CCNx in Every Sensor

Bilel Saadallah, Abdelkader Lahmadi, Olivier Festor

CCNx in Every Sensor

Bilel SAADALLAH, Abdelkader LAHMADI, Olivier FESTOR

Context and Motivation

- **Growth of interest in ICNs**
- **Large expansion of IoT and WSN**

Can CCNx be used as the core protocol in WSN applications?

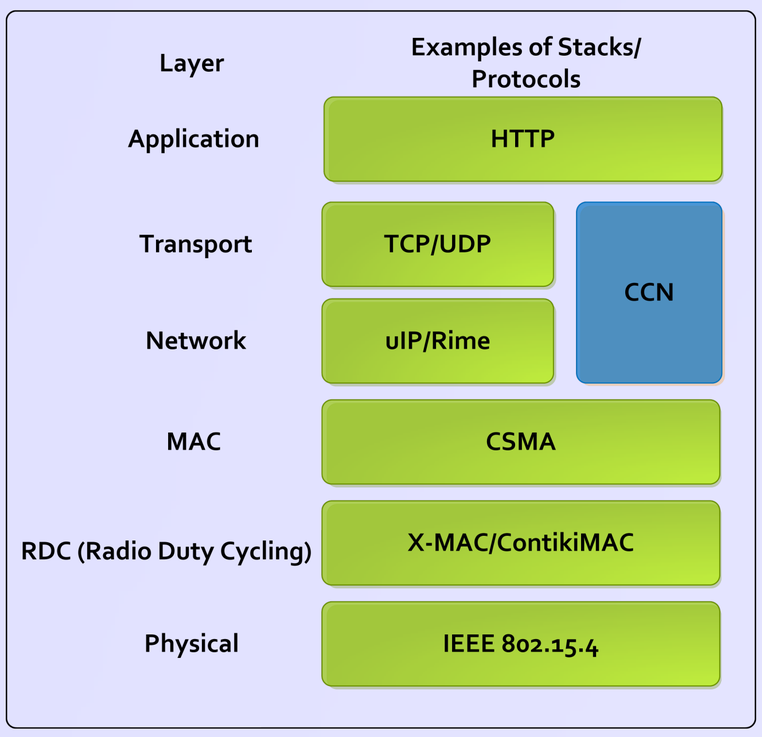
Approach

Contiki Operating System: *http://www.contiki-os.org/*

- The Open Source Operating System for the Internet of Things
- Contiki allows tiny, battery-operated low-power systems communicate with the Internet
- Providing several communication stacks

CCNx Stack in Contiki

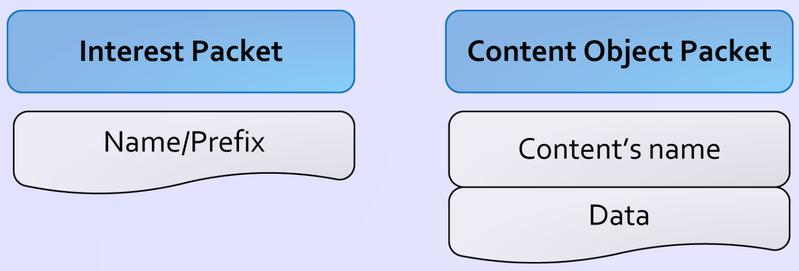
- Implements CCN processing functions, message forwarding and data caching
- Manages event posting to processes
- CCN Driver: handles message exchange with the lower layer



- Hierarchical names with components of arbitrary lengths
- Names in URI representation
- Content storage at intermediate nodes

Message Adaptation

- Simplify Interest and Content Object messages



Summary

- C language CCNx stack in Contiki

Interested in getting the code, please contact:
abdelkader.lahmadi@loria.fr

Challenges

Protocol adaptation

- Develop a fully operational CCN communication stack
- Adjust CCN messages and optimize Interest for reused data collection
- Overcome the IEEE 802.15.4 frame size limit (102 bytes for payload)

TelosB : TPR2400CA/TPR2420CA

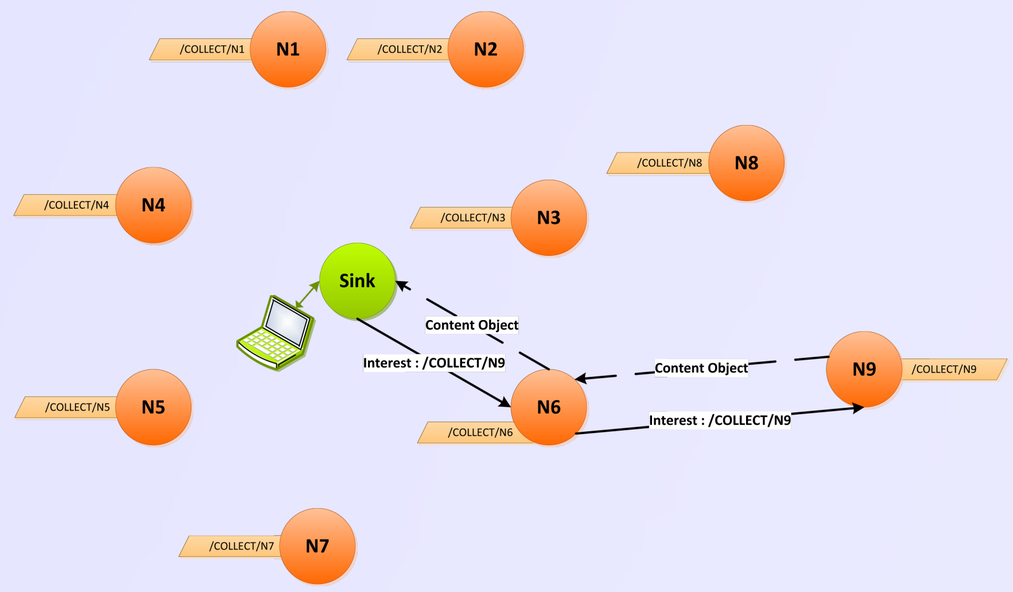
Processor Performance	16-bit RISC
Program Flash Memory	48K bytes
Measurement Serial Flash	1024K bytes
RAM	10K bytes
Configuration EEPROM	16K bytes
Frequency band	2400 MHz to 2483.5 MHz
Transmit (TX) data rate	250 kbps
Outdoor Range	75 m to 100 m
Indoor Range	20 m to 30 m

Management applications adaptation

- Collect desired management content
- Monitoring-oriented naming scheme
- Apply in-network processing functions for data aggregation
- Minimize communication overhead
- Improve delivery speed and reduce congestion

Experiment & Results

Topology



Parameters

- Available content at each node N_i: temperature, humidity, battery, light, RSSI, ETX
- 180 seconds between two collection requests to every node
- Interest with prefix /COLLECT/Ni to collect data from node i
- Content updated every 70 seconds at each node N_i

Delivery Delays

Delay (ms)	**1st Round**	**2nd Round**	**3rd Round**
Node 1	250	250	250
Node 2	328	258	250
Node 3	250	258	258
Node 4	328	258	258
Node 5	328	265	258
Node 6	461	265	258
Node 7	461	258	258
Node 8	461	258	258
Node 9	328	258	258