UAV based SDN system for wireless sensor networks

ABSTRACT

Abstract goes here.

1 INTRODUCTION

Why we are SDN ?

Software defined network is able to support flexible network programmability by using programmable data plane and centralized network controller.

OpenFlow focus on wired networks.

Challenges and opportunities of SDN for WSN:

Challenges: Limited resources of WSN nodes:

- energy
- processing
- memory
- communication

Opportunities:

- Improve resource reuse
- Implement node retasking
- Node and network management
- Enable experiments with new protocols

Existing SDN for WSN:

- Flow-Sensor
- Sensor OpenFlow
- SDWN
- TinySDN
- SDN-WISE

All of these are evaluated by simulations

Flow-Sensor [MahmudandRahmani2011], Sensor OpenFlow [Luoetal.2012] SDWN [Costanzo et al. 2012] TinySDN [de Oliveira et al. 2014] SDN-WISE [Galluccio et al. 2015]

Why and How we can implement AI ?

How we combine Ai with other applications?

2 RELATED WORK

Introduction goes here.

3 ARCHITECTURE

The architecture of the UAV based SDN system for wireless sensor networks.

Listing 1: An example of deploy routing algorithm

```
topology = get_topology();
//calculate routetable for each node
//based on topology
for(node in nodeset){
        node.routetable =
                calculate_route(topology);
}
//set route for each node
for(node in nodeset){
        UAV fly to node;
        for(route in node.routeTable)
                set_route(route);
}
```


MobiCom'18,

DOI: 10.475/123_4

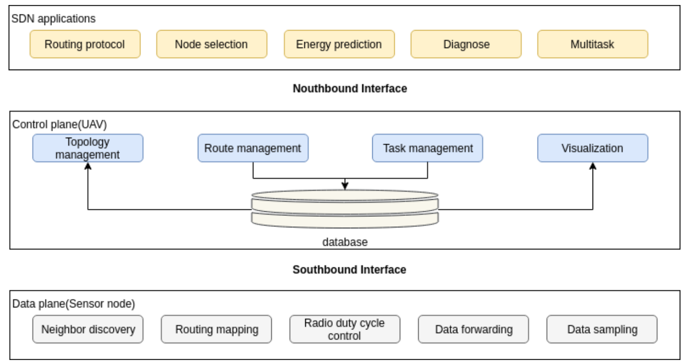

Figure 1: Architecture of the system.

4 APPLICATIONS

4.1 Overview

Traditional applications can not achieve complicated and efficient goals due to the limited processing power and memory space of sensors.

In our system, applications for wireless sensor networks are inspired by greater potential with the UAV based SDN controller. The central controller helps sensors execute complex calculations such as AI model training, as well as store global information. Besides, UAVs have flexible features and can deploy tasks to sensors by one-hop communication directly. Thus it enables the sensor network to achieve much more intelligent applications.

In our system, applications can be found for a variety of purposes, including routing, AI node selection, Ai energy prediction, multitasks and network diagnosis. We design all these applications and provide easy-to-use interfaces to users as in Table **??**.

4.2 Routing

4.3 AI Node Selection

4.3.1 Motivation.

4.3.2 Design. Greedy selection algorithm.

SRSSS AI algorithm selection.

****

Table 1: System API

Structure && Function	Description
Sensor Control Interface	
struct node	Sensor node structure
struct nodeset	A set of sensor nodes
struct neighbor_list	Neighbor infomation
struct energy_item	Energy statistic information
struct flow_table	Flow table
struct duty_cycle_table	Duty cycle control table
struct sensor_enable_table	All the nodes's states. Node state: {on,off}
switch_node(node,state)	Turn on or turn off the node
get_node_info(node)	Get node's information, including node's position, duty cycle, power, etc.
set_node_attr(node,attrTag,value)	Set node attribute, including duty cycle, radio strength, etc.
get_neighborlist(node)	Get the neighbor list of a node
UAV Application Interface	
Routing	
get_topology()	Get the topology of the network
get_flow_table(node)	Get the flow table of a node
set_flow_table(flow_table,node)	Set the flow table of a node
AI Node selection	
nodeset simple_selection(nodeset)	Select sensor set by location information
nodeset SRSSS_selection(dataset)	Select sensor set by AI algorithm based on sensing data
AI Energy Prediction	
model_selsct(modeltype)	Select an AI model
model.train(dataset,ratio)	Train an AI model with learning ratio on the data set
model.test(dataset)	Test the AI model on the data set
model.predict(node)	Do the energy prediction for a node
Multi-tasks	
create_scheduler()	Create a task scheduler
scheduler.create_buffer()	Create a task buffer
scheduler.task_buffer_add(task,nodenum)	Add a new task to task buffer
scheduler.task_schedule()	Schedule the tasks in the buffer
Diagnosis	
detect()	Detect problematic region with probes
get_topical_topology(nodeset)	Construct topical topology
diagnose_network(topology,nodeset)	Diagnose the failure nodes or lossy links

Algorithm 1 Greedy Selection Algorithm

Input: Sensor set N, Selected set M, Target area Ω, Covering area Φ;
Initialize : $N = \emptyset$, $\Phi = \emptyset$
while $M \neq N$ **do**
  if $\Phi = \Omega$ **then**
    break; \\ selected set has been found
  end if
  Find n_i : $argmax(\Phi \cap range(n))$, $n_i \in (N - M)$;
  $\Phi = \Phi \cup n_i$
end while

$$
\begin{aligned}
&(\mathbf{W}_{k+1}, \mathbf{z}_{k+1}) = \\
&\arg\min_{\mathbf{W},\mathbf{z}} \sum_{i=1}^{k} \mu^{k-i} \|\mathbf{X}_k^i \mathbf{D}_\mathbf{z} \mathbf{W}(\mathbf{I} - \mathbf{D}_\mathbf{z}) - \mathbf{X}_k^i (\mathbf{I} - \mathbf{D}_\mathbf{z})\|_2^2 \\
&+ \alpha \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 |W_{ij}| - \beta \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|_2 z_i z_j + \lambda \|\mathbf{W}\|_F^2 \\
&s.t.\ \mathbf{z} = [z_1, \ldots, z_n] \in \{0,1\}^n, \mathbf{c}^T \mathbf{z} \leq P \qquad (1)
\end{aligned}
$$

Figure 2: Objective function.

AI helps creating smarter sensor systems.

AI systems have been improving, and new advances in machine intelligence are creating seamless interactions between people and digital sensor systems.

In sensor systems, applications can be found for a variety of tasks, including selection of sensor inputs, interpreting signals, condition monitoring, fault diagnosis, machine and process control, machine design, process planning, production scheduling, and system configuring. Some examples of specific tasks undertaken by expert systems are: * Assembly * Automatic programming * Controlling intelligent complex vehicles * Planning inspection * Predicting risk of disease * Selecting tools and machining strategies * Sequence planning * Controlling plant growth.

AI can increase effective communication, reduce mistakes, minimize errors, and extend sensor life.

The tools and methods described have minimal computation complexity and can be implemented on small assembly lines, single robots, or systems with low-capability microcontrollers. These novel approaches proposed use ambient intelligence and the mixing of different AI tools in an effort to use the best of each technology. The concepts are generically applicable across many processes.

minimum energy, data loss, reliability, robustness, etc., in place during the design and operation of wireless sensor networks

a specific set of protocols for medium access, localization and positioning, time synchronization, topology control, security and routing are identified based on the current configuration of the network, the requirements of the application and the topology of their deployment.

4.4 AI Energy Prediction

4.4.1 Motivation.

4.4.2 Design.

4.5 Multi-tasks

4.5.1 Motivation. A software defined sensor node can perform multiple tasks.

wireless sensor network(WSN) are generally comprised of a group of spatially dispersed sensors, monitoring and recording the physical conditions of the environment. WSNs measure environmental conditions like temperature, sound, sunlight, humidity, etc.

A deployed wireless sensor networks are usually assigned

In our system, a sensor node can perform multiple tasks with different sensing targets simultaneously.

A sensor node may have different sensing ranges for different tasks.

only the sensor node loaded with a program for task can sense targets.

There are several practical requirements.

Different tasks have different requirements, including time, sensing range, sensing ratio, etc.

For example tasks like sunlight collection only need to be carried out during the daytime.

Our system provide a task scheduling to

Sensors are usually assigned multi-tasks.

4.5.2 Design. Sensors are assigned tasks to monitor a specific area.

Different tasks have different requirements, including time, density, etc.

Task scheduler do the arrangement.

Task buffer.

Task queue.

Scheduling table.

...

4.6 Network Diagnosis

Diagnose the network.

5 IMPLEMENTATION

Implementation goes here.

6 EVALUATION

Evaluation goes here.

7 CONCLUSION

Conclusion goes here.