TDMH-MAC: Real-time and multi-hop in the same wireless MAC

Federico Terraneo, Paolo Polidori,
Alberto Leva, William Fornaciari

Politecnico di Milano, Italy

RTSS 2018
Nashville, 13 December 2018

Quick context

This talk is about a Medium Access Control (MAC) protocol

- targeting real-time applications
- for wireless networks
- built on top of the IEEE 802.15.4 PHY layer
  - the one also used by Zigbee

This work is part of a research line on time deterministic distributed systems (previous RTSS papers)

2014 FLOPSYNC-2 clock synchronization
sub-μs, sub-μA, off-the-shelf hw components

2015 propagation delay compensation
brought sub-μs precision to large scale multi-hop networks

2016 quantization-aware clock synchronization
relevant when only a low resolution timebase is available

2017 high accuracy, low power timebase
synchronizing RTC and high resolution clock on same device

Why a MAC?

Network time uncertainty going through the TCP/IP stack

- PHY
  - ns to μs
  - time jitter of sending a symbol
- MAC
  - ms
  - time for accessing the channel (statistical multiplexing)
  - 3-6 order of magnitude increase from previous layer!
- IP/TCP
  - 10's to 100's of ms
  - multiple hops, internetworking
  - only 1-2 orders of magnitude increase from previous layer

Why a MAC?

On the Internet

- huge number of devices
- scalability is a primary concern
- statistical multiplexing is a necessity
  - makes networking equipment (almost) stateless

Real-time communication is needed in local networks

- not as many hosts as the Internet
- can make the network stateful
  - route streams, not packets
- added complexity paid off with time determinism
  - and with simplification of upper layers
- manageable also on microcontrollers

Desired feature #1

guaranteed end-to-end latency

- without statistical multiplexing

Why we want it:

- no self-interference, no frame queues
- worst case latency close to average case latency
- can push the PHY to its limits

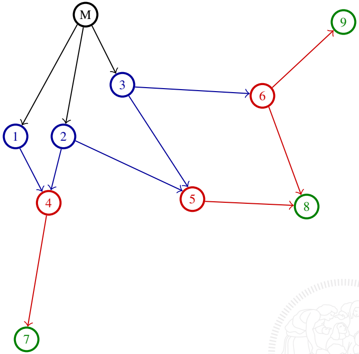
Desired feature #2

full mesh topology

- not a star network
- and no spanning tree on the mesh
- and dynamically updated too

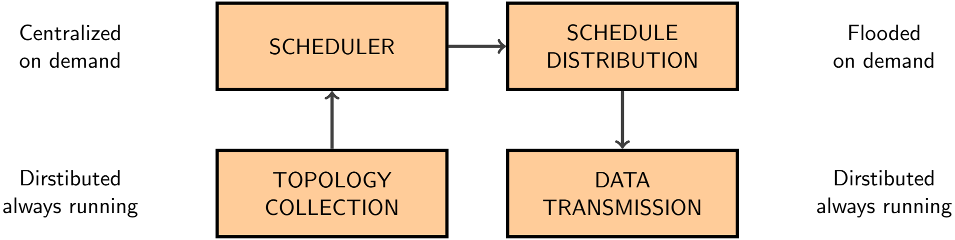
Why we want it:

- every link counts
- can do spatial redundancy
- can balance load
  - for increased throughput
  - for energy balancing



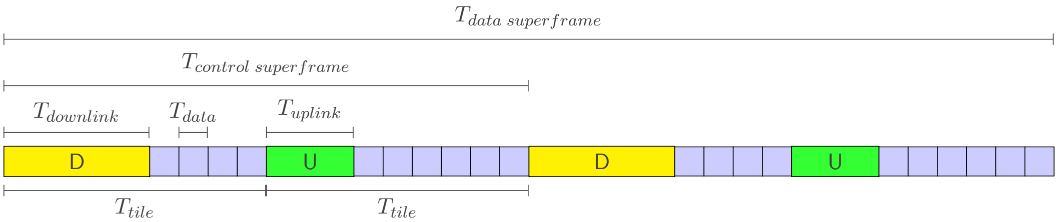
TDMH-MAC architecture

TDMH-MAC is a centralized TDMA protocol.



Main challenge is efficient topology collection

TDMH-MAC architecture



TDMH-MAC is temporally organized in *tiles*

- each tile begins with a control part for network management
  - downlink, for clock synchronization and schedule dissemination
  - uplink, for topology collection and stream management
- followed by a number of slots for data frames

Tiles repeat periodically

- control superframe
  - the shortest repeating sequence of downlink/uplink tiles
- data superframe
  - the shortest repeating sequence of data transmission (schedule)

The role of constructive interference flooding

flooding broadcasts a frame across all hops of a network

- is performed using solutions suh as Glossy...
- ...that use constructive interference
  - all nodes of a given hop transmit simultaneously
- to function, Glossy needs to embed a hop counter in the frame
  - nodes become aware of their distance in hops from the flood originator
  - TDMH-MAC makes use of this previously overlooked information

In TDMH-MAC the master node floods control downlink frames using Glossy

- used for clock synchronization and schedule dissemination
- by participating in the flood, nodes become aware of their distance in hops from the master

Topology collection

Goal: collect information on the current network topology and forward it to the master node

Control uplink slots are used

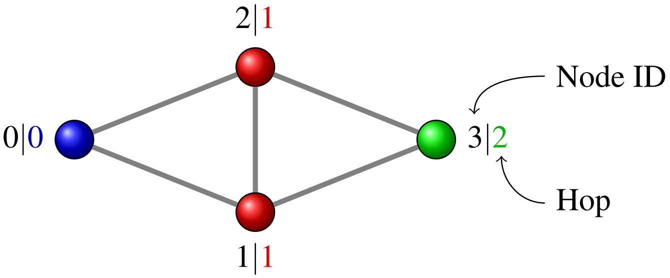
- at every control uplink only one node transmits
  - round-robin scheme
  - possible because nodes are already synchronized
  - frame is *not* flooded
    - only nodes in radio range receive it

Transmitted frame contains

- node ID
- hop it belongs to
- node ID of a forwardee, which will forward this frame towards the master
- the list of its neighbors (bitmask-encoded)
- zero or more forwarded topologies

Topology collection

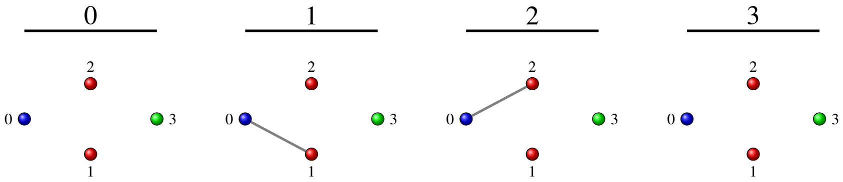
An example is shown with the following topology



We will see a full *network formation* example

Topology collection

Network knowledge of each of the four nodes



Immediately after the first clock synchronization flood

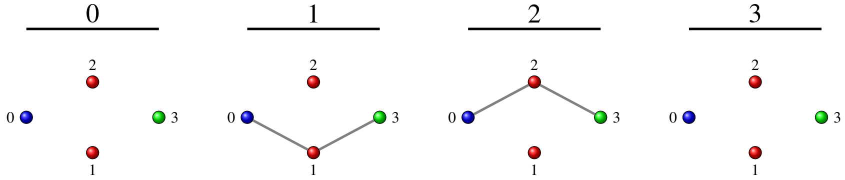
- all nodes know their hop
- node 0 and 3 have no topology information
- node 1 and 2 know that they can reach the master node
    - they infer it from belonging to hop 1

Topology collection

The round robin topology collection starts from the highest numbered node

Node 3 transmits an uplink frame with:

NodeID $= 3$, Hop $= 2$, Forwardee $=$ none, Neighbors $= \emptyset$, Forwarded $= \emptyset$

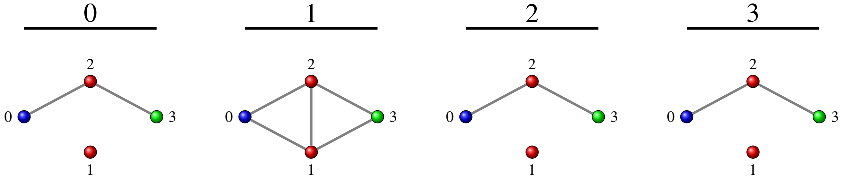


- nodes 1 and 2 overhear the frame and know they can reach 3

Topology collection

Node 2 then transmits an uplink frame with:

NodeID = 2, Hop = 1, Forwardee = 0, Neighbors = {0,3}, Forwarded = ∅

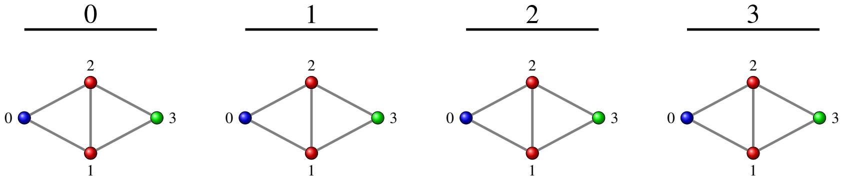


- node 3 now knows it can reach 2
  - it now has a predecessor to forward its topology to
- node 1 reaches full knowledge of the network
- other nodes reach a partial network knowledge

Topology collection

Node 1 finally transmits an uplink frame with:

NodeID = 1, Hop = 1, Forwardee = 0, Neighbors = {0,2,3}, Forwarded = $\emptyset$



- all nodes reach full network knowledge
- the round robin restarts so as to keep the topology information updated

Topology collection

In the general case

- forwarding of topology information may be necessary
- at each forwarding, data is guaranteed to get closer to the master
- only the master will converge to full network knowledge
  - other nodes may only have partial network knowledge
  - but this is not an issue, since TDMH-MAC is centralized

The data phase

Goal: transmit application data

In TDMH-MAC, application data is organized in streams

- logical point-to-point links between nodes
- can be opened and closed dynamically

Streams have the following properties

- dedicated bandwidth
- guaranteed period between application-level packets
  - latency bound equals the period
- a redundancy level

The data phase

The master node schedules streams in data slots

- breaks down a stream in individual transmissions between hops
- guaranteed no self-interference
- spatial redundancy can be supported

Nodes just execute the given schedule
For every data slot each node either

- takes a frame from upper layers and transmits it
- receives a frame into a local buffer
- transmits a frame from a buffer
- receives a frame and passes it to upper layers
- or sleeps

The TDMH scheduling problem

The TDMH-MAC scheduler is thought as a customization point

- the scheduling problem has been formalized in first order logic
  - allows to handle the problem *in abstracto*
- an OMNeT++ simulator for TDMH is available
  - allows to implement new schedulers
- TDMH-MAC runs on WandStem nodes
  - the same codebase runs both on the OMNeT++ simulator and on the nodes

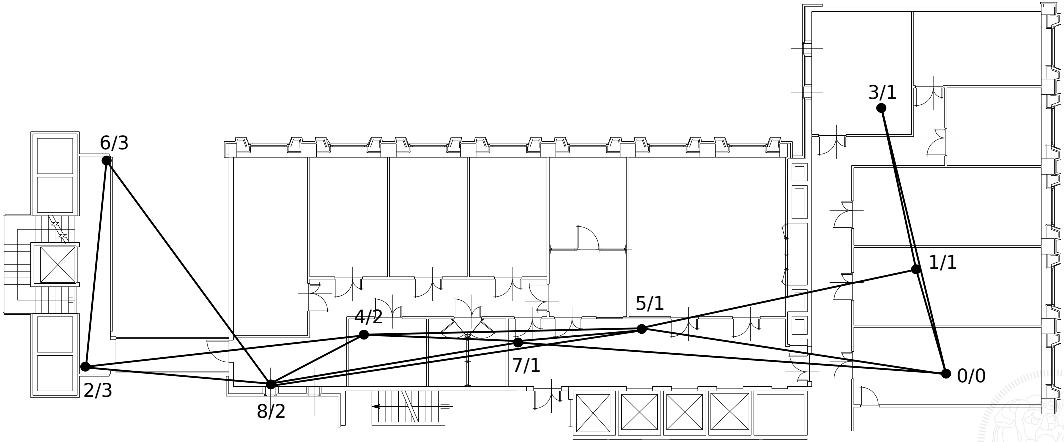
Scalability considerations

TDMH-MAC has been characterized through simulations and experiments

- it can scale to beyond 100 nodes and 10 hops
- further scaling would require a faster PHY that 802.15.4
  - main limitation: 250kbit/s data rate
- we will here show the result of an experiment
- for the simulation characterization, please see the paper

Experimental evaluation

TDMH-MAC was tested on a network of 9 nodes at Politecnico di Milano

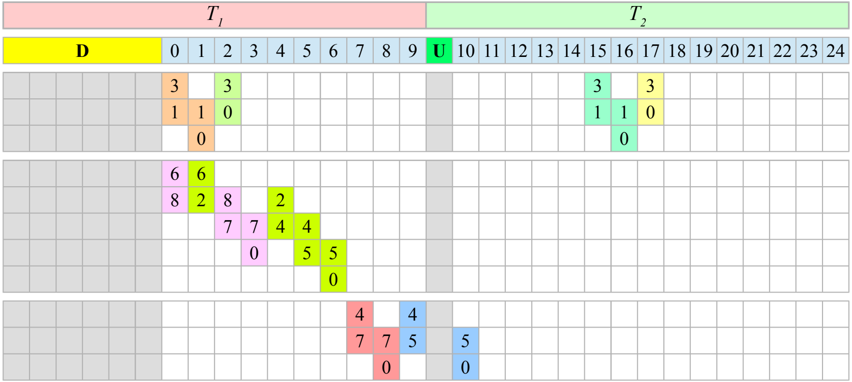


The topology shown is the one produced by the topology collection algorithm

Experimental evaluation

Three streams were opened

Source node	Dest node	Period
3	0	100ms
6	0	200ms
4	0	200ms



Experimental evaluation

Two one day experiments were performed

- without redundancy
- with double spatial redundancy

Link reliability was as follows

Stream	Without redundancy	With redundancy
$3 \rightarrow 0$	99.56%	99.67%
$6 \rightarrow 0$	95.40%	99.97%
$4 \rightarrow 0$	97.45%	99.00%

Conclusions

- We proposed TDMH-MAC, a wireless MAC fo real-time applications
- Compared to the current state of the art, our solution
  - provides latency bounds in multi-hop networks
  - supports full-mesh networks efficiently

Feature	TDMH-MAC	TSCH	DSME	LLDN	rt-WiFi
Multi-hop	✓	✓	✓		
Guaranteed period	✓			✓	✓
Spatial redundancy	✓				
Temporal redundancy	✓	✓	feasible	feasible	✓
Management	C	C/D	D	C	C
Topology	mesh	ct	ct	star	star

[1] centralized [2] distributed [3] cluster-tree

Thank you

federico.terraneo@polimi.it
paolo.polidori@mail.polimi.it
alberto.leva@polimi.it
william.fornaciari@polimi.it

Code can be found at github.com/fedetft/tdmh