Towards Evaluating Named Data Networking for the IoT: A Framework for OMNeT++

Amar Abane[1 2], Paul Muhlethaler[3], Samia Bouzefrane[1], Mehammed Daoui[2], and Abdella Battou[4]

[1] Conservatoire National des Arts et Métiers - Paris, France
[2] University Mouloud Mammeri - Tizi-Ouzou, Algeria
[3] Inria - Paris, France
[4] National Institute of Standards and Technology - MD, USA

Pisa – Italy – September 05-07 2018

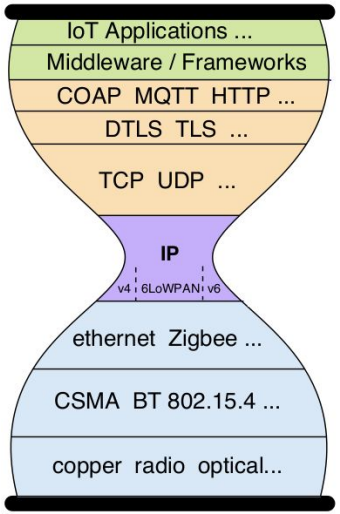
Outline

1. Named Data Networking & IoT
2. NDN-OMNeT design
3. Use case example

Named Data Networking & IoT

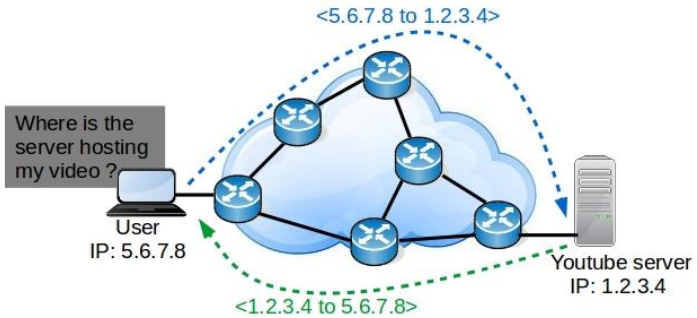
Current IP stack

- Most apps are content-based (e.g. facebook, youtube, skype, etc.)
- DNS, P2P, CDN to support content-based applications
- The applications view DNS names as their namespace
- The network layer views IP addresses as its namespace
- Need name resolution
- Need middleware



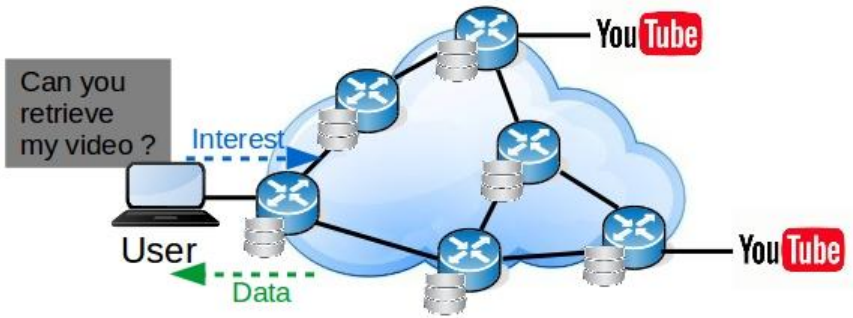
Paradigm shift

Focus on delivering packets



Host-based Networking

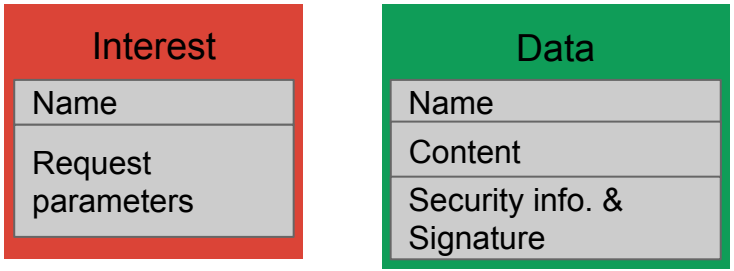
Focus on retrieving content



Information Centric Networking (ICN)

Named Data Networking

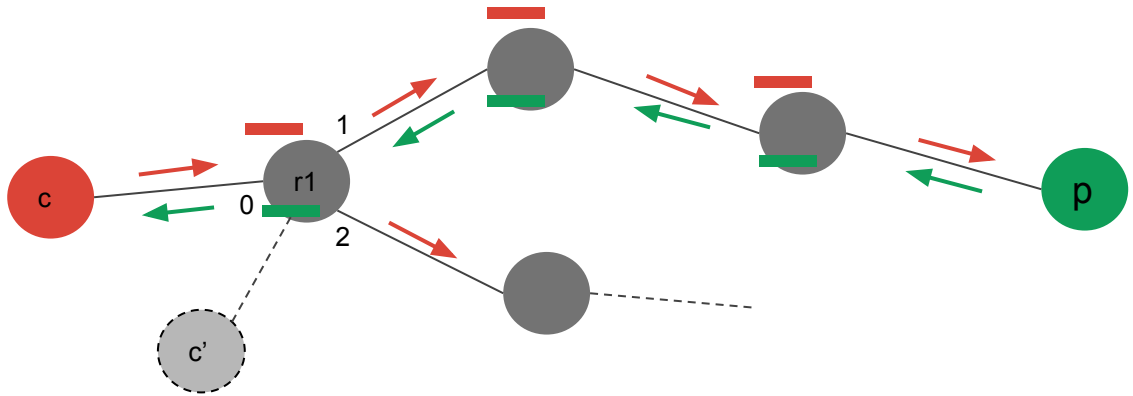
- ***Hierarchical*** *names (e.g. /UniPisa/campusA/room1/temperature)*
- ***Packet routing/forwarding directly on names***
- ***Two packet types: Interest & Data***
- *Content, name and producer bind with crypto-signature*



NDN communication

Interest	/UniPisa/campusA/room1/temp	47fa2d0c	5
	Name	Nonce	HopLimit

Data	/UniPisa/campusA/room1/temp/3pm	22°C	ECDSA	0A...65
	Name	Content	SignatureT	Signature



r1's FIB

prefix	face
.../room1/*	1, 2

R1's PIT

Interest	face
/UniPisa/...	0

r1's CS

Data
/UniPisa/...

Opportunity for the IoT

NDN provides a native support for IoT

- **Security**
  Secure IoT data directly.
- **Mechanism**
  Mobility support, asynchrone, natural names (close to CoAP)
- **Lightness**
  Implementations (e.g. NDN-RIOT) show that NDN can be lighter than 6LoWPAN on IoT devices.
- **Projects**
  NDN Building Automation System, Home automation, etc.

NDN for low-end IoT: challenges

Considering a low rate/power wireless technology (e.g. IEEE 802.15.4)

- **Wireless forwarding**
  Native NDN (over L2), reduced overhead, feasibility with current IoT devices.
- **Constrained devices**
  Packet processing/size (small MTU).
- **Naming**
  Name size/processing & semantics, FIB management, etc.
- **Device management**
  Trust model, bootstrapping, service discovery.

Evaluating NDN-IoT solutions

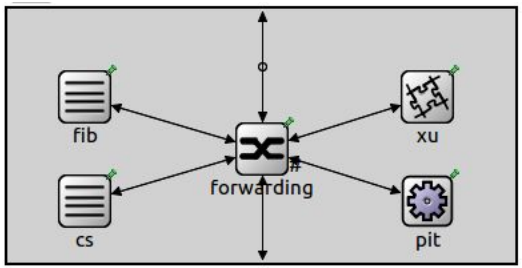
ndnSIM (ns-3) is widely used, but...

- **Visualization**
  For understanding and teaching purposes
- **Not only networking**
  Need to evaluate memory consumption, etc.
- **Quick simulations**
  Need to test features with minimal coding
- **OMNeT++/INET**
  Simulate system/network interactions, NDN data structures, etc.

NDN-OMNeT design

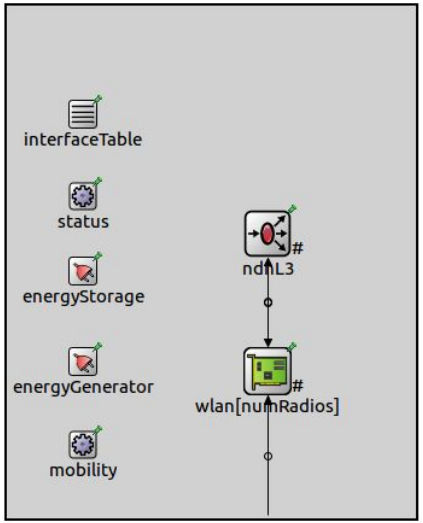
NDN core

- NDN as an L3 protocol (*NdnL3*)
- Based on INET 3.5
- Compound module that includes
  - Pending Interest Table (*IPit, PitBase*)
  - Forwarding Information Base (*IFib, FibBase*)
  - Content Store (*ICs, CsBase*)
  - eXperimental Unit (*IXu*)
  - Forwarding strategy (*IForwarding, IForwardingBase*)
- Communication by module access or messages



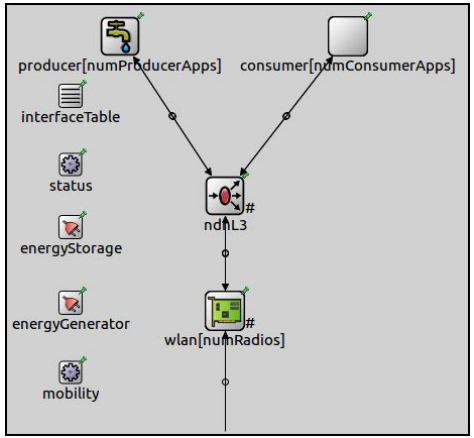
Hosts

- A typical wireless IoT device (*NdnWirelessHostBase*)
  - Basic NDN host
  - Includes NDN core as a network layer
  - Ready to act as relay node
- A typical IoT end-device (*NdnWirelessHost*)
  - Extension of the basic NDN host
  - Consumer and/or producer apps
  - Ready to act as end-device (e.g. sensor)



Applications

- Consumer app (*ConsumerAppBase*)
  - Sends Interests under a given prefix
  - Parameters: prefix, #Interests, lifetime, sendInterval, length, etc.
- Producer app (*ProducerAppBase*)
  - Responds to incoming Interests with a Data packet under a given prefix
  - Parameters: prefix, length, freshness, etc.



Packets

- NDN uses TLV packet representation
- NDN-OMNeT supports
  - Straightforward packet definition (i.e. extension of cPacket)
  - TLV representation and size computation (for packet processing evaluations)
  - Non-NDN fields are used for evaluation purposes

Use case example

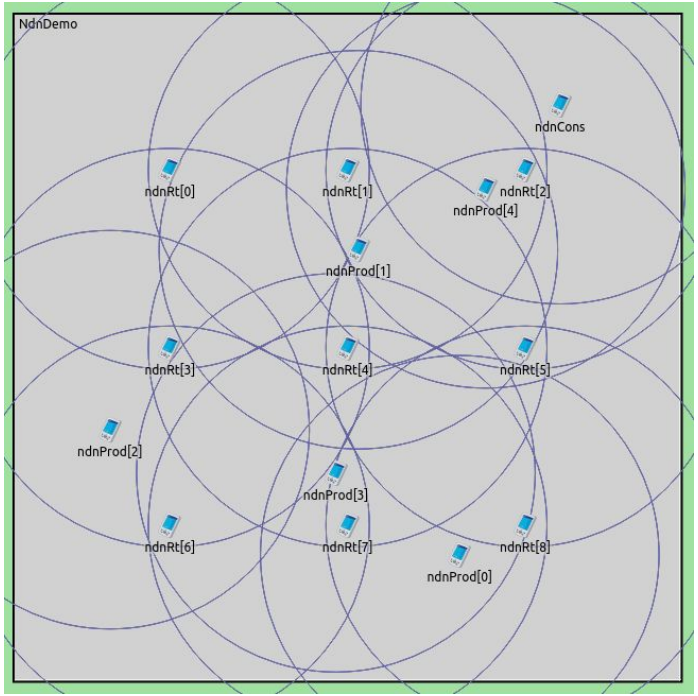
NDN wireless forwarding

Basic approach (related work)

- First Interest is broadcasted (flooding)
- Nodes keep/update temporary FIB entry after getting Data
  - In the FIB: NDN prefixes mapped to MAC addresses
- Flooding triggered by consumer after Interest timeout
- Delayed retransmissions
  - To reduce useless broadcasts
  - If a node overhears packet with the same prefix, the delayed reTx is canceled
- Different NDN-to-MAC mapping: (parameter in Forwarding module)
  - IUDU: Interest Unicast Data Unicast
  - IBDB: Interest Broadcast Data Broadcast
  - IBDU: Interest Broadcast Data Unicast
  - IUDB: Interest Unicast Data Broadcast

Simulation

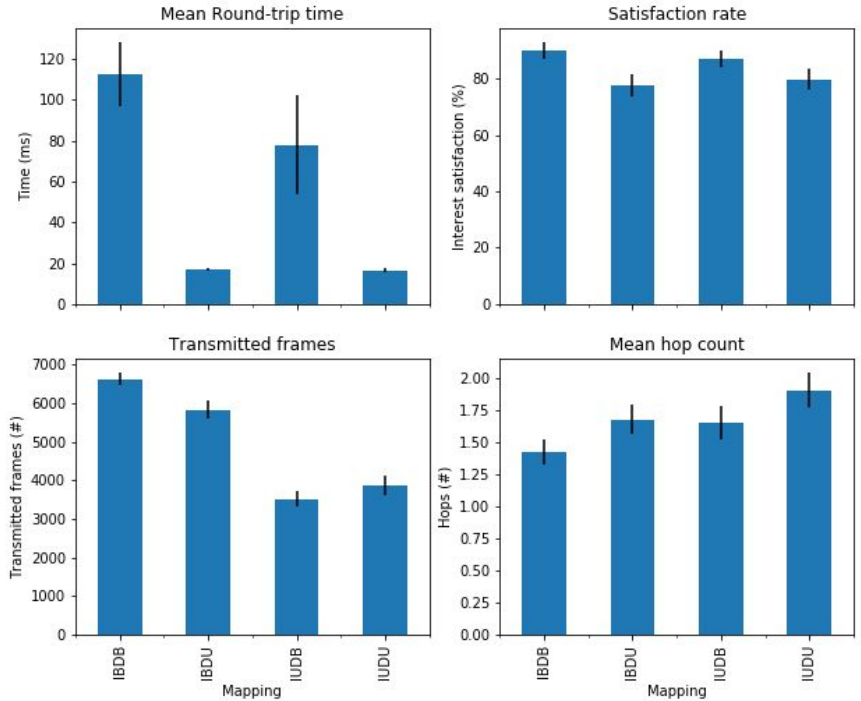
Topology



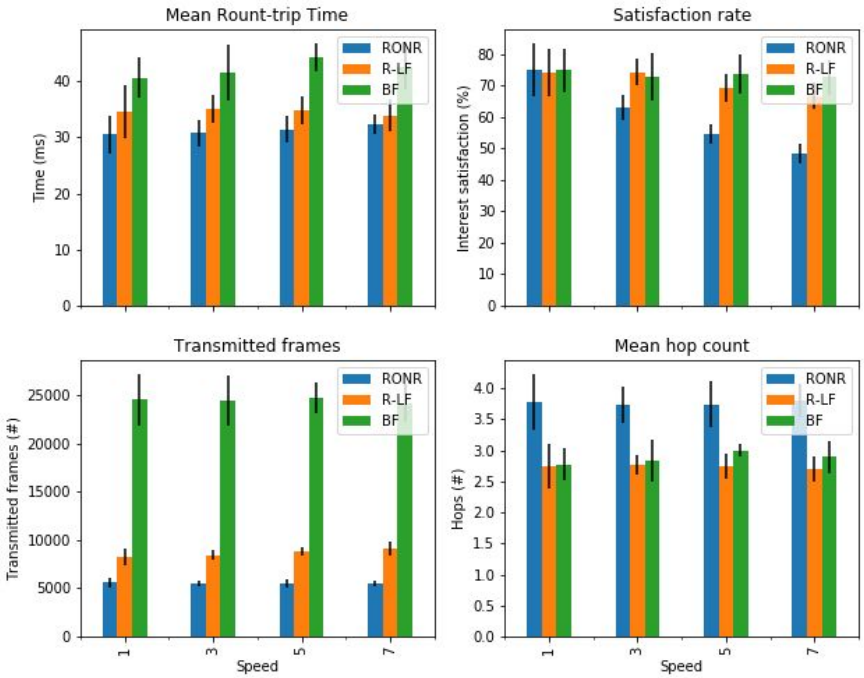
Metric

- Collisions number
- Satisfaction rate
- Interest-Data RTT
- Total transmitted frames
- PIT size/lookups
- ...

Results (1)


Mean Round-trip time
Time (ms)
0
20
40
60
80
100
120
Satisfaction rate
Interest satisfaction (%)
0
20
40
60
80
Transmitted frames
Transmitted frames (#)
0
1000
2000
3000
4000
5000
6000
7000
Mean hop count
Hops (#)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
IBDB
IBDU
IUDB
IUDU
Mapping

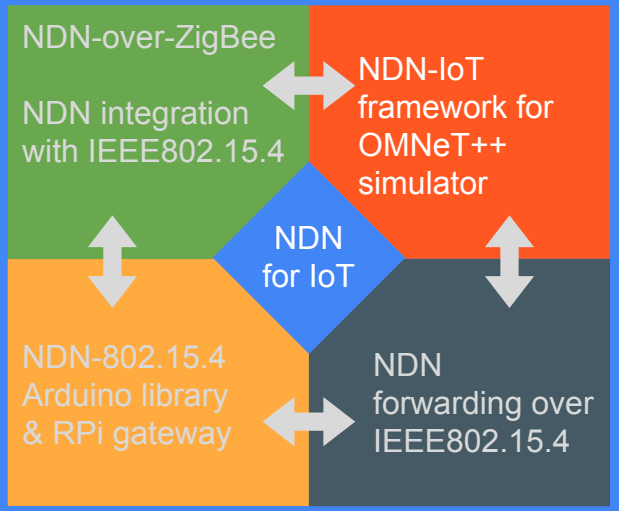
Results (2)


Mean Rount-trip Time
Time (ms)
RONR
R-LF
BF
Satisfaction rate
Interest satisfaction (%)
Transmitted frames
Transmitted frames (#)
Mean hop count
Hops (#)
Speed
1
3
5
7

Conclusion & Future work

NDN-OMNeT	• Extend OMNeT with ICN paradigm • A tool for evaluating NDN-IoT solutions
Forwarding	• Other strategies already included • Need a fully-customizable forwarding module
Future features	• Support NDN TLV packet processing • Memory/processing models for NDN data structures
Compatibility	• Deal with OMNeT/INET versions • Other integration/compatibility suggestions...

This work is part of:
A realistic NDN integration in the IoT



Thank you!

Repo:

https://github.com/amar-ox/NDNOMNeT

Contact:

amar.abane@inria.fr