Wireless Sensor Network, 2012, 4, 177-184
http://dx.doi.org/10.4236/wsn.2012.47026 Published Online July 2012 (http://www.SciRP.org/journal/wsn)

Problem of Standardization of Configuration Data in Wireless Sensor Networks: The Case of Routing Protocols

Thomas Ndie Djotio, Judith Soulamite Noutat Nouho

Management and Security of Network and System Services, University of Yaoundé, Yaoundé, Cameroon
Email: {tdjotio, noutatsoulamite}@gmail.com

Received May 12, 2012; revised June 10, 2012; accepted June 28, 2012

ABSTRACT

The exploitation of Wireless Sensor Networks (WSNs) is constrained by limited power, low computing power and storage and short-range radio transmission. Many routing protocols respecting these constraints were developed but, it still lacks formal and standardized solutions being able to help in their configuration. The configuration management that responds to this concern is very important in this type of network. It consists of the definition of data models to configure and is very necessary for the good network performance. Tangible results were obtained in traditional networks with the emergence of NETCONF and YANG standards, but on the best of our humble knowledge there are none yet in WSNs. We propose in this paper WSN-routing-protocol, a YANG data model for routing protocols configuration in WSNs. Following our model, we propose two YANG configuration data models based on the latter: they are respectively aodv for AODV and rpl for RPL.

Keywords: Sensor Network; Routing Protocols; Configuration Data; Model; YANG

1. Introduction

Technological advances in microelectronics, wireless communications, coupled with the efforts of miniaturization and cost reduction of electronic components production, have enabled the development of new generations of wireless networks: the Wireless Sensor Networks (WSNs). These networks consist of a large number of small devices called sensors operate autonomously. Each sensor has a unit able of sensing physical data (temperature, moisture, gas…) from a target environment to monitor. These data, after a numerical-analogical transformation, are communicated to a base station called sink by radio waves through the network. The sink is responsible of data aggregation for the purpose of specific treatment of analysis [1]. WSNs can be very useful for military or civilian applications when it comes to collect and process information from the environment. For example, in the military field for monitoring the movement of enemy forces or estimating of damages before or after a military operation, in the medical field to monitor patients at home and in the environmental field for fires detection in public places. WSNs differ from traditional networks by their large node density, energy autonomy, their dynamic topology and low computing power and storage [2]. They are usually deployed in thousands in a sensing area. These areas are often hostile and inaccessible, making it difficult or impossible to replace batteries of sensor nodes. These constraints posed several challenges for WSNs design and management. These challenges include configuration management in such networks.

Researches around the configuration management in traditional networks led to the construction by the IETF (Internet Engineering Task Force) of standards YANG [3] and NETCONF [4]. YANG on one hand for the configuration and state data modeling handled by NETCONF which, on the other hand, allows the remote configuration management of network equipments. The YANG data models of interfaces, routing and system management were developed in the traditional networks. In the context of WSNs, the problem remains open with more complex and important issues. This problem is partly based on equipments and services (location, hardware and software configurations, states (on, off, busy, listen, etc.)) and partly on network operations (initialization, identification and handling). Based on NETCONF and YANG standards produced by the IETF which are gaining ground in traditional networks for configuration management, we ask ourselves the question of how to adapting these results in WSNs?

In this paper, we propose with its implementation WSN-routing-protocol, a YANG data model for configuring and managing routing protocols in WSNs. It inherits from the YANG model ietf-routing [5] available for performing same tasks only in traditional networks. More precisely, we take into account specific parameters for routing protocols in WSNs to define WSN-routing-protocol as the extension of ietf-routing in WSNs. The WSN-routing-protocol model results from a depth study of the most recurrent routing protocols and existing taxonomies in the literature. We developed a new taxonomy which consists of 7 classes and 19 subclasses and identified generic configuration data for each subclass of Protocol. These data are organized in groups of nodes in the meaning of YANG. Our model handles the types derived from standard models including ietf-yang-types [6] for yang types and ietf-inet-types [7] to define INET types. In Section 2, we make a state of the art of works related to the configuration management in networks. Section 3 focuses on WSN-routing-protocol, our configuration data model for routing protocols in WSNs. We propose its description and its formal specification in YANG language. In Section 4, we validate our model and propose respectively for AODV [8] and RPL [9] routing protocols two derived configuration data models: aodv and rpl. Before concluding our approach and presenting some prospects for our model, we discuss our results in Section 5.

2. Configuration Data Standardization: State of the Art

With an aim of standardizing networks configuration interfaces, it is important to define a set of data models. These models are defined in a specific language and are consumed or manipulated by protocols. Several Standards were defined for this purpose: SNMP [10] and ASN.1 on the one hand, NETCONF [4] and YANG [3] on the other hand.

2.1. SNMP and ASN.1

SNMP [10] (Simple Network Management Protocol) is a protocol that allows network administrators to manage network devices and diagnose network problems. To operate SNMP, management objects of managed equipment must be defined and stored in the MIB (Management Information Base) [11]. SNMP messages are described in ASN.1 (Abstract Syntax Notation 1). SMI (Structure of Management Information) [5] defines rules for describing administration's information using ASN.1 and so for objects stored in the MIB. K. Korte and A. Sehgal [12] defined management objects of the RPL routing protocol in Low Power and Lossy Networks. They use SMI mechanisms to describe the MIB module of nodes monitoring, implementing the RPL routing protocol. However, SNMP is more used in performance monitoring applications and almost not used for configuration data modifications because, it presents limits related to the use of UDP protocol and lacks security standard definition.

2.2. NETCONF and YANG

NETCONF (Network Configuration protocol) [4] has been standardized for the purpose of network equipment's configuration. It remotely allows installing, manipulating, and deleting the configuration on remote network equipment. It has been defined to cover the SNMP and CLI (Command-Line Interface) lacks in network configuration functions. The NETCONF protocol is remote procedure call (RPC) based. A client encodes a RPC message in XML and sends it to a server using a secure connection method. YANG [3] is the data modeling language used by NETCONF. Furthermore, YANG can be used to define events notification format emitted by network elements and allow defining the remote procedure calls signature that can invoke network elements via NETCONF protocol. To exploit NETCONF, configuration data models must be defined in YANG. Several models currently studied for standardization were defined in the traditional networks. We have for example: the data model (ietf-routing) for routing configuration defined by L. Lhotka [13], the data model (ietf-interfaces) for interfaces configuration defined by M. Bjorklund [14] and the data model (ietf-system) for system management defined by A. Bierman and M. Bjorklund [15]. Each model describes a basic YANG module for network equipment configuration.

However, on the best of our knowledge, no data model has been yet defined in WSNs. But, the inherent constraints to these types of networks such as limited resources require a correct and efficient configuration approach for sensors nodes.

3. Our Configuration Data Model in WSNs: WSN-Routing-Protocol

To design our WSN-routing-protocol data model, we start by proposing routing protocols taxonomies that we think are relevant to WSNs, resulting from a deep study of those proposed by researchers and industrial communities. Then, for each class and subclass of protocols, we have identified generic data that finally allowed us to formally specify our configuration data model in YANG. We use taxonomy because it allows us to do a collaborative study of information processing by grouping the various elements in classes or categories. This method allows reducing the area of study while treating a large number of elements at a time. A case by case study approach, for example studying each routing protocol individually, is more extensive and less visible. The taxonomy is scalable as new elements and new classes can be added at any time. For an example, for a new routing protocol, it is sufficient to determine its characteristics, to identify the classes that match those characteristics and to link the item as a member of these classes of taxonomy.

3.1. Taxonomy of Routing Protocols in WSNs

Many routing protocols developed for WSNs are proposed in the literature. However, it is difficult to study them individually. Besides that, several of these protocols have common characteristics and can be grouped into class. Thus in this part, we elaborated a taxonomy for grouping in classes protocols presenting common routing techniques. To achieve this, we studied the following taxonomies [16-21]. These different taxonomies more generally are for routing protocols comparison objective in WSNs. From these comparisons, we can obtain important features that must be considered when designning and evaluating new routing protocols in WSNs. In our case, we will exploit the features from our taxonomy to determine generic configuration data for routing protocols in WSNs. We find in existing taxonomies criteria such as mobility [19], flat topology, hierarchical [16-18, 20,21] and geographical [16], the data-centric [18,20,21] and nodes centric [21] communication paradigms, QoS and multiple paths operation modes, aggregation based and query-based [16,17,21]. However, these features are incomplete, because some new protocols and routing techniques implement new mechanisms that were not included in these taxonomies. For example, the study of RPL [12] routing protocol allowed us to identify a new type of topology, "DAG (Direct Acyclic Graph)". From the study of the 6 Low Pan stack [22], we identified a new class of routing protocol, "routing layer" that allows determining the routing level (adaptation or network) protocols using this protocol stack. The extended taxonomy resulting from the synthesis of existing taxonomies and our study is presented in **Figure 1**.

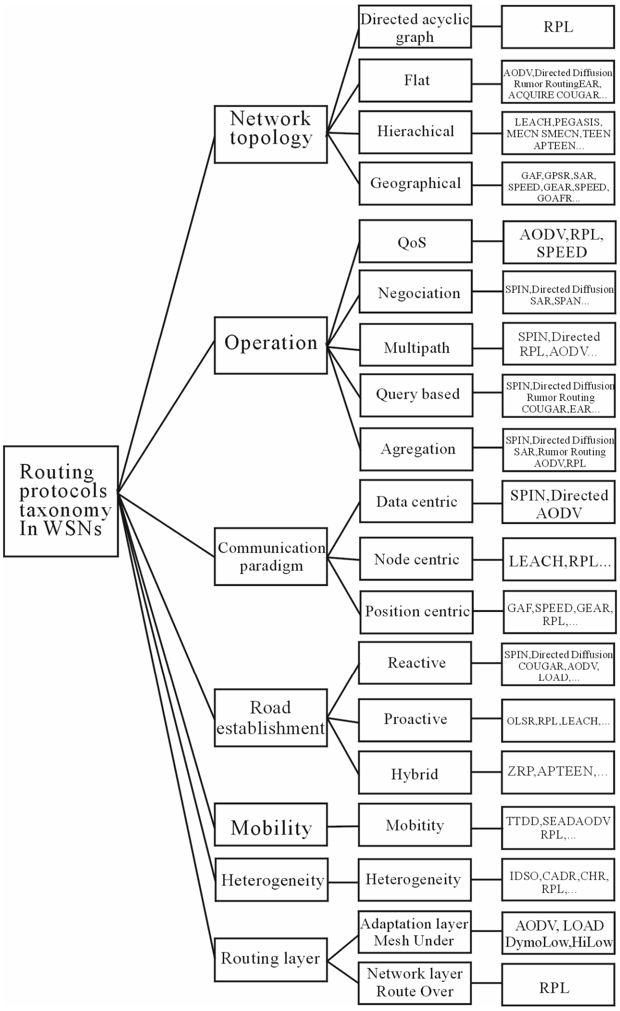

Figure 1. Taxonomy of routing protocols in wireless sensor networks.

This taxonomy includes 7 principle classes and 19 sub-classes and some examples of protocols. For each sub-class of protocols, we identified the specific configurable data.

3.2. Configuration Data Identification and Description

Configuration data consist of a set of parameters accessible in writing mode and necessary to transform a system from its initial state by default into its current state [4]. On the other hand, state data consist of a set of additional read-only data that allow obtaining state information (node state: battery level, power communication, etc., Link state: active, asleep, on, off, etc.) and statistics necessary for the proper functioning of the configured equipment and the network [4].

We will first describe the generic configuration information for all routing protocols. Then, for each class of routing protocol studied, we describe the generic configuration data and state data necessary for characterizing this class.

Our study led us to note that the following parameters are common to all routing protocols: they constitute to this effect our basic generic parameters.

Name: Specifies the name of the routing protocol.

Description: Gives a short description of the routing protocol.

Topology_Type: Indicates the topology type implemented by the protocol (flat, hierarchical, geographical, directed acyclic graph).

Operation_Type: Shows the type of operation of the routing protocol (QoS, negotiation, multipath, based query, aggregation).

Communication_Paradigm: Indicates the communication paradigm implemented by the node (data centric, node centric and position centric).

Method_Determining_Route: Shows how of establishment of the road (proactive, reactive, hybrid).

Mobility: Indicates whether the routing protocol supports node mobility.

Heterogeneity: Indicates whether the routing protocol supports the heterogeneity of nodes.

Routing_Layer: Indicates the routing layer protocol (layer adaptation or network layer).

3.2.1. Network Topology Based

The topology determines the organization of sensors in the network. There exist four topologies in routing protocols for WSNs: directed acyclic graph (DAG), flat, hierarchical and geographical. For each subclass, we have the following generic configuration data:

- DAG

  OCP (*Objective Code Point*): Identifies the objective function that the protocol must use.

  Metric_Data: Contains the order, content and code metrics.

  Node_Type: Defines the node type.

- Flat

  Hello_Interval: Shows the time after which a node must broadcast hello packets.

  Lifetime_Hello: Shows the time after which if a node does not receive a hello message from a neighbor node, it deletes it from its neighbor table.

- Hiérachical

  Function_of_Node: Determines the function of a node.

  Round_Interval: Shows the time after which the next cycle begins.

- Geographical

  Beacon_Interval: Specifies the time after which a node must broadcast control packets to indicate its position.

  Lifetime_Beacon: Shows the time after which if a node does not receive a message “beacon” of a neighboring node, it deletes it from its neighbor table.

  Signal_Power: Indicates the signal strength of the transmitting node.

3.2.2. Depending on the Protocol Operation

Based on routing protocols functionalities, there are four categories: on the QoS (Quality of Service) based routing, query-based routing, multi-path based routing, negotiation-based routing and aggregation based routing [14]. For each subclass, we identify the generic data following configuration:

- QoS-Based

  Delay_Required: This parameter specifies the delivery time required from start to finish.

  Threshold_Speed: This parameter defines the transmission speed threshold in the network.

  RPDR (*Required Paquet Delivery Ratio*): This parameter indicates the desired data delivery rate in the network.

  Level_Battery: This information given the battery level of a node.

- Query-Based

  Request_Wait_Time: This data indicates the time after which a node can rebroadcast a request which has not been answered.

  Request_Retries: This data fixes the number of times a node can retransmit a request that it does not get an answer.

- Multi-Path-Based

  Alternative_Route_Lifetime: This parameter determines the lifetime of an alternative route in the routing table.

- Negotiation-Based

  Descriptor_Data: This parameter gives the description of the data to transmit or receive.

  Data_Cache_Lifetime: This parameter specifies the lifetime of a data or an interest in the cache.

- Aggregation-Based

  Operation_Name: This parameter specifies the name of the operation.

  Operation_Description: This parameter describes the operation.

3.2.3. Communication Paradigms Based

This class of routing protocol has three subclasses: node-centric routing protocol, data-centric routing protocol and position-centric routing protocol. For each subclass, we identify the following generic configuration data:

- Node-Centric

  Prefix_Destination: This data indicates the prefix or the address of the destination node.

- Data-Centric

  Data_Attributes or Event_Attributes: This parameter defines desired or to be transmitted data or event attributes.

  Attribute_Lifetime: This parameter specifies the lifetime of an attribute in the cache or in the event table.

- Position-Centric

  Signal_Power: This data indicates the signal strength of the transmitting node.

3.2.4. Based on Route Estalishement

Depending on how the creation and maintenance of roads when routing data, routing protocols are classified into three categories: proactive protocols, reactive protocols and hybrid protocols [17]. For each subclass, we identify the following generic configuration data:

- Reactive

  Route_Request_Wait_Time: This parameter specifies the time after which a node transmits a route request message.

Route_Request_Retries: This parameter indicates the number of times a node can transmit a route request.

Active Route Timeout: This parameter specifies the time after which a route expires the routing table.

- Proactive

  Information_Topology_Delay or

  Routing_Table_Delay: This parameter specifies the time after which a node must transmit information about the network topology or its routing table to its neighbors.

- Hybrid

  The parameters used result from the combination of settings of proactive and reactive mode.

3.2.5. Mobility Based

In this class of routing protocol, the sink node randomly moves from time to time in the network to balance the energy expenditure of different network nodes. It identifies the following configuration parameter:

Signal_Msg_Delay ou Request_Timeout: This parameter specifies the time after which if a node did not send a signaling message or did not respond to a request, then it is considered to have being moved.

3.2.6. Based on the Heterogeneity and According to the Routing Layer

These two classes of protocol does not require special configuration. They are configured in the global settings and are included in the protocol implementation.

The configuration data model WSN-routing-protocol we propose is based on these different identified data.

4. Formal Specification of Configuration Data in YANG

4.1. Implementation and Validation of the Model WSN-Routing-Protocol

WSN-routing-protocol is the configuration data model we have implemented with an aim of standardizing the routing protocols configuration in WSN. The modeling is done in YANG and the validation is made using the PYang tool [23].

4.1.1. YANG

Yang is the modeling language of configuration data manipulated by NETCONF protocol [4]. YANG can be used to model the configuration data and state data from network elements. YANG is a modular language representing data structures in a XML tree format. The keyword <leaf> shows a tree leaf. If a tree node is not a leaf, it is designated by the keyword <container>. A YANG module can refer to other modules. Yang defines a set of reference types for data description [3,24]. It also allows indicating, with the <must> directive, constraints that must comply with data. Other specific application data types can be derived from integrated data types. YANG syntax uses nested groups, delimited by braces. But an equivalent XML syntax, exists known as YIN (YANG Independent Notation) [3].

4.1.2. WSN-Routing-Protocol. YANG

We have implemented the basic WSN-routing-protocol module with respect to the YANG language syntax. We used the <grouping> principle which consists in defining a set of reusable nodes either in the same module, or in another module. This basic module allows the overall configuration of routing protocols in WSNs. We present below the structure YANG of WSN-protocol routing.

```
+--rw routing-protocol
   +--rw name?
   +--rw description
   +--rw mobility
   +--rw heterogeneity
   +--rw topology-type
//grouping dag-topology
   |    +--:(DAG)
   |    |   +--rw OCP
   |    |   +--rw metric-data
   |    |   +--rw node-type?
//grouping flat-topology
   |    +--:(flat)
   |    |   +--rw hello-interval
   |    |   +--rw lifetime
//grouping hierachical-topology
   |    +--:(hierarchical)
   |    |   +--rw function-node
   |    |   +--rw round-interval
//grouping geographical-topology
   |    +--:(geographical)
   |    +--rw beacon-interval
   |    +--rw lifetime-beacon
   +--rw operation-type
//grouping QOS-Based
   |  +--rw QoS-based
   |  |   +--rw delay-required
   |  |   +--rw threshold-speed
   |  |   +--rw RPDR
   |  |   +--ro level-battery
//grouping query-based
   |  +--rw query-based
   |  |   +--rw request-wait-time
   |  |   +--rw request-retries
//grouping multipath-based
   |  +--rw multipath-based
   |  | +--rw alternative-route-lifetime
   |  |   +--rw alternative-route-max
//grouping negotiation-based
   |  +--rw negotiation-based
   |  |   +--rw descriptor-data
   |  |   +--rw data-cache-lifetime
//grouping aggregation-based
   |  +--rw aggregation-based
   |        +--rw operation-name
```

```
     |         +--rw operation-description
     +--rw communication-paradigm
     |    +--rw (paradigm)
//grouping node-centric
     |    +--:(node-centric)
     |    |    +--rw adress-destination
     |    |    +--rw id-node
//grouping data-centric
     |    +--:(data-centric)
     |    |    +--rw attribute
//grouping position-centric
     |    +--:(position-centric)
     |         +--ro signal-power
     +--rw method-determining-route
//grouping reactive
     |    +--:(reactive)
     |    |    +--rw route-request-time-out
     |    |    +--rw route-request-retries
     |    |    +--rw active-route-time-out
//grouping proactive
     |    +--:(proactive)
     |    |    +--rw interval-routing-information
//grouping hybrid
     |    +--:(hybrid)
     |         +--rw route-request-time-out
     |         +--rw route-request-retries
     |         +--rw active-route-time-out
     |         +--rw interval-routing-information
     +--rw routing-layer
          +--rw (method)
               +--:(mesh-under)
               +--:(route-over)
```

Figure 2. Structure YANG of WSN-routing-protocol.

4.2. PYang

PYang [23] is a validation tool written in Python. It can be used to correct and fix YANG modules, to transform them into other formats (DSDL, RSD, and UML tree), and to generate code (YIN) from the valid modules.

We used PYang to generate the tree model (tree-output) of various implemented yang modules. A tree-output shows the configuration data hierarchy where the list of question marks indicates optional nodes, configuration data are labeled by <rw> and state data by <ro>.

4.3. Construction Principle of a Specific Model

The majority of routing protocols in WSNs implement the specific configuration settings. YANG language allows building specific models from a basic model with two key concepts: “import” and “augment” [3]. The keyword “import” allows importing the basic module into the specific module and the key word “augment” allows extending the data model with a protocol specific configuration data. However, the ietf-routing module [13] provides an open frame to define multiple instances of routing protocols. So, we will use it to define the specific modules of each routing protocol in WSNs. Therefore, the data model of a specific routing protocol in WSNs will consist of the WSN-routing-protocol basic model, the ietf-routing model and of augmentation of its specific parameters. **Figure 3** illustrates this construction.

Based on this design principle, we constructed specific models aodv and rpl for configuring routing protocols AODV and RPL, respectively. We use PYang to validate the syntax of each specific model by generating the tree model (tree-output).

4.3.1. Module Aodv: Tree-Output

Figure 4 shows the pattern tree-output generated by PYang.

4.3.2. Module Rpl: Tree-Output

Figure 5 shows the pattern tree-output generated by PYang.

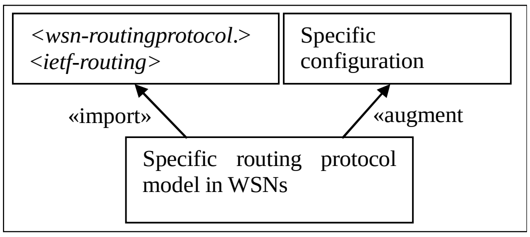

Figure 3. Construction principle of a specific model.

```
module: aodv
+--rw hello-interval?                  uint16
   +--rw lifetime?                     uint16
   +--rw delay-required?               uint16
   +--rw threshold-speed?              uint32
   +--rw RPDR?                         uint16
   +--ro level-battery?                percent
   +--rw alternative-route-lifetime? uint16
   +--rw alternative-route-max?        uint8
   +--rw operation-aggregation-name? string
   +--rw operation-aggregation-descript? string
   +--rw adress-destination* inet:ipv6-address
   +--rw id-node?             yang:phys-address
   +--rw route-request-time-out?      uint16
   +--rw route-request-retries?       uint8
   +--rwactive-route-time-out?        uint16
   +--rw (method)?
   |    +--:(mesh-under)
   |    +--:(route-over)
   +--rw aodv-configuration
        +--rw aodv-ttl
        | +--rw ttl-start?             uint8
        | +--rw ttl-increment?         uint8
        | +--rw ttl-threshold?         uint8
        | +--rw local-add-ttl?         uint8
        | +--rw max-repair-ttl?        uint8
        | +--rw min-repair-ttl?        uint8
        +--rw net-diameter?            uint16
```

Figure 4. Aodv tree-output.

```
module: rpl
   +--rw OCP?                                    uint16
   +--rw metric-data*                            string
   +--rw node-type?                              string
   +--rwdelay-required?                          uint16
   +--rw threshold-speed?                        uint32
   +--rw RPDR?                                   uint16
   +--ro level-battery?                         percent
   +--rw alternative-route-lifetime?      uint16
   +--rw alternative-route-max?              uint8
   +--rw operation-aggregation-name?     string
   +--rw operation-aggregation-descript? string
   +--rw adress-destination* inet:ipv6-address
   +--rw id-node?                   yang:phys-address
   +--ro signal-power?                           uint16
   +--rw interval-routing-information? uint16
   +--rw (method)?
   |    +--:(mesh-under)
   |    +--:(route-over)
   +--rw router-configuration
   |    +--rw dis-configuration
   | | +--rw dis-mode?                          string
   | | +--rw dis-retries?                     uint8
   |    |    +--rw delay-dis?                      uint8
   |    +--rw dao-configuration
   | | +--rw delay-dao?                          uint8
   |    +--rw route-information
   | | +--rw preference?                        uint8
   | | +--rw prefix-length?                   uint8
   | +--rw prefix?                  inet:ipv6-address
   |    +--rw rpl-instance-id?                   uint8
   | +--rw prefix-information-exist?    boolean
   | +--rw list-of-Prefix*          inet:ip-prefix
   |    +--rw valid-lifetime?                    uint32
   |    +--rw preffered-lifetime?              uint32
   |    +--rw mode-of-operation?                 uint8
   +--rw root-configuration
   |    +--rw trickle
   | +--rw dio-interval-doubling?             uint8
   | +--rw dio-interval-min?                    uint8
   | | +--rw dio-redundancy-constant?      uint8
   | +--rw path-control-size?                 uint8
   | +--rw min-hop-increase?                  uint16
   | +--rwdodag-preference?                       uint8
   | +--rwdodag-id?                 inet:ipv6-address
   +--rw not-dodag-root-configuration
        +--rwtarget-prefix?            inet:ip-prefix
```

Figure 5. Rpl: tree-output.

5. Discussion

The ROOL Working Group works [25] define a set of configurable metrics for determining the road path in the RPL routing protocol. In our WSN-routing-protocol model, these metrics are defined in the form of lists in the DAG subclass of the Topology class. This list type helps to configure so much useful metrics for the routing. The YANG ietf-routing model for the routing configuration in classical networks [13] is a model that allows configuring routing protocol, routing table and routes; while WSN-routing-Protocol is only used to configure routing protocol in WSNs. This restriction is due to constraints inherent to WSNs that require several configurable parameters and the diversity of routing protocols in such networks. Because of the limited energy factor, many routing protocols do not maintain a routing table and the method of road construction setting of a protocol is sufficient for determining routes that will be used by the protocol to transmit information in the network. However, we have extended our model by importing the ietf-routing model in the specific model of routing protocols.

6. Conclusions

In this paper, we have developed a taxonomy of routing protocols in WSNs. For each class and subclass of that taxonomy, we determined the generic configuration parameters needed for the proper functioning of the class that allowed us to define the WSN-routing-protocol model. The main contribution of this paper is the proposal with an aim of standardization, of the first YANG data model WSN-routing-Protocol in WSNs. This model is a basic model for the configuration of all routing protocols in WSNs. We can build the specific model of each routing protocol in WSNs, as respectively illustrated with the specific aodv-protocol and rpl-protocol models proposed in this paper for the configuration of the AODV and RPL routing protocols.

However, other YANG data models must be defined in WSNs. Our next work will be centered on the definition of YANG data models for the configuration of sensors nodes interfaces of and for the system management. The development of NETCONF clients in this context is also considered.

7. Acknowledgements

We thank Ladislav Lhotka, Senior researcher, Head Department, CESNET, z.s.p.o in Praha, Czech Republic (author of PYang) for his relevant remarks and advices about our research.

REFERENCES

[1] Y. Khenfouci and A. Badaoui, “Authentication Approach in Wireless Sensor Networks for the Pedagogy,” End of Studies Thesis in Computer Science, 2009.

[2] K. Beydoun, “Design of Hierarchical Routing Protocol for Wireless Sensor Network,” Ph.D. Thesis, UFR Des Sciences Et Techniques De L’Université De Franche-Comté, Besançon, 2009.

[3] M. Bjorklund, "YANG—A Data Modeling Language for the Network Configuration Protocol (NETCONF)," Internet Engineering Task Force (IETF), Request for Comments: 6020, October 2010.

[4] R. Enns, M. Bjorklund, J. Schoenwaelder and A. Bierman, "NETCONF Configuration Protocol," Network Working Group, Request for Comments: 6241, June 2011.

[5] M. Rose and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-Based Internets," Request for Comments: 1155, May 1990.

[6] J. Schoenwaelder, "Common YANG Data Types," RFC 6021, October 2010.

[7] http://www.netconfcentral.org/modules/ietf-inet-types

[8] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," Request for Comments: 3561, July 2003.

[9] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik and J. P. Vasseur, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks," Draft-Ietf-Roll-Rpl-19, March 2011.

[10] J. Case, M. Fedor, M. Schoffstall and J. Davin, "A Simple Network Management Protocol (SNMP)," May 1990.

[11] K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-Based Internets," Request for Comments: 1156, May 1990.

[12] K. Korte and A. Sehgal, "Definition of Managed Objects for the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)," Draft-Sehgal-Roll-Rpl-Mib-00, October 2010.

[13] L. Lhotka, "A YANG Data Model for Routing Configuration," Draft-Lhotka-Netmod-Routing-Cfg-00, March 2011.

[14] M. Bjorklund, "A YANG Data Model for Interface Configuration," Draft-Ietf-Netmod-Interfaces-Cfg-01, May 2011.

[15] A. Bierman and M. Bjorklund, "YANG Data Model for System Management," Draft-Bierman-Netmod-System-Mgmt-00, June 2011.

[16] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *Wireless Communications*, Vol. 11, No. 6, 2004, pp. 6-28.

[17] D. Benchaira and A. Bencheikh, "Security of the Data Dissemination in a Wireless Sensor Network: Case of Tiny Diffusion Protocol," Master in Computer Science, Alger, 2009.

[18] S. K. Singh, M. P. Singh and D. K. Singh, "Routing Protocols in Wireless Sensor Networks—A Survey," *International Journal of Computer Science & Engineering Survey*, Vol. 1, No. 2, 2010, pp. 63-83.

[19] S. Pal, D. Bhattacharyya, G. S. Tomar and T.-H. Kim, "Wireless Sensor Networks and Its Routing Protocols: A Comparative Study," 2010 *International Conference on Computational Intelligence and Communication Networks*, Bhopal, 26-28 November 2010, pp. 314-319. doi:10.1109/CICN.2010.71

[20] A. Kemal and Y. Mohamed, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Networks*, Vol. 3, No. 3, 2005, pp. 325-349. doi:10.1016/j.adhoc.2003.09.010

[21] R. V. Biradar, V. C. Patil, S. R. Sawant and R. R. Mudholkar, "Classification and Comparison of Routing Protocols in Wireless Sensor Networks," *Ubiquitous Computing Security Systems*, Vol. 4, Special Issue, 2009, pp. 704-711.

[22] N. Kushalnagar, G. Montenegro and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6Lowpans): Overview, Assumptions, Problem Statement, and Goals," RFC 4919, August 2007.

[23] http://code.google.com/p/pyang/

[24] J. Schoenwaelder, "Common YANG Data Types," Request for Comments: 6021, October 2010.

[25] J. P. Vasseur, M. Kim, K. Pister, N. Dejean and D. Barthel, "Routing Metrics Used for Path Calculation in Low Power and Lossy Networks," Draft-Ietf-Roll-Routing-Metrics-19, March 2011.