6 against IEEE 802.15.4 MAC

OS DESIGN FOR CONSTRAINED DEVICES

- **Structure of the Kernel**
  - **Monolithic**
    - Lacks modularity and becomes complex once size exceeds a limit
  - **Layered Approach**
    - Segregation between the kernel mode and user mode
  - **Micro Kernel**
    - This method is preferred as only limited set of tools runs under Kernel Mode and most other drivers and modules runs under User mode which very easy to handle and kernel will not be overloaded.
- **Programming Model**
  - **No memory segmentation**
  - **Every process runs in its own thread and has its own memory stack**

OPERATING SYSTEMS

- **TinyOS**
- **Contiki OS**
- **RIOT OS**
- **Free RTOS**
- **NuttX**

TINY OS

- Monolithic Kernel
- Uses a programming language nesC
- It provides algorithms, protocols, device drivers, file systems and a shell
- Stopped its support and is not extensively maintained

tinyos.net

CONTIKI OS

- Layered architecture
- Uses C and partial C++ support
- Provides device drivers, communication and sensor data handling as services
- Also comprises uIP Stack, a device driver loader and a Protothreading system
- Protothreads are simple light weight multi threading system that uses a stackless implementation
- Code replacement during runtime

contiki-os.org/

RIOT OS

- Micro Kernel Approach
- Tickless scheduler (It switches to idle thread during idle)
- Designed in such a way that the kernel functions are scheduled under a low clock speed.
- Full support of C and C++ is available and partial support of POSIX is ensured
- Full support for RPL, 6LoWPAN, IPv6, TCP, UDP, etc. It's completely modular

riot-os.org

RIOT OS

- Few 100 bytes of RAM and ROM is enough to load the OS into the Motes
- Support multi threading and Real Time
  - through POSIX
  - Zero latency interrupt handlers
  - minimum context switching times
  - the kernel will never crash because of a error prone device drivers

NuttX

- Created by Gregory E. Nutt
- First public release: Feb 2007
- Supports 8 to 32 bits uCs/uPs
- Release under BSD License
- Small footprint
- Very customizable
- Inspiration from Linux/Unix:
  - VFS
  - MTD
  - PROCFS
  - NuttShell

nuttx.apache.org

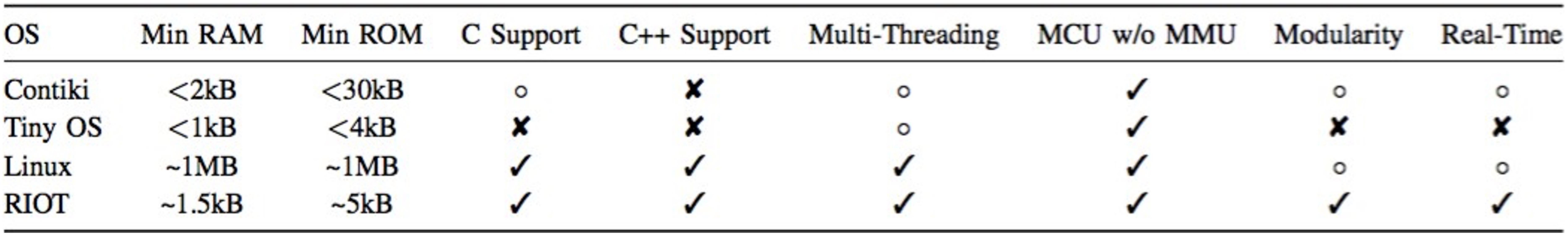
NuttX

- POSIX complaint
- Fully preemptible
- Virtual File System (VFS)
- Loadable kernel modules
- Symmetric Multi-Processing (SMP)
- Realtime scheduling (FIFO, RR, SPORADIC)
- Tickless operation support (lower power consumption)
- Pseudo-terminals (PTY) and I/O redirection

OS COMPARISONS

OS	TinyOS	Contiki OS	RIOT OS
Availability	For older Sensors, Mica Family, ATMega, Telosb, etc	For older and Recent Sensors like Mica, Telosb,	For all new sensors
Support	Support has been stopped	Still actively maintained	actively
Programming	nesC	Partial C	C,C++
Simulator	TOSSIM	native emulation is available	native method
RAM,ROM	<1kb, <4kb	<2kb, <30kb	~1.5kb, ~5kb
Multithreading	No	No (But Protothreads)	Yes
Modularity	No	Partial Support	Yes

OS COMPARISONS



OS	Min RAM	Min ROM	C Support	C++ Support	Multi-Threading	MCU w/o MMU	Modularity	Real-Time
Contiki	<2kB	<30kB	○	✘	○	✓	○	○
Tiny OS	<1kB	<4kB	✘	✘	○	✓	✘	✘
Linux	~1MB	~1MB	✓	✓	✓	✓	○	○
RIOT	~1.5kB	~5kB	✓	✓	✓	✓	✓	✓

NuttX OS DEMO

- Simple Hello World Example
- Simple LED Blinking & Serial Shell Example

REFERENCES

1. *Jiakai Li, Gursel Serpen, "Simulating Heterogeneous and Larger-Scale Wireless Sensor Networks with TOSSIM TinyOS Emulator" Procedia Computer Science 12 (2012) 374 – 37*
2. *Danilo F.S. Santos, Hyggo O. Almeida, Angelo Perkusich, "A personal connected health system for the Internet of Things based on the Constrained Application Protocol", Computers and Electrical Engineering 44 (2015) 122–136*
3. *Zhengguo Sheng et al, "Lightweight Management of Resource Constrained Sensor Devices in Internet of Things", IEEE Internet of Things Journal, Vol 2, no.5, October 2015, pp 402-411*
4. *Charity Perera et al, "A Survey on Internet of Things from industrial Market Perspective" IEEE Access 2014*
5. *Daniel Willmann, "Contiki - A Memory efficient Operating System for Embedded Smart Objects", 2009*
6. https://www.nsnam.com/
7. https://www.youtube.com/watch?v=4bSr5x5gKvA