Designing low latency, fault tolerant Sensor Networks using Complex Networks Analysis

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea Politehnica Timişoara
în domeniul Calculatoare și Tehnologia Informației
de către

ing. Alexandru IOVANOVICI

Conducător ştiinţific: Prof. dr. ing. Mircea VLĂDUȚIU
Referenţi ştiinţifici: Acad. Mircea PETRESCU – UPB Bucuresti
Prof. dr. ing. Liviu MICLEA – UTC Cluj-Napoca
Prof. dr. ing. Mircea POPA – UPT Timișoara

Ziua susţinerii tezei: 08.04.2015

BUPT

Părinților mei

This work was partially supported by the strategic grant POSDRU/159/1.5/S/137070 (2014) of the Ministry of National Education, Romania, co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013

Iovanovici, Alexandru

Designing low latency, fault tolerant Sensor Networks using Complex Networks Analysis

133 pagini, 55 figuri

Cuvinte cheie:
sensor networks, complex networks analysis, relay placement, energy optimization, fault tollerant networks, inteligent transportation systems.

Rezumat,

Cercetarile din prezenta teza se axeaza pe doua directii principale din domeniul retelelor de senzori: pe de o parte se propun o serie de algoritmi de inspiratie originala pentru plasarea nodurilor de tip releu intr-o retea de senzori in vederea minimizarii consumului energetic si pe de alta parte se analizeaza toleranta la defectare a retelei nou-obtinute prin prisma conexitatii si stabilitatii diametrului. Ca si element de noutate se remarca utilizarea unor metrici si concepte de teoria retelelor complexe precum *beetweenes* si structura de comunitate, peste care se construiesc algoritmii propusi.

Lucrarea se evidentiaza si prin caracterul ingineresc-aplicativ, continand de asemenea si un studiu de caz referitor la proiectarea unei retele se senzori dedicata monitorizarii in timp real a parametrilor de trafic rutier urban folosind metodologia si algoritmii propusi.

BUPT

Everyone takes the limits of his own vision
for the limits of the world

— Arthur Schopenhauer

Acknowledgments

This thesis is a personal endeavor which started almost four years ago as an idea.

The concepts of large sensor networks and Internet of Things just started to emerge and naturally there were a lot of grand challenges and risks with adopting the new paradigms of which most of the experts had little idea, mostly inspired or taken from the related fields such as computer networks. After numerous hours of reading and discussing various concepts and ideas I've synthesized the core concepts I had to prove by experiments I had to design. You might think that this is hard things to do, but the really hard stuff is to now what to do.

For this I had to thank a lot of people but first and foremost it is Professor Mircea Vladutiu who deserves all my gratitude and recognition, my adviser and mentor throughout all this, firstly for believing in me. When we first met I had a vague idea regarding what I would like to do, but this constant guidance and his immense knowledge in the broad field which is Computer Science helped me a lot. I should also mention his constant support during any bumps I've encountered in my research and all the well thinks advices for navigating the treacherous waters of the institutional bureaucracy. Professor, I thank you!

My colleague and friend, Lucian Prodan, deserves all my appreciation for spending countless hours on vivid discussions regarding the best way of approaching a particular problem or for suggesting an even better experiment or simulation method for testing a particular hypothesis I had. Also we dedicated himself completely for helping me and teaching on how to write a scientific paper which is capable of carrying the most valuable of the results in the most pleasant to read and understand manner and I'll remember forever the enthusiasm and joy we shared together when my first paper to an international conference was accepted (ITST2013, Tampere, Finland).

All this years — and even before — I was a member of the Advanced Computing Systems and Architectures research group and much of the results I show you now are possible as a result of numerous interactions I had with my colleagues. I'd like to express my gratitude towards

BUPT

Mihai Udrescu for bringing under attention the entire field of Complex Network Analysis, Alexandru Topirceanu for helping my managing the quirks of the Gephi software and implementing the plug-in for the algorithms I will present here, to my fellow “brothers in arms”, Ph.D. students, Andreea Bozesan and Cristian Cosariu for valuable cross-domain discussions, ideas and even *advocatus diaboli* position defenses sometimes from which stemmed novel angles on each of our individual research topics.

At last I have to remember and thank to people which were around me, for the countless minute things they’ve done, for their support in moments when I had nothing else in mind except how to improve my research. I thank here to my parents and my girlfriend for bearing with me during this endeavor.

BUPT

Contents

BUPT

BUPT

List of Figures

BUPT

BUPT

BUPT

BUPT

List of Algorithms

BUPT

1 Introduction

Almost a decade ago, Raymond Kurzweil coined the term *technological singularity* seen as the moment in time when the "non-human intelligence" is going to surpass the human one, or the time when computers and "robots" in general are going to take control over the world as we know it Kurzweil [65]. Even is the proposed time horizon is somewhere around 2045, which most of the mainstream scientists consider too early, what is worth noting is the fact that we are witnessing a more and more accelerated development of the computational power and the direct consequences are mostly at the application level, providing innovative means of crunching data in the form of information.

As with many of the technological advances we are witnessing trough the history of the humanity processing large data sets and interpreting them is having it's roots in the military and intelligence gathering sector. After the events of 9/11, many turned their opinions towards mass surveillance and data harvesting on a large scale, from classical and almost obsolete cell phone interception to more exotic side-channel approaches such as audio surveillance trough laser light reflectionLi et al. [73] and even extracting RSA private keys by audio surveying the CPU operation Genkin et al. [42]. Of course mining the traffic over data networks was not left outside the new doctrine of intelligence and the silent war known as *cyberwarfare* began.

In the realm of military battlefield surveillance one of the major technological hot topics appear to be *sensor networks*. Presented as a collection of simple devices capable of collecting data regarding the environment in near proximity and some communication capabilities they spear to be the ubiquitous element of modern sensing and automation systems. Sensing devices are called *nodes* and are usually capable of reading some elementary data such as temperature, light, sound level or presence of chemical compounds while the communication is done either wired like in underground or marine surveillance systems of wireless throughout a variety of physical media such as optical (infra-red, laser) or electromagnetically (radio-waves, microwaves). On the other side the collected data are locally processed or most of the time funneled and aggregated via an up-stream data path towards on or more aggregation nodes, called *sinks*. Depending on the size and properties of the network some nodes are in place in order to *relay* the signal. Of a smaller visibility but of equal importance in some of the sensor

BUPT

networks are the *actuator* nodes, responsible of directly acting in the surrounding environment based on the collected data and the decisions implemented in the application. Examples comprise, but are not limited to chemical plant automation, automatic weapons systems or even full body, wearable healthcare systems.

Important issues, which are going to be addressed in my thesis, arise when a sensor network scales up to hundreds or thousands nodes and/or when the nodes are mobile. If for smaller networks traditional managing techniques inspired from computer networks can be successfully applied at each of the protocol levels, the shear number of nodes imposes a performance wall which is not to be conquered with classical approaches.

In this context the *topology* of the network represents the physical and logical way the nodes are interconnected and pass the data between themselves or to the central node. Topology imposes limits regarding technological and management issues which in the end have a large impact on the performance of the network. At this moment it is work noting about the general term of *performance* when we speak about sensor networks. The term has many facets but two are of particular interest in the field of sensor networks: energy consumption and bandwidth. Based on strict energy constraints in which sensor networks are designed to operate (forests, deserts, underwater, hostile enemies) there is a really great deal of interest in the field o designing and optimizing low and extremely low power nodes and communication protocols.

On the other side of my research is the application of complex networks analysis in the field of sensor networks, more especially the optimization of the topology and routing strategies in order to reduce the delay and congestion by using concepts and methodology for the field of "new networks science". Established after the 2^{nd}World War and gaining momentum in the last decade with the addition of a new sub-field, the Social Network Analysis, CNA, deals with networks, as in graphs, but with irregular topological properties which are hard to be described deterministically in a complete and clear manner. Almost anything which can be describes as a relationship between two entities can be represented in a form of network and consequently most of the human activities are benefiting from applying CNA on their specific problems. Successful examples can be observed in the field of human sciences, sociology, biochemistry, medicine and even scientific authorship, but problems in the fields such as transportation networks, electrical engineering of even counter-terrorism are approachable with methodologies form CNA.

Besides metrics and concepts from graph theory, CNA deals with something much more hard to formalize such as node centrality, com-

BUPT

munities, modularity, notions which are easy to map on the way we, humans, see the structure of the societies. The "explosion" of online social networks has provided a good testing ground for experimenting with SNA but the same methodology when applied to sensor networks as I'm going to present in this thesis is giving as the tools for radical improvement in terms of cost-efficiency and reliability of the networks.

1.1 Research path

In this section I'm going to provide a general outline of my thesis, emphasizing on the key contributions on the subject, while allowing the reader to get an overview of the entire body of work and select the topic which presents interest. The "path" I followed can be seen in Figure 1.1.

It all started with my previous interest and involvement with the technological solutions for improving and optimizing the quality of road traffic in urban environments (cities). I already had some existing work in this field from previous years but I lack the necessary infrastructure for collecting large data sets in order to conduct the simulations and run various algorithms I had in mind. So relaying on the concept of crowd-sourcing and crowd collaboration I developed a simple mobile application for collecting near real-time data regarding the mobility (especially) driving patterns of the volunteers. The data were anonymized and used for building a "state of the traffic" map which was available online. This solution was presented at the ITST 2013 Conference in Tampere, Finland [49].

In the meantime I started reading more and more on the topic of Sensor Networks (SN) as briefly described above and after discovering some of the important issues regarding the organization of configuration of the SN I began contemplating the idea of applying some bio-inspired techniques which would be suitable to manage such a large number of nodes. The next step was to integrate theoretical aspects of the Complex Network Analysis (CNA) and especially Social Network Analysis (SNA) into the problem of Wireless Sensor Networks (WSN). This was concertized by a new algorithm I designed, laying at the crossroads between WSN and SNA, called SIDeWISe which was presented in [53] at SOFA 2014 and in [52] at ICSTCC 2014.

The intial form of the SIDeWISe algorithm was dealing with so-called "flat networks" which are a specific form of WSN, logically consisting of a single level in the hierarchy of the nodes, form source to sink. Practical aspects in designing and deploying WSN imposes often the need of a hierarchical structure with multiple levels consisting of intermediary "sinks" responsible for aggregating and processing data and even

BUPT

taking action in their near vicinity. In order to achieve such a behavior I've improved the original algorithm adding a recursive mechanism which allows processing arbitrary large networks with arbitrary deep hierarchies. This is the STiLO algorithm, presented in ICSTCC 2014 [50].

The next logical step in the path I began walking was to integrate this somehow theoretical approaches into a specific application in order to validate the assumptions and provide a test bed for the field assessment of the performance and even find some pitfalls which might arise. Consequently I turned my attention back to the field if ITS and I tried to design a specific application in the area of ITS which might benefit from implementing SIDeWISe and h-SIDeWISe. This was the problem of optimizing the flow of urban road traffic by dynamically assigning green phases on traffic lights installed in key intersections. The problem is not novel, much literature existing on this field and much attention was given to it from booth academia and the industry, but as it can be easyly seen, most of the existing solutions - which, as a matter of fact, are centralized from a logical and technological point of view - are reaching their limit quite fast. Together with my colleagues, I proposed also in [50] and refined in [30] a decentralized, locally managed *intelligent traffic light* (i-Traffic Light) which would extend it's influence in a local "community". At this level, the connection with WSN and h-SIDeWISe was obvious, so I decided to use this in the simulation and validation step of my algorithms.

A small problem arise during the design of the experiments, represented by large information packet storms which were generated from time to time by the original implementation of h-SIDeWISe. Even if the design of the algorithm required strict community delimitation, managing cross-comunity-border information was necessary form time to time (similar with the GSM hand-over) which required a lot of communication at the upper level. In order to provide and elegant solution to this issue and with taking into consideration the way we usually deal with information on an everyday basis I introduced an new improvement to the h-SIDeWISe, adding a bio-inspired taste to it. More explicitly it consisted of a bioinspired element for an uniform distribution of the main parameters caracthersing the network: average path lenght and centrality [101].

Regarding the structure and outline of this this theis, I'm going to present in more details aspects regarding the foundations of my work and the design, testing and analysis of the two major algorithms which I have devised. The next chapter is dedicated to a much more in-depth presentation of the concepts and notions regarding everything related to "The New Network Science" from terminology and definitions and up to metrics, properties and analysis which can be mad using this

BUPT

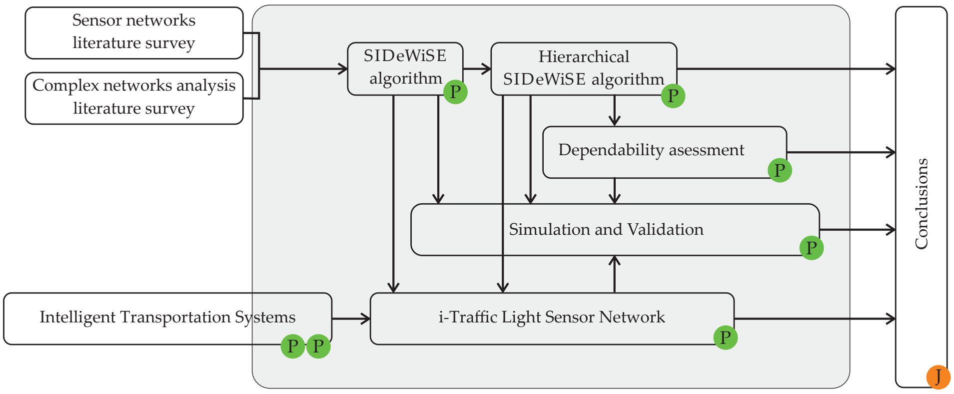


Figure 1.1: Outline of my Ph.D. research path with delivered research papers

approach. A special interest is given to the aspects related to centrality, modularity and community structure which are heavily used in my research and need clarifications. Next, a distinct chapters is devoted for presenting in great detail SIDeWISe and h-SIDeWISe algorithms. Starting with the formal definition of the problem (in terms for booth network science and sensor networks) and presenting an analysis of all the steps of my methodology I make some parallels in terms of performance an cost effectiveness with literature-classical solutions. The next chapter deals with issues regarding the dependability of sensor networks and targets the issues with networks designed using the algorithms I've proposed above while measuring two specific metrics of network fault-tollerance: diameter stability and connectivity. The impovements are emphasized in terms of coverage versus the number of lost nodes/connections. A specific chapter is putting everything togheter in an engineering fasion, being targetted with the design and implementation of a sensor neotwork for real time road traffic monitoring. A draw some conclusions regarding costs and performance in specific terms. The final chapter makes a synthesis of the main contributions presented in this thesis and in the same time provides some points of further investigation in booth theorethical/fundamental and engineering directions.

BUPT

2 Sensor Networks

2.1 Networks and computer networks

Living in our modern highly technological world exposes ourselves to a plethora of digital devices, equiped by bigger and bigger displays, higher battery autonomy and a large variety of accesories and functions, but a part that is becoming less visible even if it's the esential aspect of everything we use today is the *network*. We are considering as a given fact that we can send a message to the other side of the World of even to Mars in the press of a key, but the undelaying levels of standardization and interoperativity are the result of decades of social and technological evolution.

What started as a set of military operated machines designed to whitstand a nuclear war in the Cold War era soon expanded as the most simple and efficient way of exchanging any kind of information. From simple news reading, sending emails and up to industrial command and control systems of international financial transactions are handled by the massive system which is now known as Internet, the network of networks. Classical taxonomies which make a clear distinction between LAN and WANs are becoming more and more fluid, in the last decade the margin in the extent of the two being not so clear. The large majority of wires LANs are now based soleley on Ethernet while the WANs are almost exclusively part of the Internet [cite]. Some special applications such as real time command and control systems, which I'm going to discuss next, fall outside of this rough classification but the general idea still holds.

In the context of the general computer networks there is a special field which started initialy as part of the computer networks but which in the last two decaded becam a specific topic of interest by it's own and this is related with the domain of intelligent sensors and actuators.

2.2 Sensor networks

There are various definitions of what a sensor network is, but at the highest level it can be seen as a technological system of locaigally interlinked elements (nodes) which are capable of gatering data regarding the physical environment in which they operate (sensing), carry some

BUPT

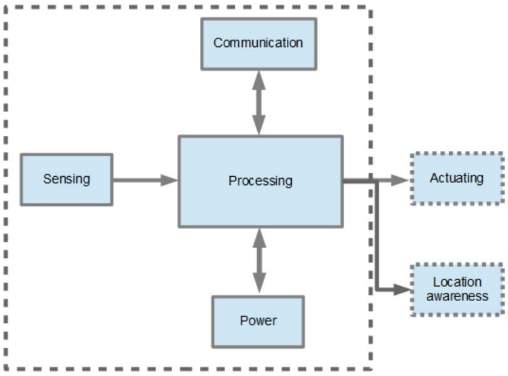


Figure 2.1: The high-level architecture of a sensor node

computation on the collected data in order to take some decissions and apply he decissions back into the environment trough the set of actuators. For the rest of this thesis, I will offer an unitary apporach for dealing with booth sensing nodes and actuating ones, refering booth of them by the term of *nodes*.

The emphasis and part of the power and andvanteges of any sensor network falls on it's shear size, seen as the number of nodes and the strong capablities of exchanging information betwoeen nodes. On the other side, of euqal importance is the hardware architecture and physical caractherisitcs of these nodes. In this case I'm speaking mostly of the resiliance of the nodes to any imaginable sources of perturbance.

From the physical point of view the inteconnection of the nodes fall into two major categories: wired (even fiber optics) and wireless (even visible line of sight).

With all these prerquisites we can say that a node consists of a sensing element (sensor), a microcomputer, one or more transceivers and a power supply (figure 2.1). Depending on the task at hand the sensor will be selected in order to booth meet the requirements in functional terms (observed phenomenon, sensitivity, resollution) but also to cope with the necessary environment factors (heat, chemical substances, dust etc.). The processing unit is selected based on booth the actual functional task of the nodes, but also based on the architecture of the netowk, the partitioning of the information processing and decission and not in the last place, based on the power requirements and availabilities. More and more networks are designed to operate in full autonomous mode, with wireless conectivity and in this case, the low power requirement is a must. I will elaborate on this later.

There is a great deal of networks and application which only require monitoring and data gathering without direct intereference with the environment.

BUPT

The organization of the logical interconections between nodes is forms what is called the *topology* of the network. From this point of view and dependant on the task ast hand correlated with the particularities of the environment, there is a clear distinction between enginnerid networks where the placement of each node is well known in advance and is considered fixed for the lifespan of the network and the *ad-hoc* networks which organize themselves after deployment being able to provide on-line reorganization to cope with partial loss of the network while still keeping functional requirements. In the same time applications which require mobile nodes fall into a distinct category, most of the time treated as a special case of ad-hoc networks. My research and the methodologies I present are dedicated towards fixed nodes, but with possibilities of providing partial online reconfiguration.

2.3 Challanges on designing sensor networks

The last years were the witness of an increased interest in the area of real time systems for sensor networks. This arises from specific applications where the functional requirements are augmented with time constraints such as: fire detection, chemical alerting or emegency shut-down of industrial plants. In the same time various closed loop control algorithms require struct timing caractheristics. In this case it is the responsability of each of the nodes to meet the deadline in a consistent manner, which translates into a similar requirement for the interconnection network. In the same time the network as a whole has to guarantee some metrics of scalability, fault tollerance and low power requirements on which I'm going to elaborate later.

The defining caractheresitcs of most sensor netoworks is their size, seen as number of nodes. In the same time the geographical span can be taken into consideration but it is of smaller importance. From this point of view there is a great deal of interest in minimizing the costs associated with the deployoment of the network as a whole which translates into minimizing the costs associated with each node and the costs of the interconections. On the long term there are costs associated with keeping the network up and running which arise from the limited power autonomy in the case of battery operated nodes and from this point of view there is a great deal of interest from optmizing the power consumption on a node level.

The power optimization is a two fold apporach: on the one side it is carried on the hardware level, by choosing low power microcontrollers and communication modules and on the other sied it is handled in the software level by implementing specific protocols for minimizing the energy consumption while still keeping the desired levels for the func-

BUPT

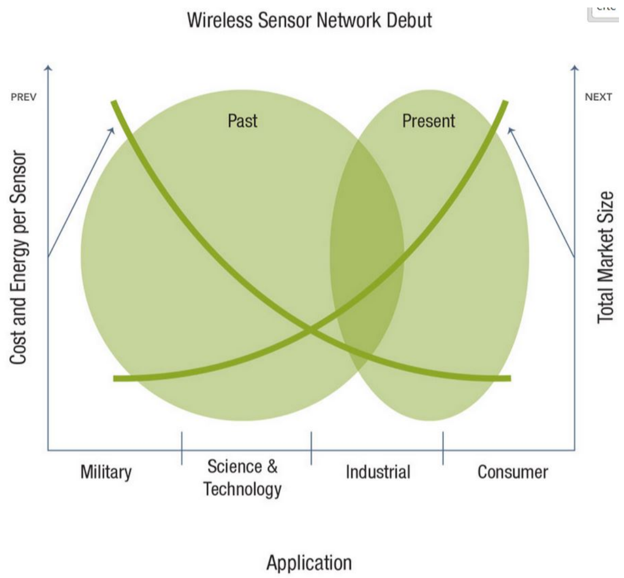


Figure 2.2: Distribution of application areas for sensor networks and thier maket share, courtoisy [27]

tional requirements. Moder microcontroller are providing a great deal of flexibility for the application designer to directly interact with functional blocks of the MCU in order to dinmaically swithch the desired units on order to tailer the power consumption.

The advances in the area of microelectroncis and low-power hardware design provide the industrial traction necessary for designing larger and larger networks with increased cummulated computational power while keeping the costs at lower and lower levels. What was in the last decade the apanage of some highly funded sectors such as military and scientifical research shifted steadily towards the industrial sector - which as allways put a great emphasis on keepeg the costs low and after 2010 there is a clear penetration of the consumer market. In figure 2.2 we witness the steep decrease of the costs strongly correlated with the clear adoption of the sensor networks in the area which until then had a higher doubt based mostly on the cost aspects. The treshold between *past* and *present* in figure 2.2 is in 2010.

2.4 Technological advances in sensors

The keyword of any sensor network is the sensor. From it's initial goal of translating a signal from the analog domain of the real worls into the digital one required for further processing it has evolved and diversified into tehns and even hundreds of subcategories, aplication

BUPT

Table 2.1: Technologies involved in designing and building sensors

Underlaying technology	Typical sensors
Analog resistive	NTC termistor
CMOS	temperature, pressure, humidity
MEMS	gyro, accelerometer, magento
LED	light, distance, surface reflection
Laser	high precission distance, particle

areas, resolutions and sensitivities in order to cover allmost any possible imaginable scenario. The so-called classical sensors were designed based on CMOS technologies and widespread examples are those for measuring temperature, humidity, capacitive proxomity or even chemical composition. Advances in area of MEMS (Micro ElectroMechanical Systems) gave birth to a new set of more specialized sensors such as gyroscopes, accelerometers, magnetometers, pressure sensors and so on. Optoelectric sensors represent another large category comprising various LED based elements such as: light sensing, optical proximity and chemical composition (particle detection).

A typical application of sensor network of building monitoring is the one tasked for optimizing the consumption of the electrical energy required for HVAC and building lighting. Classical systems are based on timers, without taking into consideration the actual presence of humans in the perimeter. Using a combinet data set from temperature, humidity and PIR presence sensors corelated with and real-time clock modern systems are capable of optimizing the duty cycle of building HVAC and indoor lighting in order to obtain a decrease between 7 and 15% of the total energy consumption [cite].

All these advances are strongly interlinked with the advances in the semiconductor and microelectronics. Modern highly integrated ans specialized sensors are a stand-alone module comprising booth the actual sensing unit and the electrical conditioning and interface in order to simplify the required interfacing but with translated into higher initial costs and sometimes higher power consumption, the designer not being able to fine-tailor the device according to his requirements.

In figure 2.3 I present in a comparative manner typical sensors for measuring the distance trough optical priciples. In subfigure (a) we can see a typical Sharp-type IR distance sensor suitable for measuring distances of up to 50cm with a tollerance of 3cm. There is a clear evidence of cheap manufacturing with exposed electronical parts and plasic lenses are hard to clean and provide numerous surface for build-up of debries witch in time would clog the sensor. This design is most of the time not suitable for long term autonomous operation but is

BUPT

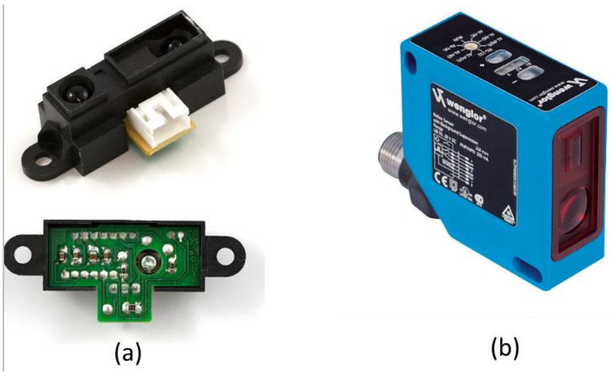


Figure 2.3: Optical distance sensor for *hobby* use and for industrial environment

consistent with a <7EUR price. In subfigure (b) we see the same type of sensor, but in this case designed for industrial environment. The tough construction with M12 threaded screw withstands the use and abuse specific for the environment while the smooth finishing of the optical surface does not pose risks of dust buildup. This a highly integrated module with it's on local processing being able to be interfaced via almost all the standard industrial interfaces such as: RS-232, RS-485 and analog output. The operating voltage if +24V DC specific with the industral domain, while the list price is 485 EUR.

It is of clear evidente the good corelation between the advances in microelectronics and system integration and the advent of larger sensor networks. With the developement of SoC type devices, the designers no longer relayed on bulky discrete components which in the end had the greatest impact on lowering the power requirements of system. Modern SoC designs are capable of integrating booth the MCU and the RF transceiver in the same chip, providing the periperials for implementing all the sensing, processing and communication required for the functional requirements of the node.

For example one of the most promissinng module at the moment is the EM35x Ember ZigBee from Silicon Labs. It includes an ARM Cortex-M3 with a IEEE 802.15.4-2003 (ZigBee) transceiver, an AES crypto module, and most interesting sub-μA sleep modes and everything is packaged in a 7×7 mm QFN chip. The reference design for this module would require just a battery an the specific sensor, with the optional requirement of an antenna for having a fully working WSN node. From the ZigBee part of the protocol it is capable of being coordinator, router or end device.

BUPT

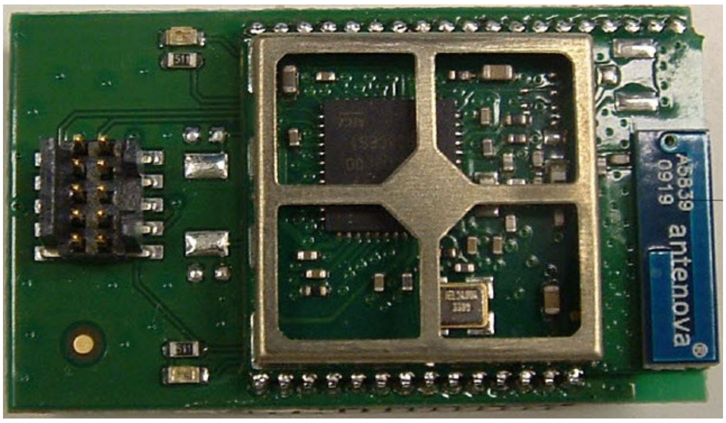


Figure 2.4: Breakout board of the EM35 ZigBee module

2.5 Power and energy aspects regarding sensor networks

Sensor networks can be defined even for wired applications, but this subfield is a small one with few specialized area when the enevironment is hostile to other alternative. Most modern application in the are of sensor networks are targeting various wireless technologies as comunication mediums. In this context there is a great demand for a reliabale and capable power source. Classic approach mandate puting a battery for each node, but there are limiting factores with this approach comming from the reduced lifetime and higher costs of operation and maintenace (figure 2.5).

We are witnessing a tremendous development in the technology and chemistry of various type of batteries. The large variety of Lithium based batteries allow a fine tailoring of power source based on the particular application, with it's characteristics and operating specificity. Two extreme cases are represented by the lithium-manganese batteries ($LiMnO_2$) suitable for application characterized by small temperature variations and short lifetime (days to weeks) while the Lithium-Thionil-Chloride ($LiSOCL_2$) are preferred when dealing with large temperature range and even decades of service lifetime) [27].

Most of the application designers and network engineers would thnik first of batteries when deciding on the power source for a wireless sensor network, but novel alternative energy sources are being developed and solutions such as energy scavenging and harvesting are good alternatives for some applications. Trough a Peltier thermoelectric element, power can be generated by the difference in temperature between two surfaces. Vibration and kinetic energy can be converted into electrical energy trough MEMS technology sensors or even piezoelectric elements

BUPT

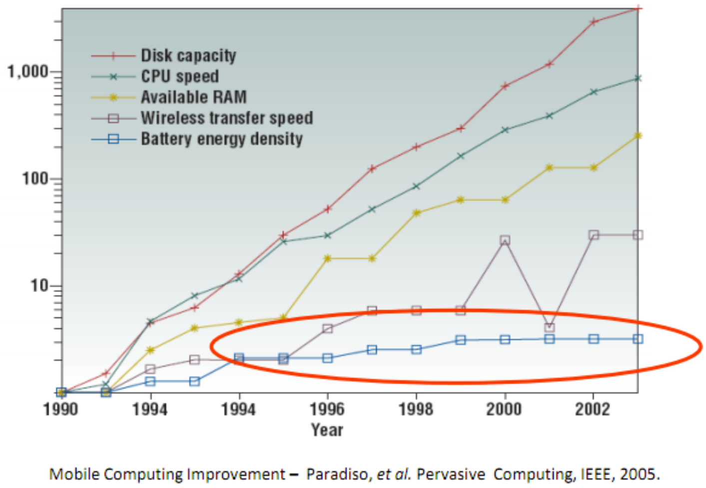


Figure 2.5: Evolution of various enabling technologies for sensor networks smphasizing on the low trend of the battery energy density. Courtesy [88].

Table 2.2: Overview of the caracthersitics of the main energy harvesting sources, Courtesy [88]

Energy source	Carctheristic	Efficiency	Typical harveste
Light	Outdoor: Sun, Indoor: secondary sources	10-24%	$1000mW/c$
Thermal	Human activity, industrial processes	$[0.1, 3]$%	$60\mu W/cm$
Vibration	Human activity: Hz, Industrial: kHz	25-50%	$4\mu W/cm$
RF	GSM: 900MHz, WiFi: 2.4GHz, ...	~50%	$0.1\mu W/cm$

(EnOcean wireless, battery-less light switches). Even conversion of the RF energy is an option when the power requirements are low (in the order of μW) such as for ubiquitous anti-theft tags. Not at last the technology for manufacturing solar photovoltaic panels is improving from $6-8$% in 1975 to around 46% with the currently state of the art multijunction cells [44].

Various approaches for mitigating the energy problem exists in the literature and most of them are allready employed in industrial applications. Of all the energy consumed by a typical sensing node, 68% is spent for communication purposes, the actual task of the node requirinng only the remaining 32% [68].

In [96] authors identify an overhead of less than 1% for routing the messages, which is consistent with the 0.8% presented in [68] and provide the bases of my furthe rinvestigations regarding optimzation of the network topology. While the impact of building more complex routes is allmost negligable there is no significat decrease in perfor-

BUPT

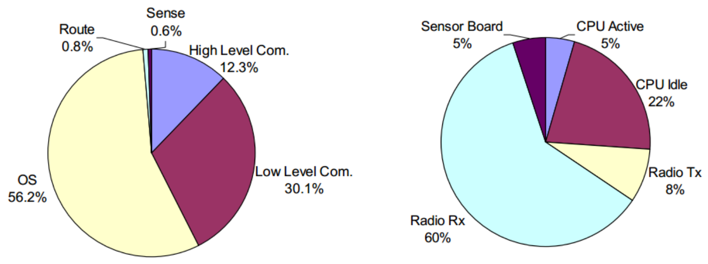


Figure 2.6: Breakdwon of the energy consumption in the main blocks of a Mica2 sensor node. Taken from [68]

mance.

There is a trend in attacking the problem of energy optimization in two directions: one is based on more and more advanced low power hardware modules for booth processing unit and transciever while the other of the application and protocol level which tries to improve the energy efficiency by implementing novel communication protocols, with less overhead for the acknowledgement of the messages. Improvement in the real time operating systems have their impact on the overall decrease in energy requirements. For example Hill et al.'s work produced a new kind of perating system for sensor network nodes, capable of fitting into 178 bytes of memory and uses messages of 12 bits length, with only 6 bits of context [45].

Another important issue to be taken into consideration is the dependance of the required power and energy when dealing with a large numbe of nodes. In this case we have for the beginning two distinct subcases:

- high density network with high number of nodes: when most of the nodes are in direct coverage of each other. In this case each of the node is going to make a single-hop transfer and the power is almost constant regardless of the destination node. Nontheless significant energy loss can incurr because of the chanell saturation and nodes would require multiple medium access trials before getting acces to the air medium;

- low density network: in this case verry few nodes are going to be in direct connection with each other and the burden of energy consumption if going to fall onto the packet routing over multihop links. Small density means there is going to be less chanell overlapping and the acces to medium can be made faster with less energy consumption.

Most wireless sensor networks fall in the second category because low-power designs imply small wireless range and modern routing proto-

BUPT

cols are highly efficient, designers prefering to have multiple hops than spectrum sturation. In the same time, typical tranciver powers offer a range in the order of tens of meters in urban envirnoment and having a high density of nodes in such a small area is the signature of only few networks: building automation or security systems.

For gaving some estimates regarding the actual power and perfomance of a small wireless sensor network, I'll present the figures for a hypothetical communication between two nodes placed at a distance of 3 km in-beween. Typical receiver power consupution of 10mW is assumed and a sensitivity of -60 dBm gives us a link budget of 80 dBm. For the transmision side of the link I'll take into consideration just the power of the final amplifier wich is an order of magnitude higher than the rest of the components. Having a single transmiter-receiver we need 110mW for covering this distance while if I install a single repeter/relay in the middle of the distance the power drops at 70 mW, giving us an improvement of 36%. Of course having a relay requires a network topology suitable for this and a protocol capable of handling the message routing and forwarding. This can be extended by adding multiple relays, while modern technology allows us to incorporate booth node functionaly and relaying into the same module, practically adding more value to the network.

From figure 2.7 one can observe that it is beneficial to add more and more relays (when they have also functional capabilities it is even better) but we should not forget that the deployment and operational costs also increase. It is well known from the industry [27, 25, 24] that the expense of deploying and replacing a single node of a actual sensor network can be more than 10 times the actual cost of the physical node. In this terms having a large network pose significant chalanges which have to be balanced by the designer, between the quality of the data and the available resources, taking into consideration actual conditions of the installation.

Similar investigation are carried also in the area of high performance computing, where the issues are targeting the dynamic minimization of energy consumption by reconfiguration and selective enabling of computational units, dependant on secific tasks to be performed. I've caried investigations in thies area and presented my findings in [48].

Metcalfe's Law

Formulated in 1993 by Gilder and addapted by Metcalf for Ethernet based networks, it states that "the value of a telecomunication network is proportional with the square of the number of users connected to the system" [95]. Modern adaptions exist for Internet, social networking and World Wide Web.

BUPT

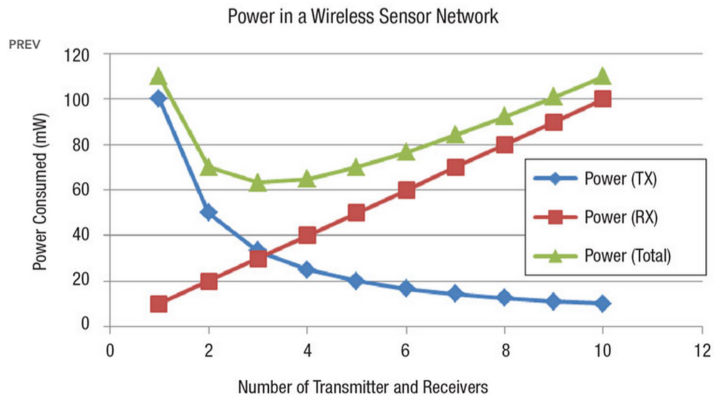


Figure 2.7: Total power consumption versus the number of nodes. Taken from [27]

It is based in actual graph theoretical metrics which I'm going to repsent in a separate chapter, but suffice to say for now that for a network of n nodes there is a number of $2^{\frac{n(n-1)}{2}}$ distinct possible connections, this being assimptiocially proportional with $O(n^2)$.

Limitations of this law start from the fact is quantized the maximum possible number of connection without taking into consideration that most of the time the actual number of connection is much lower. Another critique that can be made is that it assumes all the node a having the same importance to the network which is not the case in actual designs.

With all these Metacalf propsed an revised version of it's law which tries to map better to the real wold scenario which uses an $n \times log\, n$ approximation [95].

2.6 Network topology

Until now I've presented the basic aspects regarding the network and the end elements of it, which are the nodes with all the challenges and requirements for designing good performing ones. Next I'll talk about another important part of a sensor network, the one which defines the emergent behavior of all the simple atoms, which is interconnection of these nodes.

From a technical perspective much of the framework from computer networks still holds when dealing with sensor networks.

From a topological point of view the roots of this analysis are based in graph theory and I'll present the connections with complex networks analysis in the next chapter.

BUPT

Figure 2.8: One way sensing or actuating configuration

There are few classical regular topologies which are mostly possible to apply in engineered static networks while the topologies in the area of ad-hoc networks (seen as self organized) fall most of the time the area of complex networks.

2.6.1 One-way point-to-point

The simplest way of implementing data collection or actuation if the one way sending of data from the sensor to a processing node or the otherwise, but with no possibilities of exchanging the data flow direction (see figure 2.8). Applications of this type can be identified in the are of environment monitoring or various types of remote controls.

This kind of architecture exhibit the advantage of low cost per end node but the lack of proper acknowledgment, and protocol statelessness make them less suitable for modern application when the costs for more capable devices are equally low.

From the reliability of this solution the network as a whole relay on the massive number of nodes which are expendable on their own but there are still enough left so that the functional requirements of the network can be met.

2.6.2 Bi-directional

In figure 2.9 I present the natural extension of the one-directional solution which in this case is capable for carrying a full-duplex communication between the node and the sink. Aside from advantages regarding modularization and integration of booth functionalities in the same circuit there are some significant improvement in the area of reliability. From the application point of view, in this case we are able to implement more robust communication protocols for dealing with uncertainties regarding package arrival, fragmentation along the route and even cryptographic authentication.

Most of the solutions developed in the last decade fall into this category, alt least because of the above presented reasons or even from the functional characteristics of the application such as a plant monitoring system which requires sampling a variety of parameters in the process and actuation of various valves for regulating the process, where everything is done in a networked manner.

BUPT

Figure 2.9: Bidirectional topology for booth sensing and actuating

2.6.3 Star

Booth the above presented topologies are simple source-sink pairs. This can reach their limit in terms of functionality and advantages quite fast. Intuitively there are not much application to be developed on top of these architectures besides reading some parameters in a reduced are of the space and take a decision based on the data which has to be carried by the same module that took the readings. In this case the networked aspect of the solution is standing only because of the maybe higher computational power required, but with the advent of modern chips even this is fading out.

As I've stated above, the great advantages of the sensor networks (wired or wireless) arise from the massive number of nodes which can be deployed. At that moment we are capable of carrying computations and take decisions based on a complete picture of the phenomenon while the reliability of the network is positively influenced by the shear size of it. Loosing a small number of nodes is having a lesser impact over the functioning of the network as a whole, while the same is true for the possible rogue behavior of some of the nodes, which send erroneous data, but I'll get into more details regarding this aspect in later chapters.

A classical topology for this kind of networks is the star. Inspired from computer networks it has a *central* node acting as a sink (the destination of all the messages and the controller of the entire network) and all the nodes do the talking only with this one.

There are big disadvantages of this topology which arise from the requirement that all the nodes are in direct range to the sink. Considering the technical implementations this is a hard to follow requirement because we have to balance the short rang of the radio links (tens to hundreds) of meters with the large number of nodes (tens to hundreds or even more). Having such a large node density is required only in few applications.

The one way links are in this case suitable only for sensing application where the low costs are a primary concern while the bidirectional links are more preferred for all the other applications (see figure 2.10).

BUPT

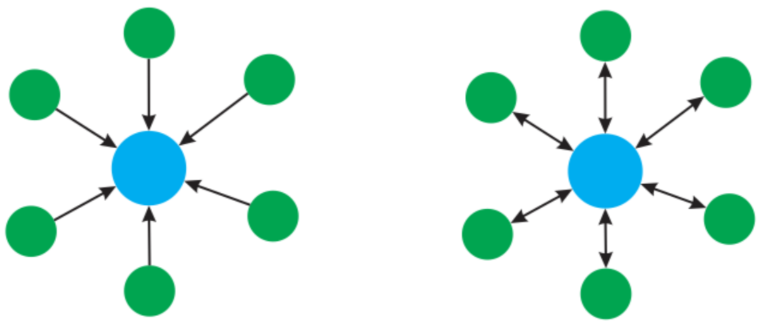


Figure 2.10: Sensing and sensing and actuating star topologies

2.6.4 Mesh

The successor of the star topology in terms of reliability which also minimizes the problem with the high density of nodes is the *mesh*. As seen in figure 2.11 in this case each of the nodes is in direct connection with all the neighboring nodes an the sink only communicate with it's own direct neighbors. In this case the network can cover a much larger geographical area while keeping the same simple and regular topology.

All the nodes should be capable of implementing a more complex networking protocol designed for message forwarding and some simple way of routing.

Disadvantages come from the requirement of having more complex nodes with a more expensive hardware. Each of the nodes is going to carry more computation and use more energy. In this case the low-power policies are harder to implement because the inactivity (sleep) times are shorter or even nonexistent.

From the reliability of the network there is a great improvement because the mesh fabric is tolerant to partial loss of nodes while the networking protocol is handling the message rerouting.

It is worth noting the building a span-tree of this kind of network shows us that the nodes closer to the sink (in topological terms) are under heavier load because they are required to handle a greater number of messages with translates into a higher energy consumption. Design decisions have to be made accordingly.

2.6.5 Cellular clustered network

As discussed above, all the previously presented topologies had some issues regarding either performance or reliability or cost/complexity of engineering and deploying.

BUPT

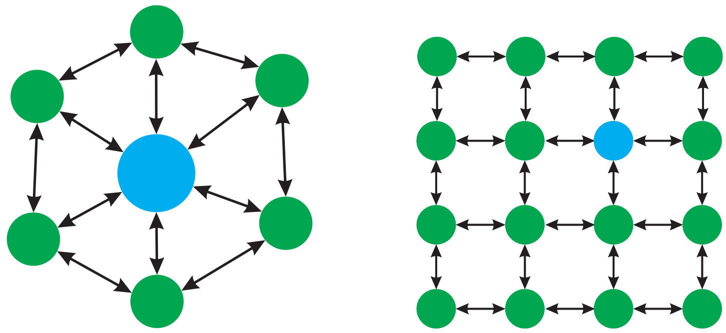


Figure 2.11: Complete 2D triangular (left) and rectangular (right) meshes, with the sink colored in blue

In order to mitigate this, inspired by GSM architecture, a new topology is introduced in the form of the *cellular-clustered* one. This uses elements of the mesh network on multiple levels, as seen in figure 2.12.

The hierarchical structure provides a good regularity at each of the levels thus requiring a simpler and less expensive engineering while still keeping a good fault tolerance. This topology is suitable in situations when there are a few disjoint regions which has to be monitored and all the data are fused together at higher levels.

From the reliability point of view there are advantages because at the lower levels nodes can be lost while keeping the integrity of the network. Still there are vulnerabilities in the upper levels, any of the high speed links (red lines in figure 2.12) being lost the integrity of the network being in danger. This kind of issues are addressed by my work.

A new aspect that's worth mentioning are the orange nodes. In this topology some of the nodes have a special role, besides sensing/actuating, that's the one of managing the operation of each of the clusters they supervise and to forward messages between clusters or the sink. They are the *relay* nodes.

2.7 Sensor network node placement

Until now I presented the background and prerequsites of actually deploying a sensor network with a strong wireless component. In more complex topologies such as the clustered-cellular one some of the links can be and are wired, but the prevalence falls onto the wireless ones.

In this context and as part of my further investigations the problems is regarding the optimal placement of the nodes in a network.

Mostly dependent on the task of the network but also of the particular

BUPT

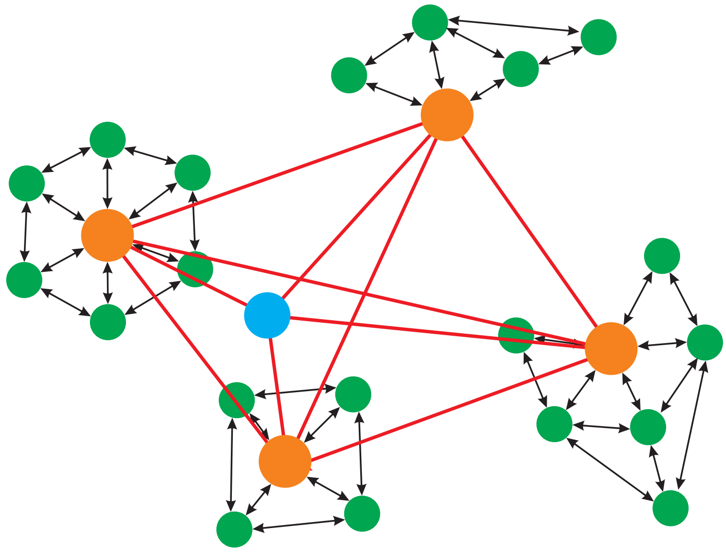


Figure 2.12: Clustered cellular topology consisting of four clusters and highspeed interconnection links. Relays are figured by orange nodes

conditions and the type of sensors there are two major strategies in placing the nodes of a sensor net-work: deterministic and random. The first one, when possible, can ensure great coverage with careful placement of the nodes and even the logical topology of the network can be established at deployment time [9].

Because of the adverse condition on the field there are situations where the single possible option for deploying nodes is in a random manner. This has adverse effects on the main metrics of a WSN [8]. In any situation where there is a large distance between two adjacent nodes, we witness a low throughput and high energy consumption.

Rich literature exists on the topic of optimal node placement [10], which is considered an NP-hard problem [11] and some non-deterministic approaches were proposed, which provide sub-optimal results [12].

Much because the current approaches in deterministic placement of the nodes proven themselves problematic but also because some of the typical WSN deployment scenarios presented both in the literature and also in the real life scenarios, such as wild fire prevention, battlefield monitoring or disaster rescue, require a random distribution of the nodes, even if there are some possibility of controlling the density of the nodes [11] my efforts is geared towards investigating the problem of relay placement strategies in this case.

Another interesting approach, which is also the starting point of my investigations, is the one presented by Xu et al. in [13]. The authors take into consideration a two-tiered topology in which nodes are clus-

BUPT

tered around relaying nodes which further communicate directly with the sink. The authors also consider a "multiple-hop communication case" which presume the existence of a hierarchy of relay nodes, connected in a tree manner to the sink. The authors propose a weighted random distribution, which increases the number of nodes as we move further away from the sink. One of the issues identified by the authors is that the random distribution can leave some parts of the network disjoint, actually partitioning the network, and their solution to the problem consists of the multihop deployment strategy [13].

Another research direction with practical application is the exploration of the issues arising when scaling the network. From an economical perspective, much of the sensor network deployments consist of incremental stages with more nodes being added (see previous section). Aside from the issues regarding possible flow congestions (packet storms) there is a great need to know the optimal placement of the relay nodes so that, with minimal costs, the new nodes are going to benefit of the existing conectivity infrastructure [11].

Early work has considered the coverage as being the paramount of the research [14], but because modern sensor node are running at the threshold of the energy requirements their coverage is largely diminished. Flat, 2D sensor networks usually consider relay nodes to be simply another node, but with higher transmission power and/or energy autonomy. The problem is getting interesting in two and multitier sensor networks where sensors are usually clustered in what can be called subnets, as presented by Chen et al. in [15]. Each subnet is sending data to a relay (called aggregation-and-forwarding node (AFN)) which in turn send the data via a multihop connection to the base station (sink).

One other approach is to use a deterministic approach in placing the relay nodes of a randomly deployed sensor network, such as presented in [15]. The problem is formulated in terms of initial set of nodes and their position, the task being to find the optimal placement of a set of relay nodes, so that the network lifetime and connectivity are maximized. Authors prove that the problem is still NP-hard, but provide a polynomial time algorithm. Tackling the problem of fault tolerance, there is an approach of maximizing reliability by placing relays at the intersection of two neighboring nodes.

Deterministic node placement is suitable in carefully controlled enviroments and requires precise placement of each of the nodes at predetermined possitions of a grid (figure 2.13). This kind o policy is suitable for applications such as building management and automation, permiter surveilance or external reference localization and mapping.

Semi-deterministic placement is a policy in which the nodes are placed in a random manner onto the grid but in precisellly known areas of the

BUPT

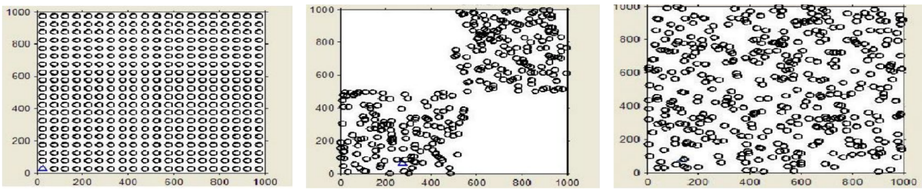


Figure 2.13: Types of node distributions: deterministic grid (left), semi-deterministic (center), random uniform (right). Adapted from [110]

grid (which are deterministic). Aplications of this topology fall unde the type of environmental monitoring, chemical detection where area of space have to be strictly monitored, but there is no requirement for precise positioning inside that specific area, the number of sensors counting for reliability and fidelity of the data.

Non-determinisc placement is seen as a random placement with various density distributions depending on the modelled phenomenon of deployment. This is actually on of the most prevalent method of engineering networks because the cpntraints of the physical world would impose it, biasing far away from the ideal deterministic model. Based on the method of deployment and wether the sensors are fixed or are capable of moving the models emplyed can vary in complexity.

2.7.1 Coverage issues in sensor networks

Much attention was given into the literature to the problem of optimal/-maximal coverage of a specifc area in space. Definitelly it is one of the goals of any sensor network designer to provide a methodology trough which one can achieve maximal coverege trough minimum investment in technological resources and minimal operational and maintenance costs. In this terms there are two approaches for treating the coverage requirement:

- single coverage: each point of the monitored space has to be (and most of the times is) monitored by at least one sensor;
- multiple coverage: in which each point in the target space has to have k sensors monitoring it, defined as k-area coverage. An area is k-covered if exists at least k distinct sensors which provide full coverage of the targeted point in space [110]. This poses significant interest in the subfield of fault-tolerance for sensor networks and I'm going to detaild further in a separate chapter.

BUPT

3 Complex Networks

Since the grammar school we were taught to express things by using substantives and the relationships between them by using verbs. At a higher level anything of the same form can be represented in a similar fashion. In the middle of XVIIth century, the great Swiss mathematian Lheonard Euler wrote his famous treatise on *Seven Bridges of Konigsberg* which is considered the semnial paper and the moment of birth for a completely novel approach called *graph theory*. We had to wait until 1878 when James Joseph Sylvester published a paper in Nature and introduced for the first time the term "graph". Another important date in this extremily short hostory of graphs is theyera 1969 when American mathematician Fran Harary published the first book, consideret at that time "definitve", on graphs and every thing related. This book provided for the first time the common framework of investigation and discussion between scientists in various areas of understanding from mathematics and physics to socilogy and linguiestics. The second part of the XXth century is dominated by mostly theoretical contributions in the graph with little practical applications besides electrical engineering with representation of the circutis as graphs.

A special place in the general theory of graphs is represented by the graphs proposed by Paul Erdos and Alfred Reny, called *random graphs* which gave rise to a completely new branch of research called *random graph theory*.

3.1 *Classical* graphs

In the canonical assumption of graph theory, a *graph* G is seen as an ordered pair of sets, one of *nodes* $n_i, i = \overline{1...N}$ and the other a set of *edges* V - called also *vertices* - which connect pairs of nodes.

3.1.1 Teminology and notations

There are a few specific terms and definitions of them which are specific to the fieeld of graph theory and which I'm going to present in the following lines.

1. *Degree* of a node is the number of nodes which are conected to the particular one;

BUPT

2. *Adjiacency* is the property of a node of being connected to another one;

3. *Path* is a sequence of censequtively connected edges*;*

4. *Connectivity* is the property of a graph so that between any two node there is a path.

3.1.2 Random graphs

The theory of random graphs lies at the corssroads between the graph theory and probability theory. The term *random* is usually applied to the probability distribution of various properties of the particular graph but in the same time can be used to describe the process for generating the graph. Theoretically they are used in order to answer to questions regarding the properties of *typical graphs* but in the same time, practical applications are to be found in the special field of *complex network analysis* on which I'm going to refer myself in following pages. Of particular interest for the mathematical approach is the Erdos-Reny random graph.

Starting from the above ideea of generating random graphs in order to analysie various properties of graphs, the generation of a random one starts with a set of nodes (vertices) and the succesive addition of edges between pairs of existing nodes until a particular property of the graph appears.

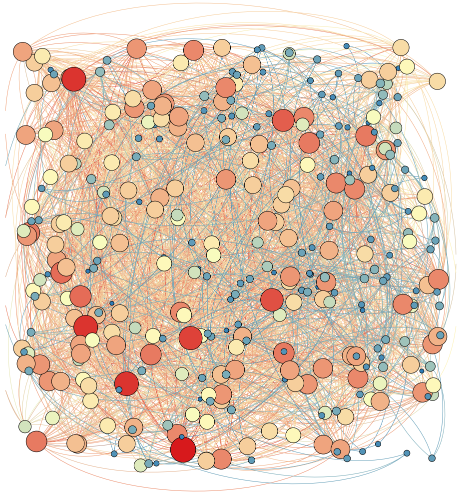
3.2 Complex networks

As the name suggests a complex network a specific type of network - also representable as a *graph* - for which we have hard to define topological properties. These proprties do not occur in other types of networks such as lattice or random graphs.

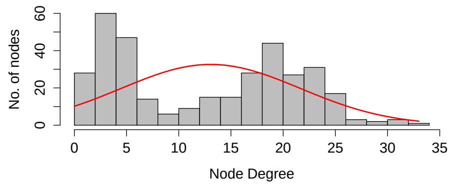
As defined by Homle and Kim [47], complex networks have patterns of interconections "that are not purely regular nor purely random". Specific to complex network analysis is the "heavy tail degree distribution", high clustering coefficient, communtiy structure and hierachical organization all of which I'm going to present more in the following sections.

Literature [98, 81, 102] mentions two big clasees of complex networks, these being the *scale-free networks* and *small-world networks*. Scale-free networks are defined by their power-law node degree distribution while samll-world ones are caractherized by short path length and high clustering coefficient [2].

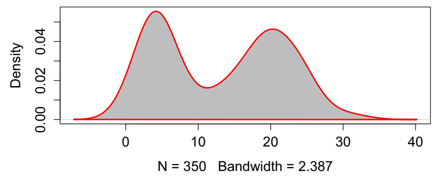
BUPT



(a) Visualization of the random graph with nodes colored and proportionaly sized by degree



(b) Degree distribution for the nodes in the random graph showing the normal distribution overlapped.



(c) Density plot of the degree distribution

Figure 3.1: Example of a random graph with 350 nodes and an average degree of 6.531

Complex networks lie at the crossroads of many diverse sciences, like biology, economics, geography, computer science, political sciences, psychology and are atracting at the present time a great deal of interest from booth academia and industry practitioners. Interesting results arise from the application of complex analysis metrics in economical and financial area and similar investigations I've carried in this field are presented in [15, 16].

3.2.1 Scale-free networks

In the 60's the studies regarding the networks of academic citations in scientific papers showed that the distribution of degrees (papers are nodes and citations are edges) folows a Paretto distribution or power-law distribution. Recent revigoration of the research in this field was observed in the beginning of the new century with the work of Albert Laszlo-Barabasi and his team regarding the mapping of the World Wide Web. They observed that some of the nodes (called in their paper, *hubs*) have a significantly more connections (links) than others and globally the distribution of WWW links follows also a power-law distribution. Barabasi continued his researches in the field and found that some other type of networks such as biological and social networks follow a similar type of distribution for the node degress thus introducint

BUPT

the term of *scale-free network*.

Of particular interest in the mechanism of creating such networks which Barabasi and Albert explain trough a methodology which they call "preferential attachement". In a simplified explanation the mechanism might be called the "rich-gets-richer" approach being the same one observed in social networks where a small number of people have a high number of followers/subscriptions.

From a more formal apprach the work of Li et al, is of particular importance because they introduced a numerical metric and a methodology of computing it such as the values range from 0 to 1, the latter being the perfect scale-free network. Using classical notations for graphs having, the following equations are extracted from their paper:

$$s(G) = \sum deg(u) \cdot deg(v) \tag{3.1}$$

where u and v are two distinct nodes. So the scale-freeness of the graps is described as the sum of products for each pair of distinct nodes. In order to nermalize the values in the $[0, 1]$ range the authors divide the value by the maximum value of the s obtaining:

$$S(G) = \frac{s(G)}{s_{max}} \tag{3.2}$$

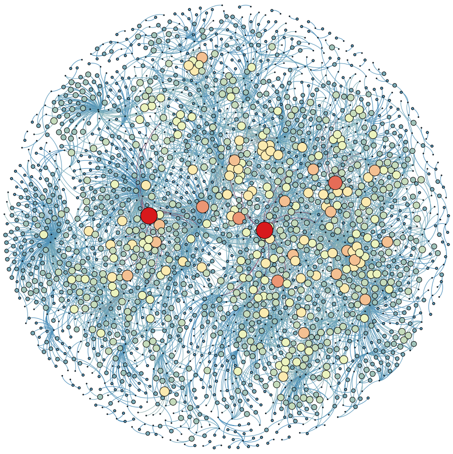
From a qulaitative point of view, most of the assumptions regarding the type of networks is usually done using the graphs of the distribution of the node's degree.

3.2.2 Small word networks

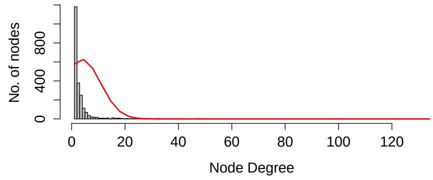
3.3 The New Network Science

The novelty introduced in my thesis is the usage of complex network analysis (SNA) principles to enhance the properties of a wireless sensors network topology. Initially emerged from techniques proposed by modern sociology [78], statistics and graph theory, SNA etc. It analyzes complex networks, which consist of nodes (individuals, actors inside the network) and ties between these nodes. The ties can represent physical links, but also friendships, organizational ties or any other type of relationship between individuals [39, 103]. Developed from complex network theory [59], a social network is a complex network which is mainly analyzed from a social point of view. The elements/people, which represent the set of nodes, form a complex set of binary ties (i.e. the smallest possible social group, formed out of two people) [34].

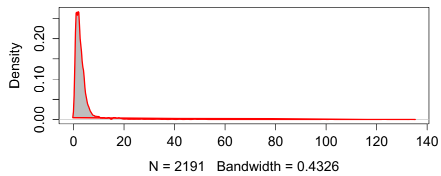
BUPT



(a) Visualization of a scale free graph with nodes colored and proportionaly sized by degree. There are very few nodes with high degree (the big one) and a lot of nodes with a very low one (small blue "dots"). The are taken from the Drug Category Network (DCN) project [?]



(b) Degree distribution for the nodes in the scale graph with the normal distribution trend line overlapped in red.



(c) Density plot of the degree distribution

Figure 3.2: Example of a scale free network with 2191 nodes and an average degree of 1.922

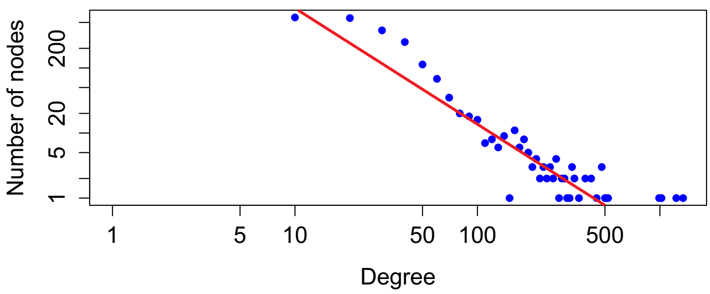
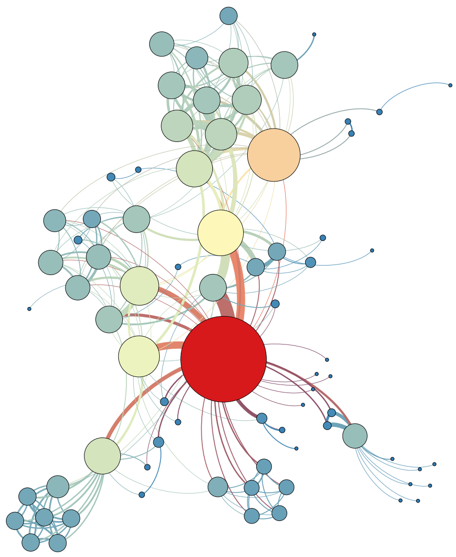
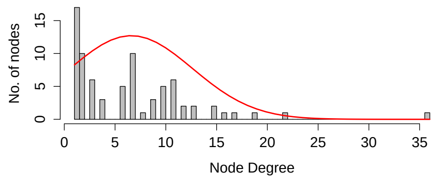


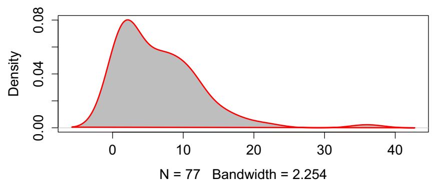
Figure 3.3: Logarithmic plot of the nodee degree distribution from Figure 3.2b for a scale free network. The linear trendline is overlaped in red.



(a) Visualization of a small world graph with nodes colored and proportionaly sized by degree. The is a much better uniformity in the distribution of the degrees (similarly sized nodes and continuos shades of color). This is the “Les Miserables” dataset compiled by Donlad Knuth showing the interactions among the caracthers of the novel with the same name.



(b) Degree distribution for the nodes in the scale graph with the normal distribution trend line overlapped in red.



(c) Density plot of the degree distribution

Figure 3.4: Example of a small-world network with 77 nodes and an average degree of 6.597

BUPT

Through measurements performed over raw, state of the art sensor networks my goal is to propose an optimal coverage of physically-linked relays over any given network so that we maximize the throughput and reliability and minimize the number of relays and cost of interconnection.

3.3.1 Complex Networks metrics

Troughout much of the subsequent work and as a quantifier for various algorithms I've developed I've measured the basic network metrics: network size (nodes and edges), average path length, clustering coefficient, average degree, network diameter, density and modularity, and also the distributions of the degrees, betweenness, closeness and (eigenvector) centrality [19].

After performing complex network analysis I've concluded that an optimal way to decide the relay placement is through community detection and centrality algorithms which I'm going to detail further.

The average path length of a network is the mean distance between two nodes, averaged over all pairs of nodes.

The average value of the degrees, measured over all nodes, is called the average degree of the network.

The diameter of a network is the longest of the shortest paths between any pair of nodes in the network.

3.4 Centrality in complex networks

As the term implies, *centrality* is a general way of speaking of how "central"is a node to a network, and depending of the case it might even map to the node's importance in that particular network. Starting with this vague presentation of the term, it's obvious that there a few more precise definitions. It is worth noting the fact, centrality was first intriduced as a metric for social network analysis and consequently many related notions can be found in the area of social sciences Newman [83].

Giving a more precise definition of what centrality is we have to start by tring to answer to a question given by Borgatti Borgatti [18]"What characterizes an important node?". In the world of network analysis there are two major approaches in measuring importance: one is regarding the flow trough the network and the other is dealing with the cohesiveness of the network.

It is worth noting s subtle observation presented by Lawyer and Glenn in Lawyer [69] regarding the effective usage of any of the centrality metrics. The problem is the "vertex centrality indicate the ralative

BUPT

importance of vertices only in a relative way". Any of the centrality metrics is designed to provide a ranking of the nodes, but they do this well eneough only for the top-tier of the nodes and loose of their power for the rest. Also the diffrence in absolute value does not map into difference in importance of the nodes Lawyer [69]. Authors give a hint regarding possible eplanations if this behaviour which lie in the heterogenous structure of the network itself. While, usually, the network is caractherized by a small set of "important" nodes, in the end she as a whole is allmost always heterogenous and consequntly we cannot apply the same centrality metric for any subset of nodes expecting consistent results Borgatti [18]. This is a verry important conclusion for my work which is explored in depth for designing STiLO algorithm.

Next, I'm going to present some of the most important centrality metrics which I have used in my work with qualitative illustrations.

3.4.1 Degree centrality

Having all the data presented until now, someone may consider solving the problem in a simple manner and call the most important node, the one with the most connections, which translate in the highest degree for that particular node. While this might be true for some situations and questionable for others, ideed it's the simples way of defining the centrality.

In terms of flow analysis, presented abowe, the degree centrality can be expressed as being the dependance in which the node is going to be traverssed by anything tha traverses the network.

Using the notations introduced in 3.1.1 the degree centrality (C_D) can be computed as follows:

$$C_D(v) = deg(v) \tag{3.3}$$

so, it's actually the degree of the vertex. Going a step further we can define the a similar metric for a subgraph. If ve consider $v^+$to be the node with the highest degree in the graph G and we have a subgraph $G' \subset G$ having itself also a node with highest degree, denoted by v'^+, then according to Borgatti [18]

$$C_D(G') = \frac{\sum[C_D(v^+) - C_D(v_i)}{\sum C_D(v'^+) - C_D(v'_i)} \tag{3.4}$$

It can be seen that the denominator of the fraction is maximized when we have a star structure with only one "central node" and all the others being directly connected to it.

In figure 3.5 I present the application of the above presented metrics on a random network consiting of 50 nodes and 145 edges giving us an

BUPT

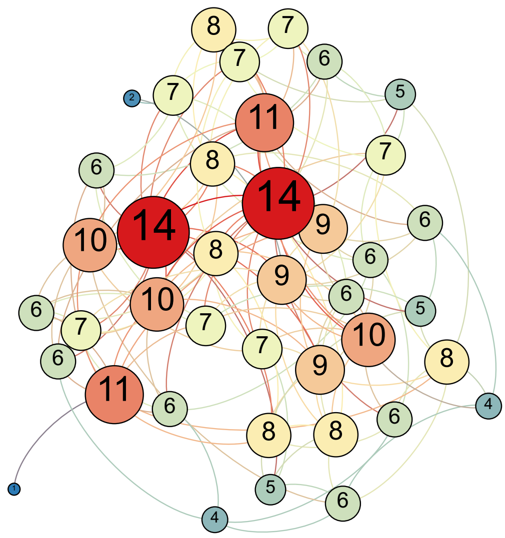


Figure 3.5: Degree centrality illustrated on a random network with 50 node and an average degree of 3.625. Node as colored from highest degree centrality (reddish) and up to the lowest (blue) and sized proportionally.

average degree of 3.625. Each of the nodes is sized and colored proportionally to it's centrality. The renedering layout used is Force Atlas 2 which I'm going to use a few more times during my investigations, but in a few simple words, it is an algorithm for drawing graphs, which consider nodes bodies with mass and the edges are elastic springs, after which it runs a gravitational attraction algorithm to find the begaviour of the system.

Knowing the inner working of the rendering engine and the degree of each of the node, we can observe the fact highest degree nodes in figure 3.5 are indeed the ones which at least "look" to be important, but as it can be seen in the simple counterexample from figure 3.6 this is not always the case. The flow apporach to this situation clareley shows the middle placed node as beign much more important even if it has a lower degree. This makes me introduce another metric for centrality, which is the *betweeness centrality*.

3.4.2 Betweenss centrality

While de above presented degree centrality was taking into consideration the number of links (connection) of every node, in this case we are speaking of the number of paths which pass trough the node. Inspired by social sciences it was first intrudoced by Freeman in order to describe the "influence" of a person in a social networks, the person

BUPT

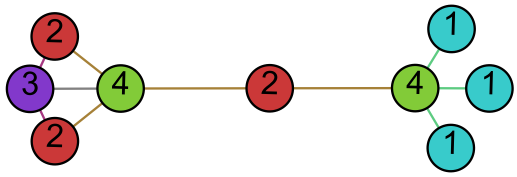


Figure 3.6: A simple counterexample for the degree centrality. Green node are having the highest degrees (4) followed closely by the purple one (3), but from the data flow perspective the reddish "centrally placed" node of degree 2 is much more important.

being seen a communication bridge Freeman [39].

Informally the betweness of a node is the number of shortest paths, between any two nodes in a network, with pass trough that particular node.

In a mathematical way, this can be described, consistently with the above formulae, as:

$$C_B(v) = \sum \frac{\sigma_{\overrightarrow{st}}(v)}{\sigma_{\overrightarrow{st}}} \quad (3.5)$$

where, $\sigma_{\overrightarrow{st}}$ denotes the total number of shortest paths between any two nodes in the graph and $\sigma_{\overrightarrow{st}}(v)$ is the number of paths from any pair of nodes, $s$and t which pass trought node v.

On a simple algorithmic level the computation of the betweeness centrality, can be expressed as follows:

- compute the shorthest path in the graph, for each pair of distinct nodes, s and t;
- taking the node under consideration v count how many of the shortest paths found in the previous step, pass troug that particular node;
- normalize the value in the $[0, 1]$ range by computing the result over the total number of the distinct paths in the graph.

In regard to the previouslly discussed degree centrality, when using betweness we take into account a much better semantics of the *network flow*, because it's close connection with the paths in a graph. In figure 3.7 and 3.8 we can observe the application of the definition on the two networks discussed above when we defined the degree centrality. While havin the same network, when we apply the new definition we observe a much better "correlation" with our intuition regarding what an important node in a network is.

BUPT

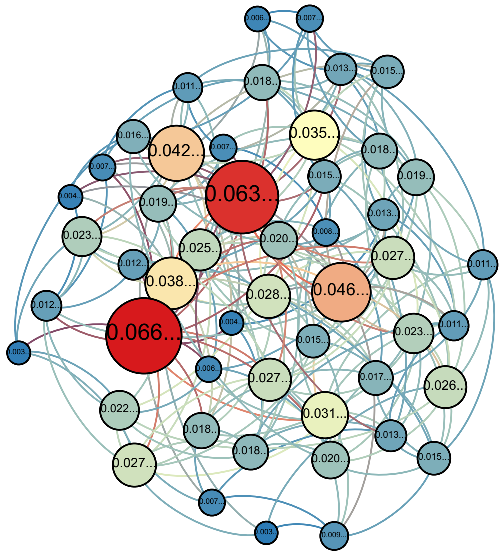


Figure 3.7: Random network from figure 3.5 with nodes colored and sized by betweeness centrality.

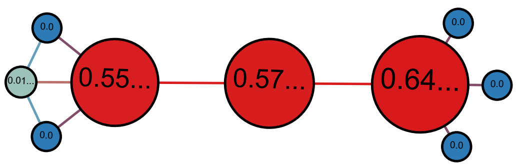


Figure 3.8: In a simple network (same as in figure 3.6) there is a clear difference between degree centrality and betweeness cetrality, defined as the number of shortest path that cross the particular node. Each node is labeled, sized and colored corespondingly to it's betweeness.

BUPT

The simple algorithm presented above requires computing the shortest path between all distinct nodes in the graph. Classical approaches with the Roy-Floyd algorithm would have a polynomial complexity of the form $\mathcal{O}(n^3)$ while when having sparse graphs (small number of edges in report to the complete graph, also denoted by *network density* metric) we can use the Jhonson algorithm which would have a complexity of the form $\mathcal{O}(|V|^2 logV + VE)$ Johnson [57].

3.4.3 Closeness centrality

Another metric of centrality, this time, more related to the topological concept of being in the center is the *closeness centrality*. For any node v we can compute the network distance of that particula node to all the other nodes as the number of edges which have to be traversed on the shorttes path on irder to read the far node. This direct value is, by the way, called *farness* and the closeness of a node is simply the inverse of this value. In this apporach a small flow-type componenent can be identified if we define the closeness as being proportional with the time required to disseminate the information regarding a specific event taking place in an arbitrary node v trough the network.

For an intuitive way of approaching the problem this may be seen as the best and most accurate way of computing centrality, but in the same time it is not consistent with the flow approach and takes into consideration only the static aspects of the network, ignoring the fact that in the end this network is only the fabric which provides more complex "services" built on top of it.

It can be easally seen that for an unconcted network, the centrality metrics for any and all the nodes would be zero. Consecquently there is an updated model, defined by Dongachev in Dangalchev [35] which takes into consideration also the situations where the networks is unconcetd. His approach requires identifing the connected components of the graph, computing the closeness centralities of every node in each of the connected component and agregate the partial results in order to compute the closeness centrality of the entire graph.

For my investigations the possibility of applying the concept of closeness centrality to weighted graphs is of particular importance and the work of Opsahl Opsahl et al. [86], continued by Boldi and Vigna Boldi and Vigna [17] provide some insights into metrics suitable for this case. They provide a simple formula,

$$C_H(v) = \sum \frac{1}{d(w,v)} \tag{3.6}$$

and introduce the convention by which $1/\infty = 0$ so, because the weight (distance) of a pair of unconcetd nodes would be $\infty$the closeness

BUPT

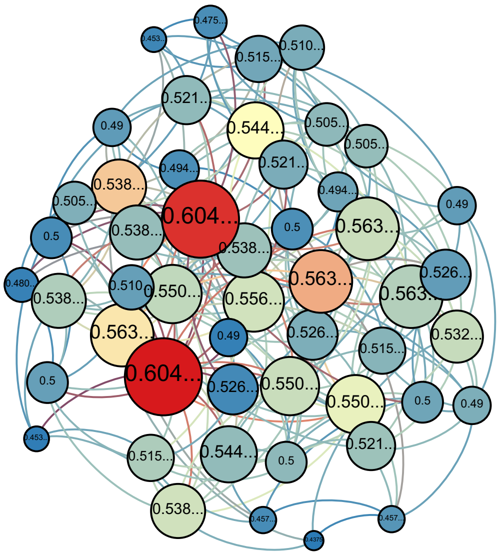


Figure 3.9: Random network from figure 3.5 with nodes colored and sized by closeness centrality.

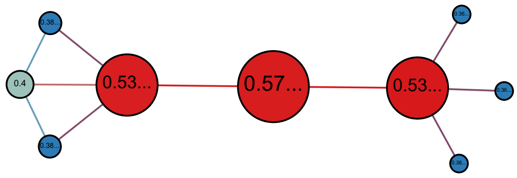


Figure 3.10: Closeness centrality of the nodes of a simple network, introduced in 3.4.1.

of that particular pair of nodes would be 0.

For the graph I allready have used as example in the previous sections, I've plotted the application of the closeness centrality rule, which can be seen in figure 3.9. One can observe a simillar, but non identical distribution of the nodes, and while using the same algorithm for layout, we have to tahe into consideration that the emphasized nodes are far from being the same as for the beetweeness distribution.

In figure 3.10 I've plotted the values for the closeness centrality on the simple graph I've allready used as vehicule for discussion in the above sections. One can perceive the fact graph being almost symetrical, the values follow the structure of the network. The intuitively central node has indeed the highest value, while the peripherial nodes have much smaller values.

BUPT

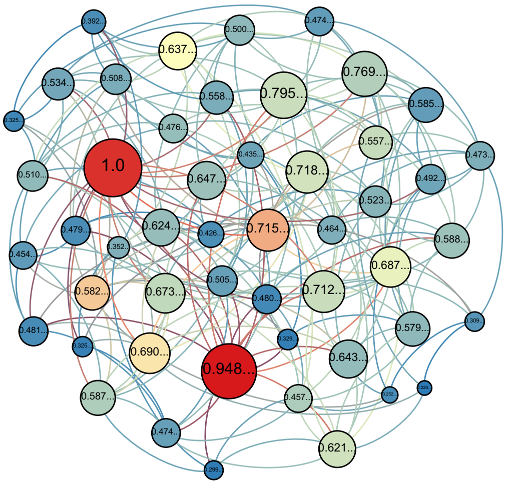


Figure 3.11: Random network from figure 3.5 with nodes colored and sized by eigenvector centrality.

3.4.4 Eigenvector centrality

The last metric of centrality I'm going to discuss, but far from covering with this the extensive range of metrics existing in the literature is the *eigenvector centrality*. Simply put, the eigenvector centrality tryies to compute "the value of the influence of a node in a network" Opsahl et al. [86]. Each node has an absolute valued score, with node having higher value, being considered more imprtant. The is a well know metric, derived from the eigenvector centrality and this is the Google PageRank score Newman [85].

As I've discussed before, my apporach is going to be from an algorithmic point of view and in this case we start from the well known description of a graph, as an adjacency matrix (A). Newman in Newman [85], defines the centrality of a node v as being:

$$C_E = \frac{1}{\lambda} \sum$$

which also as described in Newman [85] would yield us:

$$Ax = \lambda x \tag{3.7}$$

Literature recommends using the power interation algorithm for computing the eigenvalues which in term also guarantees us to find the highest value, if multiple exists Newman [85] .

Of interest for the practitioner is the fact that there is guaranteed that always is going to be a single node with absolute value of 1 (the most influent in the network). Figures 3.11 and 3.12 show the application of

BUPT

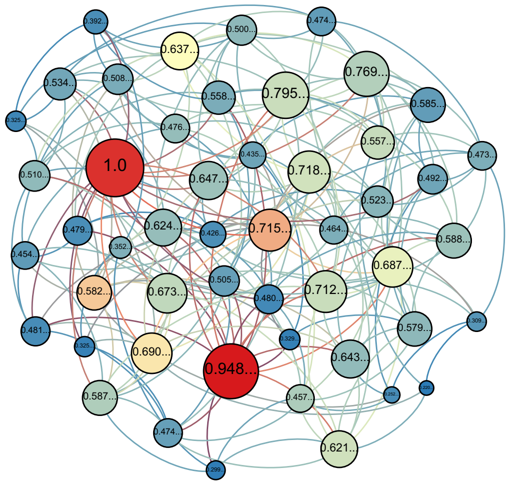


Figure 3.12: In a simple network (same as in figure 3.6) there is a clear difference between degree centrality and betweeness centrality, defined as the number of shortest path that cross the particular node. Each node is labeled, sized and colored corespondingly to it's betweeness.

the eigenvector centrality on the same two networks discussed above. This time for the simple network in figure 3.12 we can perceive that is a greater shift in structure, the highest value node, being the one with 1.0 in the left side of the picture. This is because that node is having most direct or short distance connections to most of the other nodes.

On the influence flow side of the interpretation, for example an event occuring in the node with *EV-c* 1.0 is having the highest propability of being spread fast trought the entire network while one occuring in the smalest blue nodes from the far side of the figure is going to need a much more travel route in order to be completely propagated with the corresponding diminishing in the influence of the "message".

This observations are going to be used in my algorithm for information diffusion in the sensor networks.

3.5 Comunity structrure in complex networks

What can be seen as a natural consequence of the way the nodes and edges of a network are organized, and translates into a tighter grouping of nodes around some areas of the network in regard to other areas,

BUPT

was formalized and called *community structure*. So to speak, in any large enough network one can perceive a spatial clustering of nodes, corresponding to various particular situations of the network, while in other parts of the same graph there are almost no edges.

From a more formal point of view, the community strucutre is a property of a network as a whole of which we can say it has it if the nodes can be grouped in sets (subgraphs) which are densely connected internaly [84]. There is a valid possibility that some of the nodes (and their respective communities) overlap.

From the practical point of view, communities are more often to be found in netrwoks which represent pehnomenas where there are allread some kind of community structure. For example there si a clear ovservation of this for all the graphs which represent social networks. In this case there si a verry good match between the actual real life connections (friedship, business contact) of the people and the results obtained by the community detection algorithms of which I'm going to speak later. Similar situations can be found in metabolic networks (Brabasi's Disieasome) [9] where there are connections and groupings dependant of the same gene or the citation networks where comunities are related to the same topic of interest[80].

There is an interest in developing algorithms and strategies for better and faster identification of communities because this approach in analysing large scale networks is capable of providing a bettwer insight into the functioning of the system represented and prodict the outcome of changes made to the network.

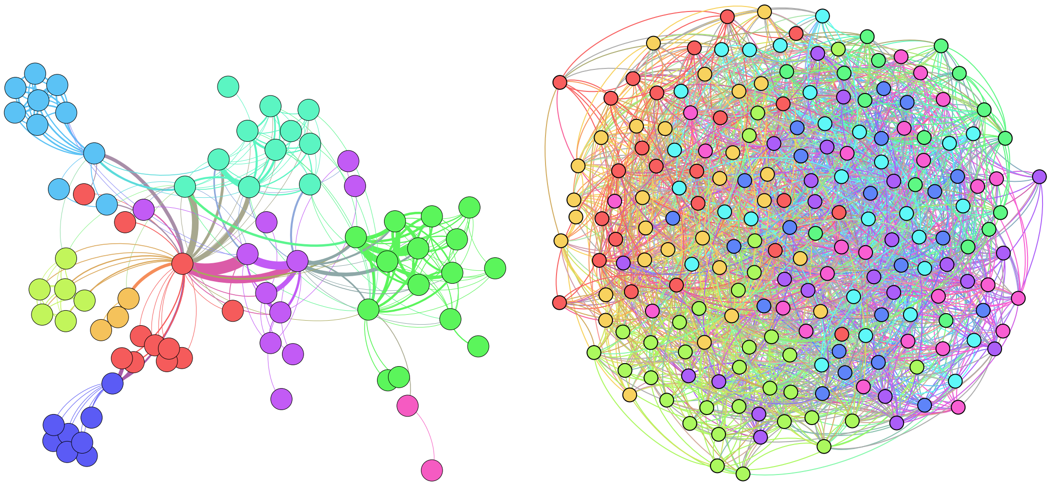
In the same case, it is not true the fact there are communities in any network and current algorithms can produce misleading results as can be seen from figure 3.13b. I've generated there a synthetic network with 200 nodes and a uniform random connection propability of 0.1. After running the community detection algorithm there are 7 identified communities, each depicted with a different color, but there is no interpretation to be made with the actual domain being a random network. So, there is the role of the practitioner to decide if and when to apply the algorithm and what interpretation to give to the results.

3.5.1 Modularity of a complex network

Until now I've discussed about the general property of a complex network of having the capability of being divided into smaller parts in a way simillar to a community breakdown. The next question which is rising is how do we measure this property.

In this case we can relay onto the *modularity* which is one of the metrics that measure the structure of a network, more exactly the strength of the partitioning/division of a graph into clsuters/communities [31].

BUPT



(a) Community structure of an artificial social network (the caracthers of *Les Misserables).*Taken form [11]

(b) Community structure of a 200 nodes random network

Figure 3.13: Large network can exhibit clustering of nodes in forms of communityes which map onto the particular properties of the real-life domain-inspired network. In figure 3.13b the results are wrong being the case of a syntheically generated random network.

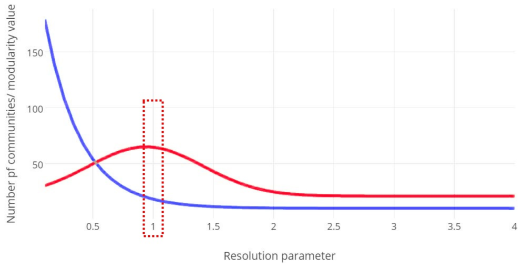
The higher the modularity number, the better community structure the network has. The value of it is in the range of $[-\frac{1}{2}, 1]$[84].

The computation of the modulariy relays on comparing the actual number of edges from a cluster with the number of adges that would fit into the same cluster if they would be distrubuted randmly, but keeping the distribution of node degrees unchanged [4].

Classical applications of the modularity metrics are in the field of community identification but there are issues with the resolution parameter: it is not capable of good identification of small comunities, this being called the *resolution limit*. The cause lies in the definition of the metric: if the communities are too small there are not many ways of randmly reconnecting the nodes so the make a clear distinction.

For the particular discussion of my work the interests falls into the particular problem of answering wether a given network can subsequently be divided or if a given algorithm should be recursvely applied, actually providing stopping conditions. If the network is not having anymore a community structure it is presomptuous to still apply methodologyes which relay on it.

BUPT



Resolution	No. of communities	Modularity resolution
0.1	182	0.055
0.2	128	0.07
0.3	91	0.081
0.4	70	0.092
0.5	56	0.098
0.6	45	0.115
0.7	36	0.112
0.8	27	0.12
0.9	21	0.129
1	17	0.137
2	9	0.044
3	9	0.044
4	9	0.044

Figure 3.14: The dependance between the resolution and the number of communities in the Modularity Algorithm. For each of the analyzed networks I've found an inflection point where the behavior of the function changes from exponential to almost linear. The practitioner has to choose on experimental bases the suitable value inside the red window.

3.5.2 Community detection and resolution

There are numerous approaches formalized into algorithms for finding communities, usually dependent of the specific application and on the relevance of link density [38, 32, 31].

The community detection algorithm as described in [83] [14] and implemented in *Gephi* uses a single parameter, the resolution; the lower the value, the larger the number of identified communities (each with fewer nodes). The default value is 1.0. Therefore a consistent methodology for choosing the resolution is required. I've carried out my investigations on a set of representative networks which model the typical literature-referenced sensor network topologies, but in the same case the methodology still hold even when applied to urban transportation networks. Figure 3.14 plots the number of identified communities against the resolution value. The networks used vary in size and acutal topology: mesh, star, celullar, random.

For each of the data sets one can perceive allmost exponential decay of the number of detected communities right after increasing the resolution from the default value of 1 (the left part of the plot), and an inflection point marking the area from where it becomes almost flat (the area defined by the red box in figure). For the rest of the algorithms and methodology I've designed, the practitioner has to choose a suitable value for the resolution inside the marked area.

3.6 Concluding remarks

This chapther was dedicated for introducting the basic terminology and concepts I'm using in my research, specific to the complex networks

BUPT

and graph theory. Much of the subsequent work is based on the two notions of centrality and community structure/modularity. In this terms I've dedicated some space presenting the specificities of the two, emphasizing on the pitfalls and limits of them.

Regarding centrality in graphs/networks it is worth noting that most of these can be adapted and fine tuned for various specific networks using the experince and understaning of the investigator while, in the same time, concepts can be extended to other types of networks, such as directed graphs of weighted ones Opsahl et al. [86]. Work has been done also on dynamic, time dependeant networks, in which consequently the centrality of evolving over time Hill and Braha [46].

On the other side the abstract term of community which has valid significance for social and human aspects of life is equaliy hard to define and quantize in computation terms. For this, there are several algorithms and methodologies which exhibit some degree of uncertanty and require parametrization from the paractitioner, him being the only one capable of deciding wheter the actual results are suitable for a particular case of study. For this I've dedicated a specific section discussing the limitations of the resolutions parameter in regard with the community size.

In the next chapter I'll present the integration of the concepts presented earlier in the form of the main algorihm I've designed as part of my research for optimal palcement of relay nodes in a sensor network.

BUPT

4 Optimal relay placement

Many applications require the deployment of an array of nodes which have to operate unattended for long time, on limited power supply (usually batteries). Major deployments consist of hundreds of nodes interconnected at a logical level, in accordance to topologies and also at a physical level, wirelessly or even wired [79]. Running on batteries, one of the major issues that can be identified is the depletion of the battery's energy [24]. Such situations can cause issues with network's topology because some of the relaying nodes are not able to forward data causing partitioning of the network and disruption of the services [61].

Applying social network analysis principles in order to analyze and optimize sensor networks is nothing but natural as the social perspective provides an innovative means of analyzing the structure of entities with a social-like structure [104, 103]. Thus, we can detect influential nodes, patterns of communication and also study dynamics inside the network. This strongly relates to wireless sensor networks as it is important to determine which sensor nodes are critical for the data throughput, which are more central so that relays can be placed at those positions, and also model growth as the network coverage spreads in time.

Much of the research in the field of sensor networks was oriented in the last years to maximize what is called "functional requirements", such as data latency [72], real time-ness and "non-functional requirements" such as data integrity [77], always taking into consideration the main constraint: limited energy supply. Minimizing energy requirements, is seen most of the time as the main solution in prolonging the life of a particular node and consequently of the entire network [74]. Because most of the sensor network deployment is in tough and adverse environments there are many more hazards than just energy loss [1].

In all of the following discussion I'm going to differentiate between regular nodes, responsible for gathering data and/or acting upon received commands and relay nodes which collect data from the nodes in the direct area of coverage and send them upstream to the sink. The scope of this research is to propose an optimization solution for choosing the number of required relays and their optimal position so that we maximize the performance of the network while keeping the

BUPT

overhead at a minimum.

In the design of a WSN the practitioner has to balance the costs involved with the solution, with the performance, and one of the key performance metrics is the average delay from node to sink. This part of the research is part of a larger endeavor of designing and deploying a near real-time sensor network for monitoring and reporting data regarding road traffic conditions and consequently dynamically adapt the state of the traffic lights which is going to be presented in Chapter 6 of this thesis.

4.1 Problem statement

Given any two-dimensional WSN, we model it as graph $G = \{N, E\}$, composed out of nodes N and edges E. The set of edges consists of all wireless links between all pairs of sensors inside each node's coverage area, like in an ad hoc network. The requirements are as follows:

- assign one sink for the network: $s \in N$,
- assign an optimal number of relay nodes: $R \subset N, |R| \ll |N|, s \in R$

in order to balance a maximal performance and a minimal cost for G.

Another assumption considers the relay nodes R interconnected using cable links with negligible latency and infinite power supply.

The performance is expressed in terms of number of hops required to reach the nearest relay (relay-to-sink communication is considered negligible, as mentioned above) and the cost is expressed in the number of required relays $|R|$.

4.2 Socializing the Network

In this section I describe the methodology of enhancing a wireless sensor network using the SNA theoretical principles described in Chapter 3. The enhancement is done at the physical level, of determining where and how many relays to place over the wireless sensor network physical monitored area, as well as how to interconnect those relays.

4.2.1 Network processing

In order to generate our input data I've used the *WSNet* topology generator [21]. It produces a set of nodes (sensors) with 2D geographical data. The next step is to convert the information into *gdf* file format which can be imported in Gephi [11], the leading tool in large graph data visualization.

BUPT

```
nodedef> name VARCHAR, x DOUBLE, y DOUBLE
0, 218.74862670898438, 96.19173431396484
1, 100.27324676513672, 146.0537567138672
2, 207.51651000976562, 252.3385772705078
3, 342.5207824707031, 41.2176399230957
4, 0.8681038618087769, 332.48284912109375
5, 112.9320068359375, 349.3313293457031 ...
```

Figure 4.1: Example data dump of the WSNet topology generator

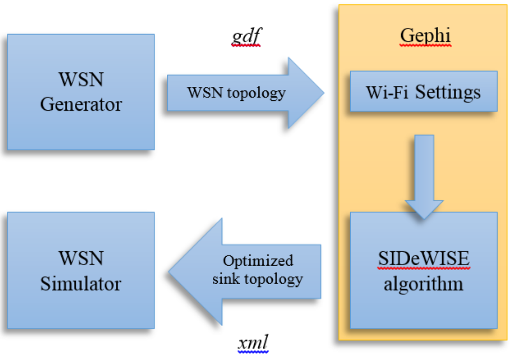


Figure 4.2: The network processing workflow using WSNet for topology generation and custom Gehpi plugin for import and implementation of my algorithm

Using WSNet, I've generated geographic random mesh configurations for testing purposes, but any other layout of sensors in a geographic space can further be processed by my algorithm. Figure 4.2 describes the network enhancement process. To apply the algorithm one first needs to generate a topology of wireless sensors which have positional data (x, y) attached to them. Once the topology file is transformed into *gdf* format it can be imported in Gephi using my custom implemented plugin. The wireless coverage area can be set after which the enhancement algorithm, called SIDeWISe, processes the topological data. The resulting topology is the initial sensor network with an additional overlapping layer of *optimally placed relays* which are all connected to a sink through *minimum cost physical links*. Once the algorithm finishes it can be rerun by customizing its parameters (discussed in Section 4.4) and the output file can be exported in XML format for further processing into a network simulator of choice, like OPNET, OMNet++, NS-3 etc.

Algorithm 4.1 SIDeWISE Algorithm: Build mesh network

Input: Sensor network represented as graph $G = \{N, R\}$ with geo-positional data of the nodes (n_i, n_j).

A: Link all nodes in wireless range

$E \leftarrow \{\}$
foreach node n_i in N:
  foreach node n_j in N, $i \neq j$:
    if distance$(n_i, n_j) < r$:
      $e_{ij} \leftarrow$create edge between (n_i, n_j) **if** none exists
      $E = E \bigcup e_{ij}$

Output: Mesh sensor network $E = \{N, R'\}$ with complete conections in the close proximity $(< r)$ of each node

4.3 The SIDeWISE Algorithm

In this section I will present the SIDeWISE (SocIally enhanceD WIreless Sensor nEt-work) algorithm which enhances any given sensor network by overlapping an additional physical network of optimally placed relays and a single sink over it.

The enhancement process is presented using the graph visualization tool Gephi, which offers standard graph analysis tools, and I've implemented a plugin designed as a Java Gephi plug-in to express my algorithm. This process is depicted in Figure 4.3. We start with a sensor network senn as a 2D graph and a given wireless coverage range r, and the first step (**step A**) is to create point-to-point edges between all pairs of nodes that are within one's range (in the fashion of an ad-hoc mesh sensor network). Thus, we obtain the complex network (graph) $G = \{N, E\}$, defined by the set of vertices (N) and edges (E) between those vertices.

The distance (n_i, n_j) is defined as the Euclidean distance between the two points (x_i, y_i) and (x_j, y_j).

Step B of the algorithm implies determining which node would best fit as being the (single) sink of the network. For this I've choosen to measure the betweeness centrality of the network and assign the node with the highest centrality as the sink, in contrast to the geographically centered approach proven less effective [108, 110]. The centrality is represented as a floating point value inside $[0, 1]$, with 1.0 always corresponding to the most "central"/"important" node. As in a social context, the most central node is the one being closest to all other nodes on average. This is measured by having the shortest average

BUPT

Algorithm 4.2 SIDeWISE Algorithm: Compute the location of the *sink* and assign the *sink*

`Input: Sensor network represented as graph` $G = \{N, R\}$ `with geo-positional data of the nodes` (n_i, n_j).

B: Assign sink

foreach node n_i in N:
  $C[n_i] \leftarrow$ compute_centrality$\{G, n_i\}$
find n_s in N where $C[n_s] = 1.0$
$R = \{n_s\}$
$n_s \rightarrow$ sink for G

`Output: Identity and location of the sink node`

paths to all other nodes in the graph. Once this node is determined it becomes the sink.

Using Gephi, the centrality of all nodes is measured and the node n_s with the highest centrality (1.0) is chosen as the sink for graph G, and is also added to the set R of relays.

Step C determines the clusters of sensors which are relevant to the network from the throughput perspective. While it is not a common practice to determine communities in sensor networks, communities are highly relevant in social networks and other types of complex networks as I've presented in the previous chapters. As such, in order to determine the optimal number of needed relays I've ran a community detection algorithm on the network G, by measuring its modularity. A community detection algorithm is a method for grouping individuals (nodes) into clusters in which all elements share one or more common properties. In this case, the commonly shared property is the position of each sensor: they have to be tightly grouped toghether and have a higher distance in relation to other nodes . A parameter named resolution can influence the number of detected communities. In comparison to the default resolution value of 1.0, a custom resolution < 1.0 will determine smaller/more communities and a resolution > 1.0 will determine larger/less communities. I'll discuss the impact of using a custom resolution in the next section.

Measuring the modularity of a realistic physical network (i.e. not regular, not evenly spread) results in a high number of communities with various sizes. As there are always small communities formed out of several stranded nodes, I ignore all communities with a total size smaller than a fraction λ of the total population, merging them with the closest large community.

BUPT

Algorithm 4.3 SIDeWISE Algorithm: First level community detection

Input: Sensor network represented as graph $G = \{N, R\}$ with geo-positional data of the nodes (n_i, n_j).

C: Detect communities

$Com \leftarrow$ community detection algorithm $\{G, resolution\}$
foreach com_i **in** Com:
  if $|com_i| < \lambda \times |N|$:
    $Com = Com \setminus \{com_i\}$

Output: Set of nodes, each with it's own community id

It is important to mention that discarding does not mean the sensors are removed from the network, it means that those groups of sensors will be considered irrelevant for the next step of the relay-placement algorithm.

Step D is an iterative process identical to **step B**, but it is applied on each individual community previously determined. The number of relays is determined by the number of relevant communities (i.e. $size > \lambda$ fraction of the population) during step C while the relays themselves are chosen during this step. Measuring the *centrality distribution* of each community, we choose the most central node as a relay. As mentioned before, the central node is the closest to all other nodes in its community. This is relevant to wireless sensors because the existing edges are determined by position, and so it becomes straightforward and efficient to choose a relay to whom any sensor requires the minimum number of hops to reach.

Consequently, the resulting set of relays R is composed out of the initial sink and one relay added per relevant community. More precise, the total number of relays is equal to the number of communities of $size > \lambda$ fraction of the total population, and contains the most central nodes in each described community.

Considering that now most wireless sensors have a relay in their vicinity, **step E** processes set R in order to create a secondary, overlapped graph of edges that connect all relays and the sink. The edges represent physical links, like broadband cable or fiber connections. Coverage of the set S with edges is done using Kruskal's minimum spanning tree (*MST*) algorithm [64].

The algorithm is applied on the set of relays, each with geographic coordinates, by iterating through all possible edges between all pairs of edges. The total number of possible edges between $|R| = r$ nodes

Algorithm 4.4 SIDeWISE Algorithm: Assign relays

Input: Sensor network represented as graph $G = \{N, R\}$ with geo-positional data of the nodes (n_i, n_j).

D: Assign relays

foreach community com_i **in** Com:
  foreach node n_{ik} **in** com_i:
    $C[n_{ik}]$←compute centrality $\{com_i\}$
    find n_{ir} in com_i **where** $C[n_{ir}] = 1.0$
    R = R $\{n_i r\}$
    n_{ir} →relay for com_i

Output: Set of relay nodes (R), each with it's own community id

Algorithm 4.5 SIDeWISE Algorithm: Create the MST of the relay network

Input: Realy network graph $R = \{N_R, \emptyset\}$ with geo-positional data of the relaynodes (n_i, n_j).

E: Create MST for relay-graph

$E_R \leftarrow \{\}$
foreach relay r_i in R:
  foreach relay r_j in R, $i \neq j$:
    e_{ij} ←create edge between (r_i, r_j) if none exists
    $E_R = E_R \bigcup \{e_{ij}\}$
$G_R\{R, KE_R\} \leftarrow Kruskal\{E_R\}$
$E_R = E_R \setminus KE_R$

Output: Relay interconnection tree, E_R

is $\frac{r\times(r-1)}{2}$. Kruskal's algorithm is applied on set E_R which results into the graph G_R composed out of the relays (as nodes) and $r-1$ edges (KE_R) with minimum cost. The cost of an edge is represented by its Euclidean distance, thus shorter edges are preferred over longer ones. Also, the set E_R of all possible edges, except the ones already selected by Kruskal's algorithm, is kept for the final optimization of the relay-graph.

The final step of the algorithm is **step F**. Although we have obtained an optimal set of relays for the wireless sensor network, as well as a minimum cost coverage network for the relays, the problem that arises is that such an MST is rarely an optimal throughput network and in the same time it has a low fault tollerance as I'm going to detail in the next chaper. Suffice to say for now that by loosing any relay node or any high-speed connection would partiion the network. As the network G_R has a sink, it is relevant that G_R itself is optimized in order for the sink to be the most central node, so that information incoming from all relays is gathered with minimum traffic congestion. The problem stated is explained in figure 4.4. It can be observed how a randomly chosen network covered with an MST resulting from step E (Kruskal) places the sink in an eccentric position. The smallest (gray) nodes represent wireless sensors, the red nodes represent relays and the single larger red node is the sink. The red edges are the physical links connecting the relays. On the right side of Figure 4.4 it can be observed how the sink is made more "central" by adding two more edges to the MST resulting from Kruskal's algorithm.

The SideWise algorithm can be summarized into the dataflow chart, presented in figure 4.3.

4.4 Simulation and results

In this section I'll exemplify the functionality of the SIDeWISE algorithm on realistic wireless sensor networks, as well as explain how the algorithm's various parameters affect the results. Figures 4.5 and **??** present the main steps of the algorithm. All sensor nodes are colored according to the community they belong to, as determined by the community detection algorithm implemented in Gephi. There are a total of 7 communities, as such, 7 relays are assigned to the whole network, one of them being the sink. Figure 4.5 displays the resulting MST (up to step E). As it is a minimum spanning tree, there are no redundant physical links which connect the sink to any distant relays. Although cost effective, if we were to add an additional edge to the MST, the congestion rate will decrease significantly, while the cost will only increase by a small amount (step F). In figure 4.4b, the first edge added

BUPT

Algorithm 4.6 SIDeWISE Algorithm: Centralize the *sink* node

Input: Realy network graph $R = \{N_R, E_R\}$ with geo-positional data of the relay nodes (n_i, n_j).

F: Centralize the sink $(C[Sink] = 1.0)$

while $C(sink) \neq 1$:
  foreach edge e_i in E_R:
    $KE_R = KE_R \bigcup e_i$
    $C(sink) \leftarrow$compute centrality$\{G_R, sink\}$
    fitness(e_i) $\leftarrow$ $C(sink)$
    $KE_R = KE_R \setminus e_i$
  find e_r in E_R where fitness(e_r) is maximal
  $KE_R = KE_R \bigcup e_r$
  $E_R = E_R \setminus e_r$
  $C(sink)$ $\leftarrow$fitness(e_r)

Output: Realy interconection graph with maximized sink centrality (1.0)

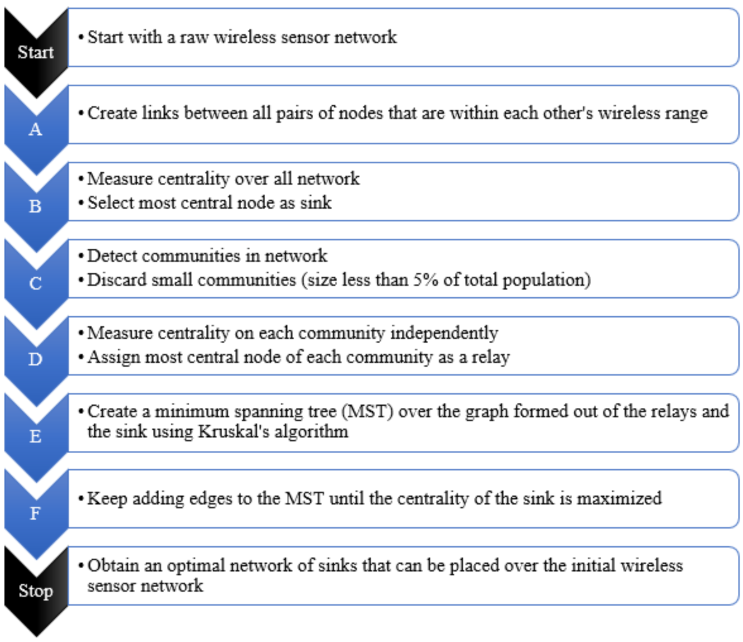
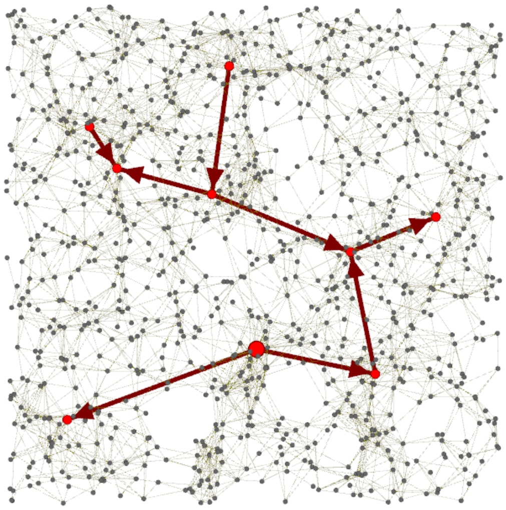
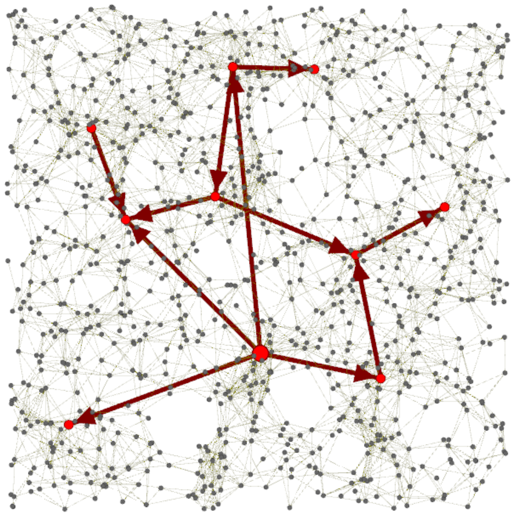


Figure 4.3: The network processing workflow of the SIDeWISe



(a) A relay network covered with a MST obtained by SIDeWISe



(b) The same MST from subfigure (a) but with an additional two edges so that the sink (big red node) becomes the central node of the network.

Figure 4.4: The results of running the SIDeWISe algorithm on a typical random sensor network

during step F of the algorithm is E1: it connects the sink with relay 4 and decreases the average path length towards the sink. Another edge is added, namely E2, so that the distance from the sink to relay 6 is reduced from 3 hops to one.

The resulting topology described above depends on two more parameters: the *resolution* of the modularity algorithm and the *wireless coverage radius*. I'll further analyze the impact of these two parameters in regard to the number of assigned relays.

4.4.1 The Number of Relays versus the Community Granularity

First I'll analyze the total number of relays as a function of the modularity resolution used to detect the communities. The scale of the modularity resolution depends on the degree of connectivity between each community and does not depend on the network structure itself [84]. Lowering or increasing the resolution is equivalent to finding an ideal tradeoff between the number of terms in a sum and the value of each individual term [67]. The experimental results depicted in Figure 4.6 show that as the resolution increases from the default value (i.e. 1.0 in Gephi's implementation of the algorithm) the total num-

BUPT

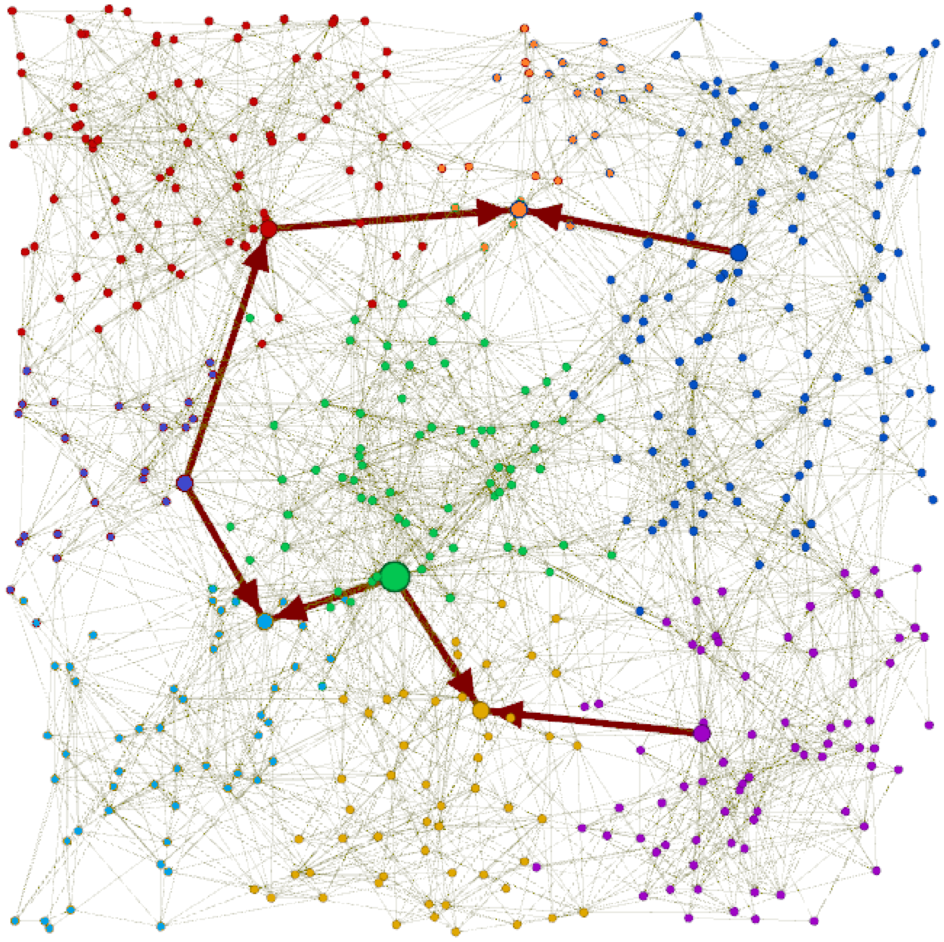


Figure 4.5: A sensor network of 1000 nodes with 7 communities, each with one assigned relay. The larger green node is the sink. Even though the coverage cost is optimal, the average path length to the sink remains high and the fault-tollerance is still low.

BUPT

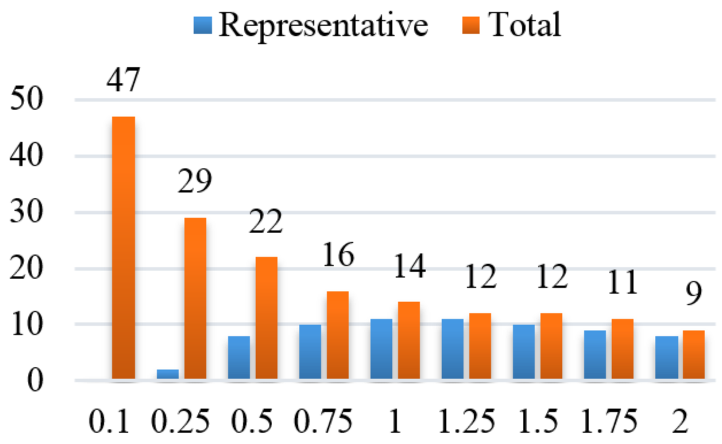


Figure 4.6: A sensor network of 1000 nodes with 14 detected communities, out of which 11 are representative (resolution = 1.0) showing the ration between total and relavant number of communities, depending on the resolution parameter.

ber of detected communities decreases slowly and converges towards a minimal value. In the chosen example, the 500 node network can be divided in no less than 9 communities, regardless of how much the resolution is increased. As the resolution is decreased from 1 towards 0, the number of detected communities increases exponentially, but their actual size decreases steadily. Because the SIDeWISE algorithm neglects communities smaller than 5% of the total population — imposed treshold for this example, but changable if needed — the number of relevant communities (blue) decreases to a state in which not a single community is considered relevant to have its own relay. Table 4.1 shows the numerical values displayed in Figure 4.6. Figure 4.7 demonstrates the fact that by increasing the resolution we obtain fewer but more significant communities, as the ratio between relevant and total converges towards 90%.

As a conclusion to this experiment, I consider that applying the SIDeWISE algorithm in every real scenario requires understanding the impact a custom set resolution has on the overall structure of relays. If the real world conditions require sparse relay placement, one can consider lowering the resolution, so that only the most truly relevant communities of sensors receive a relay in their vicinity. On the other hand, if we require a higher speed network of relays one can increase the resolution, but, as the experiments show, even though the ratio increases up to 90%, still the highest number of relays (11) is placed when the modularity is set to the optimal value of 1. Based on these observation I suggest suggest working only with resolutions between

BUPT

Table 4.1: Experimental results showing the number of relevant communities versus total number of communities in a wireless sensors network as the modularity resolution is decreased and increased from the default value of 1.

	Communities		
Resolution	**Relevant**	**Total**	**%**
0.1	0	47	0
0.25	2	29	6
0.5	8	22	36
0.75	10	16	62
1	**11**	**14**	**78**
1.25	10	12	83
1.5	10	11	90
1.75	10	10	90
2	9	9	88
>2	8	9	88

0.5 and 1.25.

4.4.2 The Number of Relays versus the Wireless Coverage Area

The second analytical approach emphasizes the total number of relays assigned as a function of the communication range of each sensor. For analytical reasons at this stage I consider that all sensors in the network have the same wireless coverage radius, thus the length of all edges in G are smaller than a given radius *r*. Similar to the modularity resolution, a small radius will determine short, local edges in the graph which leads to many small communities. Figure 4.8 represents experiments done on a standard wireless sensor network of 1000 nodes by varying the radius *r* from 5% (0.05) to 50% (0.5) of the total area covered by the network (legth of the bounding box).

While the number of detected communities falls exponentially from over 250 towards a small convergence value of 4, the number of placed relays describes a different characteristic. Small communities lead to no relays being placed, then, as *r* increases, an optimal relay layout is obtained for a coverage $r \sim 0.1$, and finally, the relay number falls again as the coverage area widens (> 0.2). The conclusion that can be drawn is that the SIDeWISe algorithm offers less improvement for sensor networks with a low area coverage, and becomes redundant if the coverage area is significant compared to the whole network surface. The optimal scenario in which to apply the SIDeWISe algorithm is in a

BUPT

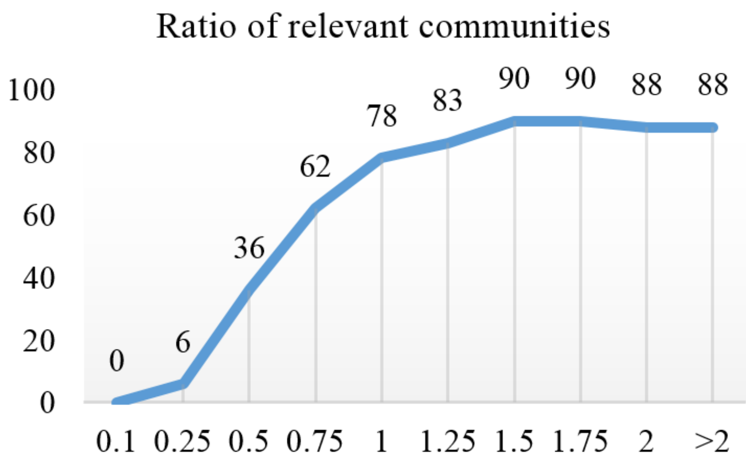


Figure 4.7: The ratio between relevant and total detected communities on a network with 500 nodes as the resolution is increased from 0.1 to values above 2. A low resolution yields poor results (<50% relevance) and high resolutions all converge towards the same result (>80% relevance).

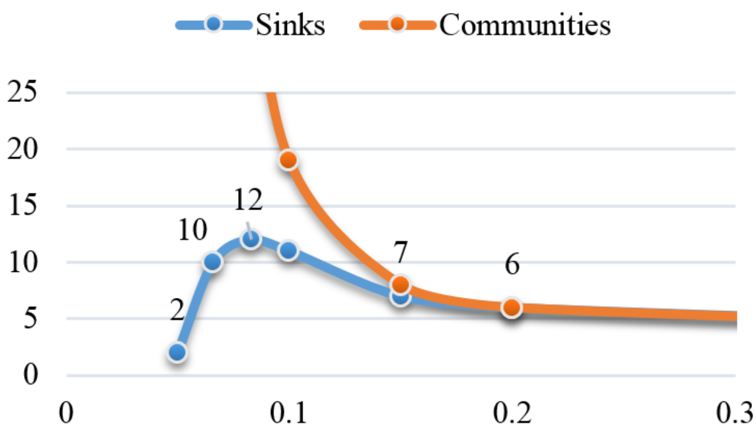


Figure 4.8: The impact of the wireless radius of sensors in relationship to the total number of communities detected (orange) and the actual number of relays assigned by the algorithm (blue). The figure is zoomed-in on the relevant portion of the graphics. The radius on the Ox axis is expressed as a percentage of the total area covered by the network.

BUPT

Table 4.2: Experimental results showing the number of relays placed by the SIDeWISE algorithm for a network with N nodes.

N	100	200	300	400	500	600	700	800	900
Relays	2	2	3	4	5	6	6	6	7

N	1000	2000	3000	4000	5000	6000	7000	8000
Relays	7	9	9	10	10	9	10	11

network in which the ratio between a sensor's coverage area and the whole area occupied by the network is between $6 - 20$%. However, by modifying the community size threshold of 5% to a different value, the algorithm can be adapted to more diverse scenarios.

4.4.3 The Number of Relays versus a Growing Network Size

I'll discuss in this section the total number of relays as a function of the network size and demonstrate that while the network increases linearly in size, the number of relays increases much slower, like a scale-free social network, namely logarithmic [1]. Table 4.2 contains measurements of number of assigned relays depending on the network size N. It is worth mentioning that for this discussion both the resolution and wireless coverage are kept at constant values.

The measurements confirm the fact that the number of required relays does not grow linearly with the number of nodes, but logarithmically. My algorithm has the same property as the small-world network described by Watts-Strogatz [105]. In Figure 4.9 we can observe the growth of the number of relays (N), with values of N ranging from 100 to 1000 with a step of 100, and from 1000 to 8000 with a step of 1000. The orange trend line $log(relays(N))$ demonstrates the mentioned logarithmic characteristic. This is a very important feature of the SIDeWISE algorithm because it manages to keep the number of relays relatively low, thus the cost remains low, as the overall network propagation delay is rapidly decreased. Making an analogy with the small-world properties which represent an ideal balance between the characteristics of a regular network and a random network [111], Figure 4.10 demonstrates the same principle: *the socially enhanced wireless sensor networks lie at the ideal crossroads between cost and performance*.

On the left side of Figure 4.10 is a network with just one sink and

BUPT

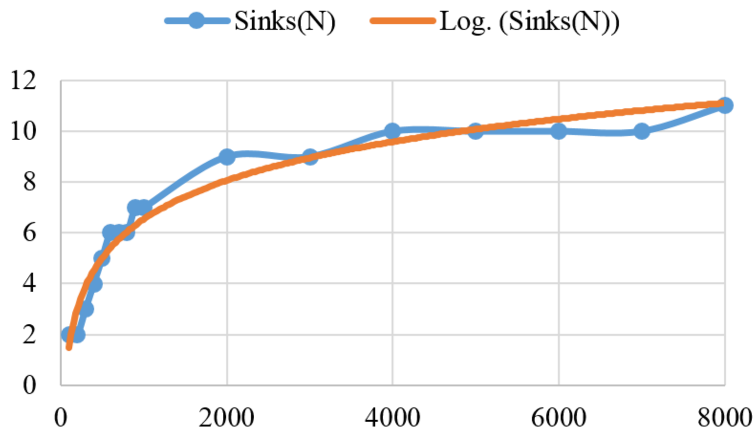


Figure 4.9: The number of assigned relays (OY axis) as a function of the network size N (OX axis). The function relays(N) has a logarithmic characteristic as the trend line demonstrates (orange).

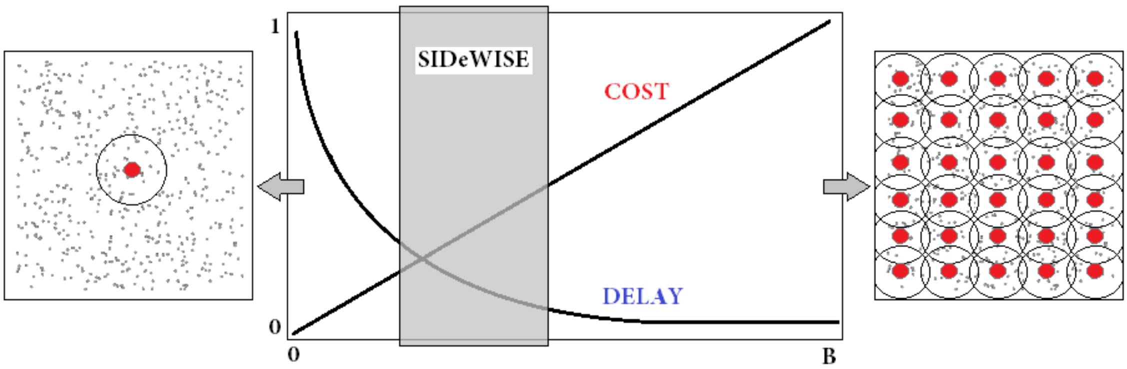


Figure 4.10: The SIDeWISE algorithm balances cost and propagation delay by optimizing the placement of the relays in a WSN. The two extreme cases are represented by a single-sink network (left) and a network fully covered by relays (right).

no relays. While being cost-optimal, it offers the worst performance as the propagation delay is maximal. On the right side is a network fully covered by relays. In this case the delay is optimal (minimum) but the cost is maximized. As the graphics of the delay and cost suggest in figure 4.10, there is a window in which we can create a network with the best possible tradeoffs: a relatively low delay (i.e. high performance) and a low cost. This is the type of enhancement which the SIDeWISE algorithm facilitates.

To exemplify the advantages, I consider a network A with 10×10 adjacent relays placed in a lattice topology like in figure 4.10 (right) and also that a wireless communication hop is equivalent to a delay of 1τ time unit and that all physical hops are equal to 0τ units of time. The cost of a network is expressed by the number of required relays. In this case, the delay on network A is 1τ and the cost is 100 (expressed in number of relays). On the other extreme is network B of the same size but with only a sink in the center. This sink only covers 1 cell out of the 100 cells in the network so the average delay for any sensor is 6.98. The cost however is equal to 1. None of the two extreme solutions – A or B – are ideal as the proposed algorithm covers the network by assigning only 7 relays while keeping the average delay at 3.62τ time units. *This yields a 92% performance improvement compared to network B and uses only 7% of the relays required for network A.*

4.5 Hierachical SIDeWISe

At this stage I propose a methodology centered around topics from the CNA, presented above which is an extension of the SIDeWISE algorithm. The first stage is represented by the recursive division of the city into topologically relevant communities. These communities contain key intersections, identified through computing the betweeness centrality. The second stage is represented by the recursive multi-level breakdown of the intially identified communities into smaller cluster and this process cand continue until a desired granularity (community size) is achiverd. The third step is represented by the bottom-up reconstruction of the relay tree structure, with an aditional optimization step of building the associated higher fault-tollerance network.

4.5.1 Issues with flat topology networks

In figure 4.11, a simple sensor network consisting of 3 communities on a two-dimensional structure is presented. A selection mechanism must be run to identify groups of master-slave nodes in order to advance from Layer 1 to Layer 2 and to change our approach from computing

BUPT

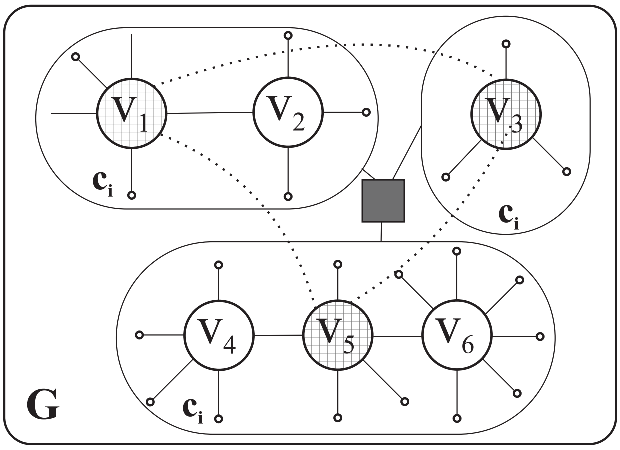


Figure 4.11: Hierarchical structuring of intelligently managed traffic light intersections. Taken from [50]

a global optimum to an adaptive mechanism (master nodes being depicted as grayed-out V_1, V_3 and V_5). The nodes identified as master will coordinate at Layer 2 all traffic movements from Layer 1 and will create, at the same time, the population for Layer 3. At the upper layer each of the communities are clustered from a logical point of view exchanging information between equipotent master nodes of distinct communities (square grey node is a logical one).

Running the algorithm at different resolutions identifies communities along with the corresponding master-slave nodes. Each of the grey-ed nodes act as relays in the entire network. Each community in figure 4.11 is associated to a topological community identified by the community identification algorithm. All other nodes, V_2, V_4 and V_8 act as lower level communities wich are going to be recursively broke-down. Inter-community communication occurs between relay nodes of each of the communities.

4.5.2 Hierachical SIDeWISe algorithm

In this section I'll present an extension of the two-tiered relay network obtained by the SIDeWISe algorithm for a multi-level structure.

Let $G = \{E, V\}$ be the graph representing the topology of the network, where $E = \{e_i \mid i = \overline{1, n_e}\}$ is the set of all edges (wireless connections), with n_e being equivalent to the number of radio links and $V = \{v_i \mid i = \overline{1, n_v}\}$ is the set of vertices, with n_v being the number of sensor nodes. The hierachical SIDeWISe (h-SIDeWISe) algorithm uses two parameters, the resolution RES (discussed previously) and the threshold value for stopping the recursion $TRESH$ (chosen

BUPT

Algorithm 4.7 The hierachical-SIDeWISe algorithm

```
input: G = E,V
define RES = 10 //see more details in Section 3.5.2
define THRESH = log_10 | G |
function assignMaster(G, RES, TRESH) {
        C=detectCommunity(G, RES)
        foreach c_i in C do {
              foreach v_i in c_i do {
                   betweness[i] ← computeBetweenes(c_i, v_i)
              }
              maxBetweenes ← max(betweenes[])
              v_master ← c_i[maxBetweenes]
              if (| c_i |> TRESH)
                    assignMaster(c_i, RES, TRESH)
         }
}
```

by the practitioner on experimental basis). The recursion parameter $TRESH$ determines the number of nodes in a community, the larger the $TRESH$, the larger the community. The recursive process is implemented by function `assignMaster(G,RES,TRESH)`. I apply the community detection algorithm as presented in [14]. Subsequently, for each of the detected communities ($c_i \in \mathcal{C}$) I compute the *betweeness centrality* for each node v_i. Next, the nodes are ranked and the master node is selected based on the *maximum value of the betweeness*. The hierarchical nature of the algorithm is implemented as a recursive process, consequently I apply the same methodology for each of the "smaller" communities of $c_i$until the number of nodes in a community is less than the specified threshold value ($TRESH$).

Figure 4.12 shows the behavior of the h-SIDeWISe algorithm on a synthetically generated random mesh network with triangle tessellation. The blue nodes represent ordinary sensing/actuating nodes, while the red one is the central sink — identified by using the SIDeWISe algorithm. The networks was subsequently divided into communities and each of the first level-communities are delineated by a differently dotted polygon. Next, the hierarchical way of approaching the problem requires the continuation of the recursive breakdown, which is represented by the magenta sub-communities shown in the figure. For the sake of visual clarity I've stopped the process after two iterations but for larger network it should be continued until the desired level of granularity is reached.

BUPT

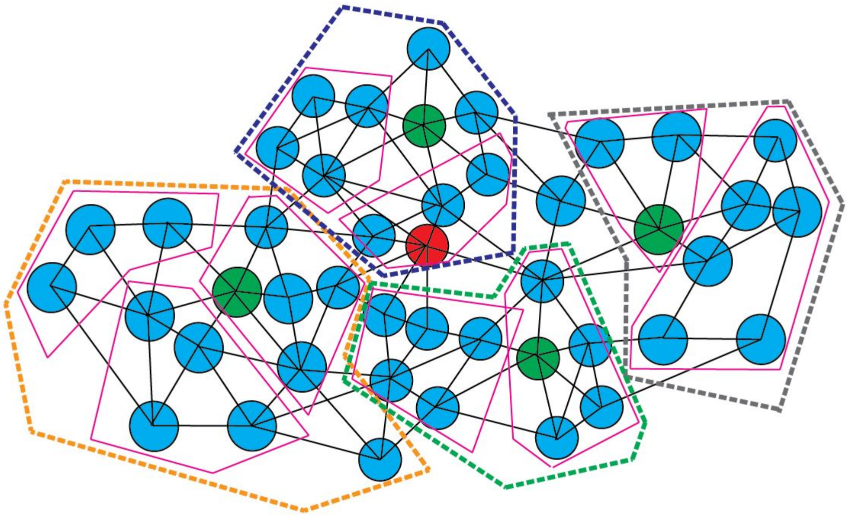


Figure 4.12: Application of the h-SIDeWISe algorithm on a random mesh network

4.6 Conclusions regarding the SIDeWISe algorithm

My work represents a novel approach in designing the placement of relay nodes in a sensor network. By using concepts from the area of social network analysis and mapping them to the already classical field of sensor networks I've succeed to add improvements to the costs implied with deploying the infrastructure. My research in done around the algorithm I've devised, called SIDeWISE.

A particular assumption is represented by the multi-tiered relaying architecture with relays linked by low-latency cable connections, which I've have stated in section 1.2. I consider my research as a framework for a much in depth analysis involving detailed physical characteristics of the network, buy the analysis is spanning in two major directions defined by the two varying parameters: the community granularity, which provides in insight into the smallest possible cluster of nodes and the already classical topic of the wireless coverage area of each node. The logarithmic behavior of SIDeWISE is of particular importance for the demanding applications of modern day sensor network with large number of nodes and with assumptions of a growing trend. Taking two classical reference examples (single central relay/sink and a regular mesh of sinks) I have shown the location of my algorithm in the design space. In regard to the mesh placement SIDeWISE provides an improvement of 92%. The extension of the initial algorithm is concern-

BUPT

ing the idea of a multi leveled topology in which there is a tree style hierarchy of relay nodes, each datagram traversing the hierarchy for reaching the diametrical extremities of the network. This is formalized into another algorithm called h-SIDeWISe and uses the same concepts of complex networks analysis further adapted for the required task. The multi-level structure proposed has the advantage of exploiting the localization of data traffic on geographical basis and providing means of interconnection between clusters only trough relays, while the lower importance radio connections which can arise from the mesh structure are used for improving the fault-tolerance.

I've also tackled the problem of finding a methodology of choosing the resolution parameter for my experiments and form this point of view I've presented a procedure based on experimental findings for determining the best value mostly dependent on the specific scenario at hand. The h-SIDeWISe uses this findings in order to efficiently identify the important nodes of the a sensor network and assign sink roles to them, afterward using a genetic optimization approach.

BUPT

5 Building fault tolerant sensor networks using SIDeWISe

Mostly dependent on the task of the network but also of the particular conditions and the type of sensors there are two major strategies in placing the nodes of a sensor net-work: deterministic and random. The first one, when possible, can ensure great cover-age with careful placement of the nodes and even the logical topology of the network can be established at deployment time [77].

Because of the adverse condition on the field there are situations where the single possible option for deploying nodes is in a random manner. This has adverse effects on the main metrics of a WSN [72]. In any situation where there is a large distance between two adjacent nodes, we witness a low throughput and high energy consumption.

Rich literature exists on the topic of optimal node placement [55], which is considered an NP-hard problem [10] and some non-deterministic approaches were proposed, which provide sub-optimal results [89].

Much because the current approaches in deterministic placement of the nodes proven themselves problematic but also because some of the typical WSN deployment scenarios presented both in the literature and also in the real life scenarios, such as wild fire prevention, battlefield monitoring or disaster rescue, require a quasi random distribution of the nodes, even if there are some possibility of controlling the density of the nodes [108] I'we decided to investigate the problem of relay placement strategies in this case.

Another interesting approach, which is also the starting point of my investigations, is the one presented by Xu et al. in [108]. The authors take into consideration a two-tiered topology in which nodes are clustered around relaying nodes which further communicate directly with the sink. The authors also consider a "multiple-hop communication case" which presume the existence of a hierarchy of relay nodes, connected in a tree manner to the sink. The authors propose a weighted random distribution, which increases the number of nodes as we move further away from the sink. One of the issues identified by the authors is that the random distribution can leave some parts of the network disjoint, actually partitioning the network, and their solution to the problem consists of the multihop deployment strategy.

Another research direction with practical application is the exploration

BUPT

of the issues arising when scaling the network. From an economical perspective, much of the sensor network deployments consist of incremental stages with more nodes being added. Aside from the issues regarding possible flow congestions there is a great need to know the optimal placement of the relay nodes so that, with minimal costs, the new nodes are is a growing interest in the area of network connectivity [61]. Early work has considered the coverage as being the paramount of the research [110], but because modern sensor node are running at the threshold of the energy requirements their coverage is largely diminished. Flat, 2D sensor networks usually consider relay nodes to be simply another node, but with higher transmission power and/or energy autonomy. The problem is getting interesting in two and multitier sensor networks where sensors are usually clustered in what can be called subnets, as presented by Chen et al. in [22]. Each subnet is sending data to a relay (called aggregation-and-forwarding node (AFN)) which in turn send the data via a multihop connection to the base station (sink).

One other approach is to use a deterministic approach in placing the relay nodes of a randomly deployed sensor network, such as presented in [22]. The problem is formulated in terms of initial set of nodes and their position, the task being to find the optimal placement of a set of relay nodes, so that the network lifetime and connectivity are maximized. Authors prove that the problem is still NP-hard, but provide a polynomial time algorithm. Tackling the problem of fault tolerance, there is an approach of maximizing reliability by placing relays at the intersection of two neighboring nodes.

5.1 Reliability in sensor networks

Since the beginning of the 1950's there was an intense research and development in the field of reliability of computers, communications and storage systems. These developments were conducted both in the academia and the industry. One of the main driving forces behind was the recognition of the fact that as the complexity of computing devices and systems increases, fault-tolerance will gain more importance. At the dawn of the 9th. decade of the past century the objective of the fault tolerance has lost of importance. This situation arises form the progress in the manufacturing area, where we could obtain individual components (pieces) which have a very good reliability factor so that the final system will have also at least a decent reliability. One aspect worth to mention is the more and more innovative packaging and new cooling mechanisms which tremendously reduced the stress factor om computation systems. The main step in the fault-tolerance testing

BUPT

procedure that has received a great deal of attention was the on-line testing, especially in the industry.

In the last few years we see a revival of the interest in fault tolerance and related techniques such as *self-repair*. One of the main driving forces behind this state of fact was the rapid growth of the Internet in the last decades. Internet requires a very high availability (called also up-time) and therefore much interest was involved in the development of fault-tolerant data-centers and associated data networks. Another field of computer engineering which benefits from an increased fault tolerance are the wireless sensor networks. WSN rise some unique conceptual and technological challenges. We know that at least two components of a sensor node will directly interact with the environment. These are the *sensors* and the *actuators*. Interacting with the environment they will be under constant and various physical, chemical and biological stress. Therefore they have a significantly lower individual reliability compared to the classical integrated circuits in fully enclosed packages. Also we should not forget that each node and the entire wireless sensor networks is an exceptionally complex system where a large number of not always homogenous nodes interact in a complex manner. Even a bigger challenge would be represented by the shear number a the nodes that can be part of a WSN, and we speak of hundreds and maybe thousands of them. All these form a large distributed embedded network system that handle a variety of sensing, actuating, communicating, signal processing, computation and communication tasks. The development of WSN is strongly linked with our capability of building low cost devices which directly influences their reliability.

Another reason for the importance of fault-tolerance in respect to WSN is that applications operate most of the time in an autonomous mode without human presence and/or interaction. One of the biggest aspects is therefore the one of *security*, *safety* and so-called *graceful degradation*. The main concern as we speak of graceful degradation is the impact on humans and the environment in case of an error, in particular when we have a loop which involve actuators. The area of debugging will rise also significant challenges mostly because of the almost impossible task of replicating the conditions and here we speak of the condition of the entire sensor network as a whole where we have to take into consideration all the aspects: geography, climate, weather, stress factors, power supply and so on, compared to the classical debugging where we regard the device as individual element (or a limited number of well known interconnected devices).

The final reason I will present here is that WSN are still a young topic of interest - especially in regard to an even newer concept called Internet of Things - field and there is not very clear how we can address a

BUPT

particular problem, so we have a large playground for research, development and testing. Even worst WSN are used in different context and environments and purposes so it is very difficult to find the best way to treat the fault-tolerance which can be applied on any of the scenarios.

5.2 Sensor networks

In the following I will try to make a surrey in the field of fault-tolerance in sensor networks and according to [37] I will consider FT at *four levels of abstraction* starting from the hardware and system software and going to the middleware and application layers. Also I will consider FT at each level of six individual components of a node: *computing*, *communication*, *storage*, *energy supply*, *sensors* and *actuators*. Also I shall speak on FT regarding an individual node as well as the network itself.

A WSN can be seen as a system composed of small wirelessly communicating nodes, where each node have some well defined but not necessarily unique function. In particular each node has a computation and communication part a power supply element and most of them have some sensing devices and less of them have actuating (influencing the environment) elements.

A WSN can be seen as a symbiosis of physical world with the Internet and computations. One of the biggest concerns in WSN is the *power supply*. On a node the power supply is very limited and the replacement of batteries is impractical or impossible because of the topological conditions or simply because of the shear size of the network. Therefore we see that energy is the most constraining factor in the operations of a WSN. The mainstream technique for conserving energy is the short-range communication between adjacent nodes, instead of a direct long-range link with the central access node [99].

From a theoretical point of view sensor networks are often modeled as as graphs where each vertex of the graph corresponds to a wireless node and there is an edge corresponding to the communication between two nodes [75]. The communication between nodes can be either 1-to-1 or 1-to-N. This is a very simplified model and providing a reasonable and practical model for sensors and actuators is a much more complex undertaking, mostly because of the large variety of functionalists and underlying technologies involved.

From the application point of view there are a lot of envisioned applications for WSN. One of the classical one is in the field of military science where they can be used to detect and spy the enemy territory providing valuable information for the deploying forces. Another field of application is the intelligent security systems in perimeter defense.

BUPT

A similar field is the monitoring of large areas of inhibited environment for detecting dangerous situations (eg. volcanoes, landslides, . . .) or contexts which require observation over a long period of time.

Later in my thesis I will develop and analyze a specific application area, of real time monitoring of urban road infrastructure in order to optimize the traffic flow in city environments.

5.3 Background in fault tolerance

In the dawn of design of digital computing systems one of the main concerns was the fault-tolerance. The relay-based computer built at Bell-Labs was exploiting so-called temporal redundancy by performing the same calculation multiple time, using exactly the same inputs and the same algorithm, and by comparing the results they were able to detect potential transitory malfunctioning. Also UNIVAC 1, built in 1951 used both *parity checking* and *arithmetic unit replication* to enhance reliability. In that time both Moore&Shannon and VonNeumann conducted studies on how to design systems that preserve functionality after a subset of components manifest failure. One of the key development directions was the enhanced serviceability features, manifested by a very *good modularity* which allowed rapid replacement of the faulty module. On of the extreme cases was the Apollo program which used triplicated computers (units) in order to lower the probability of failure [6].

In the following decades the fault-tolerance started to diverge between hardware (VLSI design) and software (especially in the database area) and now one of the hottest topic is the self-repairness of hardware units. The reliable design is discussed at three stages of the product life cycle: *design*, *manufacturing* and *usage*. Before continuing I will present some common language for the field of fault-tolerance, according to [7].

Fault is an incorrect state of hardware or software as a consequence of the failure of a component. *Permanent faults* are the ones that are continuous and stable in time. For example, permanent hardware faults are consequences of irreversible physical alteration within a component. An *intermittent fault* is one that has only occasional manifestation due, for example, to unstable characteristic of the hardware, or as a consequence of a program being in a particular subset of space. Finally, a *transient fault* is one that is the consequence of temporary environmental impact on otherwise sound hardware. One of the most interesting causes of transient fault is the impact of cosmic radiation [94].

Error is the manifestation of a fault inside a component/module. One

BUPT

of the most important aspects is that error can occur not only at the fault site but also at some distance form the fault (both in space and in time).

Fault-tolerance takes into consideration three types of concerns: fault models, fault detection and resiliency mechanisms [62].

Each level of abstraction has its own types of faults. One of the most common examples is the *stuck-at* model that was used with success in the physical testing phase at the gate level. This model supposes that the value on the input (or output) of a elementary gate is always stable (1 or 0). Another classical model is the *bridging fault* model where two or more neighboring signal lines are physically connected making a wired AND (or OR). *Shorts* and *opens* are another class of faults corresponding to missing or additionally introduced connections respectively [7]. It is interesting to notice that most of the testing approaches assume a single fault model. This is based on statistical analysis where was proven that the single-fault model is most likely to occur in practice and the double (or multiple) fault is extremely rare [71].

In the field of reliability we have the following phases.

1. Firstly we have to confine the fault, so to limits it's effects in a particular area and therefore the contamination of adjacent area is prevented;

2. Fault detection, independently from confinement, is the acceptance of the existence of a fault.

3. Fault latency is the time lag that is between the actual (physical) appearance of the fault and it's detection.

Usually fault detection techniques are classified in: *online* (during the production status of the device) and *offline* (when the device it's not in production) [54]. Most often we use a superposition of these two so that we have a detection phase which is online and a diagnostic phase which is ofline, using specialized tools. We can introduce another phase called reconfiguration, where we act on the device so that the fault is eliminated or, more generally speaking, the manifestation of the fault does not have impact on the correct output of the device. *Graceful degradation* is a reconfiguration techniqus where the performance of the system is reduced but the correct functionality is preserved. *Recovery* is the stage where an attempt is made to eliminate the effects of the faults. *Repair* is the stage where the failed component is substituted with another component which is operational [7].

BUPT

5.4 Fault tolerance metrics

Because fault tolerance is about making machines more dependable, it is important to have proper measures (metrics) by which to judge such dependability. In this section, I will examine some of these metrics and their application. "A *measure* is a mathematical abstraction that expresses some relevant facet of the performance of its object" [62]. By its nature, a measure only captures some subset of the properties of an object. The idea is to define a suitable measure so to keep this subset large enough so that behaviors of interest to the user are captured, and yet not so large that the measure loses focus [54].

5.4.1 Traditional metrics

We I describe the traditional measures of dependability of a single computer - in our case we speak about a node. These metrics measure very basic attributes of the system. Two of these measures are *reliability* and *availability*.

The conventional definition of *reliability*, denoted by $R(t)$, is the probability (as a function of the time t) that the system has been up continuously in the time interval $[0;t]$. This measure is suitable for applications in which even a momentary disruption can prove costly [7].

Two commonly used metrics are also the Mean Time to Failure (MTTF) and Mean Time Between Failures (MTBF). MTTF is defined as "the average time the system operates until a failure occurs" and MTBF is "the average time between two consecutive failures". There is a clear distinction between the two because of the time needed to repair the system after the first failure, which defines the Mean Time to Repair (MTTR) [7].

$$MTBF = MTTF + MTTR \quad (5.1)$$

Another frequently used metric is availability, denoted as $A(t)$ and which is the "average fraction of time over the interval $[0,t]$ of which the system is working properly". It is mostly used in applications where continuous functioning is not vital but it is not desirable to have the system down for a significant amount of time. A typical example is the residential Internet data connections: having small service disruptions is not catastrophically for the users but when this repeats numerous times it can lead to loosing customers. The concept of long term availability is defined as:

$$A = \lim_{t \to \infty} A(t) \quad (5.2)$$

BUPT

It can be interpreted as "the probability that the system will be up at some random point in time", and is meaningful only in systems that include repair of faulty components [3]. Using "discrete time", we can derive the long-term availability from MTTF, MTBF, and MTTR as:

$$A = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF + MTTR} \tag{5.3}$$

These definitions assume, of course, that we have a state in which the system can be said to be up and another in which it is not. For example a wire is either connected or has a break in it. This analysis can be hard to make on any system of the complexity we are dealing when speaking of computer or as in my case specific emebedded systems such as WSN. Any modern processor has tens millions of gates which are in various states during execution of code. For example having a single faulty gate in the divide unit can lead to a wrong quotient when performing a computation which itself maybe if executed one in every few thousand of hours of operation. In this case it is obvious the processor is not "fault-free" but it is equally hard to say that is "down".

A similar discussion but with an even greater deal of uncertainty can be made for the systems that exhibit graceful degradation traversing a wider range of levels of functionality from a perfect state when everything is running fine and until the completely broken system. Definitely after a certain point the system fails to perform even the most basic tasks and we can consider it "down" but until then it is in various "up" states. When are we going to consider it broken and until when it is "up"? In this case metrics like quality of service (QoS) are used to describe in a more fine grained manner the behavior of the system.

5.5 Fault tolerance of networked systems

Before delving in the specic problems of WSN I will try to present some of the general problems and specic fault-tolerance metrics regarding the networks in general. One of the main reasons is that at the present moment we have more research in the field of FT on general purpose networks than on WSN. For this part I have related extensively the work of Israel Koren [62].

Interconnection networks are widely used today. The simplest example is a network connecting a number of processors (typically with their own local memory) in a distributed system, allowing the processors to communicate through messages while executing parts of a common application. In this case processors and memories are linked by a collection of links and switching equipemnt, where a switchbox allows any

BUPT

component to communicate with several other ones without neding a dedicated link to each of them.

Another type of networks are the so-called wide-area networks (WANs), related mostly to computers and which connect a large numbers of computers that operate independently - running different and unrelated applications - and allows them to share various types of information. The term *packet* is often used instead of message (a message may consist of several packets, each traversing the network independently) when speaking of computer networks, and in this case the routers take care of the burden of swithcing at a logical level. The best known example and in the same time the wideest of this kind of network is the Internet.

Links and switchboxes (routers and relay nodes in computer networks) establish *one or more paths between the sender of the message (the source) and its receiver (the destination)*. These links can be either unidirectional or bidirectional from booth physical and logical point of view. The topology of the network may assure a single path between a pair of nodes an in this case any fault of that specifc link would disconect the nodes. Fault tolerance in networks is thus achieved at the simplest level by having multiple paths connecting source to destination, and/or spare units that can be switched in to replace the failed units [54]. High availability and fault tollerance is achieved providing multiple paths for some or all source destination pairs, and there is a need to evaluate the resilience to faults provided by such redundancy, as well as the degradation in the network operation as faults accumulate.

5.5.1 Metrics of network resilience

To quantify the resilience of a network or its degradation in the presence of node and link failures, we need measures, several of which are presented in this section. We start with generic, graph-theoretical measures and then list several measures specific to fault tolerance.

One of the most important tools in the analysis of networks is the graph theory [97].

Representing the network as a graph, with sensor and actuating devices as nodes and links of various types (wireless or wired and having different properties) as edges with associated weights, we can apply resilience measures used in graph theory. Two such measures are:

- **Node and Link Connectivity**. The simplest assessment of the health status of network exposed to faults is whether the network as a whole is still connected in spite of local failures. The node (link) connectivity of a graph is defined as the minimum number

BUPT

of nodes and/or links have to be removed (affected by failure) from the graph (network) in order to disconnect/partition it. We assume that when a node is removed, all associated links on it are also removed. Consequently, **the higher the connectivity, the more resilient the network is to faults**.

- **Diameter Stability** is another metric of network resilience. The distance between a source and a destination node in a network can be defined as the smallest number of links that must be traversed in order to transport a message from the source to the destination. The diameter of a network is the longest distance between any two nodes. Networks can have multiple routes for every pair of nodes but anyway an impact can be observed in this case over the diameter. *Diameter stability* focuses on how the diameter changes as nodes fail in the network. A deterministic instance of such a measure is the *persistence*, which is "the smallest number of nodes that must fail in order for the diameter to increase" [97]. For example, the persistence of a cycle graph is 1: the failure of just one node causes a cycle of n nodes to become a path of $n = 1$ nodes, and the diameter jumps to n=2.

5.5.2 Common topologies and their resiliance

I will present in this section examples of two types of network.

The first type connects a set of input nodes (e.g., sensors) to a set of output nodes (e.g., actuators) through a network composed only of switchboxes (distiant routing) and links (wired or wireless). As examples for this type, i will refer a crossbar network and use the metrics of resiliance to be bandwidth and conectivity. The second type of networks I will discuss are the ones of computing nodes that are interconnected through links. In this case there is no other sitching equipment inbetween (routers) but the nodes are equaly potent and all of them have the came computation possibilities, being capable of booth executing their own function and manage the packet flow in their vecinity. The typical networks used are the *mesh* and the *hypercube*, and the applicable measures for these networks are the *reliability/path reliability* or the *availability*, if repair is considered [62].

I have chosen this two types because the can easely mapped to the corresponding two majos categories of wireless sensor networks. The first type can easely aproximate the situation when we have a WSN with nodes that function as relays and beacons and the second type models a pure wireless mesh when each node is booth equiped with booth sensing and communication facilities, so that we don't need relays. The relays are modeled by switchboxes and the sensing nodes

BUPT

Figure 5.1: switch controlling the routing of connections in a multi-level network. Taken from [62]

by processors. Also we can approximate the central node by the corresponding memories.

Multistage networks — which I developed by running h-SIDeWISe and obtaining the hierachical tree-like structure — are commonly used to connect a set of input nodes to a set of output nodes through either unidirectional or bidirectional links. These networks are typically built out of 2×2 switching devices. These are switches that have two in- puts and two outputs each, and can be in any of the four settings depicted in Figure 5.1:

Another widely referenced topology of multistage network is the *butterfly* [54]. A three-level butterfly is capable of connecting 8 inputs to 8 outputs.

As an example we can see the three-stage buttery connecting eight inputs to eight outputs shown in Figure 5.2. Output line j of every stage goes into input line j of the following stage, for $j = 0...7$.

A butterfly network of type $2^k \times 2^k$ connectes 2^k inputs with 2^k outputs, having a total of k stages. There is a regular, recursive pattern flowing from input to output. The 8×8 network is actually composed of 2 4×4 butterfly networks and it has a supplementary input stage ok 4 crossbas switches each with two outputs. In general, the input stage of a k-stage buttery has the top output lines of each crossbar connected to an input line of of butterfly, and the bottom output line of each crossbar connected to an input line of another butterfly [54].

The **butterfly topology is not fault tollerant**: for any given input there is a single path to any given output. If any router in stage s would fail any of the 2^{k-s} would not be able to be connected with any of the 2^{i+1} outputs. Using the terminology introduced above, in out case booth the node and link conectivity is in this case unitary. We could improve the resiliancy of the network by introducing an extra stage for the input and provide switching capabilities in the form of spatial multiplexing for groing around failed stages.

The other topology I'm going to discuss here is the **crossbar network**. We can observe easily that the multi-level topology presented above limits the communication bandwidth between the inputs and outputs. If the inputs require, each of them, at the same time, acces at distinct outputs the network is not capable of sollving this requirement.

BUPT

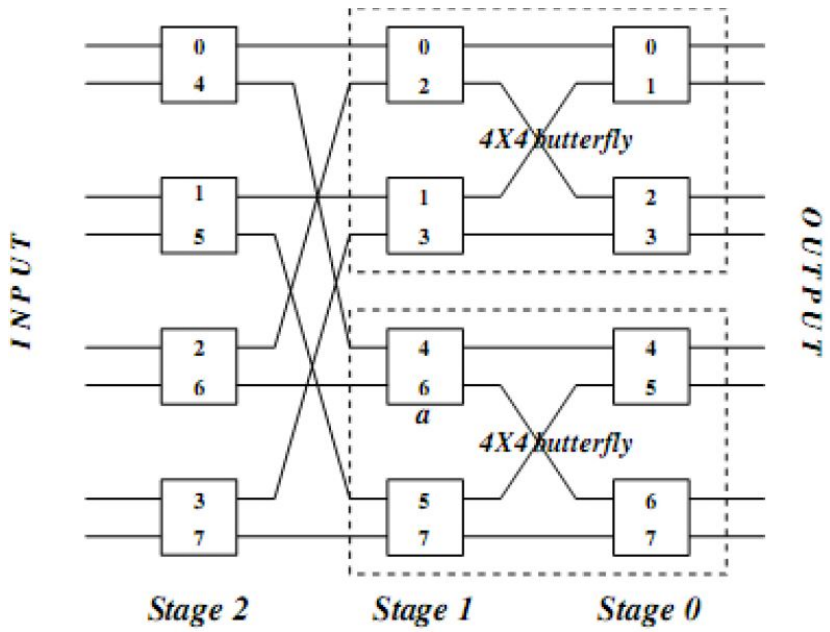


Figure 5.2: A 8I-8O butterfly network switch. Taken from [62]

In figure 5.3 we have a crossbar network which, provides a higher bandwidth seen from this point of view. As can be seen from Figure 5.3, if there are N inputs and M outputs, there is one switchbox associated with each of the $(N \rightarrow M)$ directed input/output pairings. In particular, the switchbox in row i and column j is responsible for connecting the network input on row i to the network output on column j: we call this the $(i \rightarrow j)$ switchbox. The simple aglorithm running on each of the switchboxes has to implement the following actions:

- propagate a message/packet on the same line;
- propagate a message/packet along the same column;
- diagonal propagation of a message/packet (left->top).

Each connection is able to carry one message while, in the same time, each switching node can handle two messages at the same time. For example, a node can be carring messages from left to right link at the same time as it forwards messages from its bottom link to its top link.

Equally like in the case of the butterfly we can see that the **crossbar topology is not fault tolerant**: the failure of any switching node will disconnect certain input-output pairs. The solution for this problem is similiar and requires a form a structural redundancy like seen in subfigure5.3b. Adding a row and a column of switchboxes a providing suplementary multiplexing hardware capable of selecting one of the

BUPT

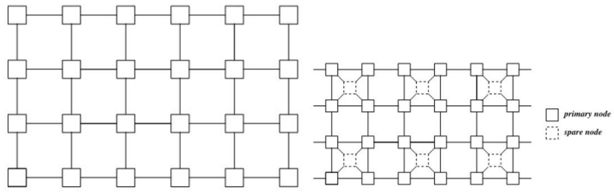


Figure 5.3: Classical and fault-tollerant crossbar switching network.

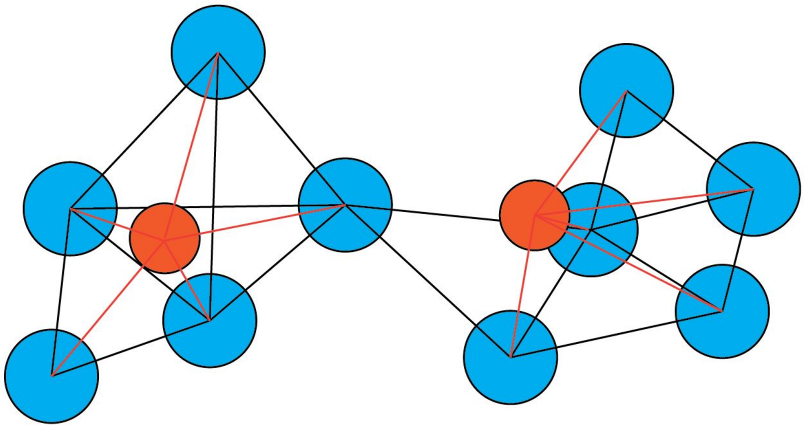


Figure 5.4: Mesh network topology and the interstitial high-redundancy variant

two distinct rows propagating further at any of the corresponding pair of columns we can improve the reliability. If any switchbox becomes faulty, the row and column to which it belongs are disconected, and the spare row and column are activated.

Mesh network topologies are a much modern approach exhibiting good bandwidth caractheristics while preserving a simple and regular topology.

Until now the above discussed topologies deal with two distinct types of entities: the ones responsible form performing actual computation (or any other task adding value to the network) and the ones used only for proving the infrastrcture fabric in which to carry data required for doing the actual task. We can consider the second ones a a burden to the actual project. In the case of a two-dimensional mesh network there is no such distinction: all the node are of the same (computing) type and all of them have the same number of topological and/or physical links (for example four). There are no dedicated switching/routing modules. Sending a message between two non-adjiacent nodes requires the identification of a path between them and forewarding the message in a multi-hop manner along the path.

BUPT

Like the previous toplogies in the case of a mesh one there is also no fault-tolerance in the classical mesh network as well. In case of failure of any of the nodes the network is going to loose it's mesh properties. The solution is also the introduction of structural redundancy in order to cope with partial failure. This mesh would require spare nodes which can be brought online to take place of their adjiacent nodes in case of failure. This is called *intersitial redundancy* [54].

Each active node has a single spare. Each spare node can serve the n (in this case four) direct neighbhours thus we have an overhead of 25% in terms of required hardware. This approach has the big advantage in case of sensor networks because it creates a close spatial proximity minimizing the latency of the network in booth operation and overhaul costs. Considering the above stated role of the nodes as having the common role for booth processing/actuating and message handling they have to be able to cope with this in order to keep the mesh properties intact. In this case there is usually a good corelation between the routing and switching policy of the algorithm an the actual topology of the network . In the case of wireless ad-hoc networks this is not always the case, but the discussion still remains [63].

5.6 Fault tolerance in WSN

In the previous section I have presented some of the main problems in the eld of classical fault tollerance. That is the area where we speak of classical computers and classical networks. But the main framework of discussion still remains valid because we can easely map computers to nodes in a WSN, nodes which have the ability to sense the environment and repsectively the switches to the relay nodes and routers in a WSN. So from the conectivity point of view there are many similarities between these two and so the above sections is usefull for the rest of the discussion.

Due to low cost associated with WSN nodes it is possible to conceive the deployment of a large scale network with potentially hundreds or thousands of nodes. One of the main characteristics required from WSN, in most of their deployment, is the dependability, so that we can rely on the correct operation and results gathered from them. In order to be considered dependable, WSNs must offer characteristics such as *reliability*, *availability*, and *maintainability*. From these, availability depends mostly on fault tolerance to keep the system working as expected. On the global level represented by the delivered service availability of a WSN means that the offered service delivered by the WSN is not affected by failures and faults in underlying components such as single nodes or node subsystems. One of the most evident arguments

BUPT

is that the failure of independent nodes is almost unavoidable. Most high-availability techniques aim to reduce MTTR (the amount of time required for detecting and recovering from a failure) to a minimum.

5.6.1 Levels of fault tolerance in WSN

I will discuss the problem taking into consideration the layered architecture of a networked system, especially a WSN [63].

Physical layer

The PHY layer is responsible for the direct communication the a particular medium between two nodes. Here we speak of modulation-demodulation and encoding-decoding. From the im- plementation point of view, usually we have some kind of software dened radio (SDR) which is a way of extending the programability to the hardware layer. A SDR offers a much bigger addictiveness to the particularities of the medium (particular configuration of the noise) which can signicantly improve the performance of the wireless network. Even the concept of SDR was used initially for the above stated problem it is well suited for solving different fault-tolerance problems. One classical example is when we have some faulty modules, used for implementing a particular encoding-decoding schema, we can use the SDR to switch to another encoding schema for which we still have hardware resources. And the adaption to noisy media can be still regarded as a fault tolerance capability.

Hardware

At the hardware level, components can be divided into two groups. One is represented by the computing, storage subsystem and power supply that are all very reliable when using modern technologies and design techniques. Of-the-shelf microprocessor, DSP processor and controllers are very reliable devices that have a very low rate of malfunctioning. There exist at least three main reasons why this does not necessarily imply that computational subsystems of sensor nodes will be exceptionally reliable [5].

- The first is that sensor nodes are very cost sensitive and therefore will not always be able to design using the highest quality components.
- The second is that strict energy constraints imply that repeated computations are often not realistic options.

BUPT

- The third is that these systems are often deployed in much harsher environments than today's computers.

Although programmability and flexibility are of high importance in sensor networks, strict energy constraints will result in extensive use of application specic designs that can have up to two orders of magnitude less energy consumption for the same functionality [5].

As we should expect sensors and actuators are elements which are most prone to malfunctioning. In the case of sensors, we can distinguish three types of faults [7]:

1. calibration systematic error;
2. random noise error and;
3. complete malfunctioning.

While the first two can be addressed through temporal redundancy, the last one is done using hardware redundancy. Currently, no scheme other that hardware redundancy is envisioned for actuators but the modeles should addopt the established metrics and models from the mechanical domain where the specific device is comming from [60].

System software

The system software of a node in a sensor network consists of the operating system (not necesarely to be used but present most of the time) and utility programs. Probably the most promising technique to implement fault-tolerance is through software diversity, where each program is implemented in numerous diferent variants hoping that will not exhibit identical bugs [109] The subsystem that can mostly benefit from fault-tolerance realized at the system software level is the communication unit. For example, one can reroute messages using diferent paths in a multihop network.

With respect to sensors and actuators, the most important piece of system software is the one related to calibration. Recently, a number of schemes have been proposed for this task [23, 40].

A very important component of system software is the one that supports distributed and simultaneous execution of localized algorithms. For example, in the case of energy minimization under functionality constraint requirements, several protocols have been developed for the coordination of distributed actions [76]. It is important to note that when communication protocols are considered, there is a clear trade-off to be made between it's complexity and its effectiveness taking also in consideration the energy efficiency.

BUPT

Application

Finally, fault tolerance can be addressed also at the application level. For example, if one wants to identify a particular person, he can try to measure using the sensors a variety of biometric features of that person. Each feature and possibly a combination of features will be sucient to identify that person. While addressing fault tolerance at the application level maybe very effcient, unfortunately any given application will require a customized way to properly address the issue [93]. On the other side, an additional advantage of application level fault tolerance is that it can be used to address faults in essentially any type of resource.

5.7 Fault tollerance of SIDeWISe generated networks

The specific area of interest regarding fault tollerance from my point of view is about asessing the general state of the networks I design after running the SIDeWISe methodology (booth flat and hierachical).

Starting with this and taking into consideration all the issues existing regarding the concept of fault tollerance when we speak about networked systems and moreover wirelessly connected ones I took under scrutiny two types of metrics, discussed above: conectivity and diameter stability.

From my point of view there are clear arguments for measuring connectivity because this is the major qualitative aspect which gurantees the emerging properties of the network as a whole: if there is no more direct connection between any two points of the network, depending on the spefific application running on top of it, major setbacks can occur. In the same direction of taught the diameter stability express the impact of the failiure of nodes over one of the major metrics I took under consideration when designing the SIDeWISe algorithms: average path length. For most application the latency of the packets traverssing the network is an important qualitative metric and the degradation of it could signfy important issues for the performance of the network as a whole.

Regarding the nodes, in the typical networks I analyise here, there are three types of them: sensing/actuating nodes, relays and the sink. Also the connections are of two types: wireless point-to-point connections between direct radio connected nodes and wired high speed connections between relays controlling communities.

My investigations started with the generation of a set of random mesh networks and running all the steps of the SIDeWISe algorithm in order

BUPT

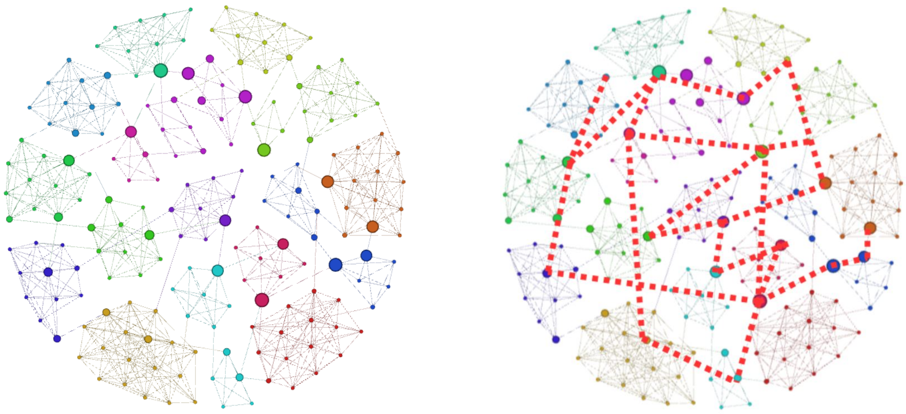


Figure 5.5: Synthetic mesh nework with 185 nodes and 1135 edges and the corresponding set of relays after running the SIDeWISe algorithm

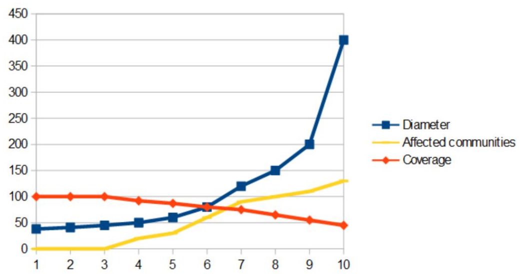


Figure 5.6: Coverage area versus number of lost connections between first level relays

to find the set of relays and the corrsponding networks of high speed connections.

For all the networks a custom script was built on top of Gephi for simulating various types of node and connections loss. Each of the events required the recompuating of the two reliability aspects I have monitored: conectivity and diameter stability. The simulation stoped when the network became disconnected a first level. That ment that first level communities were not able to exchange data anymore renedering the network unsuable.

For the the synthetic network presented in 5.5 the simullations gave promising results in the form of a 62.5% area coverage with 50% coonection loss and 67% coverage until the network became disconnected.

The next stap is to analyse the three major network topologies using the same terms. I took simulation data regarding equally sized network which difered only by topology: mesh (as above), celullar/star,

BUPT

Failed relays	Diameter	Affected communities	Coverage
0	38	0	100
1	41	0	100
2	45	0	100
3	50	2	92
4	60	3	87
5	80	6	80
6	120	9	75
7	150	10	65
8	200	11	55
9	400	13	45

Table 5.1: Dependency of the coverage and conectivity vs. the number of lost connections

random.

BUPT

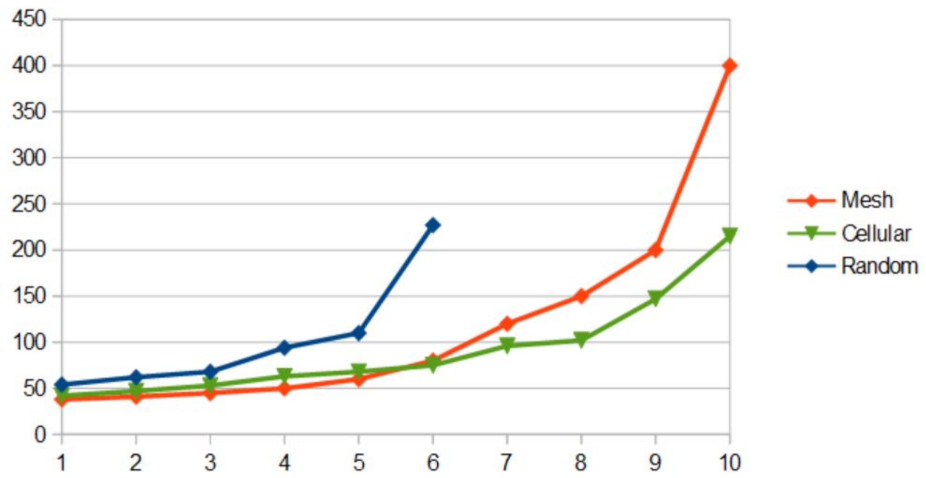


Figure 5.7: Diameter stability for three major topologies using same policy for failure simulation

BUPT

6 Road monitoring sensor network

Allowing a metaphor, it can be said that present day road traffic is reaching the point of singularity: in the very foreseeable future our current infrastructure will lose its ability to sustain high traffic demands. In the last few decades in urban environments the road traffic sustained an exponential growth, while the transportation infrastructure followed a sub-linear development [12, 13]. As a consequence, this will bring an increased number of traffic jams. Considering the critical importance of the road traffic for our current living and economy, the worst option would be a complete city-wide grid-lock. However a range of techniques are available to prevent such situations.

Since the late 70s highly industrialized nations, already beginning to feel the effects of the exponential growth of the number of cars, began to take into consideration different solutions for managing the traffic issue. From a naive point of view the traffic problem could be addressed by adding infrastructure (that is, building more roads) [66]. However, this may prove to inflict unacceptable costs and not actually solve anything, as the newly added infrastructure will also quickly reach its maximum transportation capacity. Extending the road network represents a logistic and technical challenge due to local terrain geography, which is further made complex by existing architectural constraints.

An intelligent transportation system (ITS) is a symbiotic conglomerate of hardware, software and people responsible with monitoring and reacting to various conditions on transportation networks. There are various approaches in designing an ITS for road traffic, but most of the time there is a need for collecting data regarding to the number of vehicles and/or average speed. Transportation systems generally focus on the maximum capacity available for traffic overlaid onto the current implementation of road management systems [90]. Perhaps the effective transportation capacity might be improved by applying some traffic management techniques. From a theoretical point of view most of the traffic theory was based on the background of ideal fluids, at most taking into consideration the compression properties [87]. All these approaches have major problems when applied to real-life traffic, or otherwise stated: real road traffic is neither an ideal fluid nor it behaves like one.

BUPT

Various approaches exists towards solving this problem but one of the key prerequisites for most of them is the capability of good and adequate modeling and prediction of traffic flow. Travel demand, driver choices and even the psychological and complex aspects of the driver's behavior are taken sometimes into consideration but even then the results can be far from expected [#bazzan1999agents]. The social nature of traffic conditions implies a variety of parameters that impact each driver's traffic behavior making them unpredictable and hard to simulate. Unexpected phenomena could arise, such as road rage, when a sudden reaction of even a single driver can lead to serious traffic problems.

During the last years, the mathematical models for road traffic simulation have been improved to take into consideration additional information, such as meteorological factors, day and time of the driving, surrounding environment and even psychological aspects for the typical drivers. Most of the classical models, inspired by gas or fluid behavior in pipes give non-realistic results in modern traffic situations and are considered inappropriate [33], but in the last decade we witness a refactoring of these models and implementation in simulation tools [8]. Responsible for this effect is the nonlinear and chaotic character of the systems that describe road traffic, the so-called"butterfly effect" [106]. Even the slightest changes or disturbances (some incident) in the traffic conditions on a road a few miles upstream the point of observation will induce ripple effects in quite a short time, and current models are neither able to cope with, nor are able to give accurate "what-if" simulations.

A common characteristic of current implementations for real traffic monitoring is their centralized character: usually some state (or regional) authority manages the entire system and shares data recorded (if it does so) with third-party entities or with the general public. As a consequence, citizens and researchers can only access some data already processed and filtered, but not the raw data to do independent analysis. For these systems, primary data is represented by the number of vehicles passing on a road segment over a given time period (possibly also the distribution by categories: cars, trucks, bicycles, pedestrians etc) and the average speed on that given segment of road at any given time of day and any given day of week [82]. Additional data can be represented by the average acceleration and deceleration when entering and exiting the road and even the statistical distribution of the weight of the vehicles and the number of traffic incidents/accidents.

The problem of improving the capacity of the existing transportation infrastructure was previously addressed from applying the mathematical models presented above [12, 33, 41, 31] (developed for identify-

BUPT

ing the critical areas in an existing topology or to predict problems in a proposed one) to performing the simulation and validation (finding the maximum traffic capability) of any particular intersections or road segments. All these approaches require real data to be gathered for developing initial models or validating existing ones

Even if the relationship between the layout of the streets and the character of the urban traffic can be regarded as straightforward, until recent years [36] there was no significant involvement in analyzing it in a systematic manner. The importance of network topology is vital to a traffic network planner [107]. Even if at microscopic level this aspect could be ignored, at the macroscopic level we identified how aspects from complex networks apply and influence traffic behavior. I believe that allready proven metrics for describing the structure of a complex network can be used successfully for finding the key intersections in any urban network. These intersections are subject to intense traffic flow and require careful regulation in order to maintain this flow.

The rest of this chapter is organized as follows: Section 2 provides an overview of the most relevant research initiatives in the area of my work, while Section 3 consists of a short presentation of the key concepts and metrics used in my research, with adaptation to the specific case of city monitoring. Section 4 describes the GIS independent location algorithm and infrastructure for massive distributed monitoiring while in section 5 I present the core of my investigations in the form of a novel strategy for deplying a sensor network for real time traffic monitring, designed using SIDeWISe and h-SIDeWISe.

6.1 State of the art

Traffic regulation and control is commonly based on a combination of traffic rules, traffic lights, and monitoring. A common approach for monitoring the traffic is to use induction loops installed in intersections or in a particular highway spots [#nerem2001global]. This is an efficient solution and gives good results, but the costs involved could prohibit a massive deployment (for instance, in 2012 for Romania the costs for a single installation are estimated around 500Euros). Data from such implementations are used in consumer level products such as Google Maps Traffic Layer (http://goo.gl/0H0EX).

6.1.1 Road traffic monitoring tools and systems

With the advent of modern tools and methodologies for computer aided urban planning and the rapid spread of (Geographical Information Systems) GIS tools in the public services and administration, significant

BUPT

research has been carried towards finding alternative approaches in analyzing the structure of cities. The approach based on graph theory was a clear choice and much work was put into this segment. Such an approach was presented by Alger et al. based on anonymous data collected from BTS handover events produced within a GSM network [3]. The experimental results were gathered in Southern Germany through the participation of Vodafone DE. Using data from the cell phone towers they experimentally derived the speed of each vehicle traveling over the highway and also determine the number of vehicles passing in any particular interval. An essential issue in this approach is the coarse graininess of the cell phone topology that affects its usability. The particular scenario discussed in the paper considers only a particular road (the highway) and consequently the cell's coverage can be intersected with the road's layout yielding a good approximation for the intended purpose, but not adequate in urban situations, with a much higher density of both roads and vehicles. Even if the costs involved are very low (almost zero), this approach still involves a centralized authority (represented by the telephony provider), while collecting data regarding one's mobility without the explicit consent of the cell phone owner can be regarded as privacy violation.

Another issue in this approach, also acknowledged by Alger et al. is the need of a calibration stage. The location of the boundary between two adjacent communication cells is based on both maps and EMF characteristics. For each particular deployment there is a need to check the generated boundary model against data from induction loop. Practically in the early phase we have to install expensive and intrusive equipment in the roads. Similar solutions are also reported in [13, 26, 43].

Approaches such as presented by Alger et al. are not adequate for urban road traffic monitoring because they are not fine grained enough (the order of magnitude for the resolution obtained from the handover events is measured in miles). Even if their granularity is adequate for highways (because vehicles can be monitored quite well on highways based on mobile phone communication) we need a solution capable a resolution closer to tens of meters.

Such levels of precision are achieved usually via GPS or similar external reference systems. In this context it's worth mentioning the work of Calabrese et al., presented as a case study on urban mobility in Rome [20]. An interesting aspect in their work is the use of cell phone data (as seen in [3]) in conjunction with GPS traces from various independent providers, such as the public transportation system of Rome (ATAC) and a private taxi company. Their approach, similar to the one in [3], also requires dedicated post-processing and filtering of the data coming from the cell phone carrier which introduces an over-

BUPT

head and a new level of uncertainty in the quality of data. Of particular interest for urban planning and architectural analysis is the ability of statistically visualizing large masses of citizens traveling throughout the city. However, for my analysis I needed street-level resolution which was not addressed. I still consider that monitoring pedestrians at individual level brings a computational overhead that yields no particular improvement on the quality of the final result: monitoring in real-time the quality of traffic on city roads. The artistic aspect of the project is acknowledged by the authors themselves, the target being the exhibition at the Venice Bienniale.

As we can see, existing solutions are either centralized (induction loops send all the data to a single entity) or provide coarse grained data (highway-related in [3] or statistical cloud in [20]). None of these is adequate for monitoring traffic at street level in urban environments; however, an approach inspired by [20] (using GPS receivers) can be interesting. A particular aspect not addressed by any of the reviewed solutions is the direct feedback of the users, regarding the driver's experience on the road. Such layer of social data can augment the hard data regarding average speed and number of cars therefore allowing for a much more complete and humane experience, including crowd-voting (possibility to acknowledge a previously reported status on a particular segment of road). One of the advantages which can be derived from driver's feedback is the possibility of choosing personal routes, taking into consideration real-time date from traffic. Consequently the system will self-regulate avoiding grid-lock.

Such an application that includes in some form all of the above is WAZE (http://www.waze.com/) with their social approach in crowd mapping and traffic conditions reporting. Founded in 2008, their goal is to provide real-time traffic data, providing drivers the information necessary for best routing. It is also possible to report additional things, such as traffic incidents or road conditions (beyond average speed). However, the real-time approach in Waze is potentially misleading, the quality of data being affected by the number of users driving on that particular area. Not showing statistical data regarding the number of different users from which the data are derived leads to situations such as the one presented in Figure 6.1

Figure 6.1 reflects a particular situation reported by the latest version of WAZE (3.6.0.0) in the vicinity of my university. The red segment indicates a portion of road on which the average speed is below 6 km/h and therefore a "jam" is indicated on that portion. In fact the jam extends much further, to the intersection on the far right side (marked with "Bd. CorneliuCoposu"). This particular situation is caused by simultaneously having a small number of Waze users (only one, in this case) that are moving with low speed. Such situations may mislead

BUPT

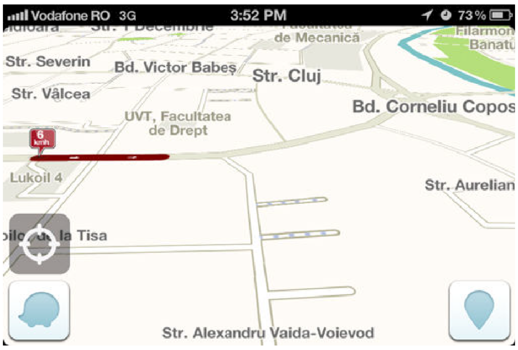


Figure 6.1: WAZE has issues when there is not enough data available

other Waze users to avoid that route, when every day experience would suggest ignoring this specific jam as it may potentially clear away much sooner than an update is available in Waze.

Relying on unrestricted number of users may also create an exposure to a malicious attack in which someone wants to induce a jam on a particular road segment. A set of users equipped with mobile devices are sent to key points of the city with the task of moving as slow as possible, although the road allows superior speed and traffic. At the same time, another user has the task of moving as fast as possible over the same road segment, therefore indicating that traffic is optimal tricking the system into considering that this road segment is free of traffic and all the users are going to reroute themselves via that particular road, creating a jam. Carefully deigning the points of placement for this two kinds of actors can generate a city wide grid-lock.

6.1.2 Road network modelling and analysis

Several research directions put together traffic behavior and complex interaction in the effort to create the "most real artificial driver", which acts as close to a real driver as possible [58]. In this context, separating the driver from its driving context cannot be achieved because the adaption process is closely related to the environment. Understanding how drivers interact and how road networks are created around specific points of interest (schools, shopping centers, concert halls, sports arenas) could lead to identifying the patterns that can apply at different scales over several road networks to achieve increased traffic flow and consequently, less congestion.

Aida et al. focus in [1] on the analysis of macro traffic in a mobile network as a way of investigating complex networks. They have investigated the impact of human relations and the obtained cluster structure and concluded that based on their values they should be able

BUPT

to obtain the behavior of a service process by observing traffic volume values.

The work of Jiang and his team presented in [56] represents a particular interest because of the complex network behavior identified by the authors in the structure of the urban traffic. More specific the authors identify a classical 80/20 behavior because roughly 20% of the streets account for more than 80% of the urban traffic. Another interesting and important aspect presented by the authors and which is also the subject of [100] is the *hierarchical view* of the urban structure. There is a clear distinction between some important streets (few) and some which are less important (many). In the same time the less important ones behave as feeders for the important ones, in a way that is similar to a central nervous system [100]. The authors also take into consideration the complex behavior of the actors (drivers and pedestrians) with respect to route learning and adapting, which can lead to a daily dynamic behavior.

Much work was done in the application of complex networks theory and metrics in the analysis of urban environment. Porta et al. present in [90] a comprehensive methodology and a framework for analysis of urban environments in terms of both classical graph theory but also using complex networks specific metrics and algorithms. Of particular interest for my study is the quite abstract concept of *centrality* and which is the key subject for Crucitti et al. in [32].

Perhaps the most common approach for traffic regulation is installing traffic signal lights which act as main actuators routing traffic between two adjacent intersections. Being the most versatile elements in the road infrastructure, much research was geared towards finding adequate strategies and algorithms for improving the quality of the driving experience by means of dynamically adapting the characteristics of traffic lights [19, 28, 36, 70]. The classical debate between centralized and distributed control is well addressed in [36], however, most of the currently deployed solutions are still centralized.

Of particular interest is finding intersections that present high importance in terms of traffic aggregation (the more cars have to pass through that specific intersection, the more importance it has). Because of this we have chosen to use as a key metric the so-called *betweeness centrality* which is defined as the number of minimum length path between any two nodes in the network. The empirical findings in [90] show that, from a statistical point of view, drivers instinctively choose routes as short as possible, which makes betweeness centrality a viable metric for finding important intersections [91].

In theory, a *community* of a in a complex network is a *subgraph* having its nodes densely connected internally, but with few connections outside, between the subgraphs [84]. This approach allows community

BUPT

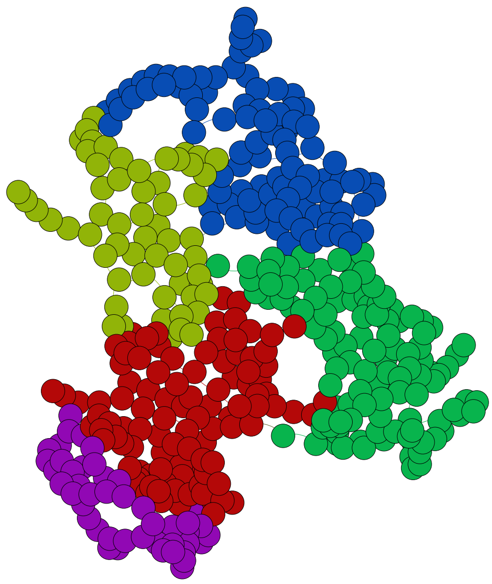


Figure 6.2: Community structure of an urban network, with possibilities of a node belonging to more than one community

overlapping (the same node belongs to more than one community), which is a common case (see example in figure **??**).

6.2 Methodology and results

I propose a methodology centered around topics from the CNA, presented above and the application of the SIDeWISe algorithm I have devised and presented in Chapter 4. The first stage is represented by the recursive division of the city into topologically relevant communities. These communities contain key intersections, identified through computing the betweeness centrality. The second stage is represented by the hierarchical assignment of traffic sensing elements and the associated controllers to key intersections as found by the centrality metric.

6.2.1 Theoretical background

We define a city topology as the set of interconnected nodes (intersections) in which possible optimizations are viewed as a three-layer stack [29]. All nodes may implement, but only some of them will

BUPT

do it, car counting, queue estimation sensors and and traffic lights (semaphores). Each traffic light installed can generate local optima values in terms of traffic flow, representing Layer 1 from [29]. However, not all traffic lights can communicate and coordinate their actions, but those who do generate Layer 2, which will consist of master and slave nodes.

In figure **??**, a simple road network consisting of 3 communities is presented. A selection mechanism must be run to identify groups of master-slave nodes in order to advance from Layer 1 to Layer 2 and to change the approach from computing a global optimum to an adaptive mechanism (master nodes being depicted as grayed-out V_1, V_3 and V_5). The nodes identified as master will coordinate at Layer 2 all traffic movements from Layer 1 and will create, at the same time, the population for Layer 3. Hatched nodes (V_1, V_3, V_5) represent master type intersections organized at Layer 2 of the hierarchy, as obtained by our algorithm. Each master traffic light has authority of changing dynamically the green time on any of the traffic lights in it's community (c_i). At the upper layer each of the communities are clustered from a logical point of view exchanging information between equipotent master nodes of distinct communities (square grey node is a logical one).

Running the SIDeWISe algorithm at different resolutions identifies communities along with the corresponding master-slave nodes (see Section 4.2). If the master does not use a traffic light, the methodology determines whether a traffic light is required. Each community in figure **??** is associated to a topological community identified in figure 6.3. All other nodes, V_2, V_4 and V_8 work as slaves. Inter-community communication occurs between master nodes in order to implement Layer 2 control required for optimizing Layer 3. Each time a master node makes adjustments on its timing plan, on a specific direction, it will send a message notifying the directly connected masters on that direction about the changes. Receiving master node may take into consideration to adjust its timing plan as a reaction to changes made by its neighbors only if its local conditions allows it. An acknowledge message containing the response (whether this is positive or negative) will be sent back to the originating node, in order to notify it about the new timings if they were taken into account or not. If the response is negative, the sending master node will not make any further changes on that direction until a positive one is received.

Going further with the detailed presentation and validation of the algorithm falls outside the scope of this thesis and was presented here only for creating the necessary context for deveoping the monitoring sensor network.

BUPT

6.3 The Timisoara case study

Timisoara is the second city in Romania in terms of booth population and urban density, placed in the western part of the country. Founded during the medieval and lieying at the crossroads of multiple cultures times the city witnesed numerous changes in administration, from otomans to austro-hungarians being transfered into Romanian administration in 1918. Each of the rulers imposed their set of regulation over the urban and architectural developeent of the city, now beign able to distinguish two almost disjoint sets of urban layouts inside the same city. One is represented by the old city centre organized during the austro-hungarian administration and clustered around a central public square with a radial-concentric topology and the other is more recently built, during the communist era, whith wider boulevards and narrow maze like streets spanning between these boulevards. Moreover, the city is divided in two almost symetrical parts by the Bega canal, the two sides having 16 bridges, of which 10 are suitable for car passing and the rest are for pedestrians and bicyles.

All of my subsequent investigations were carried-out using *Gephi* version *0.8.2*, one of the leading open source tools for large graph analysis.

The graph data were obtained by parsing the OpenStreetMap (OSM) XML export via custom written Python script. For each city I've defined and stored in a flat database a bounding rectangle specified by the geographical coordinates of opposite corners. I've used a shell script to parse the database and build appropriate queries for the Overpass API which provided raw XML with the semantics of OSM. The file contains relevant data for my investigation in the form of $nodes(id, lon, lat)$ and $ways(id, nd(nodeIs), \{tags(key, value)\})$ represented as an ordered list of nodes from start to stop. The list of $tags$ is used to specify various attributed for ways in a key-value format. From this I've used the *road type* attribute to filter pedestrian lanes and the *number of lanes* attribute in order to associate weights to each edge. I've normalized the graph representation of OSM by eliminating the intermediary nodes which were used in the original data set in order to represent curved roads in the physical domain, as a sequence of dense points. Consequently my data represents now strictly the topology of the interconnections and not the shape. The filtered data set is written by the Python script in the proper form of a *Gephi* compatible $GEFX$ file. The algorithms — SIDeWISe and h-SIDeWISe — were implemented in Java SE as a plugin for *Gephi* and all the measurements we done on a Intel i5-3320M, with 4GB RAM.

I present in Table 6.1 some of the specific metrics established in the field of CNA applied on the Timisoara graph dataset. The average node degree signifies that most of the intersections have at most two other

BUPT

connecting roads — classical T crossroads — which is specific for the block quarters. The network diameter is defined as the longest of the shortest paths in the graph and signifies the maximum number of intersections trough one has to cross in order to traverse the city. Together with the average path length I consider this as being some of the defining metrics for the quality of the urban road network.

In figure 6.3 we have an overview of the layout of the urban structure of Timisoara in form of a graph where each "dot" represents an intersection. Each community is represented with a different color. As my algorithms are hevily relaying on the concept of *community* as seen in the field of CNA I've applied the community detection on the entire graph in order to obtain what is defined as *first level communities* (Figure 6.3). Of particular interest with not much importance for this research is the fact that topological communities map really well onto the traditional quarters of the city, validating the assumption that the topology imposes even complex organization of urban environments. Figure 6.4 shows traditional quarters of the city identified as complex network communities and the main topological metrics.

Of particular importance for the *quality* of the urban road network is the average path length and the average degree. I have shown in each of the subfigures the value for this parameters and one can observe the fact that the Circumvalatiunii (figure 6.4a) quarter is having the highest average degree (3.309) which correspond to a large number of X crossroads. The low value of the average path length in Giroc and Chisoda villages (figure 6.4d) is consistent with the almost bipartite structure of the two subgraphs.

As can be seen, the road network contains the master nodes (obtained after applying h-SIDeWISe algorithm) indicated by high value of betweeness centrality. These nodes support high traffic values and these are the places that are witnessing traffic jams on a daily basis. Therefore, it is imperative to identify master nodes not only to implement traffic lights (these are necessary to enforce order since traffic is so heavy that leaving traffic only to drivers abilities and will to obey regulations would certainly result in traffic jam), but also to achieve traffic optimization through coordination of traffic data. Since local optimizations have been addressed previously [29], establishing a coordination infrastructure between nodes implementing traffic lights (both master and slave) can be regarded as a final, more demanding stage.

The case study over the city of Timisoara clearly indicates a match between master nodes identified by this methodology and the nodes with significant traffic problems. Figure 6.5 shows the recursive character of the approach with subdividing each of the communities and assigning the local master traffic light. Recursive application of the SIDeWISe algorithm onto the community presented in Figure 6.5, rep-

BUPT

Table 6.1: Main topological parameters for Timisoara.

Parameter	Value
Number of nodes	4070
Number of edges	5542
Average degree	1.434
Diameter	61
Average path length	42.916
Modularity ($res = 10$)	0.938
No. of 1^{st} level communities	39
Clustering coefficient	0.043
Eigenvector centrality	0.9236

resenting the Circumvalatiunii and Mehala quarters. In subfigure (b) one can observe the inner community structure of the upper level community presented in (a). Subsequent stages present the application of the same recursive process on smaller and smaller sets of nodes. In subfigure (f) we show the partitioning of a community of 71 nodes shown in subfigure (e) into 3 sub-communities. For the sake of clarity I've stopped the illustration at level 4 of recursion but depending of the TRESH parameter of the algorithm one cand obtain even more granularity if necessary. At each level I've emphasized with a red (larger) circle the node with the highest betweeness, the one which is to be considered as a relay, in my algorithm.

6.4 Conclusions

My investigation was geared towards finding a suitable algorithm for hierarchical placement of traffic lights under the form of master and slave nodes. I have proposed a novel approach based on mapping aspects from complex network analysis onto issues from urban transportation networks. The key aspects used in this methodology are the betweeness centrality and community detection.

I have presented a systematic procedure for determining the best value for the resolution of the community detection algorithm. I've extended the empirical findings trough a new algorithm that efficiently identifies master nodes —important intersections — in the city where traffic lights are required and where coordination could further optimize traffic. After successive runs, the algorithm will generate the set of master and slave nodes in the network that can be used to implement

BUPT

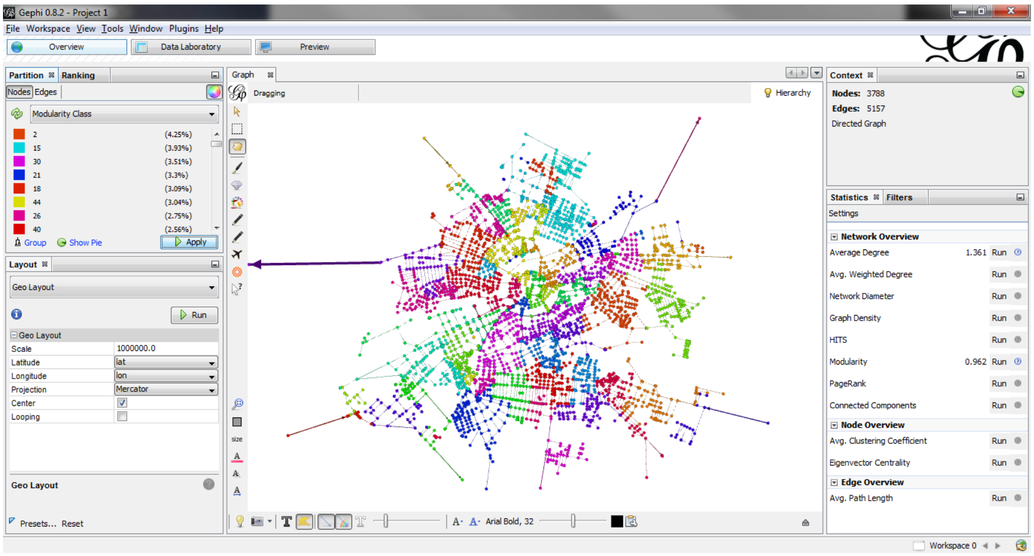


Figure 6.3: The network structure of Timisoara's street infrastructure with communities colored distinctively. The topological community clustering is mapping very well onto the traditional neighborhoods of the city.

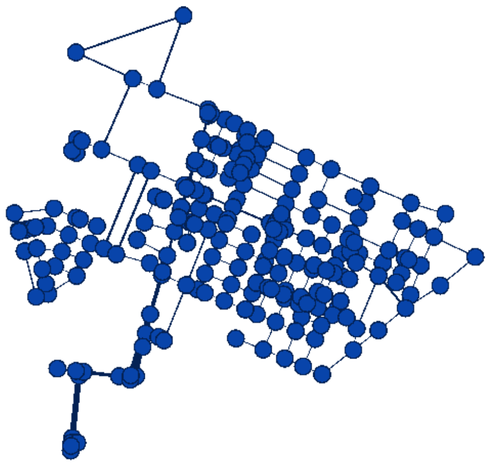
Layer 2 and Layer 3 of the control stack.

Our case study on Timisoara city confirms the correct identification of master nodes as the most congested intersections in Timisoara. Also, attaching each master node from each community to the central intersection provides the suitable policy for selecting the nodes of Layer 2. Intelligent traffic light control is part of modern ITS systems which are an important ingredient for the environmental policies. Efficient placement of semaphores reduces deployment and maintenance costs for the infrastructure and also reduces the impact over the environment trough lower emission levels as a consequence of a better traffic flow.

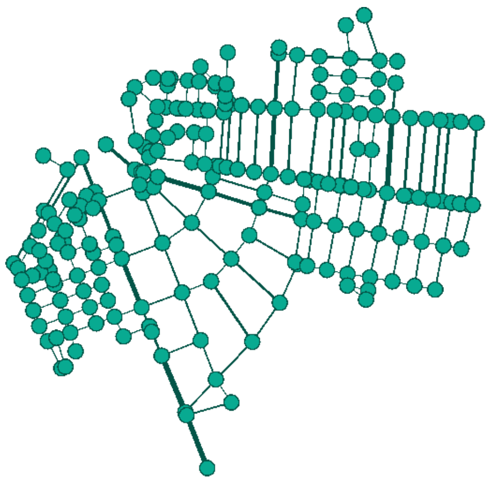
Future work will focus on the design and implementation of the communication framework between master nodes and also on proposing dynamic internal community layout changes.

BUPT

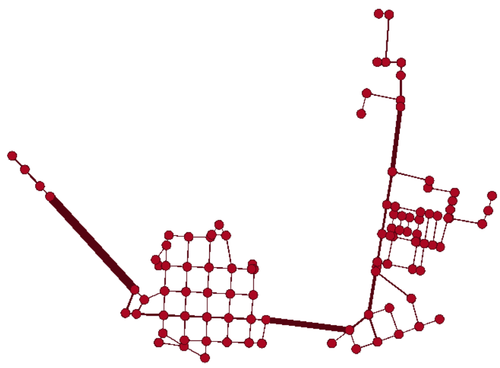
(a) **Circumvalatiunii**362 nodes, Avg. degree=3.309, Avg. path length=12.62



(b) **Aradului**: 297 nodes, Avg. degree: $=2.863$, Avg. path length:$=13.33$



(c) **Girocului quarter**: 206 node, Avg. degree$=2.951$, Avg. path length$=10.589$.



(d) **Girocului and Chisoda villages**: 88 nodes, Avg. degree$=2.951$, Avg. path length$=9.683$

Figure 6.4: In the side figures we emphasized some of the communities and the corresponding quarters of the city, presenting the key network metrics.

BUPT

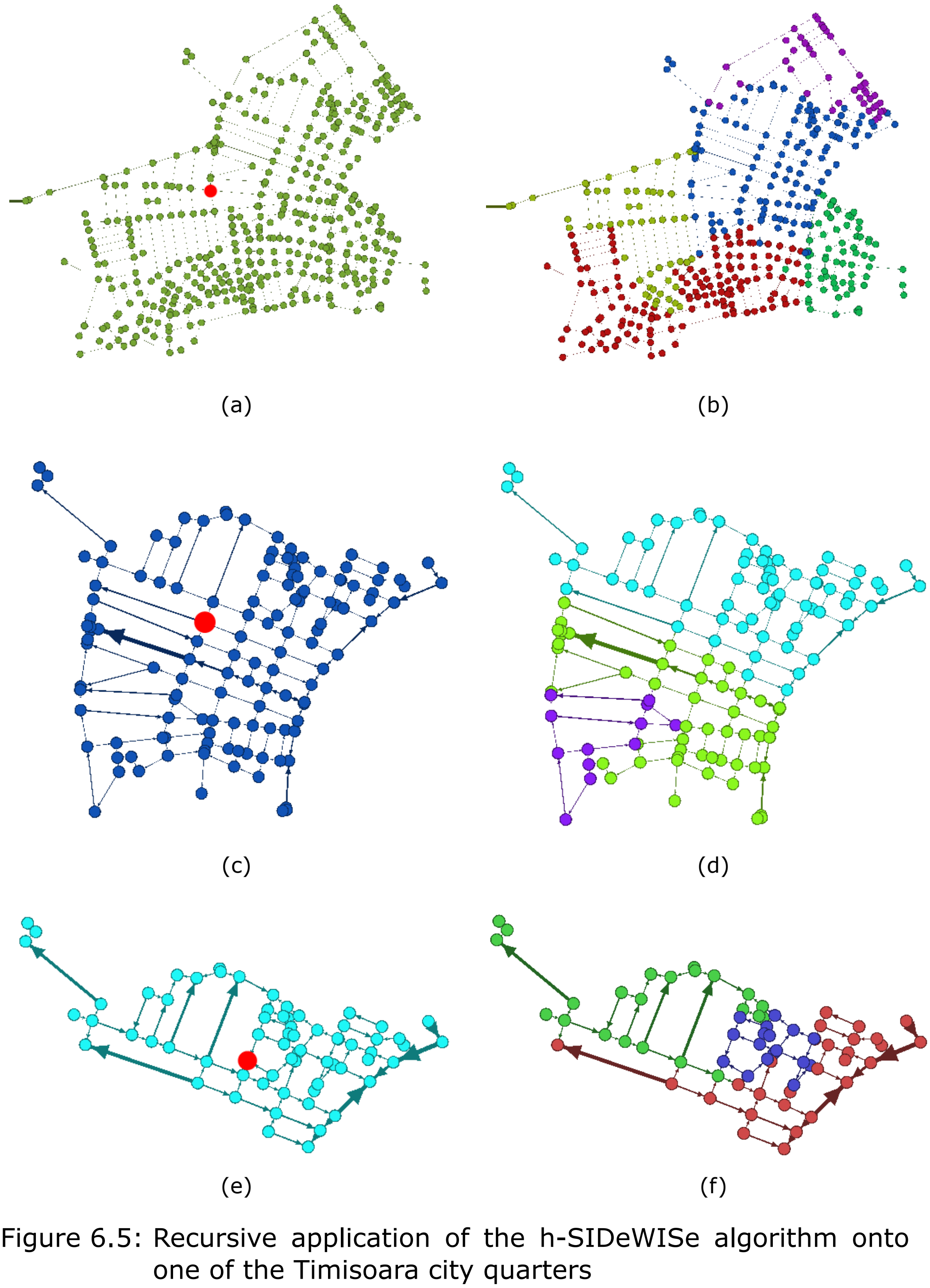


Figure 6.5: Recursive application of the h-SIDeWISe algorithm onto one of the Timisoara city quarters

BUPT

(a) (b)

Figure 6.6: First level monitoring network for Timisoara (a) and the fault tollerant version of it (b)

BUPT

7 Conclusions

In this chapter I will try to argue about the extent I fulfilled the goals outlined in Chapter 1, while in the same time synthesizing the original contributions added on top of the state-of-art. I will try to give a dual perspective over my research: on from a more algorithmic and scientific way regarding the design and testing of the two algorithms I developed and the other from a more engineering approach regarding the design and implementation of a sensor network designed for near real time monitoring of urban road traffic.

The main goal of this thesis is to find novel methodologies and algorithms for placing relays of a sensor network in order to minimize the total energy consumption and improve the reliability of the network. The interdisciplinary approach of this work is based on integrating concepts and metrics of the Network Science in order to find "important" spots of the network which are suitable for placing relays. This meant that a thorough exploration of the field of Network Science had to be done in order to clearly identify the best suitable metrics and in the same time to have a good understanding of the behavior of the algorithms which were to be reused:

- properties and differences between regular and random topologies: were used in order to find the adequate tools specific to each of the cases; in my situation the real life network exhibits more of a random behavior than regular;
- the *centrality* metrics were heavily used in my research and consequently I had to compare the practical behavior of the most important ones in order to find suitable adaptations to the specific cases needed when designing sensor networks; *betweeness* was used for finding the heavy load nodes (lots of data traffic) while *eigenvector* was used for finding the nodes which are topologically closest to all other nodes;
- community structure is another important aspect from the realm of complex networks analysis and I have used it in order to breakdown the network into smaller sub-networks; for one of the algorithms this was done on a single level (central sink + *n* communities) while the extension of this algorithm allows a multi-level (recursive) breakdown of the network until a desired level of granularity is reached.

BUPT

All the above presented aspects are representing the starting points of the work I've carried out, but throughout the thesis I've mentioned points where original adaptations were made in order to accomplish the desired goals.

7.1 Research path and results

This thesis represents an individual contribution to the effort of creating tools and methodologies for designing better sensor networks, capable of keeping with scaling up the number of nodes and in the same time preserving the functional requirements in imposed limits. In the same time the work carried in the area of designing solutions for the road traffic monitoring was done together with other members of the ACSA Research Group and consequently I'll try to put in light only my own endeavor in this field.

After a brief introductory chapter desired to present to the reader the general context and problems in the field of sensor networks, Chapter 2 was intended to provide essential background to the state-of-the art of sensor networks, introduce the terminology and present some typical literature and industry referenced projects where sensor networks provide solution to major social problems. In the same time in this chapter is introduce the problem of optimal relay placement. Various methodologies of finding the best way for placing this nodes exists and specific projects, depending mostly on their environmental conditions, have decided to use different approaches. This is a topic on which rich literature exists and a lot of research groups around the world have investigated this topic. As a consequence in this chapter I've made a review of the main results emphasizing of those which are going to represent terms of comparison for my further presentation

Chapter 3 contains an introduction in the novel topic of Complex Networks and by extension the area of Network Science. When I first approached the topic of optimal relay placement I the major approaches were based either on specific instruments of the mathematics (combinatorial optimization, linear programming, graph theory) or in numerical simulations and pure brute force or with some genetic optimization step. Having some experience and understanding in the field of Complex Networks form previous projects carried in our research group I found interesting to apply to tools and methods in order top analyze the specific type of networks which are sensor networks. In this chapter I introduce the specific terminology related to the field of Complex Networks and I provide in depth analysis of the most important topologies: regular (star, ring, bus and mesh) or random, with various types of distributions for the main characteristics of the network. The novel

BUPT

aspects regarding complex networks deal in the same time with the specific metrics which allow a better characterization of the network as a whole instead of local properties like in classical graph theory. In this case we are dealing mostly with distributions of various parameters of the node or edge space (degree distribution is a is good example of such metric) and I've used this for the genetic optimization step of my algorithms. For all the network topologies under discussion an analysis of these metrics is given, showing the specific properties for each of them. A particular interest is given to a more global property of the network which is the modularity metric capable of showing the degree in which we can identify communities inside a complex network. Based on this various community detection algorithms exist and I've done a review of some of them for laying the grounds required for may research.

Chapter 4 contains the two major algorithms of original conception which provide the core of my research. Starting with the concepts and theoretical grounds described in chapter 3 and providing a formalization of the problem presented in chapter 1 — optimal relay placement — this chapter is dedicated to an in depth analysis of the SIDeWISe algorithm. I start with a formalized description of the problem in terms and notations specific to the graph theory and I continue with the high level description of the procedure from data generation using WSNet topology generator and up to the conversion in a suitable format for Gephi. Translating to Gephi the main idea of the algorithm is explained in a top-down approach and synthesized in figure 4.3. All the steps of the algorithm are subsequently presented in a pseudocode, using the above introduced notations and relaying on metrics and properties of the complex networks. Of those, of particular importance for the novelty of my work, I would emphasize on the following:

- *Step B*: uses the centrality metric over the entire network and assign the sink to the node with the highest centrality. In this case I've used the eigenvector centrality because it maps with the idea of having the average minimum length path o all the other nodes of the network, making it a good analogy with the minimum latency in average. Investigations were made also with betweeness centrality but this was not found suitable in terms of latency in this case.

- *Step C*: uses the community detection algorithms specific to complex networks and adapted in this case for sensor networks. The goal is to determine the optimal number of relays to be placed over the covered area in order to further decrease the latency of the network. Communities represent cluster of tightly/densely grouped/linked nodes with few connections with nodes form other

BUPT

communities. The algorithm can be parametrized trough a parameter called *resolution*, and in this case one can fine tune the number of communities and consequently their size. I provide an in depth analysis of the impact of the resolution on the results of the algorithm and a methodology for finding the best values in case of sensor networks.

- *Step D*: applies the entire procedure describe in step B but on the communities identified in step C. Again the eigenvector centrality is used because the existing edges are determined by position, and so it becomes straightforward and efficient to choose a relay to whom any sensor requires the minimum number of hops to reach.

- *Step E*: uses the relays identified in step D, and builds a MST over this nodes. This is a secondary, overlapped graph of edges that connect all relays and the sink. The edges represent physical links, like broadband cable or fiber connections. The cost of each edge is represented by the euclidean distance. This section ends with creating the tree and consequently it has a low fault tolerance (a loss of any edge/connection) would partition the network.

- In *Step F*, I try to improve the low fault tolerance identified in step E. This is designed to improve the centrality of the sink node in relation to the network of relays by adding one ore more low-latency links. The node identified in Step B still remains the sink of the entire network, but there are more high speed paths to it. This has two major advantages, bu creating multiple paths in the network of relays: first it improves the bandwidth capabilities of the entire network in normal operation mode and second it provide backup routes in case some of the node are lost during operation.

This chapter continues with an multifaceted analysis of the algorithm in which I take under scrutiny each of the most important parameters and variable aspects of the SIDeWISe methodology and asess it's impact over the general performance. The discussion is made as follows:

- the radio link coverage radius imposes limits over the capabilities of building direct ad-hoc mesh networks. This is a parameter r and I've done experimental determination of the most suitable value taking into consideration the number of resulting clusters (communities) versus their size (number of nodes). The best values are around 0.1% of the length of the covered area bounding box. Consequently in this case we obtain the lowest number of required relays for covering the entire network.

BUPT

- the resolution of the community detection algorithm directly dictates the number of communities and implicitly the number of relays (there is one relay per community). I've proposed a methodology for finding the best value for the resolution by introducing a metric which links the number of communities with their size. Lowering the resolution we are going to have a lot of small communities and viceversa. By using this metric I identified the window of optimal values to be in the range from 0.5 to 1.25.
- the size of the network in terms of number of nodes has a significant impact over the performance of most of the algorithms for placing relays and I've investigated the dependence between the number of nodes and number of required relays. In the case of my algorithm there is a logarithmic relationship between the two, which is much better than the linear results obtained in [22]. This is a very important feature of the SIDeWISE algorithm because it manages to keep the number of relays relatively low, thus the cost remains low, as the overall network propagation delay is rapidly decreased.

The last part of the chapter is dedicated with introducing another algorithm of original conception, which is an extension of the SIDeWISe. Using the same notation and formalization I deal with the situations where there is a need of building multi-level networks. The original algorithm was designed for a $2+1$ tiering of the network: there were two levels on nodes (sensing/actuating and relays) and a single sink. The extension I present here allows a $n+1$ tiering of the network by allowing a recursive breakdown of the network in small communities using the concepts and metrics of the complex networks analysis. In this case the relay from level x has the role of a sink for all the nodes of the subnet of level $x+1$. Consequently there is a tree like hierarchy of relays with the root being still identified like in Step B of SIDeWISe. There are some obvious advantages of organizing the network in such a manner regarding the localization of data traffic to a single community partitioning the data on a path of *top-down-up* type. In the same time it provides the natural organization for implementing higher level algorithms which require a master-slave architecture for command and control allowing a better mach between the physical and logical properties of the network. The multi-level structure provides also a better fault tolerance allowing the localization of the fault at the level of a single "cell".

Continuing even further, Chapter 5 deals with the other part of my investigations which are regarding aspects of fault-tolerance and dependability of the network. The main interest of this endeavor is to assess my algorithms capabilities of building fault tolerant networks.

BUPT

First I provide a general overview of all the aspects regarding the real of dependability starting from the general an well known taxonomy of fault-error-failure and continuing with a focus on the special topics of fault tolerance in sensor network systems. I continue with a presentation of the traditional metrics of fault tolerance. There are specific issues with at least measuring it in case of networked systems because at first it is far from trivial just knowing whether a network is working or not. In this case the approach is more fuzzy and deals with aspects like Quality of Service trying to provide a more continuous quantization of the state of health of the network. Consequently I introduce some modern metrics designed specifically for networked systems. In my case I'll use node and link connectivity and diameter stability in order to assess the impact of loosing an increasingly high number of relay nodes and high speed connections. A section was dedicated for discussion the common network topologies and their resilience, but it has the status of a background state of the art review, having little impact over my work because none of the topologies I take under consideration, nor those created by using SIDeWISe are regular. I finalize the chapter by providing experimental findings for assessing the fault tolerance of the networks built by my algorithm. In a simulated environment I ran algorithms for randomly introducing failures of nodes and links measuring in each of the steps the affected sensing nodes (those which were not capable to perform their duties in the network) and percent of the covered area which was not connected anymore. Encouraging results arise from having 50% of the area covered even after loosing 62.4% of the relays.

7.2 Summary of contributions

This thesis is the result of almost four years of work together with my colleagues form the ACSA Research Group. During all this time I've been involved in a few other projects and research grants which were developed in the lab and some of my work was influenced by this, but the core of my research remains personal endeavor and the main contributions of this thesis are personal.

Putting together all the aspects I've approached in this work I would emphasize on the following:

1. The thesis represents a thorough and critical analysis of the main problems in the area of sensor networks, emphasizing on aspects regarding the optimal placement of the relay nodes in order to reduce to the total energy consumption of the network and improve the fault-tolerance;

BUPT

2. There is an original adaptation of the main concepts and metrics from Network Science to the topic of Sensor Networks, new metrics being introduced for describing the "importance" of a node to the network (eigenvector and betweeness centrality)

3. The concept of community, specific to the complex networks domain, is originally adapted for the area of Sensor Network in order to provide a solution to the classical design problem of algorithmically identifying topological clustering of the nodes (in communities) and the key points for placing relay nodes.

4. An original algorithm (SIDeWISe) is presented, devised as a framework for automatic design of a sensor network and optimal placement of relay nodes and of the single sink node in order to minimize the energy consumption and improve the fault-tolerance. The results are presented in relation with literature referenced algorithms the logarithmic dependency with the scale of the network being emphasized.

5. In improvement of the algorithm is introduced for allowing a multi-level analysis of the network, providing means of hierarchically organizing nodes in clusters and building a tree/graph of relays. The algorithm uses metrics and concepts originally adapted from the field of complex networks (modularity and community structure).

6. The entire methodology and all the proposed algorithms are integrated for the analysis and design of a sensor network tasked with monitoring urban road traffic, using a distributed multi-level architecture. It exploits the geographic localization of the data traffic generated by the higher level algorithms. I present estimation regarding the deployments and operation costs and also performance in terms of latency and fault tolerance in terms of area coverage preservation as the number of faulty relays increases.

7.3 Further development

The work presented in this thesis is, to the best of my knowledge, a novel approach in designing sensor networks with regard to minimizing the power consumption of the network as a whole, while preserving or improving the fault tolerance of the network. Consequently much work can be further dedicated for improving and refining the results reported here and there a low of avenues of research which require more in-depth exploration.

BUPT

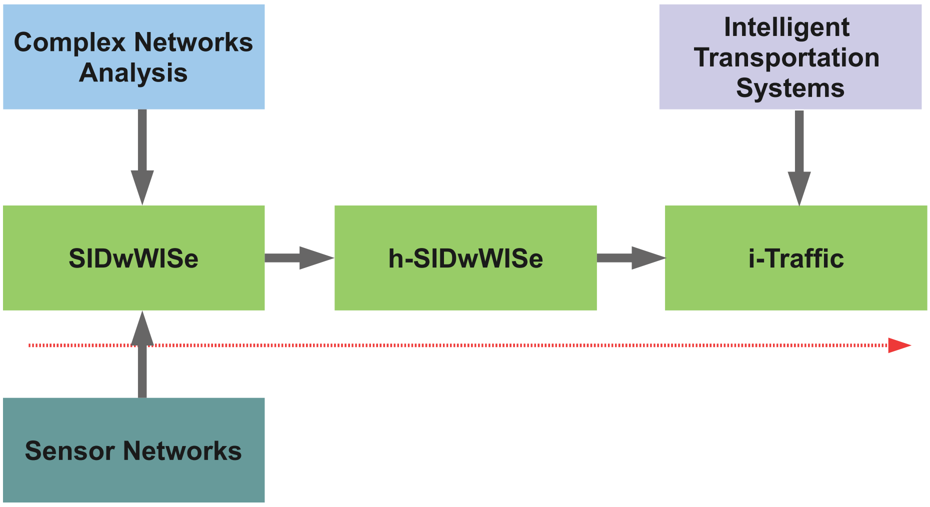


Figure 7.1: The outline of my doctoral research

In the following paragraphs I'll try to sketch a few possible directions where my work could be continued, with impact over some of the existing problems.

7.3.1 Mobile networks

The entire body of work and the SIDeWISe framework is built around a limitative assumption of having only static networks. I'm referring here to the fixed position in space for all the elements of the network in time. While this is true for a lot of major case studies referenced in the literature and in the same time it's a valid assumption for the case study I've presented in Chapter 6, there is a great deal of interest in booth the scientific and industrial area regarding mobile networks.

There are a lot of particularities in the case of mobile sensor networks as outlined in [92]. From the point of my algorithms of particular importance is the need of dynamically recomputing the the main metrics of the networks as the topology changes. Booth the centrality metrics and the community detection algorithm are dependent of the network as a whole so any change in the topology would directly impact the results: the sink and relay can change drastically.

A possible direction of investigation on this topic is further extending the h-SIDeWISe algorithm by using concepts from the multi-level and fractal networks in order to introduce a more layered architecture, and provide a balance between the static/fixed topology of the upper levels of relays and the central sink while keeping the lower layers (leafs of the tree) disorganized and consequently the computational overhead caused by the continuous reconfiguration can be localized deeper in

BUPT

the hierarchy.

7.3.2 Internet of Things

Another hot topic in booth the academia and the industry in the last years is the novel concept of Internet of Things. My contact with this field goes back around the start of my doctoral research and my interest was kept high while following the latest results. In the meanwhile part of my research activities in the ACSA Lab touched also with the aspects regarding IoT, especially the medical application trough a research grant regarding an embedded device for monitoring physical rehabilitation [51].

Seen initially just as a buzz-world and marketing concept for the already established field of sensor networks it emerged in the last years as a field of research and development by itself.

The era when millions and billions of devices are going to be strongly interconnected via IP based protocols is not far form today. Major industry players such as Cisco, IBM, Google and so on are investing huge amounts into developing viable and technologically sound platforms for the moment when everything is going to be connected with everything else. Even if we are closing the gap fast, there are still a lot of challenges in any direction of IoT, some of the most important being as follows:

- lack of standardization and interoperability: at the present moment the are a lot of relatively small industrial applications which already use various IP-based protocols for data acquisition and plant automation. The big issue is represented by the lack of standardization which places them in island like structures with no means of exchanging data on a large scale in order to build innovative services;

- shear size of the network: at it's peak it is estimated that around 20 billion "things" are going to be IP based connected to the Internet at any moment. With the introduction of IPv6 addressing scheme there is no shortage of physical IP address, but the challenges arise from finding policies and methodologies for deploying in organized an efficient manner all the elements of the network;

- lack of serious game-changing applications: at the present moment none of the major competitors from the representative industries hasn't found any really novel and disruptive idea for the IoT. There are serious developments in almost any area of human activity from e-health to agriculture and trough any industrial

BUPT

field, but there is nothing that could truly impress us in order to grant the future of Internet technology to the IoT.

I'll summarize this with a quote of Rob van Kraneburg which said: "*Internet of Things holds a lot of promises but it also holds dangers*".

BUPT

Bibliography

[1] Masaki Aida, Keisuke Ishibashi, Chisa Takano, Hiroyoshi Miwa, Kaori Muranaka, and Akira Miura. Cluster structures in topology of large-scale social networks revealed by traffic data. In *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, volume 1, pages 6–pp. IEEE, 2005.

[2] Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[3] Michael Alger, E Wilson, T Gould, R Whittaker, and N Radulovic. Real-time traffic monitoring using mobile phone data. *Online: http://www. maths-in-industry. org/miis/30 Vodafone Pilotentwicklung GmbH*, 2004.

[4] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

[5] Jahanzeb Anwer and Marco Platzner. Analytic reliability evaluation for fault-tolerant circuit structures on fpgas. In *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on*, pages 177–184. IEEE, 2014.

[6] Algirdas Avizienis. A fault tolerance infrastructure for high-performance cots-based computing in dependable space systems. In *Dependable Computing, 2004. Proceedings. 10th IEEE Pacific Rim International Symposium on*, page 336. IEEE, 2004.

[7] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, 1(1):11–33, 2004.

[8] A Aw and Michel Rascle. Resurrection of "second order" models of traffic flow. *SIAM journal on applied mathematics*, 60(3): 916–938, 2000.

[9] Albert-Laszlo Barabasi, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.

BUPT

[10] Ataul Bari, Ying Chen, Arunita Jaekel, and Subir Bandyopadhyay. A new architecture for hierarchical sensor networks with mobile data collectors. In *Distributed Computing and Networking*, pages 116–127. Springer, 2010.

[11] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *ICWSM*, 2009.

[12] Marc Bernot, Vicent Caselles, and Jean-Michel Morel. *Optimal transportation networks: models and theory*, volume 1955. Springer Verlag, 2009.

[13] Timothy Black, Vicky Mak, Pubudu N Pathirana, and Saeid Nahavandi. Using autonomous mobile agents for efficient data collection in sensor networks. In *Automation Congress, 2006. WAC'06. World*, pages 1–6. IEEE, 2006.

[14] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[15] Renata-Graziela Boar, Alexandru Iovanovici, and Horia Ciocarlie. Analyzing the structure of world financial markets trough complex network analysis. In *(2) Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on*, pages 307–311. IEEE, 2014.

[16] Renata-Graziela Boar, Alexandru Iovanovici, and Horia Ciocarlie. A spatial approach in analyzing the structure and dynamics of the romanian gdp. In *Applied Computational Intelligence and Informatics (SACI), 2014 IEEE 9th International Symposium on*, pages 259–262. IEEE, 2014.

[17] Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *Internet Mathematics*, (just-accepted):00–00, 2014.

[18] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.

[19] Larry Bull, J Shaaban, Andy Tomlinson, JD Addison, and BG Heydecker. Towards distributed adaptive control for road traffic junction signals using learning classifier systems. In *Applications of Learning Classifier Systems*, pages 276–299. Springer, 2004.

BUPT

[20] Francesco Calabrese, Massimo Colonna, Piero Lovisolo, Dario Parata, and Carlo Ratti. Real-time urban monitoring using cell phones: A case study in rome. *Intelligent Transportation Systems, IEEE Transactions on*, 12(1):141–151, 2011.

[21] Guillaume Chelius, Antoine Fraboulet, and E Fleury. Wsnet: a modular event-driven wireless network simulator, 2006.

[22] Guangting Chen and Suhui Cui. Relay node placement in two-tiered wireless sensor networks with base stations. *Journal of Combinatorial Optimization*, 26(3):499–508, 2013.

[23] Jian-Jia Chen and Muhammad Shafique. Embedded software reliability for unreliable hardware. In *Embedded Software (EMSOFT), 2014 International Conference on*, pages 1–1. IEEE, 2014.

[24] Peng Cheng, Chen-Nee Chuah, and Xin Liu. Energy-aware node placement in wireless sensor networks. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 5, pages 3210–3214. IEEE, 2004.

[25] Chee-Yee Chong and Srikanta P Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

[26] E Cipriani, S Gori, and L Mannini. Traffic state estimation based on data fusion techniques. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1477–1482. IEEE, 2012.

[27] DANIEL COOLEY. Wireless sensor networks evolve to meet mainstream needs. *RTCMagazine*, 2012. URL http://rtcmagazine.com/articles/view/102871.

[28] Francesco Corman, Andrea D'Ariano, Dario Pacciarelli, and Marco Pranzo. Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies*, 17(6):607–616, 2009.

[29] Cristian Cosariu, Lucian Prodan, and Mircea Vladutiu. Toward traffic movement optimization using adaptive inter-traffic signaling. In *Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on*, pages 539–544. IEEE, 2013.

[30] Cristian Cosariu, Alexandru Iovanovici, Lucian Prodan, and Mircea Vladutiu. Bio-inspired redistribution of urban traffic flow

BUPT

using a social network approach. In *IEEE Congress on Evolutionary Computation, (CEC), 2015*. IEEE, in press.

[31] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5):512–546, 2011.

[32] Paolo Crucitti, Vito Latora, and Sergio Porta. Centrality measures in spatial networks of urban streets. *Physical Review E*, 73(3): 036125, 2006.

[33] Carlos F Daganzo. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B: Methodological*, 29(4):277–286, 1995.

[34] Alessia D'Andrea, Fernando Ferri, and Patrizia Grifoni. An overview of methods for virtual social networks analysis. In *Computational Social Network Analysis*, pages 3–25. Springer, 2010.

[35] Chavdar Dangalchev. Residual closeness in networks. *Physica A: Statistical Mechanics and its Applications*, 365(2):556–564, 2006.

[36] Sébastien Faye, Claude Chaudet, and Isabelle Demeure. A distributed algorithm for adaptive traffic lights control. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1572–1577. IEEE, 2012.

[37] Emad Felemban, Chang-Gun Lee, and Eylem Ekici. Mmspeed: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 5(6):738–754, 2006.

[38] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[39] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979.

[40] Xin Gao, Ping Wang, Bing Zhao, Ying Liu, and Nan Huang. Design of embedded software reliability testing scheme and protocol for intellectual ammeter. In *Advanced Materials Research*, volume 945, pages 1768–1772. Trans Tech Publ, 2014.

[41] Denos C Gazis. *Traffic theory*, volume 50. Springer, 2002.

BUPT

[42] Daniel Genkin, Adi Shamir, and Eran Tromer. Rsa key extraction via low-bandwidth acoustic cryptanalysis. *IACR Cryptology ePrint Archive*, 2013:857, 2013.

[43] Paola A Gonzalez, Jeremy S Weinstein, Sean J Barbeau, Miguel A Labrador, Philip L Winters, Nevine L Georggi, and R Perez. Automating mode detection for travel behaviour analysis by using global positioning systemsenabled mobile phones and neural networks. *Intelligent Transport Systems, IET*, 4(1):37–49, 2010.

[44] Martin A Green, Keith Emery, Yoshihiro Hishikawa, Wilhelm Warta, and Ewan D Dunlop. Solar cell efficiency tables (version 45). *Progress in photovoltaics: research and applications*, 23(1):1–9, 2015.

[45] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *ACM SIGOPS operating systems review*, volume 34, pages 93–104. ACM, 2000.

[46] Scott A Hill and Dan Braha. Dynamic model of time-dependent complex networks. *Physical Review E*, 82(4):046105, 2010.

[47] Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):026107, 2002.

[48] A Iovanovici, C Visan, and M Marcu. Performance and power consumption investigation for execution of integer operations on cpu and gpu processors for multimedia applications. In *Intelligent Systems and Informatics, 2009. SISY'09. 7th International Symposium on*, pages 285–289. IEEE, 2009.

[49] Alexandru Iovanovici, Lucian Prodan, and Mircea Vladutiu. Collaborative environment for road traffic monitoring. In *ITS Telecommunications (ITST), 2013 13th International Conference on*, pages 232–237. IEEE, 2013.

[50] Alexandru Iovanovici, Cristian Cosariu, Lucian Prodan, and Mircea Vladutiu. A hierarchical approach in deploying traffic lights using complex network analysis. In *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*, pages 791 – 796. IEEE, 17-19 Oct. 2014 2014.

[51] Alexandru Iovanovici, Adina Nitulescu, and Lucian Prodan. Cloud based internet of things system for kinetotherapeutic recovery monitoring. In *Bioengineering, 1st International Conference*

BUPT

2014 (BIOENG), number ISBN: 978-605-5120-95-5, page 54, Istanbul, Turkey, 2014. DAKAM.

[52] Alexandru Iovanovici, Alexandru Topirceanu, Mihai Udrescu, and Mircea Vladutiu. Design space exploration for optimizing wireless sensor networks using social network analysis. In *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*, pages 815 – 820. IEEE, 17-19 Oct. 2014 2014.

[53] Alexandru Iovanovici, Topirceanu Alexandru, Cristian Cosariu, Lucian Prodan, and Mircea Vladutiu. Heuristic optimization of wireless sensor networks using social network analysis. In Lakhmi C. Jain Valentina Emilia Balas and Branko Kovacevic, editors, *Advances in Intelligent Systems and Computing/ 6th International Workshop on Soft Computing Applications*, number ISSN: 2194-5357, in press.

[54] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[55] Mika Ishizuka and Masaki Aida. Performance study of node placement in sensor networks. In *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, pages 598–603. IEEE, 2004.

[56] Bin Jiang. Street hierarchies: a minority of streets account for a majority of traffic flow. *International Journal of Geographical Information Science*, 23(8):1033–1048, 2009.

[57] Donald B Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM (JACM)*, 24(1):1–13, 1977.

[58] Oskar Juhlin. Traffic behaviour as social interaction-implications for the design of artificial drivers. In *PROCEEDINGS OF 6TH WORLD CONGRESS ON INTELLIGENT TRANSPORT SYSTEMS (ITS), HELD TORONTO, CANADA, NOVEMBER 8-12, 1999*, 1999.

[59] B Kantarci and V Labatut. Classification of complex networks based on topological properties. *Proceedings of the 3rd International Conference on Social Computing and Its Applications*, 2013.

[60] Kailash C Kapur and Michael Pecht. *Reliability engineering*. John Wiley & Sons, 2014.

BUPT

[61] Benahmed Khelifa, Hafid Haffaf, Merabti Madjid, and David Llewellyn-Jones. Monitoring connectivity in wireless sensor networks. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 507–512. IEEE, 2009.

[62] Israel Koren and C Mani Krishna. *Fault-tolerant systems*. Morgan Kaufmann, 2010.

[63] Farinaz Koushanfar, Miodrag Potkonjak, and A Sangiovanni-Vincentell. Fault tolerance techniques for wireless ad hoc sensor networks. In *Sensors, 2002. Proceedings of IEEE*, volume 2, pages 1491–1496. IEEE, 2002.

[64] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[65] Ray Kurzweil. *The singularity is near: When humans transcend biology*. Penguin, 2005.

[66] Myer Kutz. *Handbook of transportation engineering*, volume 1000. McGraw-Hill New York, 2004.

[67] Renaud Lambiotte, J-C Delvenne, and Mauricio Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*, 2008.

[68] Olaf Landsiedel, Klaus Wehrle, and Stefan Götz. Accurate prediction of power consumption in sensor networks. In *Proc. of the Second Workshop on Embedded Networked Sensors*. Citeseer, 2005.

[69] Glenn Lawyer. Understanding the spreading power of all nodes in a network: a continuous-time perspective. *arXiv preprint arXiv:1405.6707*, 2014.

[70] Guillaume Leduc. Road traffic data: Collection methods and applications. *Working Papers on Energy, Transport and Climate Change*, 1:55, 2008.

[71] Régis Leveugle, A Calvez, Paolo Maistri, and Pierre Vanhauwaert. Statistical fault injection: quantified error and confidence. In *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09.*, pages 502–506. IEEE, 2009.

[72] Li Erran Li and Prasun Sinha. Throughput and energy efficiency in topology-controlled multi-hop wireless sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 132–140. ACM, 2003.

BUPT

[73] Liyan Li, Hualin Zeng, Yuze Zhang, Qingshan Kong, Yan Zhou, and Yuliang Liu. Analysis of backscattering characteristics of objects for remote laser voice acquisition. *Applied optics*, 53(5): 971–978, 2014.

[74] Wei Li and Jing-Yu Yang. Comparing networks from a data analysis perspective. In *Complex Sciences*, pages 1907–1916. Springer, 2009.

[75] Yingshu Li and My T Thai. *Wireless sensor networks and applications*. Springer Science & Business Media, 2008.

[76] Anfeng Liu, Deyu Zhang, Penghui Zhang, Guohua Cui, and Zhigang Chen. On mitigating hotspots to maximize network lifetime in multi-hop wireless sensor network with guaranteed transport delay and reliability. *Peer-to-Peer Networking and Applications*, 7(3):255–273, 2014.

[77] Xiaowen Liu and Martin Haenggi. Toward quasiregular sensor networks: Topology control algorithms for improved energy efficiency. *Parallel and Distributed Systems, IEEE Transactions on*, 17(9):975–986, 2006.

[78] John J Macionis and Linda M Gerber. Sociology: Seventh canadian edition, 2011.

[79] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.

[80] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[81] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[82] P Misra and P Enge. Global positioning system: Signals, measurements and. 2011.

[83] Mark Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[84] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103 (23):8577–8582, 2006.

BUPT

[85] Mark EJ Newman. The mathematics of networks. *The new palgrave encyclopedia of economics*, 2:1–12, 2008.

[86] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, 2010.

[87] Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[88] Joseph A Paradiso and Thad Starner. Energy scavenging for mobile and wireless electronics. *Pervasive Computing, IEEE*, 4(1): 18–27, 2005.

[89] Wint Yi Poe and Jens B Schmitt. Sink placement without location information in large-scale wireless sensor networks. In *Asian Internet Engineering Conference*, pages 69–76. ACM, 2009.

[90] Sergio Porta, Paolo Crucitti, and Vito Latora. The network analysis of urban streets: a dual approach. *Physica A: Statistical Mechanics and its Applications*, 369(2):853–866, 2006.

[91] Rami Puzis, Yaniv Altshuler, Yuval Elovici, Shlomo Bekhor, Yoram Shiftan, and Alex Pentland. Augmented betweenness centrality for environmentally aware traffic monitoring in transportation networks. *Journal of Intelligent Transportation Systems*, 17(1): 91–105, 2013.

[92] Javad Rezazadeh. Mobile wireles sensor networks overview. *International Journal of Computer Communications and Networks (IJCCN)*, 2(1), 2012.

[93] Dilan Sahin, Vehbi Cagri Gungor, Taskin Kocak, and Gurkan Tuna. Quality-of-service differentiation in single-path and multipath routing for wireless sensor network-based smart grid applications. *Ad Hoc Networks*, 22:43–60, 2014.

[94] Michael Santarini. Cosmic radiation comes to asic and soc design. *EDN, May*, 12, 2005.

[95] Carl Shapiro and Hal R Varian. *Information rules: a strategic guide to the network economy*. Harvard Business Press, 2013.

[96] Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the*

BUPT

2nd international conference on Embedded networked sensor systems, pages 188–200. ACM, 2004.

[97] Martin L Shooman. Probabilistic reliability. 1968.

[98] Steven H Strogatz. Exploring complex networks. *Nature*, 410 (6825):268–276, 2001.

[99] Sameer Tilak, Nael B Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2): 28–36, 2002.

[100] Martin Tomko, Stephan Winter, and Christophe Claramunt. Experiential hierarchies of streets. *Computers, Environment and Urban Systems*, 32(1):41–52, 2008.

[101] Alexandru Topirceanu, Alexandru Iovanovici, Mihai Udrescu, and Mircea Vladutiu. Social cities: Redistribution of traffic flow in cities using a social network approach. In Springer, editor, *6th International Workshop on Soft Computing Applications (SOFA)*, in press.

[102] Xiao Fan Wang and Guanrong Chen. Complex networks: small-world, scale-free and beyond. *Circuits and Systems Magazine, IEEE*, 3(1):6–20, 2003.

[103] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[104] Stanley Wasserman and Joseph Galaskiewicz. *Advances in social network analysis: Research in the social and behavioral sciences*. Sage, 1994.

[105] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440–442, 1998.

[106] Daniel B Work, O-P Tossavainen, Quinn Jacobson, and Alexandre M Bayen. Lagrangian sensing: traffic estimation with mobile devices. In *American Control Conference, 2009. ACC'09.*, pages 1536–1543. IEEE, 2009.

[107] Jian-Jun Wu, Zi-You Gao, and Hui-jun Sun. Optimal traffic networks topology: A complex networks perspective. *Physica A: Statistical Mechanics and its Applications*, 387(4):1025–1032, 2008.

BUPT

[108] Kenan Xu, Hossam S Hassanein, Glen Takahara, and Q Wang. Relay node deployment strategies in heterogeneous wireless sensor networks: multiple-hop communication case. In *SECON*, volume 5, pages 575–585, 2005.

[109] Shigeru Yamada. *Software reliability modeling: fundamentals and applications*. Springer, 2014.

[110] Mohamed Younis and Kemal Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, 2008.

[111] Faraz Zaidi. Small world networks and clustered small world networks with random connectivity. *Social Network Analysis and Mining*, pages 1–13, 2013.

BUPT