**Wireless Network Protocols**

Classified by transmission technology and scale

Transmission technology: broadcast and point-to-point links

In wireless, all transmissions are via broadcast. All network nodes listening to the network are receiving the transmission

**Formal methods for stability analysis of network….**

A **Networked Control System** (**NCS**) is a [control system](https://en.wikipedia.org/wiki/Control_system) wherein the control loops are closed through a communication [network](https://en.wikipedia.org/wiki/Computer_network). The defining feature of an NCS is that control and feedback signals are exchanged among the system's components in the form of information packages through a network.

**Todo:**

x: Understand the Wireless networked control systems  
x: Understand the protocol 802.15.4 -> how does it work?  
x: What is a sender, what is a channel

x:Understand the PTA modelling, understand the PTA (invariant, clock, guard …)  
x:Understand how to model the PTA in PRISM  
x:Understand how the paper modelled the PTA

o: implement a simple version of the parallel network (2 simple sender and 1 channel)  
o: implement the full version of the parallel network  
o: test it with the results of the paper

o: generate a test data set  
o: use AALERGIA on it  
o: model check the AALERGIA model with the properties

o: talk to masoud

**Network control systems:**

Connectiing control systems components (sensors, controllers, actuators) via network efficiently reduces the complexity of the system. Data can be shared efficitnetly over the network. It is easy to add more sensors and other units with very little cost. They connect cyber space with physical space, making task execution from a distance easily accessible. Applications: space exploration, factory automation, hazardous environments, domestic robots, automobiles, aircraft, …

Example: Robot in Japan being controlled via a computer in the USA (robot gives feedback signal, has sensors, control signal is send from the USA)

*NCS components:*

* Acquisition (sensors)
  + Collecting relevant data using distributed sensors in the network to study system under control
  + Can be in any form (video, images, but also small numbers like temperature)
  + Important factors:
    - Bandwith requiremenets for data transfer
    - Data collection strategies for several sensors
    - Cheap, reliable and energy effictient sensors
* Command (controllers)
  + Applying the typical control strategies (PID, adaptive control, …) is a challenging task
* Communication and control (actuators)
  + Reliable
  + Secure
  + Ease of use
  + Availability

Network reliability is always an important factor for NCS:

* Problems are:
  + Delays
    - Time to read sensor, measure it and send a control signal to an actuator depens heavily on the network
    - Performance of NCS can be affected significantly by the network delays
    - Can destabilize a system (because delay affects the ability to track e.g. a robots path)
  + Jitter
  + Losses
  + Inference

Bandwidth problems are compensated with different scheduling methods, Network Security is also a factor to be considered.

backoff

 (networking)

A [**host**](http://encyclopedia2.thefreedictionary.com/host) which has experienced a [**collision**](http://encyclopedia2.thefreedictionary.com/collision) on a [**network**](http://encyclopedia2.thefreedictionary.com/network) waits for a amount of time before attempting to retransmit. Arandom backoff minimises the probability that the same nodes will collide again, even if they are using the samebackoff algorithm. Increasing the backoff period after each collision also helps to prevent repeated collisions, especially when the network is heavily loaded.

**802.15.4 Protocol: How does it work:**

Protocol for WPAN networks. Physical layer (responsible for sending bits) and the Medium access control layer (MAC layer).

Goals are:

* Low energy consumption
* Cheap hardware
* Parallel usage of several protocols

The basic framework conceives a 10-meter communications range with a [transfer rate](https://en.wikipedia.org/wiki/Transfer_rate)of 250 kbit/s. Tradeoffs are possible to favor more radically [embedded devices](https://en.wikipedia.org/wiki/Embedded_system)with even lower power requirements, through the definition of not one, but several physical layers Focuses on low-cost, low-speed ubiquitous communication between devices. It can be contrasted with other approaches, such as [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi), which offer more bandwidth and require more power.

As already mentioned, the main identifying feature of IEEE 802.15.4 among WPANs is the importance of achieving extremely low manufacturing and operation costs and technological simplicity, without sacrificing flexibility or generality.

CSMA/CA is responsible for collision avoidance.

Is used, because devices cannot see if a collision happened -> hidden node problem

* <https://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_avoidance>

FFD (full function device) acts as coordinators, RFD (reduced function devices) can only coummunicate with FFDs.

Possible topologies:

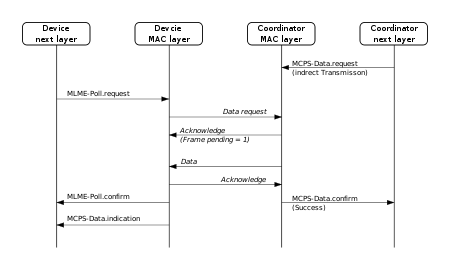
* Star
* and Peer to Peer

[Frames](https://en.wikipedia.org/wiki/Data_frame) are the basic unit of data transport, of which there are four fundamental types (data, acknowledgment, beacon and MAC command frames)

* <https://en.wikipedia.org/wiki/Beacon_frame>

The physical medium is accessed through a [CSMA/CA](https://en.wikipedia.org/wiki/CSMA/CA) protocol. Networks which are not using beaconing mechanisms utilize an unslotted variation which is based on the listening of the medium, leveraged by a [random exponential backoff](https://en.wikipedia.org/wiki/Exponential_backoff) algorithm; acknowledgments do not adhere to this discipline. Common data transmission utilizes unallocated slots when beaconing is in use; again, confirmations do not follow the same process.

Unslotted Mode:

[](https://de.wikipedia.org/wiki/Datei:Indirect.svg)

Datenübertragung vom Koordinator zum Knoten ohne Beacon

Vor jedem Sendevorgang überprüft ein Teilnehmer per CSMA/CA, ob der Kanal belegt ist und sendet seine Daten, sobald er frei wird. Optional kann er im gesendeten Paket angeben, ob er eine Antwort (ACK) wünscht, anhand derer er überprüfen kann, ob das Paket korrekt ankam oder die Übertragung gestört wurde. Dieser Modus erfordert keinerlei Verwaltungsaufwand durch den PAN-Koordinator.

Unterschieden wird zwischen 3 Kommunikationsszenarien:

1. Daten von Teilnehmer an PAN-Koordinator. Koordinator empfängt Daten und sendet, wenn angegeben, ein ACK. Um die Daten empfangen zu können, muss sich der Koordinator stets empfangsbereit halten und darf sich nicht in einen energiesparenden Modus kurzzeitig schlafen legen.
2. Daten von Teilnehmer an Teilnehmer (Peer-to-Peer-Topologie). Wie im ersten Szenario muss sich der Empfänger stets empfangsbereit halten.
3. Daten von PAN-Koordinator an Teilnehmer. Teilnehmer fragt regelmäßig nach, ob Daten für ihn beim Koordinator liegen. Dieser antwortet (ACK) und sendet die Daten, wenn welche vorliegen, oder ein leeres Datenpaket, wenn er keine Daten für den Teilnehmer besitzt. Siehe rechte Abbildung. Auch hier bleibt der Koordinator stets empfangsbereit.

Bezogen auf den Energiebedarf muss in diesem Modus folglich ein Knoten, für gewöhnlich der PAN-Koordinator, immer empfangsbereit sein, während die anderen Knoten die meiste Zeit Energie sparen können.

**Timed automaton (timed automaton)**

In [automata theory](https://en.wikipedia.org/wiki/Automata_theory), a **timed automaton** is a [finite automaton](https://en.wikipedia.org/wiki/Finite_automaton) extended with a finite set of real-valued clocks. During a run of a timed automaton, clock values increase all with the same speed. Along the transitions of the automaton, clock values can be compared to integers. These comparisons form guards that may enable or disable transitions and by doing so constrain the possible behaviors of the automaton. Further, clocks can be reset. Timed automata are a sub-class of a type [hybrid automata](https://en.wikipedia.org/wiki/Hybrid_automaton).

Formally, a **timed automaton** is a tuple *A* = (*Q*,Σ,*C*,*E*,*q*0) that consists of the following components:

* *Q* is a [finite set](https://en.wikipedia.org/wiki/Finite_set). The elements of *Q* are called the *states* of *A*.
* Σ is a finite set called the *alphabet* or *actions* of *A*.
* *C* is a finite set called the *clocks* of *A*.
* *E* ⊆ *Q*×Σ×B(*C*)×P(*C*)×*Q* is a set of edges, called *transitions* of *A*, where
  + B(*C*) is the set of boolean clock constraints involving clocks from *C*, and
  + P(*C*) is the [powerset](https://en.wikipedia.org/wiki/Powerset) of *C*.
* *q*0 is an element of *Q*, called the initial state.

An edge (*q*,*a*,*g*,*r*,*q'*) from *E* is a transition from state *q* to *q'* with action *a*, guard *g* and clock resets *r*.

The states are also called locations.

The transitions are also called switches. There is a finite amount of clocks, they can be reset. If you read the time of the watch, you get the time that passed since the last reset. Each switch can do a time constraint (guard), that only works if the actual time is in the constraints.

Invariant can be specified for each location (state). The time can only stay as long in one location, as specified in the invariant. (if no invariant is specified, system can stay as long as it wants in one location).

**Paper content:**

Goal is the stability analysis for WNCS. Communication system is a PTA, stability condition is written in PTL -> satisfaction of PTL expression is equivalent to stability of the WNCS.

Basic idea: derive stability condition from control system and convert it into a specification that can be checked. Then the communication system is modelled as a PTA and model checking is performed on it. If the specification cannot be met, the controller needs to be redesigned or the communications parameters need to be tuned.

**Thoughts:**

Apparently, I need to convert the PTA model to an MDP (because I need the outputs).

The outputs are the states that are traversed -> needed for the model!

How do I deal with the parallel models? Convert it to one model? Handle it as 3 models?

* Prism supports that

Model the PTA with modest and convert it to an MDP

Model needs to be converted by hand, with digital clock semantics

Probleme bei Umsetzung:

* PRISM unterstützt kein Random (übergang zwischen BackoffTime und Backoff)
  + -> lösung, mit aufzählung aller random optionen, muss bei modellen dann händisch angepasst werden. Siehe Bluetooth case study als beispiel (random bei module receiver)
* Verhalten von PRISM bei der Simulation? Läuft es parallel oder nicht?
  + Ich denke es läuft parallel, aber die Module/Prozesse laufen nicht deterministisch ab!
* Wie modelliert man die Zeiteinheit? Also die Sekunden?
  + Siehe WLAN beispiel, ma n nimmt eine basis einheit (z.b. 50us und dividiert die anderen Zeiten danach, rundet ab und auf oder entscheidet deterministisch welcher wert es ist (z.b basis 50, gewünschte zeit 83, kann also 1oder2 sein, da es inzwischen liegt)
* BACKOFF TIME, ist eine float einheit, PRISM kann nur integer die sich verändern?
  + Scale up and down
* Synchronisiations prozess (Siehe TICK)

Todo:

* Weitere Paper lesen über digital clock semantics -> versuchen code zu verstehen
* Bestes Beispiel: WLAN (bei PRISM -> case studies, hier ist aiuch ein code)
* Time is modelled via scaling, take a basic time unit and scale it

TimeUnits:

Tb contains 20 symbols \* Ts. One symbol has 4 bits and the bitrate is 250kbps, so Ts = 16us and TB = 320us

Packet length is nondeterministic and is Integer \* Tb

Common deadline is h -> Sampling period, each station has the same deadline

|  |  |
| --- | --- |
| Tb (Backoff Period) – Basiseinheit | 320us |
| Tcca (CCA, Clear Channel Analysis) | Tb |
| BoPeriod | Tb |
| Data (packet length) | Nondeterministic, Integer \* tb, (varying between 20Tb-30Tb, nondeterministically) |
| NB | Backoff tries, init to 0 and max default is 4 |
| Bemin | Default value 3 |
| Bemax | Default value 5 |
| (2^(Bemin)-1)\*Tb | * Bemin 3: 7 Tb * Bemin 4: 15 Tb * Bemin 5: 31Tb |

Model finished

* Nondeterministic data length needs to be implemented
* Rewards
  + **Pmin**=? [ pathprop ]
  + **Pmax**=? [ pathprop ]
* Testing

Ways to do the trace execution:

* Traces of the whole model
* Generate traces for each individual system and combine them later in prism

prism IEEE\_802.15.4\_protocol.prism -simpath 1000,vars=(s,s2,c1,c2),sep=comma path.csv

How to export the model from prism!

type: dot -Tps –v filename.dot -o outfile.ps

* Possible errors (always waits 1TB before it continues, but there is the possibility that 0TB is the Backoff duration! (in s1)
* *R*max(time)=?[true *U done*1 *∧ done*2]*. Implement it like that*
* *Deadlocks sind schuld?*
* *Schuld ist c1=c2, c2=c1 in neuem modul!*

Jedes mal bei time nimmt er nichtdeterminstisch eines von den beiden modulen anstatt beide gleichzeitig auszuführen?

* Keine collision kann durchgeführt werden!
* Abstraction? Siehe hermann 3 token! Abstraktion wie z.B. start, sending, finish, failed
* Wie Data aufbereiten, sodass das programm es lesen kann?
* Test ob es ausreicht, nur channel zu checken
* Achtung! Fehler bei der Erstellung des PRISM models bei den states im PRISM file -> state 0 sollte beginnen und nicht state 1 -> ändern! Evt nochmal mit matlab PRISM model vergleichen!
  + done
* Man kann auch matlab allergia versuchen den MDP lernen zu lassen
* Teste wie gut der code bei herman3 die properties approximiert
* Problem ist vermutlich, dass es einen übergang von den states gibt und AALERGIA lernt, dass es letztendlich immer in c3c3 landet -> der Zeitfaktor fehlt!
  + Gelöst -> siehe unten.
* WORKED:
  + P=? [ F<=80 "c3c3" ] -> Abfrage auf Model, 80 sind die Steps im dtmc (equivalent zu den time units). Wichtig ist hier, dass man die daten immer anhand der time änderungen ausgibt, dann kann AALERGIA daraus ein Model produzieren.

Kann man aus PRISM labels ausgeben?

Abstraktionsebenen:

start, sending, finish, failed

oder gleich wie herman 3 problem:

sender1:s

sender2:s

time: (0->running, 1->over)

s: 0-5 states

s2:0-5 states

c:0-2 states

Kein exakter wert: Problem, dass die Zeitsteps nicht ganz dem entsprechen, was sie sollen:

* Dem nachgehen 🡪 only time values are interesting!

**Schreiben:**

Lets do this:

First paper: AALERGIA

A way to refine AALERGIA is by using active learning techniques -> interactive data acquisition.

Statistical model checking can also be used to check probabilistic properties of the system -> this works by checking example traces of a system whtether a logical property holds with a given minimum proability. -> here, an explicit probabilistic model is extracted which supports verification without resampling the whole system…model can also be used for other MDD tasks.

By selecting different predicates or actions, different models support different views of the system (abstraction). If we tailor the observation of the properties to the properties of interest -> verification gets easier and more efficient.

Main difference of this algorithm to other alergia implementations: compatibility is defined with the distributions P\_Tqr, P\_Tqb of the original tree (and not of the current automaton.) -> better empirical interpretation

And they didn’t use a fixed threshold for the compatibility criterion, but a data-dependend threshold.

AALERGIA is run with different alpha-values and evaluated with the BIC score. The best alpha is found by the golden section search, each run optimises the BIC score.

In order to use abstraction, they generated the outputs and “sampled” it by removing certain values and just calling it “point” -> in the dice game.

Learning procedure can be extended to both inputs and outputs (-> further work)

Second Paper: Learning Markov Decision processes

1. [] **Nondeterministic**: I have no idea what nature will do.
2. [] **Probabilistic**: I have been observing nature and gathering statistics.

Algorithm can be refined by active learning to take advantage of data acquisition.

Finish paper!

Get better results!

Next up: slides for presentation

Move presentation one week behind

Instructions for Masoud to run the tool and test all the results.

* Get the right results!
* Matlab test (done->same result)
* More variables
* Check if time is right (done->is right)
* Check if model in workspace is same as sample model (done->is the same)
* Channel model produces similar results
* More variables: -> unterteilung, sollte NB bei 4 schon aufhören?

AALERGIA components, understand them:

FPTA -> Normalization (DFFA2DPFA)->Find a Search Region (calculate\_search\_region)->golden section search->AALERGIA (Merge, calculate\_compatible\_parameter, AAlergia\_compatible)->export to prism

**FPTA**:

Pretty well understood

**Normalization**:

Pretty well understood

**Find a Search Region:**

Left epsilon starts with 1 and gets halfed in each run -> *BIC score is evaluated*

We are searching for a maximum: if new score is bigger than old score, we found a new right border

If new score is smaller than old score, we found our left boarder

From the epsilon we found for the left boarder:

Epsilon gets doubled every time -> *BIC score is evaluated*

If new score is smaller than old score -> right boarder is found

If new score is bigger than old score -> we found a new left border

*BIC score evaluation:*

AALERGIA with new value -> output is a dffa -> normalize dffa to get a dpfa

Calculate likelihood takes the new DPFA (deterministic, probabilistic finite automaton) and multiplies all the probabilities of all the traces together.

BIC score is calculated with likelihood!

BIC score is BIC = likelihood – ½ \* Automata size \* log(sample size)

The bigger, the better!

**Golden Section Search:**

Searches extremum with a triple of points whose distance from a golden ration

* Calculates BIC score for the points a1 and a2
* Goes smaller and smaller until extremum is found

Explain golden section search?

**AALERGIA:**

**Merge**

**Calculate\_compatible\_parameter**

Needs a lot of insight!

**AALERGIA\_compatible**

Needs a lot of insight!

**Export to Prism:**

Needs a little bit more insight