Security in Smart Cities:

Adapting Castalia to Simulate Attacks on Deployed Heterogeneous WSNs

Jaume Guasch[1]

[1] Master Program in Security of Information and Communication Technologies Student
Universitat Oberta de Catalunya (UOC)

January 11th, 2016
Final Studies Project Presentation

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Introduction

- Human population tends to live in growing cities and metropolis
- It is anticipated that by 2030, 60% of world population will already live in large cities and urban areas
- The intensive use of Information and Communication Technologies (ICT) opens new opportunities for cities management
- Smart City concept appeared in the 90's decade as the set of initiatives to bring better services to citizens and improvements to city management
- Amongst the most studied issues, Security is highlighted in the literature from different angles and seen as a concern by users and policy makers
- Security in the context of managing heterogeneous information sources in the 'smart' environment with the confluence of a growing and diverse number of data generation elements
- Sensor Network is seen as the central key element in the 'smart' environment
- The analysis of the data associated with sensor networks opens a field of study to understand their type, possible menaces and the impact on security

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Smart Cities and IoT

From the Literature Review

- Smart Cities represent the confluence between the intelligent services that big cities or metropolis need and the growing use of ICT
- Broad and prolific field with different research disciplines from supplied services, real experiences, study of technologies and the study of main concerns to the deployment and acceptance by users
- Example: Barcelona as one of the leading smart cities in Europe and representative approach combining smart districts, living labs, e-Services, Open Data and providing examples of district transformation as the 22@ successful implementation
- Example: Santander with *SmartSantander* as one of the largest smart city test-beds in Europe and platform integration of the large amounts of data collected from all the deployed sensors
- From the new trends in the so-called Future Internet, the Internet of Things (IoT) is converging rapidly with smart cities as the combination of elements such as ubiquitous computing, sensors technology, wireless communications, Internet and the embedded devices

Smart Cities and IoT

From the Literature Review

- Smart Cities represent the confluence between the intelligent services that big cities or metropolis need and the growing use of ICT
- Broad and prolific field with different research disciplines from supplied services, real experiences, study of technologies and the study of main concerns to the deployment and acceptance by users
- Example: Barcelona as one of the leading smart cities in Europe and representative approach combining smart districts, living labs, e-Services, Open Data and providing examples of district transformation as the 22@ successful implementation
- Example: Santander with *SmartSantander* as one of the largest smart city test-beds in Europe and platform integration of the large amounts of data collected from all the deployed sensors
- From the new trends in the so-called Future Internet, the Internet of Things (IoT) is converging rapidly with smart cities as the combination of elements such as ubiquitous computing, sensors technology, wireless communications, Internet and the embedded devices

Smart Cities and IoT

From the Literature Review

- Smart Cities represent the confluence between the intelligent services that big cities or metropolis need and the growing use of ICT
- Broad and prolific field with different research disciplines from supplied services, real experiences, study of technologies and the study of main concerns to the deployment and acceptance by users
- Example: Barcelona as one of the leading smart cities in Europe and representative approach combining smart districts, living labs, e-Services, Open Data and providing examples of district transformation as the 22@ successful implementation
- Example: Santander with *SmartSantander* as one of the largest smart city test-beds in Europe and platform integration of the large amounts of data collected from all the deployed sensors
- From the new trends in the so-called Future Internet, the Internet of Things (IoT) is converging rapidly with smart cities as the combination of elements such as ubiquitous computing, sensors technology, wireless communications, Internet and the embedded devices

Smart Cities and IoT

From the Literature Review

- Smart Cities represent the confluence between the intelligent services that big cities or metropolis need and the growing use of ICT
- Broad and prolific field with different research disciplines from supplied services, real experiences, study of technologies and the study of main concerns to the deployment and acceptance by users
- Example: Barcelona as one of the leading smart cities in Europe and representative approach combining smart districts, living labs, e-Services, Open Data and providing examples of district transformation as the 22@ successful implementation
- Example: Santander with *SmartSantander* as one of the largest smart city test-beds in Europe and platform integration of the large amounts of data collected from all the deployed sensors
- From the new trends in the so-called Future Internet, the Internet of Things (IoT) is converging rapidly with smart cities as the combination of elements such as ubiquitous computing, sensors technology, wireless communications, Internet and the embedded devices

Smart Cities and IoT

From the Literature Review

- Smart Cities represent the confluence between the intelligent services that big cities or metropolis need and the growing use of ICT
- Broad and prolific field with different research disciplines from supplied services, real experiences, study of technologies and the study of main concerns to the deployment and acceptance by users
- Example: Barcelona as one of the leading smart cities in Europe and representative approach combining smart districts, living labs, e-Services, Open Data and providing examples of district transformation as the 22@ successful implementation
- Example: Santander with *SmartSantander* as one of the largest smart city test-beds in Europe and platform integration of the large amounts of data collected from all the deployed sensors
- From the new trends in the so-called Future Internet, the Internet of Things (IoT) is converging rapidly with smart cities as the combination of elements such as ubiquitous computing, sensors technology, wireless communications, Internet and the embedded devices

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Wireless Sensor Networks

From The Literature Review

- Sensor networks are usually of diverse types and from different manufacturers and might present doubts related to ensuring security, among others issues
- Requirements of both low cost and low power consumption of sensor networks that characterize the IoT networks pose an additional challenge
- Security is contemplated in the broadest sense, covering availability, the integrity and capacity of authorized access
- Security in IoT and in Smart Cities is directly related to security in WSNs
- WSNs limitations make it challenging to ensure their security when facing attacks in their function and operation
- Increasing number of applications where security is critical would condition policy-makers' acceptance and generate final users resistances
- Network security is a fundamental requirement that will demand the development of mechanisms to detect the presence of possible attacks and to estimate the level of security of a given WSN
- Simulation of real networks in lab environment as a good compromise between the fidelity of the model and the ability to emulate a wide range of situations and scenarios

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Pursued Objective

Initial Project Requirements

- Main objective: set-up of a simulation environment in order to emulate a real network of already existing nodes
- Information about several groups of nodes from different brands, characterized by a list of individual sensor node references and their GPS positions has been obtained for the city of Barcelona
- The actual received data sent from sensors during an extensive period up to 14 days has also been supplied
- No additional information about network routing or the presence of additional routing nodes or gateways has been supplied
- The approach: To send actual available data from nodes, instead of programming the nodes to send messages following certain statistical distribution
- Different scenarios to emulate anomalies and attacks might be able to be simulated in order to compare the effects on the received data when applied
- Final Output: The information coming from every simulation to be processed to feed a detection system that will learn from every scenario in different time intervals to infer when an attack occurred

Pursued Objective

Initial Project Requirements

- Main objective: set-up of a simulation environment in order to emulate a real network of already existing nodes
- Information about several groups of nodes from different brands, characterized by a list of individual sensor node references and their GPS positions has been obtained for the city of Barcelona
- The actual received data sent from sensors during an extensive period up to 14 days has also been supplied
- No additional information about network routing or the presence of additional routing nodes or gateways has been supplied
- The approach: To send actual available data from nodes, instead of programming the nodes to send messages following certain statistical distribution
- Different scenarios to emulate anomalies and attacks might be able to be simulated in order to compare the effects on the received data when applied
- Final Output: The information coming from every simulation to be processed to feed a detection system that will learn from every scenario in different time intervals to infer when an attack occurred

Pursued Objective

Initial Project Requirements

- Main objective: set-up of a simulation environment in order to emulate a real network of already existing nodes
- Information about several groups of nodes from different brands, characterized by a list of individual sensor node references and their GPS positions has been obtained for the city of Barcelona
- The actual received data sent from sensors during an extensive period up to 14 days has also been supplied
- No additional information about network routing or the presence of additional routing nodes or gateways has been supplied
- The approach: To send actual available data from nodes, instead of programming the nodes to send messages following certain statistical distribution
- Different scenarios to emulate anomalies and attacks might be able to be simulated in order to compare the effects on the received data when applied
- Final Output: The information coming from every simulation to be processed to feed a detection system that will learn from every scenario in different time intervals to infer when an attack occurred

Pursued Objective

Initial Project Requirements

- Main objective: set-up of a simulation environment in order to emulate a real network of already existing nodes
- Information about several groups of nodes from different brands, characterized by a list of individual sensor node references and their GPS positions has been obtained for the city of Barcelona
- The actual received data sent from sensors during an extensive period up to 14 days has also been supplied
- No additional information about network routing or the presence of additional routing nodes or gateways has been supplied
- The approach: To send actual available data from nodes, instead of programming the nodes to send messages following certain statistical distribution
- Different scenarios to emulate anomalies and attacks might be able to be simulated in order to compare the effects on the received data when applied
- Final Output: The information coming from every simulation to be processed to feed a detection system that will learn from every scenario in different time intervals to infer when an attack occurred

Pursued Objective

Initial Project Requirements

- Main objective: set-up of a simulation environment in order to emulate a real network of already existing nodes
- Information about several groups of nodes from different brands, characterized by a list of individual sensor node references and their GPS positions has been obtained for the city of Barcelona
- The actual received data sent from sensors during an extensive period up to 14 days has also been supplied
- No additional information about network routing or the presence of additional routing nodes or gateways has been supplied
- The approach: To send actual available data from nodes, instead of programming the nodes to send messages following certain statistical distribution
- Different scenarios to emulate anomalies and attacks might be able to be simulated in order to compare the effects on the received data when applied
- Final Output: The information coming from every simulation to be processed to feed a detection system that will learn from every scenario in different time intervals to infer when an attack occurred

Pursued Objective

Initial Project Requirements

- Main objective: set-up of a simulation environment in order to emulate a real network of already existing nodes
- Information about several groups of nodes from different brands, characterized by a list of individual sensor node references and their GPS positions has been obtained for the city of Barcelona
- The actual received data sent from sensors during an extensive period up to 14 days has also been supplied
- No additional information about network routing or the presence of additional routing nodes or gateways has been supplied
- The approach: To send actual available data from nodes, instead of programming the nodes to send messages following certain statistical distribution
- Different scenarios to emulate anomalies and attacks might be able to be simulated in order to compare the effects on the received data when applied
- Final Output: The information coming from every simulation to be processed to feed a detection system that will learn from every scenario in different time intervals to infer when an attack occurred

Pursued Objective

Initial Project Requirements

- Main objective: set-up of a simulation environment in order to emulate a real network of already existing nodes
- Information about several groups of nodes from different brands, characterized by a list of individual sensor node references and their GPS positions has been obtained for the city of Barcelona
- The actual received data sent from sensors during an extensive period up to 14 days has also been supplied
- No additional information about network routing or the presence of additional routing nodes or gateways has been supplied
- The approach: To send actual available data from nodes, instead of programming the nodes to send messages following certain statistical distribution
- Different scenarios to emulate anomalies and attacks might be able to be simulated in order to compare the effects on the received data when applied
- Final Output: The information coming from every simulation to be processed to feed a detection system that will learn from every scenario in different time intervals to infer when an attack occurred

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Simulation Environment

OMNeT++ & Castalia

- The simulation environment is build using Castalia 3.2 over OMNeT++ 4.6
- OMNeT++ is an object-oriented modular discrete event network simulation framework with a generic architecture
- Basic element of OMNeT++ is the module. Modules are simple or compound. Simple modules are lowest level of hierarchy and behaviour is programmed in C++. Compound modules are constructed by other simple or compound modules
- Modules are connected to other modules via gates that send and receive messages that are formed by arbitrary data structures
- A simulation model is composed by different simple and compound modules, that are interconnected in order to pass messages among them
- Castalia is a simulator designed for Wireless Sensor Network (WSN) and Body Area Networks (BAN) that are both characterized by low power embedded devices
- Castalia pre-defines OMNeT++ simple and compound modules to easily simulate WSNs and BANs. Modules might be created or modified to include new protocols or add new applications

Simulation Environment

OMNeT++ & Castalia

- The simulation environment is build using Castalia 3.2 over OMNeT++ 4.6
- OMNeT++ is an object-oriented modular discrete event network simulation framework with a generic architecture
- Basic element of OMNeT++ is the module. Modules are simple or compound. Simple modules are lowest level of hierarchy and behaviour is programmed in C++. Compound modules are constructed by other simple or compound modules
- Modules are connected to other modules via gates that send and receive messages that are formed by arbitrary data structures
- A simulation model is composed by different simple and compound modules, that are interconnected in order to pass messages among them
- Castalia is a simulator designed for Wireless Sensor Network (WSN) and Body Area Networks (BAN) that are both characterized by low power embedded devices
- Castalia pre-defines OMNeT++ simple and compound modules to easily simulate WSNs and BANs. Modules might be created or modified to include new protocols or add new applications

Simulation Environment

OMNeT++ & Castalia

- The simulation environment is build using Castalia 3.2 over OMNeT++ 4.6
- OMNeT++ is an object-oriented modular discrete event network simulation framework with a generic architecture
- Basic element of OMNeT++ is the module. Modules are simple or compound. Simple modules are lowest level of hierarchy and behaviour is programmed in C++. Compound modules are constructed by other simple or compound modules
- Modules are connected to other modules via gates that send and receive messages that are formed by arbitrary data structures
- A simulation model is composed by different simple and compound modules, that are interconnected in order to pass messages among them
- Castalia is a simulator designed for Wireless Sensor Network (WSN) and Body Area Networks (BAN) that are both characterized by low power embedded devices
- Castalia pre-defines OMNeT++ simple and compound modules to easily simulate WSNs and BANs. Modules might be created or modified to include new protocols or add new applications

Simulation Environment

OMNeT++ & Castalia

- The simulation environment is build using Castalia 3.2 over OMNeT++ 4.6
- OMNeT++ is an object-oriented modular discrete event network simulation framework with a generic architecture
- Basic element of OMNeT++ is the module. Modules are simple or compound. Simple modules are lowest level of hierarchy and behaviour is programmed in C++. Compound modules are constructed by other simple or compound modules
- Modules are connected to other modules via gates that send and receive messages that are formed by arbitrary data structures
- A simulation model is composed by different simple and compound modules, that are interconnected in order to pass messages among them
- Castalia is a simulator designed for Wireless Sensor Network (WSN) and Body Area Networks (BAN) that are both characterized by low power embedded devices
- Castalia pre-defines OMNeT++ simple and compound modules to easily simulate WSNs and BANs. Modules might be created or modified to include new protocols or add new applications

Simulation Environment

OMNeT++ & Castalia

- The simulation environment is build using Castalia 3.2 over OMNeT++ 4.6
- OMNeT++ is an object-oriented modular discrete event network simulation framework with a generic architecture
- Basic element of OMNeT++ is the module. Modules are simple or compound. Simple modules are lowest level of hierarchy and behaviour is programmed in C++. Compound modules are constructed by other simple or compound modules
- Modules are connected to other modules via gates that send and receive messages that are formed by arbitrary data structures
- A simulation model is composed by different simple and compound modules, that are interconnected in order to pass messages among them
- Castalia is a simulator designed for Wireless Sensor Network (WSN) and Body Area Networks (BAN) that are both characterized by low power embedded devices
- Castalia pre-defines OMNeT++ simple and compound modules to easily simulate WSNs and BANs. Modules might be created or modified to include new protocols or add new applications

Simulation Environment

OMNeT++ & Castalia

- The simulation environment is build using Castalia 3.2 over OMNeT++ 4.6
- OMNeT++ is an object-oriented modular discrete event network simulation framework with a generic architecture
- Basic element of OMNeT++ is the module. Modules are simple or compound. Simple modules are lowest level of hierarchy and behaviour is programmed in C++. Compound modules are constructed by other simple or compound modules
- Modules are connected to other modules via gates that send and receive messages that are formed by arbitrary data structures
- A simulation model is composed by different simple and compound modules, that are interconnected in order to pass messages among them
- Castalia is a simulator designed for Wireless Sensor Network (WSN) and Body Area Networks (BAN) that are both characterized by low power embedded devices
- Castalia pre-defines OMNeT++ simple and compound modules to easily simulate WSNs and BANs. Modules might be created or modified to include new protocols or add new applications

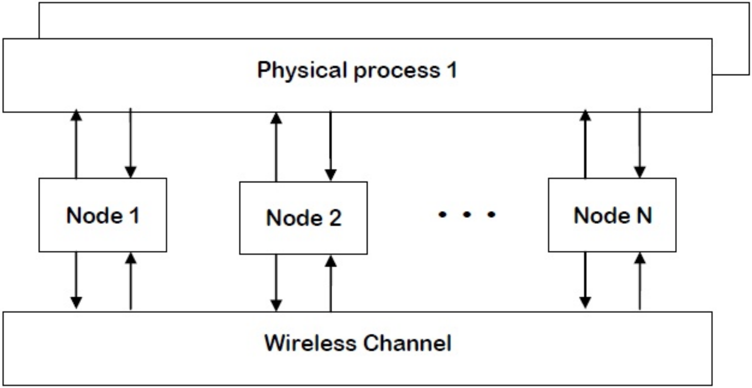
Simulation Environment

OMNeT++ & Castalia

- The simulation environment is build using Castalia 3.2 over OMNeT++ 4.6
- OMNeT++ is an object-oriented modular discrete event network simulation framework with a generic architecture
- Basic element of OMNeT++ is the module. Modules are simple or compound. Simple modules are lowest level of hierarchy and behaviour is programmed in C++. Compound modules are constructed by other simple or compound modules
- Modules are connected to other modules via gates that send and receive messages that are formed by arbitrary data structures
- A simulation model is composed by different simple and compound modules, that are interconnected in order to pass messages among them
- Castalia is a simulator designed for Wireless Sensor Network (WSN) and Body Area Networks (BAN) that are both characterized by low power embedded devices
- Castalia pre-defines OMNeT++ simple and compound modules to easily simulate WSNs and BANs. Modules might be created or modified to include new protocols or add new applications

Simulation Environment

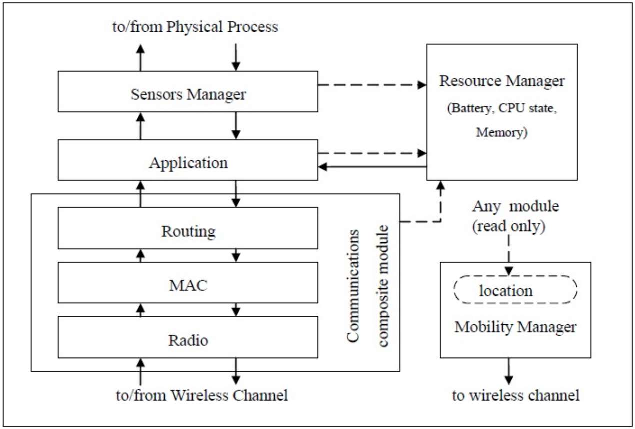
Basic Modules in Castalia



Castalia Basic Modules (source: Castalia user manual)

Simulation Environment

Node Module in Castalia



Castalia Node Module (source: Castalia user manual)

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Baseline Data

Sensor Networks Information

- Supplied data is constituted of a series of files containing the name and position for the nodes of different brands and the nodes data received during a given period of time
- The names of nodes and their position are defined in a single csv file with the structure shown bellow (actual brand names omitted):

```
x,y,Node_ID,Provider
2.1303350072,41.3852070789,MIC0001,BRAND_1
2.1300723767,41.3846812626,MIC0002,BRAND_1
(...)
2.1306075408,41.3851881008,T240714,BRAND_2
2.1299546229,41.3842444442,T240711,BRAND_2
(...)
2.1305602148,41.3840314145,71330,BRAND_3
2.1303745153,41.3838977008,71315,BRAND_3
(...)
```

Baseline Data

Sensor Networks Information

- Supplied data is constituted of a series of files containing the name and position for the nodes of different brands and the nodes data received during a given period of time
- The names of nodes and their position are defined in a single csv file with the structure shown bellow (actual brand names omitted):

```
x,y,Node_ID,Provider
2.1303350072,41.3852070789,MIC0001,BRAND_1
2.1300723767,41.3846812626,MIC0002,BRAND_1
(...)
2.1306075408,41.3851881008,T240714,BRAND_2
2.1299546229,41.3842444442,T240711,BRAND_2
(...)
2.1305602148,41.3840314145,71330,BRAND_3
2.1303745153,41.3838977008,71315,BRAND_3
(...)
```

Baseline Data

Sensor Networks Information (cont.)

- Files (one per brand) containing the received event messages follow the shown structure (see two examples for 2 different brands):

```
nodeId,"message_app",time,"timestamp_message"
MIC0003,"64.9",1442698020,"2015-09-19 23:27:00.000"
MIC0010,"65.2",1442697960,"2015-09-19 23:26:00.000"
MIC0007,"65.7",1442697960,"2015-09-19 23:26:00.000"
(...)
nodeId,"message_app",time,"timestamp_message"
"TA120-T240711-N","65.1","1442699927","2015-09-19 23:58:47.340"
"TA120-T240708-N","68.5","1442699921","2015-09-19 23:58:41.387"
"TA120-T241198-B",100,"1442699913","2015-09-19 23:58:33.466"
(...)
```

- Additionally, kml files used to represent spatial position of nodes in Google Earth are also available for all brands

Baseline Data

Sensor Networks Information (cont.)

- Files (one per brand) containing the received event messages follow the shown structure (see two examples for 2 different brands):

```
nodeId,"message_app",time,"timestamp_message"
MIC0003,"64.9",1442698020,"2015-09-19 23:27:00.000"
MIC0010,"65.2",1442697960,"2015-09-19 23:26:00.000"
MIC0007,"65.7",1442697960,"2015-09-19 23:26:00.000"
(...)
nodeId,"message_app",time,"timestamp_message"
"TA120-T240711-N","65.1","1442699927","2015-09-19 23:58:47.340"
"TA120-T240708-N","68.5","1442699921","2015-09-19 23:58:41.387"
"TA120-T241198-B",100,"1442699913","2015-09-19 23:58:33.466"
(...)
```

- Additionally, kml files used to represent spatial position of nodes in Google Earth are also available for all brands

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Simulation Set-up

Arranging Baseline Data

- First Python script developed to sort csv events file into increasing order and convert messages from nodes to numeric value in either case. Timecodes are also kept in seconds format:

```
$pyhton3 ordena.py <events_file_name>.csv
```

- The second script takes the nodes definition file, the number of intervals and the interval duration the simulation will have as input parameters:

```
$python3 setup_simulation.py <nodes_file>.csv #intervals duration
```

- Node positions are GPS coordinates. Simulator requires node placement with X,Y coordinates in meters. Conversion is obtained using the following approximation:

```
X[m] = 111.195*(long-long_origin)[deg]*COS(lat[rad])
Y[m] = 111.195*(lat-lat_origin)[deg]
```

- Simulation configuration file *omnetpp.ini* and data files for nodes are created by the second script
- Node 0 or sink node is automatically placed at the middle of the area determined by application nodes

Simulation Set-up

Arranging Baseline Data

- First Python script developed to sort csv events file into increasing order and convert messages from nodes to numeric value in either case. Timecodes are also kept in seconds format:

  ```
  $pyhton3 ordena.py <events_file_name>.csv
  ```

- The second script takes the nodes definition file, the number of intervals and the interval duration the simulation will have as input parameters:

  ```
  $python3 setup_simulation.py <nodes_file>.csv #intervals duration
  ```

- Node positions are GPS coordinates. Simulator requires node placement with X,Y coordinates in meters. Conversion is obtained using the following approximation:

  ```
  X[m] = 111.195*(long-long_origin)[deg]*COS(lat[rad])
  Y[m] = 111.195*(lat-lat_origin)[deg]
  ```

- Simulation configuration file *omnetpp.ini* and data files for nodes are created by the second script
- Node 0 or sink node is automatically placed at the middle of the area determined by application nodes

Simulation Set-up

Arranging Baseline Data

- First Python script developed to sort csv events file into increasing order and convert messages from nodes to numeric value in either case. Timecodes are also kept in seconds format:

  ```
  $pyhton3 ordena.py <events_file_name>.csv
  ```

- The second script takes the nodes definition file, the number of intervals and the interval duration the simulation will have as input parameters:

  ```
  $python3 setup_simulation.py <nodes_file>.csv #intervals duration
  ```

- Node positions are GPS coordinates. Simulator requires node placement with X,Y coordinates in meters. Conversion is obtained using the following approximation:

  ```
  X[m] = 111.195*(long-long_origin)[deg]*COS(lat[rad])
  Y[m] = 111.195*(lat-lat_origin)[deg]
  ```

- Simulation configuration file *omnetpp.ini* and data files for nodes are created by the second script
- Node 0 or sink node is automatically placed at the middle of the area determined by application nodes

Simulation Set-up

Arranging Baseline Data

- First Python script developed to sort csv events file into increasing order and convert messages from nodes to numeric value in either case. Timecodes are also kept in seconds format:

```
$pyhton3 ordena.py <events_file_name>.csv
```

- The second script takes the nodes definition file, the number of intervals and the interval duration the simulation will have as input parameters:

```
$python3 setup_simulation.py <nodes_file>.csv #intervals duration
```

- Node positions are GPS coordinates. Simulator requires node placement with X,Y coordinates in meters. Conversion is obtained using the following approximation:

```
X[m] = 111.195*(long-long_origin)[deg]*COS(lat[rad])
Y[m] = 111.195*(lat-lat_origin)[deg]
```

- Simulation configuration file *omnetpp.ini* and data files for nodes are created by the second script
- Node 0 or sink node is automatically placed at the middle of the area determined by application nodes

Simulation Set-up

Arranging Baseline Data

- First Python script developed to sort csv events file into increasing order and convert messages from nodes to numeric value in either case. Timecodes are also kept in seconds format:

```
$pyhton3 ordena.py <events_file_name>.csv
```

- The second script takes the nodes definition file, the number of intervals and the interval duration the simulation will have as input parameters:

```
$python3 setup_simulation.py <nodes_file>.csv #intervals duration
```

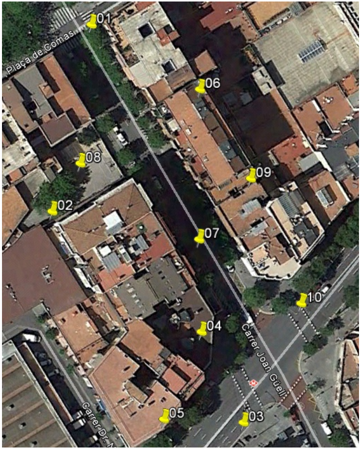
- Node positions are GPS coordinates. Simulator requires node placement with X,Y coordinates in meters. Conversion is obtained using the following approximation:

```
X[m] = 111.195*(long-long_origin)[deg]*COS(lat[rad])
Y[m] = 111.195*(lat-lat_origin)[deg]
```

- Simulation configuration file *omnetpp.ini* and data files for nodes are created by the second script
- Node 0 or sink node is automatically placed at the middle of the area determined by application nodes

Simulation Set-up

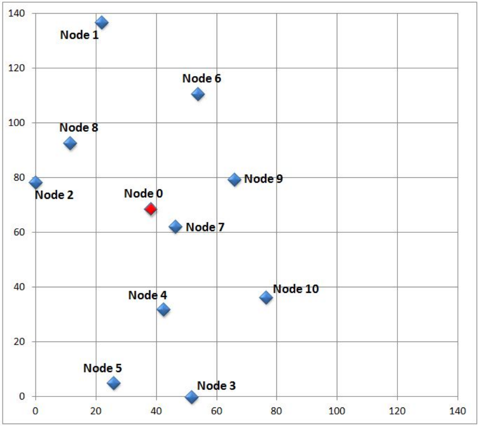
Baseline Data Example in a Map



Nodes placement on a map (from .kml file on Google Earth)

Simulation Set-up

Baseline Data Example in X,Y Coords.



Nodes and node 0 placement on an X,Y system (axes in m)

Outline

1. Introduction
2. Background and Related Work
    - Smart Cities and IoT
    - Wireless Sensor Networks
    - Pursued Objective
3. Development
    - Simulation Environment
    - Baseline Data
    - Simulation Set-up
    - Developments in Castalia
4. Analysis and Results
    - Initial Results and Improvements
    - Simulation and Output Files
5. Conclusions and Further Work
    - Conclusions
    - Further work
6. Acknowledgements

Developments in Castalia

Development of New Functionalities

- Castalia has several ways to define values at the Physical Process Module either as constant or defined by time and position
- The ability to read a file as a way to assign values to node sensors is not yet supported. A new function is required to be implemented
- The application module *ThroughputTest* is modified accordingly to accommodate external file reading:

```
int ThroughputTestNEW::readInputFile
(string file, map<long,int>& temps, map<long,int>& valors)
```

- A timer is added that is fired when the message has to be sent. The application forms the packet and sends it to the Communications Module:

```
toNetworkLayer(createGenericDataPacket(sensorRead,
numberTimesSensed), recipientAddress.c_str());
```

- If the number of available samples for a given node is achieved, the timer is cancelled and the node passes to idle state. This allows the simulation of nodes with diverse data amounts and cadences

Developments in Castalia

Development of New Functionalities

- Castalia has several ways to define values at the Physical Process Module either as constant or defined by time and position
- The ability to read a file as a way to assign values to node sensors is not yet supported. A new function is required to be implemented
- The application module *ThroughputTest* is modified accordingly to accommodate external file reading:

```
int ThroughputTestNEW::readInputFile
(string file, map<long,int>& temps, map<long,int>& valors)
```

- A timer is added that is fired when the message has to be sent. The application forms the packet and sends it to the Communications Module:

```
toNetworkLayer(createGenericDataPacket(sensorRead,
numberTimesSensed), recipientAddress.c_str());
```

- If the number of available samples for a given node is achieved, the timer is cancelled and the node passes to idle state. This allows the simulation of nodes with diverse data amounts and cadences

Developments in Castalia

Development of New Functionalities

- Castalia has several ways to define values at the Physical Process Module either as constant or defined by time and position
- The ability to read a file as a way to assign values to node sensors is not yet supported. A new function is required to be implemented
- The application module *ThroughputTest* is modified accordingly to accommodate external file reading:

```
int ThroughputTestNEW::readInputFile
(string file, map<long,int>& temps, map<long,int>& valors)
```

- A timer is added that is fired when the message has to be sent. The application forms the packet and sends it to the Communications Module:

```
toNetworkLayer(createGenericDataPacket(sensorRead,
numberTimesSensed), recipientAddress.c_str());
```

- If the number of available samples for a given node is achieved, the timer is cancelled and the node passes to idle state. This allows the simulation of nodes with diverse data amounts and cadences

Developments in Castalia

Development of New Functionalities

- Castalia has several ways to define values at the Physical Process Module either as constant or defined by time and position
- The ability to read a file as a way to assign values to node sensors is not yet supported. A new function is required to be implemented
- The application module *ThroughputTest* is modified accordingly to accommodate external file reading:

  ```
  int ThroughputTestNEW::readInputFile
  (string file, map<long,int>& temps, map<long,int>& valors)
  ```

- A timer is added that is fired when the message has to be sent. The application forms the packet and sends it to the Communications Module:

  ```
  toNetworkLayer(createGenericDataPacket(sensorRead,
  numberTimesSensed), recipientAddress.c_str());
  ```

- If the number of available samples for a given node is achieved, the timer is cancelled and the node passes to idle state. This allows the simulation of nodes with diverse data amounts and cadences

Developments in Castalia

Development of New Functionalities

- Castalia has several ways to define values at the Physical Process Module either as constant or defined by time and position
- The ability to read a file as a way to assign values to node sensors is not yet supported. A new function is required to be implemented
- The application module *ThroughputTest* is modified accordingly to accommodate external file reading:

  ```
  int ThroughputTestNEW::readInputFile
  (string file, map<long,int>& temps, map<long,int>& valors)
  ```

- A timer is added that is fired when the message has to be sent. The application forms the packet and sends it to the Communications Module:

  ```
  toNetworkLayer(createGenericDataPacket(sensorRead,
  numberTimesSensed), recipientAddress.c_str());
  ```

- If the number of available samples for a given node is achieved, the timer is cancelled and the node passes to idle state. This allows the simulation of nodes with diverse data amounts and cadences

Developments in Castalia

Development of New Functionalities (cont.)

- The attacker node is defined as the higher Id node. Original application timer is modified to serve as the attacker timer according to the attacker node *packet_rate* parameter
- Attacker node sends data packets with value 0 to broadcast and follows the chosen attack definition contained into *omnetpp.ini* configuration file

```
toNetworkLayer(createGenericDataPacket(0,dataSN),
BROADCAST_NETWORK_ADDRESS);
```

- The *Application.startupDelay* parameter is used together with the *sim-time-limit* to define the requested intervals into *omnetpp.ini* file:

```
[Config Interval0]
SN.node[*].Application.startupDelay = 0
sim-time-limit = 3600s
[Config Interval1]
SN.node[*].Application.startupDelay = 3600
sim-time-limit = 7200s
```

Developments in Castalia

Development of New Functionalities (cont.)

- The attacker node is defined as the higher Id node. Original application timer is modified to serve as the attacker timer according to the attacker node *packet_rate* parameter
- Attacker node sends data packets with value 0 to broadcast and follows the chosen attack definition contained into *omnetpp.ini* configuration file

  ```
  toNetworkLayer(createGenericDataPacket(0,dataSN),
  BROADCAST_NETWORK_ADDRESS);
  ```

- The *Application.startupDelay* parameter is used together with the *sim-time-limit* to define the requested intervals into *omnetpp.ini* file:

  ```
  [Config Interval0]
  SN.node[*].Application.startupDelay = 0
  sim-time-limit = 3600s
  [Config Interval1]
  SN.node[*].Application.startupDelay = 3600
  sim-time-limit = 7200s
  ```

Developments in Castalia

Development of New Functionalities (cont.)

- The attacker node is defined as the higher Id node. Original application timer is modified to serve as the attacker timer according to the attacker node *packet_rate* parameter
- Attacker node sends data packets with value 0 to broadcast and follows the chosen attack definition contained into *omnetpp.ini* configuration file

  ```
  toNetworkLayer(createGenericDataPacket(0,dataSN),
  BROADCAST_NETWORK_ADDRESS);
  ```

- The *Application.startupDelay* parameter is used together with the *sim-time-limit* to define the requested intervals into *omnetpp.ini* file:

  ```
  [Config Interval0]
  SN.node[*].Application.startupDelay = 0
  sim-time-limit = 3600s
  [Config Interval1]
  SN.node[*].Application.startupDelay = 3600
  sim-time-limit = 7200s
  ```

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Initial Results and Improvements

Simulation Results. Scenario 1

- A first scenario is configured for 10 application nodes defining one interval of 4 hours. Nodes have TMAC as MAC protocol and no attack is applied

```
[Config TMAC]
SN.node[0..10].Communication.MACProtocolName = "TMAC"
SN.node[0..10].Communication.MAC.phyDataRate = 250
SN.node[0..10].Communication.MAC.maxTxRetries = 5
SN.node[0..10].Communication.MAC.waitTimeout = 5
SN.node[0..10].Communication.MAC.collisionResolution = 0
[Config NoAttack]
SN.node[11].Application.packet_rate = 0
SN.node[11].Communication.Radio.TxOutputPower = "-15dBm"
SN.node[11].Communication.MACProtocolName = "TunableMAC"
```

- Results are shown in the next table:

	node 2	node 4	node 7	node 8
#received	4	31	56	43
Rec. rate	0.06	1	0.97	0.94

Initial Results and Improvements

Simulation Results. Scenario 1

- A first scenario is configured for 10 application nodes defining one interval of 4 hours. Nodes have TMAC as MAC protocol and no attack is applied

```
[Config TMAC]
SN.node[0..10].Communication.MACProtocolName = "TMAC"
SN.node[0..10].Communication.MAC.phyDataRate = 250
SN.node[0..10].Communication.MAC.maxTxRetries = 5
SN.node[0..10].Communication.MAC.waitTimeout = 5
SN.node[0..10].Communication.MAC.collisionResolution = 0
[Config NoAttack]
SN.node[11].Application.packet_rate = 0
SN.node[11].Communication.Radio.TxOutputPower = "-15dBm"
SN.node[11].Communication.MACProtocolName = "TunableMAC"
```

- Results are shown in the next table:

	node 2	node 4	node 7	node 8
#received	4	31	56	43
Rec. rate	0.06	1	0.97	0.94

Initial Results and Improvements

Simulation Results. Scenario 2

- Nodes are power limited and some of them are away from sink node (node 0). In scenario 1, many packets are lost
- A second scenario is composed adding routing information to nodes. Data added to the initial csv file and compiled with modified scripts to create configuration file with routing information [Config Multihop]
- Modified *ThroughputTest* is adapted accordingly and the packet structure is also modified to accommodate the origin node to every message
- New simulation with Multihop configuration improves receptions rates:

	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	19	62	44	30	79	16	57	45
Rec. rate	0.45	0.97	0.62	0.97	0.9	0.57	0.98	0.98

Initial Results and Improvements

Simulation Results. Scenario 2

- Nodes are power limited and some of them are away from sink node (node 0). In scenario 1, many packets are lost
- A second scenario is composed adding routing information to nodes. Data added to the initial csv file and compiled with modified scripts to create configuration file with routing information [Config Multihop]
- Modified *ThroughputTest* is adapted accordingly and the packet structure is also modified to accommodate the origin node to every message
- New simulation with Multihop configuration improves receptions rates:

	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	19	62	44	30	79	16	57	45
Rec. rate	0.45	0.97	0.62	0.97	0.9	0.57	0.98	0.98

Initial Results and Improvements

Simulation Results. Scenario 2

- Nodes are power limited and some of them are away from sink node (node 0). In scenario 1, many packets are lost
- A second scenario is composed adding routing information to nodes. Data added to the initial csv file and compiled with modified scripts to create configuration file with routing information [Config Multihop]
- Modified *ThroughputTest* is adapted accordingly and the packet structure is also modified to accommodate the origin node to every message
- New simulation with Multihop configuration improves receptions rates:

	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	19	62	44	30	79	16	57	45
Rec. rate	0.45	0.97	0.62	0.97	0.9	0.57	0.98	0.98

Initial Results and Improvements

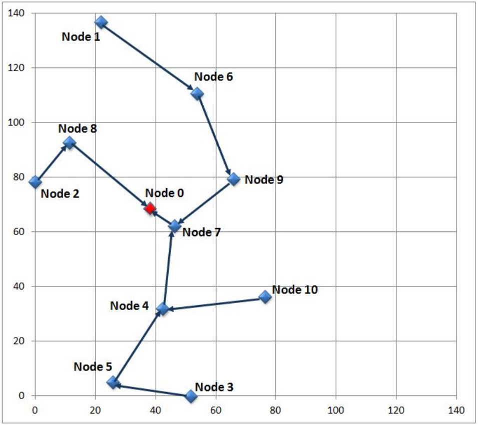
Simulation Results. Scenario 2

- Nodes are power limited and some of them are away from sink node (node 0). In scenario 1, many packets are lost
- A second scenario is composed adding routing information to nodes. Data added to the initial csv file and compiled with modified scripts to create configuration file with routing information [Config Multihop]
- Modified *ThroughputTest* is adapted accordingly and the packet structure is also modified to accommodate the origin node to every message
- New simulation with Multihop configuration improves receptions rates:

	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	19	62	44	30	79	16	57	45
Rec. rate	0.45	0.97	0.62	0.97	0.9	0.57	0.98	0.98

Initial Results and Improvements

Simulation Results. Scenario 2. Map of Routing



Nodes and node 0 placement and routing (axes in m)

Initial Results and Improvements

Simulation Results. Scenario 2. Additional Comments

- Node 9 has no packets to send since events file had no messages from this node
- Node 10 does not achieve to send packets to node 0 due to radio congestion at node 4
- Node 4 does not receive the packet from node 10 and only forwards packets received from node 5 to next node. When node 10 sends packets to node 4, the radio of node 4 is not ready
- Node 10 retries sending packet to node 4 according to the defined maximum number of retries and finally discards sending the current packet
- Congestion in certain nodes has been a big issue not totally solved in order to completely emulate the original network with the supplied event files
- Adding intermediate nodes has been discarded because no information is available about the existence of any additional nodes in the actual network. Simulations showed that congestion problem prevails or even increases with intermediate nodes

Initial Results and Improvements

Simulation Results. Scenario 2. Additional Comments

- Node 9 has no packets to send since events file had no messages from this node
- Node 10 does not achieve to send packets to node 0 due to radio congestion at node 4
- Node 4 does not receive the packet from node 10 and only forwards packets received from node 5 to next node. When node 10 sends packets to node 4, the radio of node 4 is not ready
- Node 10 retries sending packet to node 4 according to the defined maximum number of retries and finally discards sending the current packet
- Congestion in certain nodes has been a big issue not totally solved in order to completely emulate the original network with the supplied event files
- Adding intermediate nodes has been discarded because no information is available about the existence of any additional nodes in the actual network. Simulations showed that congestion problem prevails or even increases with intermediate nodes

Initial Results and Improvements

Simulation Results. Scenario 2. Additional Comments

- Node 9 has no packets to send since events file had no messages from this node
- Node 10 does not achieve to send packets to node 0 due to radio congestion at node 4
- Node 4 does not receive the packet from node 10 and only forwards packets received from node 5 to next node. When node 10 sends packets to node 4, the radio of node 4 is not ready
- Node 10 retries sending packet to node 4 according to the defined maximum number of retries and finally discards sending the current packet
- Congestion in certain nodes has been a big issue not totally solved in order to completely emulate the original network with the supplied event files
- Adding intermediate nodes has been discarded because no information is available about the existence of any additional nodes in the actual network. Simulations showed that congestion problem prevails or even increases with intermediate nodes

Initial Results and Improvements

Simulation Results. Scenario 2. Additional Comments

- Node 9 has no packets to send since events file had no messages from this node
- Node 10 does not achieve to send packets to node 0 due to radio congestion at node 4
- Node 4 does not receive the packet from node 10 and only forwards packets received from node 5 to next node. When node 10 sends packets to node 4, the radio of node 4 is not ready
- Node 10 retries sending packet to node 4 according to the defined maximum number of retries and finally discards sending the current packet
- Congestion in certain nodes has been a big issue not totally solved in order to completely emulate the original network with the supplied event files
- Adding intermediate nodes has been discarded because no information is available about the existence of any additional nodes in the actual network. Simulations showed that congestion problem prevails or even increases with intermediate nodes

Initial Results and Improvements

Simulation Results. Scenario 2. Additional Comments

- Node 9 has no packets to send since events file had no messages from this node
- Node 10 does not achieve to send packets to node 0 due to radio congestion at node 4
- Node 4 does not receive the packet from node 10 and only forwards packets received from node 5 to next node. When node 10 sends packets to node 4, the radio of node 4 is not ready
- Node 10 retries sending packet to node 4 according to the defined maximum number of retries and finally discards sending the current packet
- Congestion in certain nodes has been a big issue not totally solved in order to completely emulate the original network with the supplied event files
- Adding intermediate nodes has been discarded because no information is available about the existence of any additional nodes in the actual network. Simulations showed that congestion problem prevails or even increases with intermediate nodes

Initial Results and Improvements

Simulation Results. Scenario 2. Additional Comments

- Node 9 has no packets to send since events file had no messages from this node
- Node 10 does not achieve to send packets to node 0 due to radio congestion at node 4
- Node 4 does not receive the packet from node 10 and only forwards packets received from node 5 to next node. When node 10 sends packets to node 4, the radio of node 4 is not ready
- Node 10 retries sending packet to node 4 according to the defined maximum number of retries and finally discards sending the current packet
- Congestion in certain nodes has been a big issue not totally solved in order to completely emulate the original network with the supplied event files
- Adding intermediate nodes has been discarded because no information is available about the existence of any additional nodes in the actual network. Simulations showed that congestion problem prevails or even increases with intermediate nodes

Initial Results and Improvements

Simulation Results. Scenario 3

- Previous simulation configuration is now repeated adding an attack that is designed to affect the transmission medium
- The attacker node uses a different protocol than the rest of nodes because it is supposed not related to the existing network and free to transmit independently of the channel state

```
[Config JammingAttack]
SN.node[11].Communication.Radio.TxOutputPower = "0dBm"
SN.node[11].Application.constantDataPayload = 277
SN.node[11].Communication.Routing.maxNetFrameSize = 2500
SN.node[11].Communication.MAC.maxMACFrameSize = 2500
SN.node[11].Communication.Radio.maxPhyFrameSize = 2500
SN.node[11].Communication.MACProtocolName = "TunableMAC"
SN.node[11].Application.packet_rate = 75
SN.node[11].xCoor = 50
SN.node[11].yCoor = 80
```

- This transmission medium attack might be assimilated to a Jamming Attack. Attacker node is manually placed at desired coordinates

Initial Results and Improvements

Simulation Results. Scenario 3

- Previous simulation configuration is now repeated adding an attack that is designed to affect the transmission medium
- The attacker node uses a different protocol than the rest of nodes because it is supposed not related to the existing network and free to transmit independently of the channel state

```
[Config JammingAttack]
SN.node[11].Communication.Radio.TxOutputPower = "0dBm"
SN.node[11].Application.constantDataPayload = 277
SN.node[11].Communication.Routing.maxNetFrameSize = 2500
SN.node[11].Communication.MAC.maxMACFrameSize = 2500
SN.node[11].Communication.Radio.maxPhyFrameSize = 2500
SN.node[11].Communication.MACProtocolName = "TunableMAC"
SN.node[11].Application.packet_rate = 75
SN.node[11].xCoor = 50
SN.node[11].yCoor = 80
```

- This transmission medium attack might be assimilated to a Jamming Attack. Attacker node is manually placed at desired coordinates

Initial Results and Improvements

Simulation Results. Scenario 3

- Previous simulation configuration is now repeated adding an attack that is designed to affect the transmission medium
- The attacker node uses a different protocol than the rest of nodes because it is supposed not related to the existing network and free to transmit independently of the channel state

```
[Config JammingAttack]
SN.node[11].Communication.Radio.TxOutputPower = "0dBm"
SN.node[11].Application.constantDataPayload = 277
SN.node[11].Communication.Routing.maxNetFrameSize = 2500
SN.node[11].Communication.MAC.maxMACFrameSize = 2500
SN.node[11].Communication.Radio.maxPhyFrameSize = 2500
SN.node[11].Communication.MACProtocolName = "TunableMAC"
SN.node[11].Application.packet_rate = 75
SN.node[11].xCoor = 50
SN.node[11].yCoor = 80
```

- This transmission medium attack might be assimilated to a Jamming Attack. Attacker node is manually placed at desired coordinates

Initial Results and Improvements

Simulation Results. Scenario 3. Results

- Simulation results show the effect of the defined attack. Tables bellow compare both simulation results:

no attack	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	19	62	44	30	79	16	57	45
Rec. rate	0.45	0.97	0.62	0.97	0.9	0.57	0.98	0.98

Jamming	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	17	59	23	30	24	15	58	46
Rec. rate	0.4	0.92	0.32	0.97	0.27	0.54	1	1

- The number of received packets and the corresponding reception rate are clearly reduced when the attack is applied (see nodes 3 & 5)
- Additional configurations might be used to emulate other attacks in various conditions focusing selected areas of the network, the sink node or sub-networks corresponding to an specific brand or functionality

Initial Results and Improvements

Simulation Results. Scenario 3. Results

- Simulation results show the effect of the defined attack. Tables bellow compare both simulation results:

no attack	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	19	62	44	30	79	16	57	45
Rec. rate	0.45	0.97	0.62	0.97	0.9	0.57	0.98	0.98

Jamming	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	17	59	23	30	24	15	58	46
Rec. rate	0.4	0.92	0.32	0.97	0.27	0.54	1	1

- The number of received packets and the corresponding reception rate are clearly reduced when the attack is applied (see nodes 3 & 5)
- Additional configurations might be used to emulate other attacks in various conditions focusing selected areas of the network, the sink node or sub-networks corresponding to an specific brand or functionality

Initial Results and Improvements

Simulation Results. Scenario 3. Results

- Simulation results show the effect of the defined attack. Tables bellow compare both simulation results:

no attack	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	19	62	44	30	79	16	57	45
Rec. rate	0.45	0.97	0.62	0.97	0.9	0.57	0.98	0.98

Jamming	nd 1	nd 2	nd 3	nd 4	nd 5	nd 6	nd 7	nd 8
#received	17	59	23	30	24	15	58	46
Rec. rate	0.4	0.92	0.32	0.97	0.27	0.54	1	1

- The number of received packets and the corresponding reception rate are clearly reduced when the attack is applied (see nodes 3 & 5)
- Additional configurations might be used to emulate other attacks in various conditions focusing selected areas of the network, the sink node or sub-networks corresponding to an specific brand or functionality

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Simulation and Output Files

Last Development

- The Final goal was to prepare simulation result files in a convenient format to be used as input in anomaly detectors
- Detectors will be trained with simulation data coming from several intervals to emulate the real network formed by different nodes from various brands
- Output files will be the result of simulations run in different scenarios. The objective is to see if the system is able to detect the anomalies when they occur and even classifying the type of attack
- A single table for every simulation is created containing one row per interval and scenario (no attack, attack of type 1, attack of type 2, etc.) and columns gathering output data about simulation parameters for every node
- This later analysis is out of the scope of this work. However, the required provisions to compile the *CastaliaResults* csv file represented the last development of the present work

Simulation and Output Files

Last Development

- The Final goal was to prepare simulation result files in a convenient format to be used as input in anomaly detectors

- Detectors will be trained with simulation data coming from several intervals to emulate the real network formed by different nodes from various brands

- Output files will be the result of simulations run in different scenarios. The objective is to see if the system is able to detect the anomalies when they occur and even classifying the type of attack

- A single table for every simulation is created containing one row per interval and scenario (no attack, attack of type 1, attack of type 2, etc.) and columns gathering output data about simulation parameters for every node

- This later analysis is out of the scope of this work. However, the required provisions to compile the *CastaliaResults* csv file represented the last development of the present work

Simulation and Output Files

Last Development

- The Final goal was to prepare simulation result files in a convenient format to be used as input in anomaly detectors
- Detectors will be trained with simulation data coming from several intervals to emulate the real network formed by different nodes from various brands
- Output files will be the result of simulations run in different scenarios. The objective is to see if the system is able to detect the anomalies when they occur and even classifying the type of attack
- A single table for every simulation is created containing one row per interval and scenario (no attack, attack of type 1, attack of type 2, etc.) and columns gathering output data about simulation parameters for every node
- This later analysis is out of the scope of this work. However, the required provisions to compile the *CastaliaResults* csv file represented the last development of the present work

Simulation and Output Files

Last Development

- The Final goal was to prepare simulation result files in a convenient format to be used as input in anomaly detectors
- Detectors will be trained with simulation data coming from several intervals to emulate the real network formed by different nodes from various brands
- Output files will be the result of simulations run in different scenarios. The objective is to see if the system is able to detect the anomalies when they occur and even classifying the type of attack
- A single table for every simulation is created containing one row per interval and scenario (no attack, attack of type 1, attack of type 2, etc.) and columns gathering output data about simulation parameters for every node
- This later analysis is out of the scope of this work. However, the required provisions to compile the *CastaliaResults* csv file represented the last development of the present work

Simulation and Output Files

Last Development

- The Final goal was to prepare simulation result files in a convenient format to be used as input in anomaly detectors
- Detectors will be trained with simulation data coming from several intervals to emulate the real network formed by different nodes from various brands
- Output files will be the result of simulations run in different scenarios. The objective is to see if the system is able to detect the anomalies when they occur and even classifying the type of attack
- A single table for every simulation is created containing one row per interval and scenario (no attack, attack of type 1, attack of type 2, etc.) and columns gathering output data about simulation parameters for every node
- This later analysis is out of the scope of this work. However, the required provisions to compile the *CastaliaResults* csv file represented the last development of the present work

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

```
$python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

  ```
  $python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
  ```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

  ```
  $python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
  ```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

  ```
  $python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
  ```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

  ```
  $python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
  ```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

  ```
  $python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
  ```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

```
$python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

  ```
  $python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
  ```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

```
$python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
    - Packets received at node 0
    - Reception loss at node 0
    - Reception rate at node 0
    - Consumed energy during simulation
    - Radio Tx packets
    - MAC sent packets breakdown attending to values for:
      ACK, CTS, DATA, RTS & SYNC
    - Radio RX packets breakdown attending to for:
      Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Simulation and Output Files

Last Development (cont.)

- Last Python script has also been created on that purpose:

  ```
  $python3 convertResultsFile.py <sim_file> #Intervals #Scenarios
  ```

- Previous script will collect the following parameters for every node in the simulation from the simulator results file:
  - Packets received at node 0
  - Reception loss at node 0
  - Reception rate at node 0
  - Consumed energy during simulation
  - Radio Tx packets
  - MAC sent packets breakdown attending to values for:
    ACK, CTS, DATA, RTS & SYNC
  - Radio RX packets breakdown attending to for:
    Failed with NO interference, Failed with interference, Failed, below sensitivity, Failed, non RX state, Received despite interference & Received with NO interference
- The results file is given into a new text comma separated file (csv)

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Conclusions

- This project has shown the feasibility of adapting Castalia to simulate actual deployed WSNs taking supplied data from external files to construct and send messages
- Multihop routing configuration improves the reception rate from distant nodes compensating the lack of node transmission power
- Congestion prevails due to <radio not in RX> state when packets are sent from neighbour nodes
- The addition of routing nodes has finally been discarded due to the absence of information regarding networks topology. Simulations showed congestion also occurring in additional nodes
- Effect of applying attacks is noticeable from the collected simulation data and parameters as reception rate are modified in affected nodes

Conclusions

- This project has shown the feasibility of adapting Castalia to simulate actual deployed WSNs taking supplied data from external files to construct and send messages
- Multihop routing configuration improves the reception rate from distant nodes compensating the lack of node transmission power
- Congestion prevails due to <radio not in RX> state when packets are sent from neighbour nodes
- The addition of routing nodes has finally been discarded due to the absence of information regarding networks topology. Simulations showed congestion also occurring in additional nodes
- Effect of applying attacks is noticeable from the collected simulation data and parameters as reception rate are modified in affected nodes

Conclusions

- This project has shown the feasibility of adapting Castalia to simulate actual deployed WSNs taking supplied data from external files to construct and send messages
- Multihop routing configuration improves the reception rate from distant nodes compensating the lack of node transmission power
- Congestion prevails due to <radio not in RX> state when packets are sent from neighbour nodes
- The addition of routing nodes has finally been discarded due to the absence of information regarding networks topology. Simulations showed congestion also occurring in additional nodes
- Effect of applying attacks is noticeable from the collected simulation data and parameters as reception rate are modified in affected nodes

Conclusions

- This project has shown the feasibility of adapting Castalia to simulate actual deployed WSNs taking supplied data from external files to construct and send messages
- Multihop routing configuration improves the reception rate from distant nodes compensating the lack of node transmission power
- Congestion prevails due to <radio not in RX> state when packets are sent from neighbour nodes
- The addition of routing nodes has finally been discarded due to the absence of information regarding networks topology. Simulations showed congestion also occurring in additional nodes
- Effect of applying attacks is noticeable from the collected simulation data and parameters as reception rate are modified in affected nodes

Conclusions

- This project has shown the feasibility of adapting Castalia to simulate actual deployed WSNs taking supplied data from external files to construct and send messages
- Multihop routing configuration improves the reception rate from distant nodes compensating the lack of node transmission power
- Congestion prevails due to <radio not in RX> state when packets are sent from neighbour nodes
- The addition of routing nodes has finally been discarded due to the absence of information regarding networks topology. Simulations showed congestion also occurring in additional nodes
- Effect of applying attacks is noticeable from the collected simulation data and parameters as reception rate are modified in affected nodes

Outline

1. Introduction
2. Background and Related Work
   - Smart Cities and IoT
   - Wireless Sensor Networks
   - Pursued Objective
3. Development
   - Simulation Environment
   - Baseline Data
   - Simulation Set-up
   - Developments in Castalia
4. Analysis and Results
   - Initial Results and Improvements
   - Simulation and Output Files
5. Conclusions and Further Work
   - Conclusions
   - Further work
6. Acknowledgements

Further work

- After the developments in the present design and creation project, final findings and results do open new lines of activity that could be undertaken afterwards:
    - Use current developments to conduct systematic simulations around baseline data to feed anomaly detectors with different scenarios to evaluate detection capacity
    - Define different attacks to test effects upon position, MAC, TX power and mobility. Include multiple attacker nodes and define the systematic rules to place attacker nodes on the field
    - Revisit congestion issues to fully understand packets loss and take the necessary provisions to achieve higher reception rates in dense networks
    - Improve simulation repetitiveness in cases where adding a passive node affects simulation results, due to slightly different initialization values for several simulation parameters
    - Implement Collection Tree Protocol (CTP). Although a Castalia implementation of CTP exists, it is not maintained since 2012. Some development effort might be required to adapt it to run under the latest versions of Castalia and OMNeT++

Further work

- After the developments in the present design and creation project, final findings and results do open new lines of activity that could be undertaken afterwards:
    - Use current developments to conduct systematic simulations around baseline data to feed anomaly detectors with different scenarios to evaluate detection capacity
    - Define different attacks to test effects upon position, MAC, TX power and mobility. Include multiple attacker nodes and define the systematic rules to place attacker nodes on the field
    - Revisit congestion issues to fully understand packets loss and take the necessary provisions to achieve higher reception rates in dense networks
    - Improve simulation repetitiveness in cases where adding a passive node affects simulation results, due to slightly different initialization values for several simulation parameters
    - Implement Collection Tree Protocol (CTP). Although a Castalia implementation of CTP exists, it is not maintained since 2012. Some development effort might be required to adapt it to run under the latest versions of Castalia and OMNeT++

Further work

- After the developments in the present design and creation project, final findings and results do open new lines of activity that could be undertaken afterwards:
  - Use current developments to conduct systematic simulations around baseline data to feed anomaly detectors with different scenarios to evaluate detection capacity
  - Define different attacks to test effects upon position, MAC, TX power and mobility. Include multiple attacker nodes and define the systematic rules to place attacker nodes on the field
  - Revisit congestion issues to fully understand packets loss and take the necessary provisions to achieve higher reception rates in dense networks
  - Improve simulation repetitiveness in cases where adding a passive node affects simulation results, due to slightly different initialization values for several simulation parameters
  - Implement Collection Tree Protocol (CTP). Although a Castalia implementation of CTP exists, it is not maintained since 2012. Some development effort might be required to adapt it to run under the latest versions of Castalia and OMNeT++

Further work

- After the developments in the present design and creation project, final findings and results do open new lines of activity that could be undertaken afterwards:
  - Use current developments to conduct systematic simulations around baseline data to feed anomaly detectors with different scenarios to evaluate detection capacity
  - Define different attacks to test effects upon position, MAC, TX power and mobility. Include multiple attacker nodes and define the systematic rules to place attacker nodes on the field
  - Revisit congestion issues to fully understand packets loss and take the necessary provisions to achieve higher reception rates in dense networks
  - Improve simulation repetitiveness in cases where adding a passive node affects simulation results, due to slightly different initialization values for several simulation parameters
  - Implement Collection Tree Protocol (CTP). Although a Castalia implementation of CTP exists, it is not maintained since 2012. Some development effort might be required to adapt it to run under the latest versions of Castalia and OMNeT++

Further work

- After the developments in the present design and creation project, final findings and results do open new lines of activity that could be undertaken afterwards:
    - Use current developments to conduct systematic simulations around baseline data to feed anomaly detectors with different scenarios to evaluate detection capacity
    - Define different attacks to test effects upon position, MAC, TX power and mobility. Include multiple attacker nodes and define the systematic rules to place attacker nodes on the field
    - Revisit congestion issues to fully understand packets loss and take the necessary provisions to achieve higher reception rates in dense networks
    - Improve simulation repetitiveness in cases where adding a passive node affects simulation results, due to slightly different initialization values for several simulation parameters
    - Implement Collection Tree Protocol (CTP). Although a Castalia implementation of CTP exists, it is not maintained since 2012. Some development effort might be required to adapt it to run under the latest versions of Castalia and OMNeT++

Further work

- After the developments in the present design and creation project, final findings and results do open new lines of activity that could be undertaken afterwards:
    - Use current developments to conduct systematic simulations around baseline data to feed anomaly detectors with different scenarios to evaluate detection capacity
    - Define different attacks to test effects upon position, MAC, TX power and mobility. Include multiple attacker nodes and define the systematic rules to place attacker nodes on the field
    - Revisit congestion issues to fully understand packets loss and take the necessary provisions to achieve higher reception rates in dense networks
    - Improve simulation repetitiveness in cases where adding a passive node affects simulation results, due to slightly different initialization values for several simulation parameters
    - Implement Collection Tree Protocol (CTP). Although a Castalia implementation of CTP exists, it is not maintained since 2012. Some development effort might be required to adapt it to run under the latest versions of Castalia and OMNeT++

Acknowledgements

The author would like to thank UOC University and specially KISON research group professors and consultants as well as the MISTIC program students for their help, collaboration and constructive comments on the present work.