Deployment of flow-sensors in Internet of Things' virtualization via OpenFlow

Arif Mahmud, Rahim Rahmani and Theo Kanter
Department of Computer and Systems Sciences
Stockholm University
Sweden
{arifm, rahim, kanter}@dsv.su.se

Abstract—**A novel idea presented in this paper is to deploy the OpenFlow technology in wireless sensor networks that can lead to a significant achievement in Internet of things and cloud computing arena through network virtualization. Two new abstract layers namely Common platform layer and virtualization layer can be added at the top and bottom of a preset Infrastructure as a Service architecture. Our proposed IOT virtualization can be applicable in a random topology scenario which makes possible of the flow-sensors' resources to be shared, establishment of multi operational sensor networks and escalation of the reachability under the same platform without establishing any further physical networks. Flow-sensor achieved 39% points more reachability than the typical sensors in an ideal scenario and even better results are possible from the amount of packets generation and simulation time viewpoint for larger scale networks.**

Keywords- *Internet of things, Infrastructure as a service, Virtualization, OpenFlow, Flow-sensor, WSN*

I. INTRODUCTION

Next generation internet is highly dependable on the incorporation of regular objects found in our surroundings those can be uniquely recognizable, controllable and monitor-able such as sensors, actuators etc. into Internet of things. Just IP connectivity won't allow WSN to be included in Internet due to their limited resources like bandwidth, memory, energy and communication capabilities [1]. Dynamic internet connectivity happened to be possible through Integration of WSN and IOT. Task evaluation allows gaining benefits from network heterogeneity, remotely accessing becomes possible through efficient collaboration to achieve a certain set of future challenges [2].

Possible application includes e-health, home automation, transportation, battle field inspection, safety, failure management and in some other areas where usual and normal attempts were proven to be very expensive and uncertain [3]. Unstructured randomly sited sensors integrated into IOT also have the capabilities to offer a large amount of environmental services such as sound, pressure, temperature, motion etc.

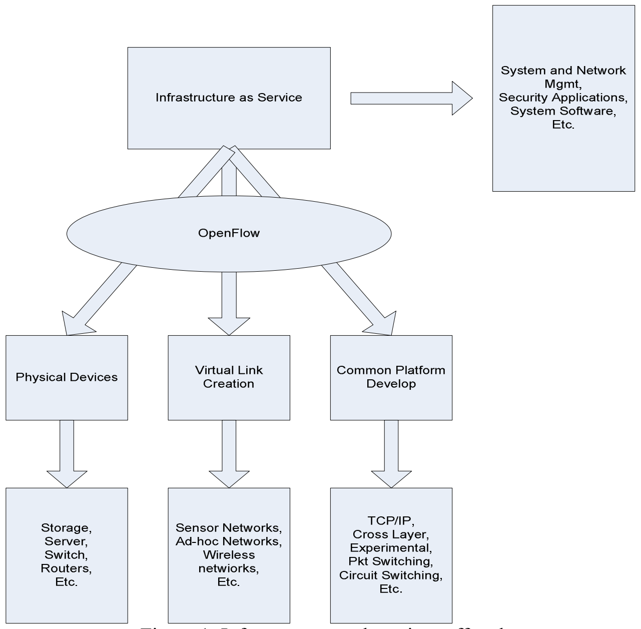
Present Infrastructure As a Service (IAAS) contain a preset architecture with location aware network mapping along with associated physical devices like different servers and storage devices, routers and switches and running routing logics and algorithms. These topologies cannot support the dynamic one where presence of sensors, intelligent devices are virtual and cannot create a runnable common platform for different kind of traffics such as TCP/IP, cross layer, experimental etc. OpenFlow programmability and virtualization feature allows two completely new abstract layers namely common platform layer and virtualization layer to be added at the top and bottom of a preset architecture. It also allows the present infrastructure running without any obstacles even after adding new layer function-abilities. So, only OpenFlow can provide a better solution through network programmability and device virtualizations and thus enable the IAAS to provide the service like security applications, system and network applications, system software etc.

We have proposed the following ideas to implement in IOT arena for the sake of achieving a common platform and virtualization with IAAS layer through deploying OpenFlow protocol:

- A completely new idea to merge OpenFlow technology with IAAS layer to make sensor data clouds more efficient from information gaining, sensor management, monitor and virtualization point of view.
- Placement of sensor node is very important for proper data transmission and reception, but in a random scenario, it is almost impossible. Many researchers have proposed highly optimized placement and transmission algorithms but these are too complicated to be implemented practically. So, why should not we try OpenFlow supported flow-sensor?
- Typical Sensor networks are formed in an ad-hoc mode to perform any specific task. So, a common platform is required and OpenFlow is able to provide that even for experimental traffics.
- Data is required to be shared and passed among different wireless objects like sensors, actuators, PDAs etc. OpenFlow is able to provide a common platform and virtualization layer for all networks and thus allow them to share the resources.

The paper is organized in the following way: Section 2 describes the Motivation and background; Section 3 presents Design and implementation of the proposed model; Section 4 describes the model checking of the new concept; Section 5

978-0-7695-4727-5/12 $26.00 © 2012 IEEE
DOI 10.1109/MUSIC.2012.41



Infrastructure as Service
System and Network Mgmt, Security Applications, System Software, Etc.
OpenFlow
Physical Devices
Virtual Link Creation
Common Platform Develop
Storage, Server, Switch, Routers, Etc.
Sensor Networks, Ad-hoc Networks, Wireless netwiorks, Etc.
TCP/IP, Cross Layer, Experimental, Pkt Switching, Circuit Switching, Etc.

Figure 1: Infrastructure and services offered

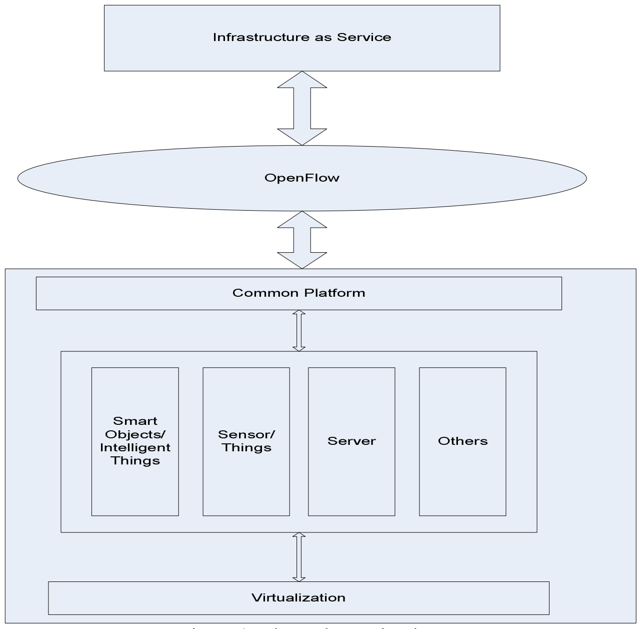


Figure 2: Three abstraction layers

presents the performance evaluation and the conclusions are provided in section 6.

II. BACKGROUND AND MOTIVATION

The OpenFlow can split the traffic path into data packet (maintained by underlying router or switch) and control packet (maintained by a controller or control server) which turn the physical device into a simple one from a complicated mode since complex intelligence programs are removed. Today OpenFlow is supported by several major switch/router vendors (especially a set of functions which are common) and can support all sort of layers (2, 3, and 4) headers [4]. It is also able to integrate the circuit and packet switching technology and these can be treated separately too. Core network also gain noteworthy benefits due to control, management schemes from cost, energy effectiveness and overall network performances point of view [5, 6].

Flow-sensor is just like a typical sensor associated with a control interface (software layer) and flow tables (hardware layer). A Flow table contains a rule (Header) with source and destination address, action that takes the decision (either to drop or to forward the packets) and a counter that maintains a statistics of control and data packet. Control interface exchanges secure messages (control packet) via OpenFlow and sensor buffer maintain typical TCP/IP with access point (data packet exchange) [7].

Infrastructure as a Service is known to be one of the most important methodologies to communicate with the services offered by cloud computing which preserve applications, information in virtual storages and servers and can be access via web browser from internet [8]. IAAS also provides a solid base to Software as a Service and Platform as a Service. It is responsible to create an abstract layer (virtual middleware environment) on physical devices like storages, servers etc. along with offered services. Opportunity is given to user to operate and configure guest OS where storage, bandwidth and other performances matrixes are previously fixed [9, 10].

III. MODEL AND IMPLEMENTATION

Different traffic flows could be treated differently and it could be achieved with a simple modification in packet header without any complex optimization algorithms.

A. *Infrastructure Model*

One of the most important characteristics of OpenFlow is to add virtual layers with the preset layers, leaving the established infrastructure unchanged. As shown in fig. 1, a virtual link can be created among different networks and a common platform can be developed for various communication systems. The system is fully a centralized system from physical layer viewpoint but a distribution of service (flow visor could be utilized) could be maintained. One central system can monitor, control all sorts of traffics. It can help to achieve better band-width, reliability, robust routing, etc. which will lead to a better Quality of Service (QoS).

In a multi-hopping scenario packets are transferred via some adjacent nodes. So, nodes near to access points bears too much load in comparison to distant nodes in a downstream scenario and inactivity of these important nodes may cause the network to be collapsed. Virtual presence of sensor nodes can solve the problem where we can create a virtual link between two sensor networks through access point negotiation. So, we can design a three a three layer platform (fig. 2) where common platform and virtualization layer are newly added with established infrastructure. Sensors need not to be worried about reach-ability or their

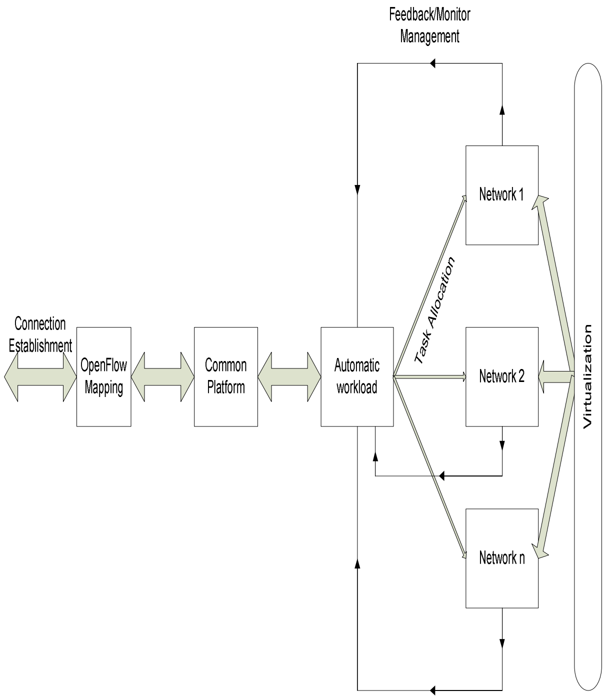


Figure 3: Work flow management

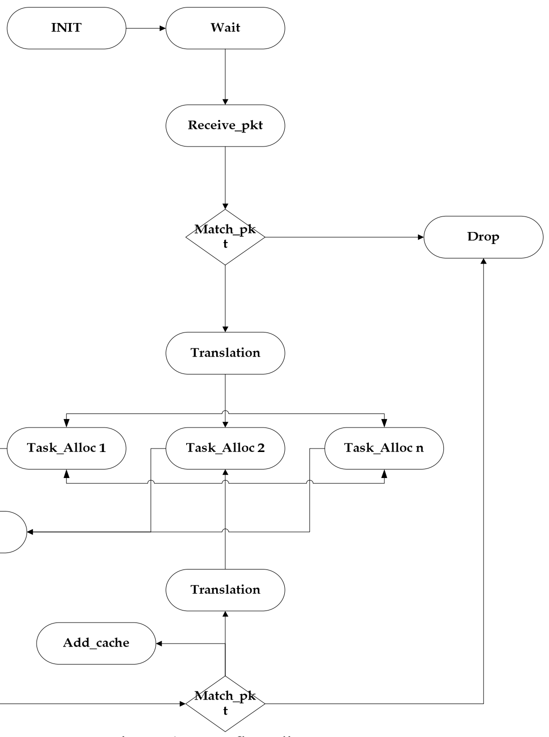


Figure 4: Data flow diagram

placement even in harsh areas. Packet could be sent to any nodes even if it is sited on different networks.

B. *Work-flow management diagram*

OpenFlow architecture allows building a common platform as found in fig. 3 for different routed-switched traffic and requires to be mapped before that. OpenFlow supports 2nd layer, 3rd layer and even cross layer traffic where source and destination addresses are needed to previously set up. OpenFlow mapping layer establish a connection between physical devices and OpenFlow table via a secured OpenFlow communication protocol. OpenFlow control server generates a tree structure and locates the position of sensor devices. It also can monitor the packet flow in downstream direction and observe the current status of each sensor on a requirement or periodic basis.

IV. MODEL CHECKING AND CONCEPT

The PROMELA and SPIN combination has proved to be a versatile and useful tool in the simulation and verification of software systems [12, 13, and 14]. Both have been extensively used in modeling and verifying communication protocols. In particular, SPIN shall be applied to simulate exhaustively the correctness of flow sensor and provide verification in Linear Temporal Logic (LTL) with respect to convergence.

A. *Definition States*

3 different states, shown in fig. 4 Match_pkt, Translation and Send_pkt can be represented by Π , μ, Γ respectively.

Definition 1: Upon receiving, data packet will be matched based on control server state, packet source and packet information. Then the packet will be either dropped or sent to the translation or mapping state accordingly.

$\Pi \models \quad D \rightarrow$ Pkt if either $\Pi \not\models \quad D$ or $\pi \models$ Pkt, where receiving of packet = Pkt and packet dropping = D

Definition 2: Translation state maps the data into the flow table and allocates the task into different sensor networks.

$\mu \models F_1 \vee F_2 \vee \ldots \vee F_n$ for $\forall$ id = {1, 2… n} and we also achieve $\mu \models F_{id}$

Different task allocations can be denoted by F_1, F_2... F_n respectively along with network id as id.

Definition 3: Packet will be sent either to cache or to translation state in case of acknowledgement or data respectively.

$\Gamma \models$ cache $\vee$ μ iff $\Gamma \models$ cache or $\Gamma \models \mu$, where Add_cache and translation state are represented by cache and μ respectively

B. *LTL formulas*

The following LTL formulas are generated for the definitions:

LTL1: $\square$ (Receive_pkt $\wedge$ Drop $\rightarrow$ $\lozenge$action ToMatch_pkt)

LTL1: $\square$ (Translation $\rightarrow$ $\lozenge$ (action ToTask_alloc 1 $\vee$ action ToTask_alloc 2 $\vee$ …. $\vee$ action ToTask_alloc n))

LTL1: $\square$ (Send_pkt $\wedge$ Drop $\rightarrow$ $\lozenge$ (action ToAdd_cache $\vee$ action ToTranslation)

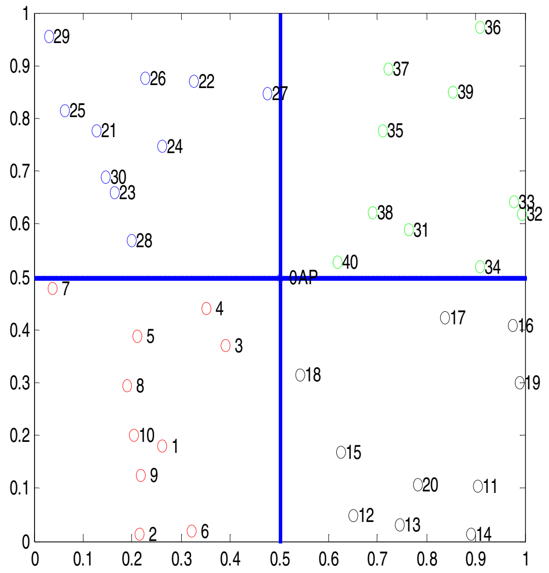


Figure 5: Four different sensor networks

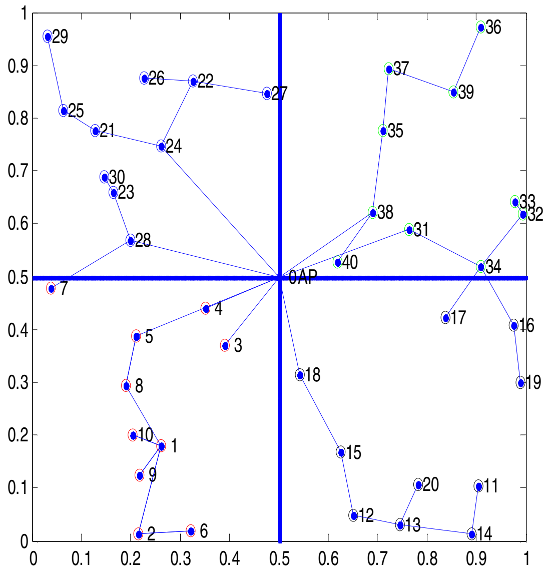


Figure 7: Sharing resources among different networks

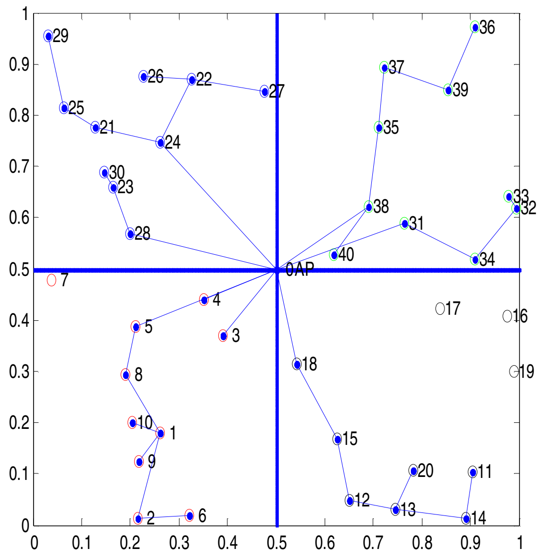


Figure 6: Individual routing for sensor networks

C. *Reference Topology Model*

4 different sensor networks have been created with an access point that can be found in fig. 5. All the sensors of different networks will use the access point as the gateway. Different sensor networks are assumed to serve different applications where each network contains 10 sensors. These sensors are randomly sited and access point is situated somehow in the middle. Red, black, green and blue sensors are assumed to the 1st, 2nd, 3rd and 4th network respectively.

At the beginning (fig. 6) all the sensors are assumed to the typical sensors and sensors of one network are not allowed to communicate with other networks. In random networks, some nodes are always sited out of state or away from the range of access point and other sensors. In 1st network node 7 and in 2nd network node 16, 17 and 19 are out of range. That's why we have 100% reach-ability in 3rd and 4th network but 90% and 70% reach-ability in 1st and 2nd network respectively.

Then in fig. 7 all the typical sensors are replaced by flow-sensors and sensors of one network can utilize sensors of other networks for data transfer. We can see node 28 of 4th network can reach node 7 of 1st network. And node 16, 17 and 19 of network 2nd can use node 34 of 3rd network as intermediate nodes in a multi-hopping scenario. So now all the 4 networks are assumed to be a single network virtually and 100% reach-ability can be achieved thereby.

V. PERFORMANCE EVALUATION

The system performance metrics include response time and total number of generated packets for varying topology scenario, sensor density and transmission power.

A. *Simulation Assumption and Parameter*

- Sensor nodes are sited to be in random but access point is placed in the middle of topology.
- We have maximum 4 networks per access point (also assumed as All Net/AP) where mobility of sensor nodes is not considered.

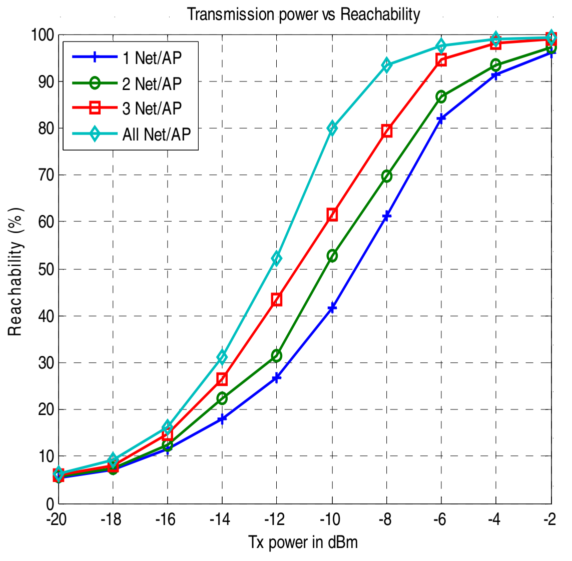


Figure 8: Reachability comparison on varying transmission power

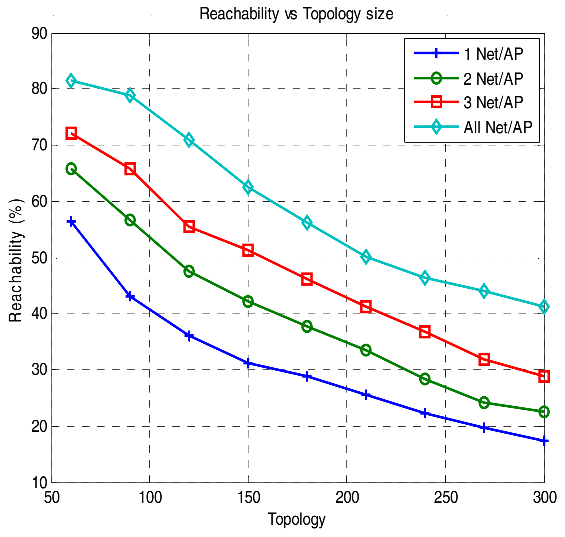


Figure 9: Reachability comparison on varying topology sizes

- Communication range of sensor nodes and access point are 10 meter and 20 meter respectively.
- Receiver sensitivity is -80.5 dBm.
- Total number of packets includes both transmitted and received packets.
- Topology size, sensor density and transmission power are 100*100, 1*10-2/m2 and -10.3 dBm respectively in an ideal scenario.
- Each of the simulation runs for 100 times where we have counted average number of packets with total simulation time.

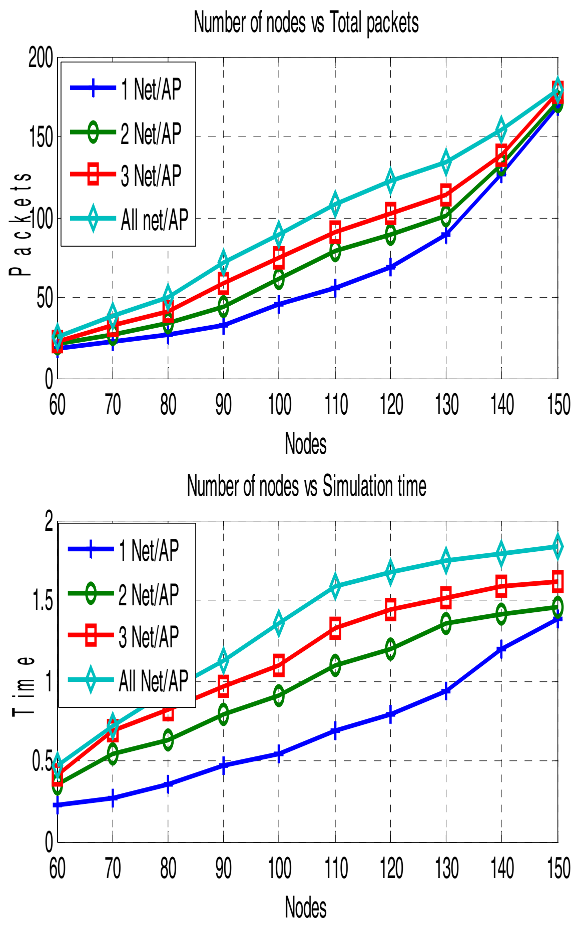


Figure 10: Total packets and simulation time comparison on varying number of nodes

B. *Simulation results*

We have compared the performance of flow-sensor and typical sensors where they are randomly sited in maximum four networks with an access point. In 1 Net/AP, sensors of one network are not allowed to communicate with sensors of other networks and they will behave as typical sensors. In 2 Net/AP, 3 Net/AP and 4 Net/AP, sensors of 2 networks, sensors of 3 networks and sensors of all 4 networks will be allowed to communicate as flow-sensors. We have counted the total number of packets (average) and simulation time (total) along with reachability based on varying topology sizes, number of nodes and transmission power.

Fig. 8 explains the reachability of typical sensors and flow-sensors based on varying transmission power. It's true that in a very low and high transmission power both of them behave equally but it is important to know about the ideal scenario activity. All Net/AP reaches 90% of reachability in -8.5 dBm whereas typical sensor requires -4.2 dBm and 2

Net/AP and 3 Net/AP require -5 and -6.8 dBm respectively. In an almost ideal Tx power scenario (-10 dBm), 1 Net/AP, 2 Net/AP, 3 Net/AP and 4 Net/AP have the reach-ability of 41.56%, 52.84%, 61.56% and 79.92% respectively.

Fig. 9 illustrates reachability counted on varying topology sizes. Reachability of both the typical sensors (1 Net/AP) and flow-sensors (All Net/AP) have been decreased with the increase of topology size. But in all the cases flow-sensors maintains better reachability in comparison to typical sensor network scenario. In a medium scale network (topology size as 180*180), 1 Net/AP, 2 Net/AP, 3 Net/AP and 4 Net/AP have the reach-ability of 28.78%, 37.81%, 46.24% and 56.16% respectively.

In fig. 10 simulation time and total number of packets have been calculated on the same varying amount of nodes. In medium scale networks flow-sensors requires more time to simulate and generate more packets than typical sensors. But in case of higher number of nodes, the difference between them gets decreased. Its true for small networks both of them bear low reachability as reflected in their simulation time and generated number of packets. In a medium scale network (number of nodes = 100), 1 Net/AP, 2 Net/AP, 3 Net/AP and 4 Net/AP have generated 45.23, 61.32, 74.71 and 89.39 packets with a simulation time of 0.55, 0.91, 1.10 and 1.36 sec respectively.

VI. CONCLUSION

Our proposed IOT virtualization can be applicable in a random topology scenario where some of the physical nodes can be sited out of state and inactivity of those nodes can make unreachable from access points. Network virtualization allows flow-sensors of different networks to be used as intermediate nodes under the same platform without establishing any further physical networks. Thus enables resources to be shared, establishment of multi operational sensor networks and escalation of the reachability thereby.

Promising results indicate that All Net/IP (flow-sensor) achieved 39% points more reach-ability than 1 Net/AP or typical sensor. Same result trend is observed in case of different network sizes. Flow sensor generated more packets and took more time to simulate in comparison to typical sensor in an ideal scenario. It surely can happen since better reach-ability requires more packets to be produced with higher simulation time. But for large number of nodes the difference gets lower and flow-sensor is found to perform better in large scale networks.

Hardware to programmable hardware, operating system to network operating system, vendor to owner specific, previous hardware defined networking (device monitoring, traffic controlling, and topology definition) turns to be software defined and where everything leads to a programmable and customized networking system. So, OpenFlow and flow-sensor makes ease for the researchers to investigate on experimental traffic, protocol and overall network as well and it can lead to a significant achievement in Internet of things and cloud computing arena through network modernization and virtualization.

REFERENCES

[1] Delphine Christin, Andreas Reinhardt, Parag S. Mogre and Ralf Steinmetz, "Wireless Sensor Networks and the Internet of Things: Selected Challenges", Multimedia Communications Lab, Technische Universit¨at Darmstadt, Merckstr. 25, 64283 Darmstadt, Germany.

[2] Cristina Alcaraz, Pablo Najera, Javier Lopez and Rodrigo Roman, "Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?" Computer Science Department University of Malaga, Malaga, Spain.

[3] Luigi Atzori , Antonio Iera , and Giacomo Morabito, "The Internet of Things: A survey", University of Cagliari, Italy, Computer Networks, Volume 54, Issue 15, 28 October 2010, Pages 2787–2805.

[4] "OpenFlow Switch Specification", http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf

[5] R. Sherwood, G. Gibb, K.-King Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "FlowVisor: A Network virtualization Layer", Technical reports, Deutche Telecom and Stanford University, October 2009

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner "OpenFlow: Enabling Innovation in Campus Networks", March 14, 2008. http://www.openflow.org/documents/openflow-wp-latest.pdf

[7] Arif Mahmud and Rahim Rahmani," Exploitation of OpenFlow in Wireless Sensor Networks", Dept. Of Information, Technology and Media, Mid Sweden University, Int. Conference on computer Science and Network Technology, IEEE 2011, Harbin, China.in press.

[8] Rajkumar Buyya, Chee Shin Yeo and Srikumar Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", Dept. of Comput. Sci. & Software Eng., Univ. of Melbourne, Melbourne, VIC, The 10th IEEE International Conference on High Performance Computing and Communications, 2008.

[9] Radu Prodan, Simon Ostermann, "A Survey and Taxonomy of Infrastructure as a Service and Web Hosting Cloud Providers", Inst. of Comput. Sci., Univ. of Innsbruck, Innsbruck, Austria, IEEE, Grid computing, 2009.

[10] Francesco Longo, Rahul Ghosh, Vijay K. Naik, and Kishor S. Trivedi, "A Scalable Availability Model for Infrastructure-as-a-Service Cloud", Dipt. di Mat., Univ. degli Studi di Messina, Messina, Italy, IEEE 2011.

[11] Deepak Ganesan, Razvan Cristescu and Baltasar Beferull Lozano,"PowerEfficient Sensor Placement and Transmission Structure for Data Gathering under Distortion Constraints", Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90095.

[12] Gerard J. Holzmann. " The Model Checker SPIN". IEEE Transaction on software Engineering ,23(5):1-17, May 1997.

[13] Gerard J. Holzmann, SPIN Online Reference. Bell Labs, http://cm.bell-labs.com/cm/cs/what/spin/Man /index.html)August 1997.

[14] Rob Gerth. " Concise Promela Refence". Technical report, Eindhoven University, (http://cm.bell-labs.com/cm/cs/what/spin/Man/Quick.html) June1997