

# ENERGY CONSUMPTION IN WIRELESS MESH NETWORKS

---

Ulrik Wilken Rasmussen  
Aalborg University



**Title:**

Energy Consumption in Wireless Mesh  
Networks

**Project period:**

Januar 1<sup>st</sup> - May 1<sup>th</sup>, 2012

**Project group:** 12gr10

**Group member:**

Ulrik Wilken Rasmussen

**Supervisor:**

Frank Fitzek  
Stephan A. Rein

**Abstract:**

This report documents the development of an analytical model of both throughput and power consumption in a wireless mesh network with the Alice and Bob topology. A MeasuringTool have also been designed to measure the actual power consumption in the network.

The analysis shows that applying network coding saves energy for high load scenarios. This is validated with the MeasuringTool.

The model derived and measurements achieved have demonstrate that network coding gives even benefits for the case of low load.

**Number of copies:** 4

**Number of pages:** 30

**Appended documents:**

(3 appendix, 1 CD-ROM)

**Total number of pages:** 37

**Finished:** May 2012



---

# Preface

---

This report is developed as an 10<sup>th</sup> semester project for the master program in Networks and Distributed Systems at Aalborg University. The project was carried out in the spring semester of 2012 under the supervision of Stefan Rein. The project was proposed by Frank Fitzek.

Attached to this report is a CD with an electronic copy of the report (high resolution figures), the developed simulation software and a electronic versions of the bibliography.

## Motivation

Wireless mesh network are evolving rapidly and is a hot topic in network communication. The master project Inter-flow Network Coding for Wireless Mess Networks introduces applied network coding in the mesh network. The report showed that when applying network coding the point of congestion is mowed, and the mesh network then is able to gain a higher throughput. Wireless mesh networks can be grouped into two groups.

First group is the wireless mesh networks that have planned configuration, and may be deployed to provide dynamic and cost effective connectivity over a certain geographic area. These mesh routers are often not limited in terms of resources.

The second group is the wireless ad hoc mesh network that is formed with wireless devices, e.g. cell phones, when they come within communication range of each other. The nodes in this wireless ad hoc mesh network acts both as routers and nodes. This requires that nodes are in promiscuous mode and awake when other devices are using the mesh network. The nodes are often constrained by its resources e.g. battery life.

This raises the interesting topic; The throughput is higher in wireless mesh networks with network coding than without network coding how does this difference affect the power consumption in the network.

This question have also been addressed in the perspective in [1] because when applying network coding, it creates the requirement that all the nodes have to be in promiscuous mode, which increase processing since all packets are filtered in software.

To evaluate the possible difference in energy consumption in the wireless mesh network without and network coding, this project designs, develops and implements an energy measuring unit to be attached to notes using the routing protocol B.A.T.M.A.N adv.<sup>1</sup> Patched with CATWOMAN<sup>2</sup>.

## Acknowledgments

In the process of writing this report, feedback and ideas have been received from several persons. I will like to thanks Martin Hundebøll for answering questions regarding CATWOMAN and especially Achuthan Paramanathan for a good corporations in the design and developing of the MeasuringTool.

---

<sup>1</sup>Better Approach To Mobile Ad-hoc Networking

<sup>2</sup>Coding Applied To Wireless On Mobile Ad-hoc Networks

---

## Notes and References

The report makes use of cross references. Figures, tables, and source code are numbered and these numbers are used in the text. In some cases, the size of the figures and tables makes it necessary to place it on the following page.

When using external sources, information about the source is given in the bibliography in appendix C on page 37. The source information is referenced by a text and year encapsulated in braces. A reference to the example source is [Example 08b].

## Author Signature

---

Ulrik Wilken Rasmussen

---

# Table of Contents

---

<b>Table of Contents</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction to mesh networks . . . . .	1
1.2 Routing algorithms . . . . .	2
1.3 network coding . . . . .	3
1.4 Coding gain . . . . .	4
<b>2 Project Description</b>	<b>7</b>
2.1 Project contribution . . . . .	8
2.2 Requirements specification . . . . .	8
2.3 Delimitations . . . . .	8
<b>3 Theoretical activity model</b>	<b>9</b>
3.1 The wireless mesh network configuration . . . . .	9
3.2 Alice and Bob routing scheme with and without NC . . . . .	9
3.3 Node activity model without network coding . . . . .	10
3.4 Node activity model with network coding . . . . .	12
3.5 Calculation the power consumption . . . . .	14
<b>4 System description</b>	<b>15</b>
4.1 MeasuringTool analysis . . . . .	15
4.2 Current measuring . . . . .	16
4.3 Voltage measuring . . . . .	17
4.4 Data collector . . . . .	17
4.5 Integration of the MeasuringTool . . . . .	18
4.6 Software design . . . . .	18
4.7 Calibration of the MeasuringTool . . . . .	19
<b>5 Testing the devices</b>	<b>23</b>
5.1 Preliminary power measurements . . . . .	23
5.2 Experiment setup . . . . .	23
<b>6 Conclusion</b>	<b>29</b>
6.1 Further work . . . . .	29
<b>A MeasuringTool source code</b>	<b>31</b>
<b>B MeasuringTool PCB layout of HW</b>	<b>33</b>
<b>C Paper - Energy Consump. Model and Meas. result. for NC M Wireless N.</b>	<b>35</b>

**Bibliography**

**37**

---

## Introduction

---

A large number of research projects have begun aiming their focus on investigating the so called green wireless communication. Some of the publications are [2], [3] and [4].

Green communication is about reducing the overall energy consumption in order to reduce the CO<sub>2</sub> emission caused by ICT<sup>1</sup>.

A real explanation of the interest in this research area is that the electricity bills coming to network and service operators are quite high. So by reducing the energy consumption and still have the same service is greeted as an interesting topic. Power saving have been a big issue for quite some time, especially in small wireless devices e.g. mobil phones, where for instance the battery capacity is small. The paper [5] deals with the energy consumption in a sensor network and how to effectively use the energy of the network to extend the network lifetime.

This report continues the work in CATWOMAN [1], that introduced network coding in the wireless mesh network of B.A.T.M.A.N adv.[6].The continuing work done in this report is by investigating the energy consumption in a wireless mesh network to derive the best energy efficiency. The report will look especially at the point of congestion and to when the MAC fairness controls the network.

Beside this report a article have been written in cooperation with Achuthan Paramanathan and with help from others. It is found in appendix C.

### 1.1 Introduction to mesh networks

In order to design energy efficient protocols, it is necessary to develop models of for the whole or some of the network the protocol is designed for. Even after designing the model, it does not always reflect to true network, which can introduce drawbacks in the designed protocol. To investigate the designed protocol in a real network it is necessary to have measurement tools to validate our protocols.

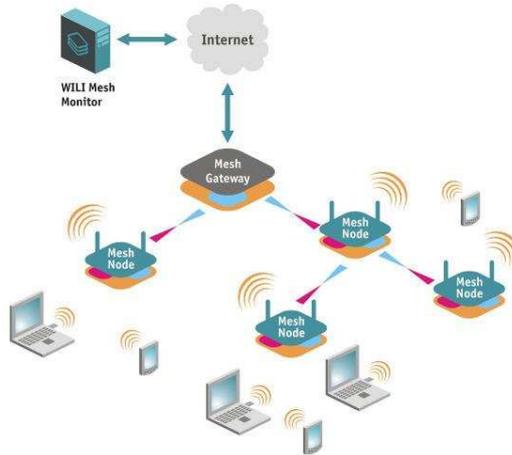
Models help us to quickly understand the behavior of simple network topologies, but they might be harder to derive for larger arbitrary networks. Measurement setups are always time consuming, but can be applied to any topology. Some measurements focused on power consumption of single devices [7]. To support large arbitrary networks energy consumption has to be measured at any node in the network.

In order to model larger networks it is necessary to understand the elementary topologies. This report will focus on the Alice and Bob topology, and compare the results from the analytical model and the measurements for the Alice and Bob scenario. Furthermore the impact caused by the usage of network coding is investigated. When writing about applying network coding to wireless meshed networks, a reference to COPE [4], this introduced a practical method of network coding.

---

<sup>1</sup>Information and Communication Technology

A typical mesh network is shown in figure 1.1. Here the network consists of both nodes



**Figure 1.1:** An example of a mesh network. [8]

The mesh topology is a type of network where each node must not only capture and disseminate its own data, but also serve as a relay for other nodes. A mesh network can for instance use flooding or a routing technique to discover other nodes in the network. A mesh network where all nodes are connected to each other is called a fully connected network. Mesh networks can be seen as one type of ad hoc network. Mobile ad hoc networks and mesh networks are therefore closely related, but normally mesh networks are in fixed locations whereas mobile networks tend to move around with their owner.

The Mesh network consists of a self-healing capability, which enables the routing-based network to operate even if a node breaks down or a connection goes bad. This means that the network is typically quite reliable, as there are large mesh networks often are more than one path between a source and a destination.

## 1.2 Routing algorithms

The following section describes some examples of different routing algorithms used in networks, and ends by explaining problems introduced in an ad hoc network.

### 1.2.1 Shortest path routing

Shortest path routing is a simple routing protocol; it calculates the number of hops to the destination, and from this calculation it selects the shortest. The number of hops to each destination can be estimated by flooding the network for the first packet to be sent to an unknown destination. When a router receives a packet and it doesn't know the way to the destination, the router uses flooding. The packet is sent out on all other lines than it is received on. The packet contains a hop counter. If the length from source to the destination is unknown, the counter is set to the worst case, which is the longest path in the subnet. The received packet is not flooded on the net if the packet has already been received once by the router, but is simply thrown away. To prevent the routing from running out of memory, because it needs to store all incoming flooding packets, the packets are normally added a counter. The router then only needs to keep track of the number of the counter, and can throw packets away with a lower number. A variation of flooding is selective flooding. The router does not send the packet out on every line, but only on the ones that seem to be in the right direction. E.g. there is little point in sending a west-bound packet on an east-bound line. While flooding is not practical in most applications, it still has some usage. In military where a line can be destroyed, the tremendous robustness of flooding is highly

desirable. In distributed database application, it can sometimes be necessary to update all the databases concurrently, here flooding can be useful. In wireless networks a packet transmitted can be received by all stations in range. This property is utilized by network coding. Flooding will in fact always chooses the shortest path, and no other algorithm can produce a shorter delay, unfortunately the overhead generated by the flooding should not be ignored.

### 1.2.2 Distance Vector Routing

Distance vector routing use a dynamic routing algorithm, because static routing does not take the current network load into consideration. The most popular dynamic routing schemes are distance vector routing and link state routing. In distance vector routing, each router contains a table, which is exchange with the neighbors. The router is assumed to know the distance to its neighbors. This can be measured by echo packets that are being time stamped.

Distance vector routing is not good in practice. It adapts to good news fast, but is poor to adapt to bad news, e.g. a router is gone. Only 1 hop per echo. This means that each echo only updates one point at a time. Furthermore the data received from another router doesn't say what route it uses, only it has a time delay to the destination. Distance vector routing was used in the ARPANET in 1979.

### 1.2.3 Routing in ad hoc networks

The difference in an ad hoc network compared to wired networks is that all the usual rules about fixed topologies, fixed and known neighbors, fixed relationships between IP addresses and locations and more are no longer valid. In the ad hoc network routes can come and go. Routing in ad hoc systems is quite different from routing in fixed networks, and several routing algorithm have been proposed for ad hoc networks. Route in a ad hoc network are usually discovery by the nodes sending a route request packet and waits for a route reply packet in return. It is not a good algorithm in large networks, since it generates a lot of packets even though the destination is close by. This can be coped with by adding a time to live on the first packet.[9]

## 1.3 network coding

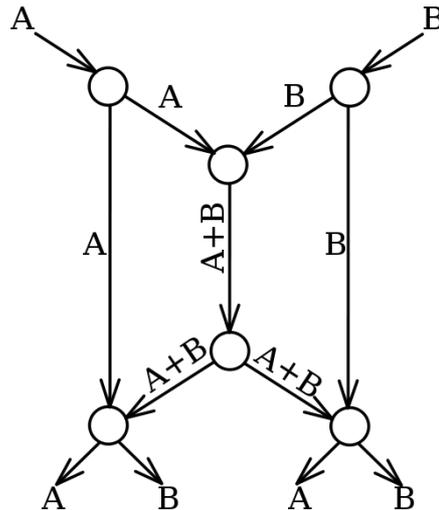
Network coding changes that way that nodes transmits packets in a multi hop network. Instead of just simply store and fording the packets. The packets can be combined an forwarded. An illustration of network coding is illustrated in figure 1.2. The figure illustrates the Butterfly network. The network consists of two sources, located in the top of the picture, each having knowledge of the packet A and B. There are two destination nodes at the bottom of the figure, which each want to know both A and B. Each relay, also called edge can carry only a single value. In each time slot the nodes can only transmit of receive one packet.

If only traditional routing is used, then the central line would be able to carry A or B, but not both. Suppose that the one source sends A through the center, then the left destination would receive A twice and not know B at all. Sending B poses a similar problem for the right destination.

Using a simple coding, as as illustrated both A and B are transmitted by the central line in a combined packet to both destinations simultaneously. The left destination receives A and A+B, and can calculate B by subtracting the two values. This is a linear code because the encoding and decoding schemes are linear operations.

Linearly network coding is just one out of many different approaches to coding network packets. In this report the network linear coding is used, therefore there will not be discussed other forms of network coding, however the list below illustrate a short selection of other types network coding.

- Linear coding
- Random coding
- Static codes



**Figure 1.2:** The Butterfly network, in which network coding is applied. The source nodes in the top transmits two packets towards the two destination nodes in the lower part of the illustration.

- Algorithms/protocols
- Cyclic networks

## 1.4 Coding gain

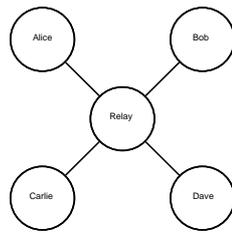
The coding gain in this report is based on the definitions given in [4] Where the Coding Gain is defined as the:

- The ratio of the number of transmissions required by the current non-coding approach, to the number of transmissions used by COPE to deliver the same set of packets.

By definition, this number is greater than or equal to 1. In the Alice and Bob topology, where for instance the packets sent from the end nodes is XOR'ed and broadcasted as a single packet by the Relay node. The network coding reduces the number of transmissions from 4 to 3, as illustrated in figure 3.4 on page 11 and 3.9 on page 13. The results in a coding gain of  $4/3 = 1.33$ . That is the maximum achievable in this setup.

The X topology and Alice and Bob topology examples can be combined to improve the benefits of network coding, as in the cross topology, see figure 1.3. Without network coding it takes 8 transmissions for each flow to send one packet to its destination. If it is assumed that all the nodes have a perfect overhearing one and another, The Relay node can XOR 4 packets in each transmission thus reducing the number of transmissions from 8 to 5, which produces a coding gain of  $8/5=1.6$

The gains shown in this section are only showing the best practice, in real mesh networks, it would not be possible to reach these coding gains, but it is possible to approach the gains.



**Figure 1.3:** Simple topologies to understand th structure of the network. This is the X topology.



---

## Project Description

---

This project is concentrated to examine the power consumption in a wireless mesh network. The wireless mesh network used in this project builds on the B.A.T.M.A.N routing protocol. The protocol has in a master's project at AAU been further development to include the possibility of applying network coding in the mesh network. This development has been named CATWOMAN and is a patch to B.A.T.M.A.N. The patch is used in this project, in order to compare the power consumption in a mesh network where network coding is both enable and disabled.

In the report the mesh network will consist of two end nodes Alice and Bob and a middle nod called the Relay. The mesh network in the report will be setup differently compared to traditional mesh networks in that way that the end nodes in the network will be banned from communication directly to each other. The end nodes Alice, Bob are therefore only allowed to communicate via the Relay nodes in the network. This means that the Relay nodes only need to be capable of encoding and route network packets.

The project will examine the power consumption of each node individually in the wireless mesh network. To do the measurements on each node a MeasuringTool is developed and produced in an amount that each network node is fitted with a MeasuringTool.

The special thing in this report is that all nodes in the wireless mesh network will have their power consumption measured at the same time. Creating data to be analyzed for the whole network and not just a single node.

The future aspects for the MeasuringTool is that it should be combined with the wireless mesh network nodes and help the network to choose the cheapest route in the mesh network

## 2.1 Project contribution

The project will develop a theoretical of the energy consumption for a simple Alice and Bob topology in a wireless mesh network. Furthermore a MeasuringTool will be developed to measure the in real life energy consumption.

## 2.2 Requirements specification

This master thesis takes its grounds in a simplified setup of a wireless mesh network consisting of an Alice and Bob topology.

It is shown in [1] that applying network coding to an Alice & Bob scenario the throughput on the network increases and that congestion does not happen before all nodes reach an equal time division of  $1/3$  of the channel time.

The idea with this report is to examine what gain, if any, is reached in a mesh network, from when congestion occurs in a network without network coding till congestion occurs in a network with network coding, when looking at the power consumption.

The goal of this project is hence to find answer to the following questions.

- Design a MeasuringTool that functions with the OM1P router as cheap as possible.
- The MeasuringTool should be precise enough to measure the difference with and without network coding.
- Developing a theoretical model of the throughput and power consumption for each nodes.
- Compare the theoretical modal with the actually measured
- Determine the possible "best performances" throughput vs. power consumption.

## 2.3 Delimitations

The theoretical model defined in this report, will be limited to cover the Alice and Bob network topology. The same limitation will also be for the MeasuringTool that too will be used in the Alice and Bob network topology. The limitations are made due to the limited amount of time available. The Alice and Bob topology should even though show the potential possibilities with the MeasuringTool in larger networks.

---

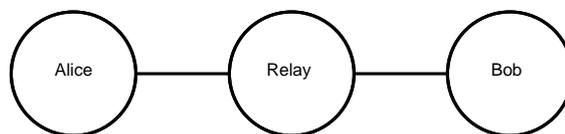
## Theoretical activity model

---

This chapter presents the wireless mesh network configuration and routing configuration, that is used in this Report. After the presentation, a theoretical model of the communication in the wireless mesh network is derived. The model will cover communication in the network for using network coding and with out network coding. The model will be used to calculate the energy consumption in the network and will later in the report be used to compare the calculated and the actually energy consumption measured in the wireless mesh network.

### 3.1 The wireless mesh network configuration

The wireless mesh network configuration used in this report will for simplicity consist of three nodes. Two end nodes named Alice and Bob and a center node called Relay, as shown in Figure 3.1.



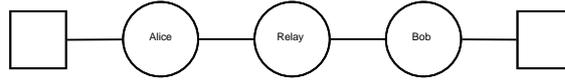
**Figure 3.1:** Illustration of Alice and Bob topology.

This is called an Alice and Bob topology where Alice and Bob are commonly used as placeholder names for archetypal characters. The Alice and Bob topology is similar to the X-topology, as described in section 1.4 on page 4. But the average with the Alice and Bob topology is that it consist of fewer nodes, which make it easier as base for developing a theoretical model that covers the network.

In this report, the network configuration is setup so it is only then end nodes, Alice and Bob who are able to create new data to be transmitted. The two end nodes are furthermore only able to reach the Relay that forwards the packet to the right destination.

### 3.2 Alice and Bob routing scheme with and without NC

To derive the routing scheme the following will present the communication route in the Alice and Bob topology. In the first scheme, the wireless mesh network is not implemented with network coding. In real live the each end node would be connected to e.g. a computer, as shown in Figure 3.2.



**Figure 3.2:** Illustration of Alice and Bob topology, where the two end nodes are connected by wire to a e.g. computer.

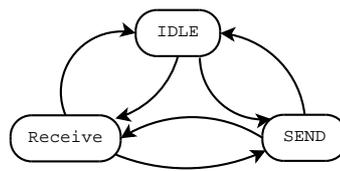
The two computers would then simultaneously act as both server and client, where the client creates data that is sent to the server. And to make the communication symmetric, both clients are requested to transmit the same amount of data at the same time with static link quality and no rapid change in error probabilities.

Now to look at how the end node, Alice, routes the packet it receives from the connected computer to the server computer connected to the end node Bob. After Alice receives the packet it looks in the routing table to find a route to the packets destination, here Bob. Alice finds an entry, saying that the source is reachable via the Relay. The packet is then transmitted to the Relay. The Relay then looks for the destination address in its routing table, finds the address of Bob and forwards the packet to Bob that forwards the packet to the computer connected to Bob. This is done the similar when the source and destination are swapped around.

When network coding is enabled in the wireless mesh network, the Relay now introduces a hold period for each packet. The hold period is defined in the network setup. Now every time the Relay receives a packet, the packet is being hold in a queue until the Relay receives a packet from a different source, making it possible to combine the two received packets to one with network coding. If this does not happen before the hold time is exceeded the packet is picked of the queue and transmitted by the Relay without being encoded with a packet.

### 3.3 Node activity model without network coding

The purpose for this section is to derive the activities scheme of each node in this given network scenario. For this each node will be distinguished in three different stages, namely Send, Receive and Idle, see figure 3.3. The reason for not having a Computation stage is that all nodes in the mesh network are in promiscuous mode forcing them to inspect all packets receive. This makes the computation time always dependent of either the send stage or the receive stage.



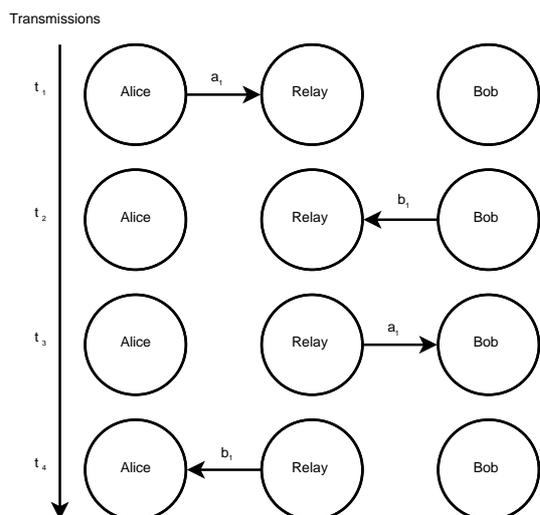
**Figure 3.3:** The nodes activity states.

The parameter  $\alpha$  describes the proportion of time in which a node stays in a given activity, whereas  $\alpha_s$  specifically represents the proportion of any time period a node is actively sending. Similarly,  $\alpha_r$  and  $\alpha_i$  describes the proportion for receiving or being idle. The node will always be in one of the three stages giving, which is expressed in equation 3.1

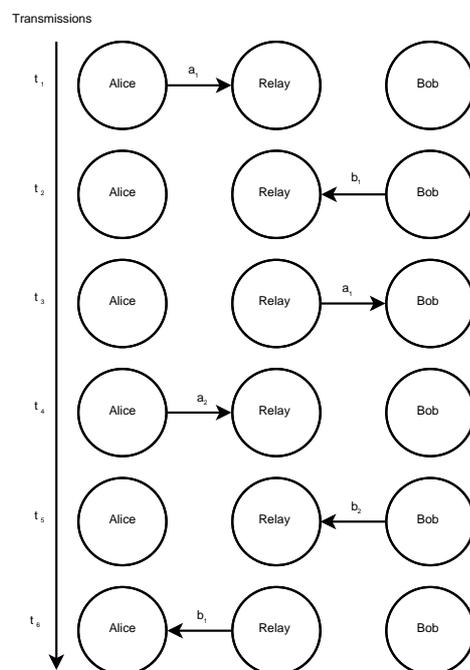
$$\alpha_{total}(l) = \alpha_s(l) + \alpha_r(l) + \alpha_i(l) = 1 \quad (3.1)$$

When the end nodes, Alice and Bob communicates symmetric, they follow the same transmission sequence given in Figure 3.4. Here each packet send from the end nodes is received by

the Relay that forwards the packet to the other end node. As shown in the figure it takes four time slots to exchange one packet from Alice and one packet from Bob. Figure 3.5 shows the symmetric communication between Alice and Bob when the network has reached congestion and MAC fairness have divided the transmission slots evenly between the three nodes. It should be noticed that when this state occurs the Relay, is not able to forward all the packets it receive so a portion of the packets will timeout before they reach their destination.



**Figure 3.4:** Communication scheme for a wireless mesh network consisting of the Alice and Bob topology.

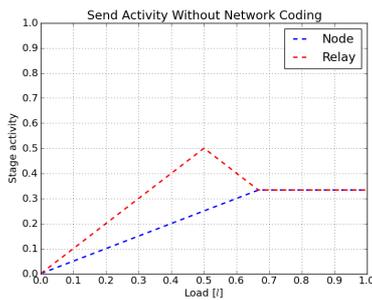


**Figure 3.5:** Communication scheme for a wireless mesh network consisting of the Alice and Bob topology, when the network is overloaded and packets are dropped.

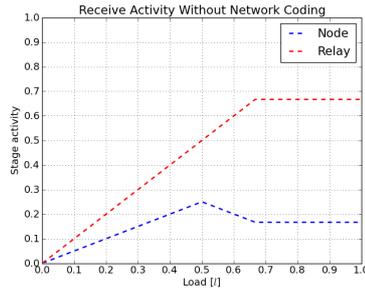
The symmetric transmission sequence makes the two end nodes follow the same pattern and the Relay its own pattern. The activity scheme is derived for an end node and a Relay node. To look at the amount of time each node is in each stage, the traffic load,  $l$ , is examined. This is done by going from no load, crossing the conjunction point and settle with the maximal possible load.

The load parameter  $l$ , expresses the unitless load factor that ranges from 0 to 1. The state activity,  $\alpha_s(l)$ ,  $\alpha_r(l)$  and  $\alpha_i(l)$  is calculated for the range of  $l$  and given in the figures 3.6, 3.7 and 3.8. The figures show the state activities for both the end nodes and the Relay node.

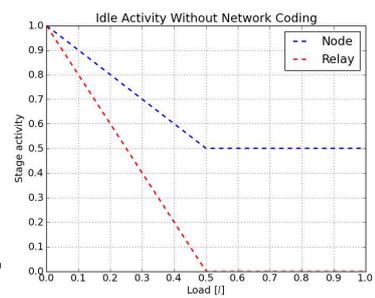
Figure 3.6 express the sending activity of the nodes Alice or Bob increasing linearly with the offered load. The sending activity for the Relay increases twice as fast until the load reaches  $1/2$ . At this load  $l$ , Alice as well as Bob are contributing with a load  $l$  of  $1/2$  resulting in an additional load contribution for the Relay of  $1/2$ . Therefore, for the approach without network coding, the network capacity is fully used by load  $l$  reaching summarized is 1. If the load  $l$  continues to increase, Alice and Bob will keep increasing their sending activity at the expense of the sending activity of the Relay. The MAC layer reacts on this congestion and introduces the MAC fairness. The result is that all three nodes are given the same sending activity from when the given load  $l$  reaches  $2/3$ . Even if load  $l$  is increase further, the activity of each nodes load  $l$  is limited to  $1/3$ .



**Figure 3.6:** The theoretical send activity for one end Node and the Relay in the AB topology. With the load going from 0 to 1.



**Figure 3.7:** The theoretical receiving activity for one end Node and the Relay in the AB topology. With the load going from 0 to 1.



**Figure 3.8:** The theoretical idle activity for one end Node and the Relay in the AB topology. With the load going from 0 to 1.

Figure 3.7 express the receiving activity by the end nodes Alice or Bob, and the Relay. The receiving activity continues to rise linearly for all nodes until the load reaches  $1/2$ . At this point the Relay node have reached its maximum send activity, see figure 3.6. From this load  $l$  the MAC fairness starts to divide the channel into 3 equal transmissions sizes, which is why the end nodes Alice and Bob now drops the time being in the receiving stage. In the continues increase of the load  $l$  (from  $1/2$  till  $2/3$ ) the end nodes reduce the amount of receiving activity, since they are allowed more send activity. When load  $l$  becomes larger than  $2/3$  the receiving activity flattens out and settles with the Relay being twice as high as the end nodes Alice and Bob.

Figure 3.8 express the amount of idle time each node has available. Until load  $l$  reaches  $1/2$  the loads continues to drop linearly for all the nodes. When load  $l$  becomes larger than  $1/2$ , the idle time reaches zero for the Relay node, whereas the two end nodes are idle for half the time. This figure also shows that the Relay node is the bottleneck in this network configuration.

### 3.4 Node activity model with network coding

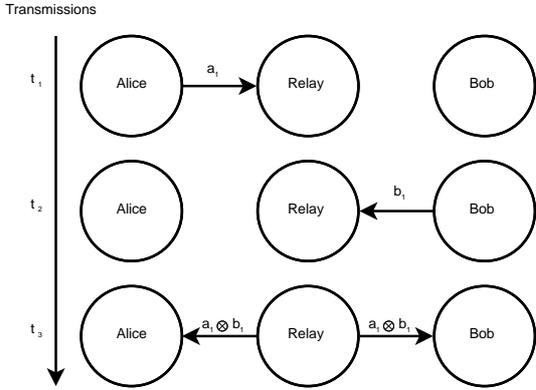
This section derives the activity scheme of each node in this given network scenario, where network coding is enabled. As in the previous section each node will be organized in the three stages Send, Receive and Idle. Even though network coding introduces more computation in the Relay node the Computation stage will still be ignored. This is justified by the type of network coding done in CATWOMAN which is a bit wise XOR operation. The XOR operation is cheap in computation.

The parameter  $\alpha_r$ , described in the previous section will also be used here to devoted the activities similar with  $\alpha_s$ ,  $\alpha_r$  and  $\alpha_i$ . When applying network coding the nodes will follow the same scheme and will always be in one of the three stages giving, as expressed in equation 3.1 on page 10

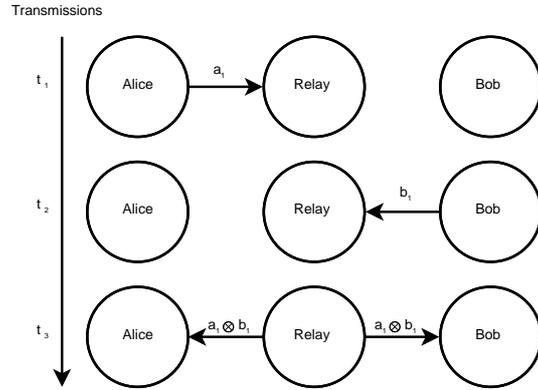
When the end node, Alice and Bob communicates symmetric, they follow the transmission sequence given in Figure 3.9. Here each packet send from the end nodes is received by the Relay. The Relay XOR the received packet with a received packet from another node. This new packet is now broadcasted to both end nodes Alice and Bob. As shown in figure 3.9, it now takes one less transmission slot to exchange one packet from Alice and one packet from Bob, and now the same amount of data is exchanged in three transmission slots.

Figure 3.10 shows the same symmetric communicates between Alice and Bob when the network has reached congestion and MAC fairness have divided the transmission slots evenly between the three nodes. It is noticed that the transmission sequence is the same as before congestion. This,

because the communication was already divided evenly between all three nodes. As noted the two figures expect that the coding gain is 1.33, see section 1.4 on page 4 for details.

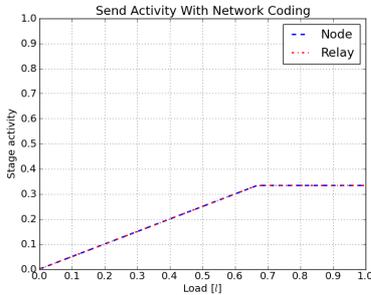


**Figure 3.9:** Communication scheme for a wireless mesh network with CATWOMAN network coding consisting of the Alice and Bob topology.

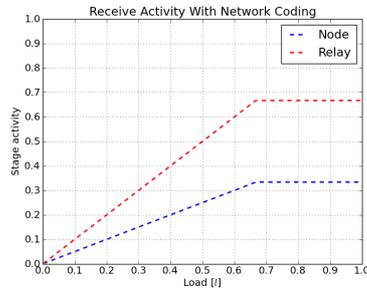


**Figure 3.10:** Communication scheme for a wireless mesh network with CATWOMAN network coding consisting of the Alice and Bob topology, with MAC fairness enabled.

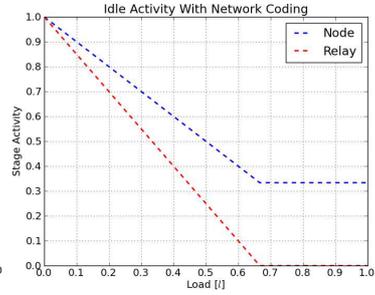
To look at the different activities of the nodes, they will again be divided in showing the activity for the end node, e.g. Alice, since the transmission is still symmetric for the two end nodes, and showing the activity for the Relay. The load  $l$  ranges from 0 to 1, and the state activity,  $\alpha_s(l)$ ,  $\alpha_r(l)$  and  $\alpha_i(l)$  is calculated for the range of  $l$  given the figures 3.11, 3.12 and 3.8. The figures display the state activities for both the end nodes and the Relay node.



**Figure 3.11:** The theoretical send activity for one end Node and the Relay in the AB topology with network coding. With the load going from 0 to 1.



**Figure 3.12:** The theoretical receive activity for one end Node and the Relay in the AB topology with network coding. With the load going from 0 to 1.



**Figure 3.13:** The theoretical idle activity for one end Node and the Relay in the AB topology with network coding. With the load going from 0 to 1.

Figure 3.11 express the sending activity of all the three nodes. Because the coding gain is 1.33, every two packets received from the end nodes Alice and Bob, results in a single packet for the Relay to send. The sending activity of Alice, Bob and the Relay is therefore equal. The sending activity increases linear until the load  $l$  reaches  $2/3$ . At the load  $l = 2/3$  the network reaches the maximum channel capacity. At this point the sending activity is stabilized at this level. The sending activity is now divided equally between the three nodes with  $1/3$  send activity.

Figure 3.12 express the receiving by the nodes Alice or Bob and the Relay. The receiving activity continues to rise linear for all nodes until the load  $l$  reaches  $2/3$ . When the load  $l$  becomes larger than  $2/3$  the receiving activity stops rising and flattens out with the receiving activity for the Relay being twice as high as for the end nodes Alice and Bob.

Figure 3.13 express the amount of idle time each node has available. Until load  $l$  reaches  $2/3$  the loads continues to drop linearly for all nodes. From this point and onwards, the idle time reaches zero for the Relay node, whereas the two end nodes will be idle for  $1/3$  of the time. This figure also shows that the Relay node is not the bottleneck, since the network capacity is divided equally between the three nodes in the Alice and Bob network topology.

		Phase	I	II	III
		Load [ $l$ ]	0-0.5	0.5-0.6	0.6-1.0
Send [ $\alpha_s$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$1-l$	$\frac{1}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
Receive [ $\alpha_r$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1-l}{2}$	$\frac{1}{6}$
		R	$l$	$l$	$\frac{2}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$l$	$\frac{2}{3}$

**Table 3.1:** Sending and receiving Activity level for all three nodes in different the phases for both with and without network coding.

Table 3.1 gives a detailed description of the activity level just derived. For the network with and without network coding the activity states are differentiate in three states. The load  $l$  is differentiate between three situations throughout the load span. The first phase describes the load between 0 and  $1/2$ , the second phase contains the load  $l$  between  $1/2$  and  $2/3$  and the third phase describe the load  $l$  from  $2/3$  until it reaches 1. These phases reflect the characteristic load points in [10]

### 3.5 Calculation the power consumption

After deriving the different activity levels it is now possible to calculate the power consumption  $P_{total}$  at any given load with equation 3.2

$$P_{total}(l) = P_s * \alpha_s(l) + P_r * \alpha_r(l) + P_i * \alpha_i(l) \quad (3.2)$$

where:

$P_s$  is the nodes power consumption when it is transmitting.

$P_r$  is the nodes power consumption when it is receiving.

$P_i$  is the nodes power consumption when it is idle.

The power levels are determined with the MeasuringTool in section 5.1 on page 23.

---

## System description

---

### 4.1 MeasuringTool analysis

This section describes the choice of components for the multifunction DAQ<sup>1</sup> MeasuringTool. The MeasuringTool must be able to measure the energy consumption in the wireless mesh network and meet the requirements given in 2.2 on page 8. Some of the most important requirements are that each node in the wireless mesh network are to be measured individually; this will be done by equipping each node with a MeasuringTool. The components chosen for building the MeasuringTool should therefore be as cheap as possible, due to the amount of MeasuringTools needed. In the following the development of the MeasuringTool is divided into the following three sections:

- Current measuring
- Voltage measuring
- Data collection (DAQ)

The Current and Voltage measuring is done at hardware level where as the data collection by software.

The MeasuringTool is developed to be compatible with the OM1P router from Open-Mesh [11]. To start with the hardware is examined. This router is used in the developed test platform for the development of CATWOMAN [1] and will be used in the tests described in this report.

The OM1P router is based on an Atheros AR2315A Wireless System-on-a-Chip, with a 200 MHz MIPS processor and 32 MB of SDRAM. It can be powered either by POE<sup>2</sup> or by a 12 volt power supply. The last option is used to supply power to the OM1P router. The technical specifications for the OM1P router does not include the amount of power it uses. So an initial measurement has been done. This measurement shows that the peak consumption is approximately 400 mA when connected to a 12 volt power supply. This is set as a requirement that the MeasuringTool should be able to measure.

In the developing of the MeasuringTool, two other DAQ tools have been examined. This is to have a reference measuring when validating the precision of the MeasuringTool. The first DAQ is the National Instruments USB-6008 which it provides basic data acquisition functionality for applications such as simple data logging. The NI USB-6008 is equipped with a 12 bit resolution ADC and 8 analog inputs with a 10 KHz sample rate. In the technical description of the NI USB-6008 is written that it requires the Texas Instrument INA 139 for measuring current. This IC will be described later in this chapter, as it is used in the MeasuringTool.

---

<sup>1</sup>Data Acquisition

<sup>2</sup>Power Over Ethernet

The second DAQ tool is the Agilent 66319D. This tool is a DC power supply developed for the mobile device and used in R&D. It is developed for testing wireless and battery powered devices and has an advanced measurement system to measure current exactly. The Agilent 66319D will be used in the validation of the MeasuringTool.

## 4.2 Current measuring

Current measuring is typically done with a shunt resistor placed in series with the current flow in a device. The resistor is specifically chosen to be high-precision and low-impedance to minimize interfere greatly with the device being measured. Since the value of the resistor is known, the current can be calculated by measuring the voltage drop across the shunt resistor using Ohm's law, see equationeq:ohm.

$$U = R * I \tag{4.1}$$

where:

$U$  is voltage

$R$  is resisters

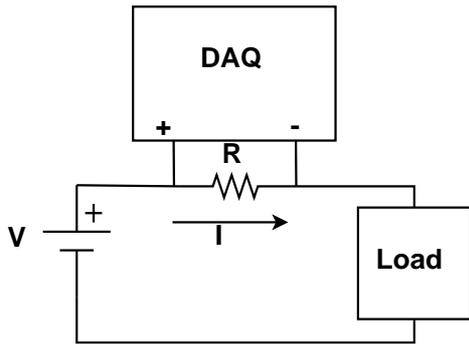
$I$  is current

Another method to measure the current is by using a Hall Effect current transducer. The transducer generates a magnetic field in the presence of current flow. The potential difference in the magnetic field corresponds directly to the magnitude of the current flow and by measuring and analyzing the potential the current can be measured in a manner similar to the other measurement methods. The Hall Effect current transducer is not as invasive as the shunt resistor method, but the price is much expensive.

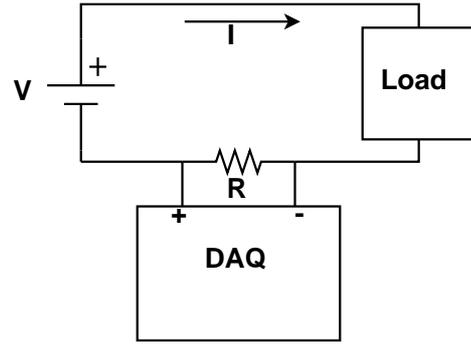
When using a shunt resistor in series with the device, there are two different configurations that can be used when measuring current. The first, referred to as high-side current measurement, involves placement of the shunt resistor in series with the supply path of the load circuit as shown in Figure 4.1. The primary advantage of using this method for current measurements is that the current draw is isolated for a specific component of the overall system with minimal losses. This means that there is not a risk of current leakage into other components of the system. The main disadvantage of this method is that in devices powered by a high voltage and will be present on both sides of the shunt resistor. This high voltage could potentially damage the device being used to measure the differential voltage across the shunt.

The second configuration is low-side current measurement to measure the current. In this configuration the shunt resistor is placed in series with the return path of the load as shown in Figure 4.2. While low-side current measurement does remove the possibility of affecting high voltage levels, this method has some disadvantages. Only current that is directly returned to the supply is measured, creates errors in measurement for loads that leak current to other grounds such as the network cable connected to the node or to the cassis.

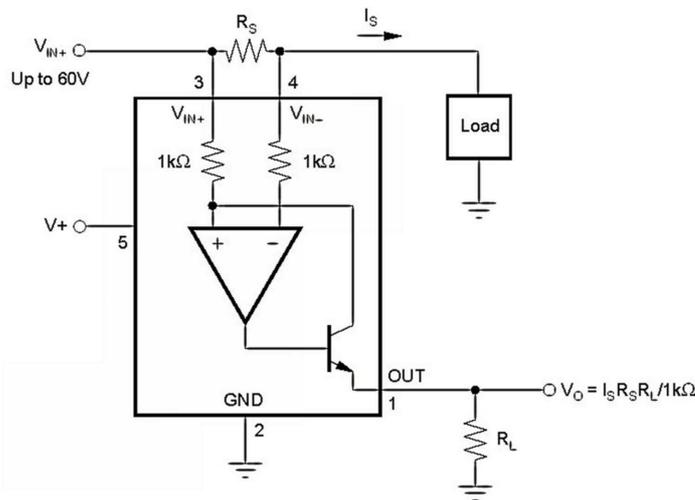
To secure the components against high voltage levels Texas Instruments has developed the TI 139 IC, see Figure 4.3. The TI INA139 is a high-side current shunt monitor. The voltage over the shunt resistor can range from 2.7V to 40V. The device converts a differential input voltage, the voltage over the shunt resistor, to a current output. This current is converted back to a voltage with an external load resistor that sets any gain from 1 to over 100. This makes the INA 139 ideal for the MeasuringTool, because it is possible to set the gain between the input/output to fit inside the input range of the ADC.



**Figure 4.1:** High-Side current measurement. The shunt resistor in series with the supply path of the load circuit.



**Figure 4.2:** Low-side current measurement. The shunt resistor in series with the return path of the load circuit.



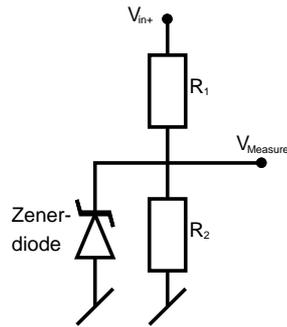
**Figure 4.3:** Texas Instruments INA139 high-side measurement current shunt monitor.

### 4.3 Voltage measuring

Voltage measuring is done by dividing the supply voltage with a resistor circuitry to fit the input range of the ADC. It should be noted that the voltage division made by the resistor circuitry does not protect against high voltage, this can be done by adding a zener diode with the same size as the desired voltage. The supply voltage measuring is done to ensure that a high current drain by a device, that can cause a voltage drop due to the power supply, cannot supply the power fast enough, and will not cause an incorrect energy measurement by the MeasuringTool.

### 4.4 Data collector

The data collection is done by having an ADC converting the analog voltage to a digital value. An MCU can then be used to transfer data to a computer for processing. The simplest way to get assemble these components is by using a development board. The requirement specification demands the tool to be cheap to build. This point in the direction of the low-cost Arduino and



**Figure 4.4:** Voltage division made by the resistor circuitry.

Digilent development boards that both are equipped with all the required components to develop the MeasuringTool. The Digilent Uno32 The Digilent chipKIT Uno32 that is equipped with the PIC32MX320F128 microcontroller has been chosen based on the technical specifications.

The items below summarizes the requirements to the MeasuringTool.

- Using INA139 for current measurement
- Using resistor circuitry for voltage measurement
- Using Digilent chipKIT Uno32 for data collecting
- Measure Voltages between 10 - 14 Volts.
- Measure Current between 100 mA - 400 mA.

## 4.5 Integration of the MeasuringTool

The MeasuringTool is designed on the basis of the TI INA139 current shunt monitor [12] and the low-cost chipKIT UNO32 development board from Digilent with a 10 bit A/D converter. The chipKit Uno32 board will function as the data acquisition board. The two above mentioned measuring components (current and supply voltage) from the monitoring circuit are connected to two separate ADC input-ports on the chipKIT UNO32 board. These inputs are sampled continually by the chipKIT UNO32 board with an 87 KHz sample rate. The PCB designed for the MeasuringTool is found in appendix B on page 33.

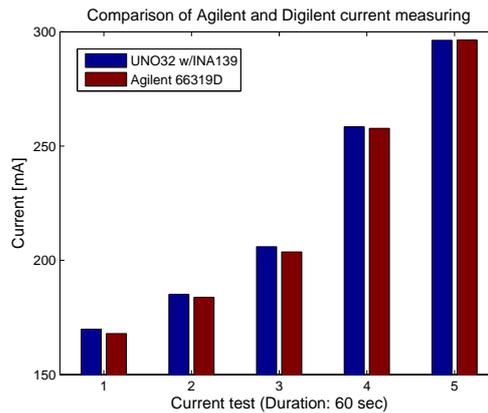
## 4.6 Software design

The ADC in the PIC MCU mounted on the chipKIT UNO32 is as mentioned above able to sample the two inputs with an 87 KHz sample rate. To reduce the large amount of measurement data, the data is averaged for every 10 K samples to a single value. The data is then stored in the memory of the board while the measuring is going on and only then read out of the UNO32 board when this is stopped. This is done to protect against interference from outputting the measured values while sampling the inputs. The software developed for the chipKIT UNO32 is created with a serial communication line, which is programmed to allows three different input commands.

- Starting sampling
- Stop sampling output data and
- Stop sampling flush data



by having different load connected to one of these two instruments at a time. Each load will be with known resistors so the power consumption will be constant during the measuring. The second step is where the MeasuringTool will be connected to a wireless mesh network node. The MeasuringTools input supply will at the same time connected to the Agilent 66319D that will function as the power supply in this setup. The Agilent 66319D will at the same time measure the power consumption of the node. In the second step the power consumption will vary. It will be high every time the network node transmits and low when the network node is in receiving mode listening for network traffic.



**Figure 4.6:** Comparison of the measuring by the MeasuringTool and the Agilent 66319D power supply. In this test setup were both tools connected to five different uniform power loads.

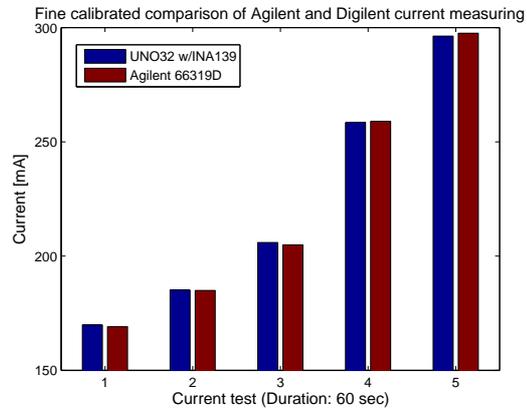
In the first step each load is measured for a duration of 60 seconds. The result from five different loads connected to the MeasuringTool and the Agilent 66319D power supply is shown in Figure 4.6. The test results indicate that the MeasuringTool and the Agilent 66319D have a constant offset. The MeasuringTool measured on the average 1.23 mA higher than the Agilent 66319D (with a variance of 0.87 mA).

As illustrated in Figure 4.6 the differences between the MeasuringTool and the Agilent 66319D power supply is small and result in marginal difference. This can be calibrated so the measurement done by the MeasuringTool will reflect the accuracy of the Agilent 66319D when a constant load is connected. Figure 4.7 shows the measurements that are now calibrated.

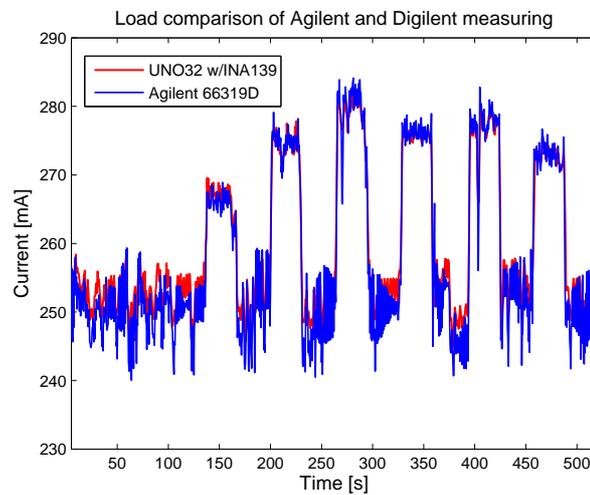
The second step in calibrating the MeasuringTool is now done with varying power consumption. The CATWOMAN test bench is setup for this test. The test bench is setup to transmit random data in five transmission steps, going from 500 kbps to 3000 kbps. The measurements received from the MeasuringTool and Agilent 66319D power supply is shown in Figure 4.8. The data received from the MeasuringTool, is in this figure corrected with the average difference found in the first step of the test.

It is notable that the measurements done by the MeasuringTool is not completely linear when it is compared to the Agilent 66319D power supply. In the comparison done in figure 4.8 the mean difference is 14.64 mA. The difference is due to the nonlinear error.

The main interest with the MeasuringTool is to be able to compare the energy consumption of different wireless protocols. With this in mind the accuracy of the MeasuringTool is sufficient to do that.



**Figure 4.7:** The measuring by the MeasuringTool and the Agilent 66319D power supply from Figure 4.6 is calibrated in this figure.



**Figure 4.8:** Comparison of the measuring by the MeasuringTool and the Agilent 66319D power supply. In this test setup were both tools connected to a node transmitting in five steps, going from 500 kbps to 3000 kbps.



---

## Testing the devices

---

To evaluate the power usage of the networks nodes, different tests will be carried out in a test environment of a wireless mesh network. The tests conducted are for UDP as described in this chapter. The description includes the test bed and the tools used to perform the tests. To begin with preliminary power measurements have been performed to determine the power consumption of the different activity stages described earlier. At the end of this section the theoretical power estimation is compare with the real power consumption.

### 5.1 Preliminary power measurements

The MeasuringTool was used to do a preliminary power measurements in order to estimate the values  $P_s$ ,  $P_r$  and  $P_i$  for Equation 3.2 on page 14. The measurement was done by connecting the MeasuringTool to a single node and then forcing the node into the three activity stages Send, Receive and Idle. The mean and the measured values from these tests, are given in table 5.1.

$P_s$	$P_r$	$P_i$
3.48 W	3.24 W	2.94 W

**Table 5.1:** A single nodes power consumption in all activity stages.

### 5.2 Experiment setup

The test platform consists of three OM1P routers, which are used to form a Alice and Bob topology. Each node is installed with B.A.T.M.A.N adv. routing protocol and patched with CATWOMAN. For information regarding the installation of the B.A.T.M.A.N adv. routing protocol, see [6].

In the experiment setup the nodes Alice and Bob are both placed apart from each other and the Relay node in between, see figure 5.1 on the following page. The two end nodes, Alice and Bob, are both configured so they are not able to communicate directly, but must route the traffic through the Relay node. In all the tests the nodes are configured to use an 11 Mbit/s fixed rate for data transmission and the rate adaptation have been disabled. For controlling the nodes and data logging of the communication, all the nodes are connected to individually laptops, as shown in figure 5.2 on the next page. The execution of the tests are handled from the test server, which also is connected to the Relay node.

A report is created from every test. The test report contains the measured data from the test run. Like e.g. the node CPU usage, amount of data packets sent, received and lost, power usage

etc.

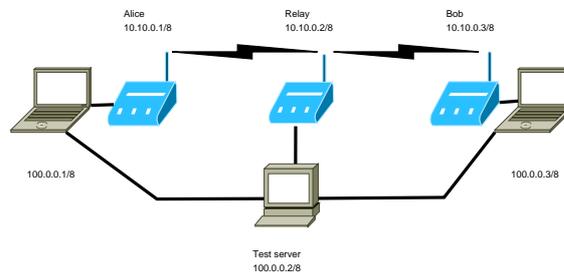
The data packets transmitted between the two end nodes Alice and Bob. The network is managed and created by the tool Iperf.[13] In each test the data is generated to be transmitted with a specific rate. This rate is varied throughout the tests going from 100 Kbit/s to 3000 Kbit/s in steps of 100Kbit/s. The transmissions are measured for a period of 60 seconds and each test is conducted five times.

From the tests the average actual current, voltage and throughput is collected and used for calculating the power and the energy per bit ratio for each node.

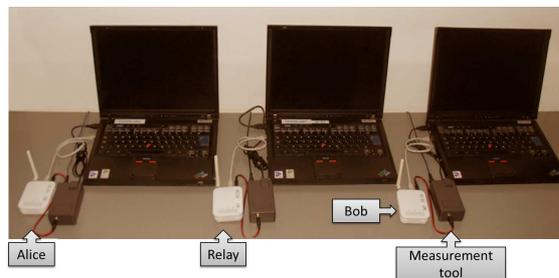
All tests were executed during nighttime to minimize interference in the wireless network.

### 5.2.1 Energy measurement results

The energy measurement result is the throughput vs. the load given in figure 5.3. The figure shows the difference throughput between the approaches with and without network coding.



**Figure 5.1:** Testbed consisting of the three nodes: Alice, Bob and Relay. The nodes are all connected to a server that controls all the test runs.

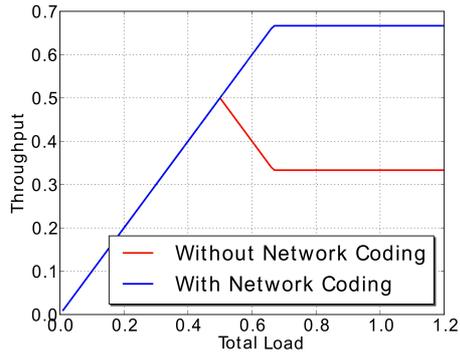


**Figure 5.2:** Picture of the real life testbed. The nodes are gathered together to show the actual setup.

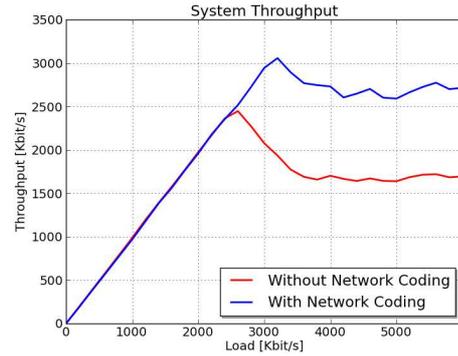
The figure 5.6 shows that the power values increase linearly for low load situations, whereas the approach without network coding yields to larger values. Later the power level for the approach without network coding is stabilizing at 9.77 Watts, and with network coding it stabilizes at 9.82 Watts.

Figure 5.6 furthermore shows that in low load scenarios network coding results in lower power values as the reduction of radio transmission is reduced.

By combining figure 5.4 and figure 5.6, the figure 5.8 can be derived.

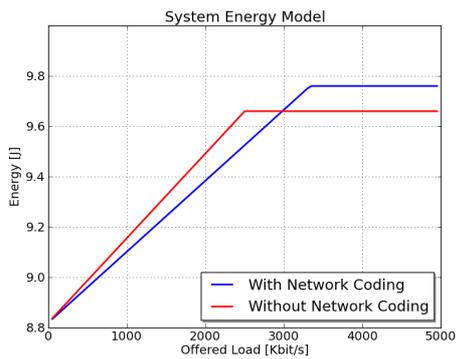


**Figure 5.3:** The theoretical throughput in the Alice Bob topology

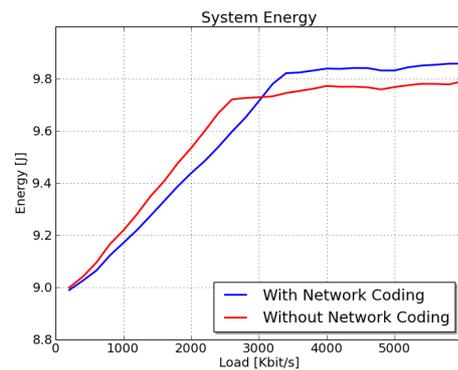


**Figure 5.4:** The measured throughput in the Alice Bob topology.

Figure 5.8 shows the energy per bit vs. offered load. It shows that for low load scenarios the two approaches yield the same energy per bit values whereas an increased offered load results in larger energy per bit values for the approach without network coding ( $6 \mu J/bit$ ) compared to the approach with network coding ( $3.8 \mu J/bit$ ).



**Figure 5.5:** The theoretical calculated power consumption in the Alice Bob topology

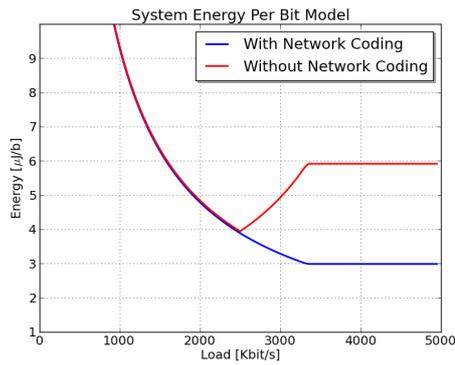


**Figure 5.6:** The measured power consumption in the Alice Bob topology.

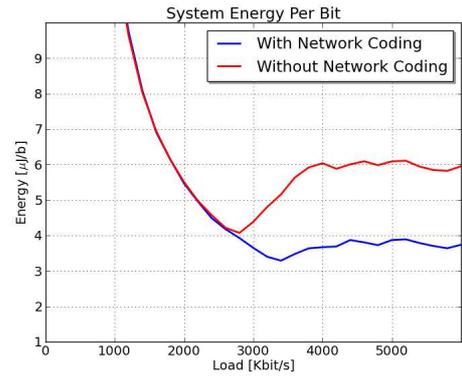
### 5.2.2 Analytical model results

By using the power values given in table 5.1 on page 23 and equation 3.2 on page 14 the power consumption vs. load plot can be derived, see figure 5.5.

For both approaches the power consumption value increases with the load in phase I and II, whereas the power stabilizes in phase III. As for without network coding the power level is slightly higher compared to with network coding in phase I and phase II. As earlier mentioned the model does not reflect the impact on the power consumption caused by the computations. It is also assumed that the nodes only receive packets which are intended for them. In the continuing the load  $l$  will be multiplied with the capacity  $C = 4.9 \text{ Mbit/s}$  in order to compare the analytical



**Figure 5.7:** The theoretical calculated energy used per transmitted bit in the Alice Bob topology



**Figure 5.8:** The measured energy used for transmitting a bit in the Alice Bob topology.

model with the real measurements.

Figure 5.3 shows the throughput for both with and without network coding, and shows the throughput vs. the load offered. The theoretical model power is given in figure 5.5.

By combining figure 5.3 and figure 5.5, the figure 5.8 can be derive. The derived figure shows the energy per bit vs. offered load.

Figure 5.8 shows the energy per bit in phase I is nearly the same fore with and without network coding and decreases when the load increases. In phase II the energy per bit figure for without network coding increases and with network coding decreases. In phase III the energy per bit stabilized to  $6 \mu J/bit$  and  $3 \mu J/bit$  for without and with network coding, respectively.

### 5.2.3 Model and Measurement Comparison

The results achieved by the model and the measurements will now be compared. Figure 5.3 and figure 5.4 represent the throughput vs. load. In general, these two plots fit very well. The expected throughput for network coding is slightly lower than expected. This is most likely due to the fact that it not completely assure that the traffic was transmitted fully symmetric. Even though the two figures show that network coding achieves a significant larger throughput than no network coding.

Comparing Figure 5.5 and figure 5.6 the power consumption vs. load, the model and the measurements also match. While the load  $l$  is lower than 3000 Kbit/s both model and measurement report a larger power value. When the load  $l$  is higher than 3000 Kbit/s the measurements show larger power values than the model.

The model expects the power values to stabilize after 3000 Kbit/s, whereas the measurements still continued a steady increase for an increasing load. The first discrepancy is explained by the fact that the model does not account for the power needed to perform network coding. Also, it does not consider overhearing for the case without network coding.

Continuing to figure 5.7 and 5.8 that show the energy per bit for the model and the measurements.

The results delivered from the model and the measured data without network coding fits very well. Both, the model and the measurement, predict that the energy per bit for high load scenarios will be around  $6 \mu J/bit$ . The match for the network coding model and measurements does not fit equally well. The model foresees  $3 \mu J/bit$  for high load scenarios. The measurements report the energy to be around  $3.8 \mu J/bit$ . This mismatch is a result of the false prediction of the throughput and the power values for the high load scenarios. Where the error in the power values is very small, the prediction error in the throughput is the dominant one.

The presented results for the model and the measurements carried out by the MeasuringTool are, even with the errors inside reasonable limits that makes the measurements easy to follow.



---

## Conclusion

---

This work presents the analytical model of both throughput and power consumption in a wireless mesh network with the Alice and Bob topology. The work also presents a tool named MeasuringTool for mounting on each network node. The analytical model covered the communication activities and power consumption for the network for both without network coding and with network coding. The analytical model has been validated by comparing it to the measurements done by the MeasuringTool. The calibration of the MeasuringTool has been done by testing it against the Agilent 66319D power supply with acceptable results.

In the report is shown how the energy measurement have been carried out The report describes how it have been possible to develop the low cost Measuring that is capable of giving quite accurate results in comparison to an Agilent 66319D.

The new hardware allows system wide energy measurement at each node in a wireless meshed network.

The analytical model for the network also shows that even a simple model is capable giving near price measurements.

In the report it is shown that applying network coding saves energy for high load scenarios. When using network coding the hold time configured in CATWOMAN is too short to keep the packet in the Relay long enough until a packet is received and used to code with. This means that not all packets were XOR'ed together when the load was low.

But even so the few packets that were coded lowered the energy consumption with almost 1.5%. It should though be noticed that 1,5% is so low that is could also be measuring noise.

For the high load scenarios the energy consumption for with network coding is slightly higher due to the computational power it requires, but as the throughput increases with network coding, the energy per bit ratio is improving so that the gain per Joule is higher.

The model derived and measurements achieved have demonstrate that network coding gives even benefits for the case of low load - in that case saving energy at the nodes

### 6.1 Further work

The analytic model should be extended to include the error prone wireless transmission resulting in more radio transmission. It should also include the possibility to model larger networks. The MeasuringTool should be implemented to transmit the measured in the wireless network to make it easier to implement.



## Appendix A

---

# MeasuringTool source code

---

This chapter contain the source code compiled on the development board in the MeasuringTool.

```
// IO setup
const int CurrentPinIn = A0; // Analog input
const int VoltagePinIn = A1; // Analog input

// Variables setup
int sensorCurrVal = 0; // ADC Curr value
int sensorVoltVal = 0; // ADC Volt value
//unsigned int testData[1415];
unsigned int currData[1415]; // array of current measures PIN A0
unsigned int voltData[1415]; // array of voltage measures PIN A1
int humanInput = 0;
int arrayIndex = 0;
int doMeasure = 0;
unsigned long startTime = 0;
int timesToMeasure = 10000;
int measureCount = 0;
int i = 0;
int count = 0;
void setup() {
    // Set serial communication to 9600 bps:
    Serial.begin(9600);
}

void resetCounters() {
    doMeasure = 0;
    arrayIndex = 0;
    measureCount = 0;
    //Serial.println("Break");
}

void loop() {
    while (Serial.available() > 0) {
        humanInput = Serial.read();
        if (humanInput == 'a') {
            doMeasure = 1;
            startTime = millis();
        }
    }
}
```

```
    //Serial.print("Starting\n");
  }
  else if (humanInput == 'b') {
    doMeasure = 2;
    //Serial.print("Stopping\n");
  }
  else if (humanInput == 'c') {
    resetCounters();
  }
  humanInput = 0;
}

switch (doMeasure) {
  case 1:
    sensorCurrVal = sensorCurrVal + analogRead(CurrentPinIn);
    sensorVoltVal = sensorVoltVal + analogRead(VoltagePinIn);
    measureCount++;
    if (measureCount == timesToMeasure) {
      //testData[arrayIndex] = int(millis() - startTime);
      voltData[arrayIndex] = sensorVoltVal / timesToMeasure;
      currData[arrayIndex] = sensorCurrVal / timesToMeasure;
      //Serial.println(testData[arrayIndex]);          // ** Just a debug line
      arrayIndex++;
      if (arrayIndex > 1413) {
        Serial.print("buffer full\n");
        resetCounters();
      }
      measureCount = 0;
      sensorCurrVal = 0;
      sensorVoltVal = 0;
    }
    break;
  case 2:
    for (i = 0; i < arrayIndex; i++) {
      Serial.print(voltData[i]);
      Serial.print(" ");
      Serial.print(currData[i]);
      Serial.print("\n");
    }
    Serial.print("*");
  resetCounters();
  break;
}
}
```

## Appendix B

# MeasuringTool PCB layout of HW

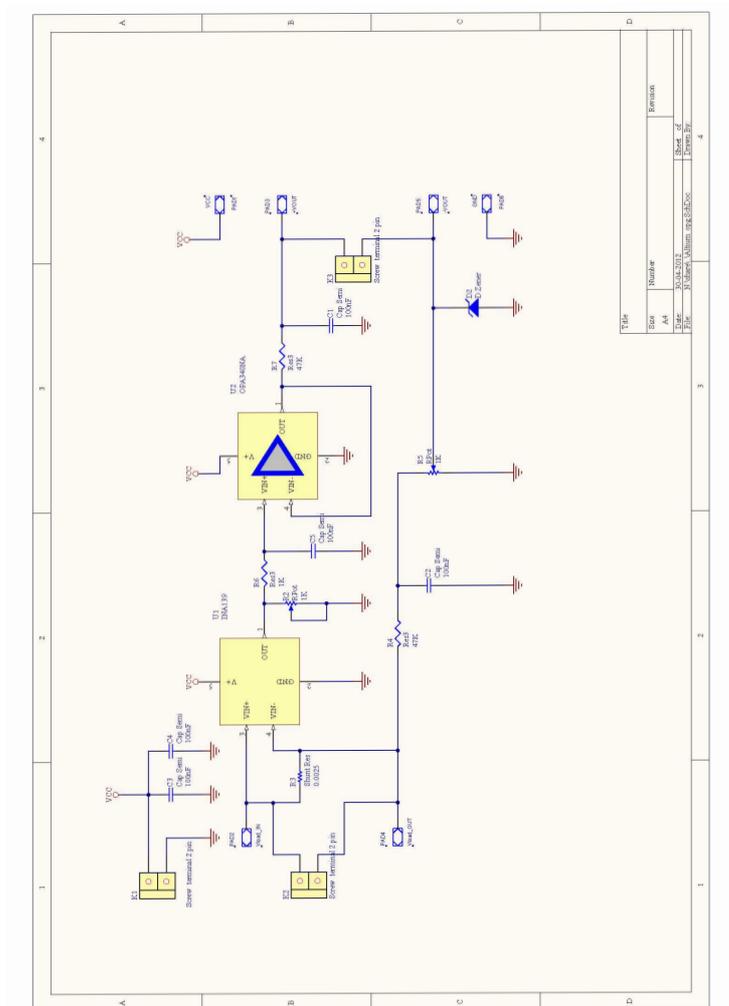


Figure B.1: MeasuringTool PCB layout of Hardware.



Appendix C

---

**Paper - Energy Consump. Model and Meas. result.  
for NC M Wireless N.**

---

Her skal artiklen sættes ind.



---

## Bibliography

---

- [1] Martin Hundebøll and Jeppe Ledet-Pedersen. *Inter-flow Network Coding for Wireless Mesh Networks*. Aalborg University, 2011.
- [2] Luis M. Correia, Dietrich Zeller, Oliver Blume, Dieter Ferling, Ylva Jading, István Gódor, Ylva Jading, István Gódor, Ylva Jading, and István Gódor. Challenges and enabling technologies for energy aware mobile radio networks. *Communications Magazine, IEEE*, 48(11):66–72, 2011.
- [3] Karina Gomez, Roberto Riggio, Tinku Rasheed, Daniele Miorandi, Imrich Chlamtac, and Fabrizio Granelli. Analysing the energy consumption behaviour of wifi networks. In *Proc. of IEEE, Online Conference on Green Communications (GreenCom)*, pp. 98-104, 2011.
- [4] Sachin Katti, Hariharan Rahul, Wenjun Hu, Diana Katabi, Muriel Médard, and Jon Crowcroft. Xors in the air: practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3):497–510, 2008.
- [5] Sofiane Moad, Morten Tranberg Hansen, Raja Jurdak, Branislav Kusy, and Nizar Bouabdallah. Load balancing metric with diversity for energy efficient routing in wireless sensor network. *ScienceDirect*, 2011.
- [6] Freifunk. B.a.t.m.a.n. advanced documentation overview. Internet.
- [7] Karina Gomez, Roberto Riggio, Daniele Miorandi, Imrich Chlamtac, and Fabrizio Granelli. Tunegreen: A distributed energy consumption monitor for wireless networks. 2011.
- [8] Wilibox. Wilibox.
- [9] Andrew S. Tanenbaum. *Computer Networks Foruth Edition*. Pearson Education International, 2003.
- [10] Fang Zhao. *Distributed Control of Coded Networks*. Massachusetts institute of technology, 2010.
- [11] Open-Mesh. Om1p 802.11g mid power mini router - single band series. Internet, November 2011. <http://www.open-mesh.com/index.php/professional/professional-mini-router-us-plugs.html>.
- [12] Texas Instruments. Ina139 - high-side measurement current shunt monitor.
- [13] Internet Performance Working Group. Iperf. Internet.

