Energy efficient data transmission and Aggregation of WSN using data processing in Map Reduce

Praveen T*, Aswini VS, Kiruthika K, Selva Karthika C
Dept. of Computer Science and Engineering, Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Anna University, Chennai, India
***Corresponding author: E-Mail: Praveen@velhightech.com**

ABSTRACT

In this paper we construct a energy efficient Wireless Sensor Network and to transmit the sensed data in a secure way, reaching the Base Station, for Query processing on the Historical data and to apply in the Map Reduce. The architecture of WSN is a hybrid type which is a combination of infrastructure and infrastructure less network. The Communication between sensor and sensor head takes place through peer to peer Architecture (Infrastructureless) and the communication between Cluster Head and Base Station is Broadcast Based (Infrastructure Oriented).This Hybrid Architecture reduces the Energy Consumption of Sensor nodes as it will be depleted soon when each sensor broadcasts sensed data to Base station as it senses. Hence a Cluster Head will be elected for each cluster on the basis of its battery, Memory and the ability to process and the Sensors will be sending their sensed data to the Cluster Head in a peer to peer manner

KEY WORDS: Wireless sensor networks, software defined networking, big data, Map Reduce

1. INTRODUCTION

Wireless sensor networks (WSNs) are expected to be the most common big data sources. In fact, the vast use of sensors, either deployed autonomously or integrated in other mobile devices, such as smartphones, has rendered them an indispensable part of a big data value chain. As a result, more and more applications are relying on information received from WSNs to perform analytics; and often the same information is leveraged in several contexts.

Alike applications have been existing for quite some time, before the so-called big data era. Mobile crowd sensing, participatory and opportunistic sensing have emerged and are in the spotlight since many years now. Experience from past research in these areas suggests that there is a need to move from the fully decoupled architectures, where sensors are just sending data to the cloud, to hybrid or even completely integrated approaches, where part of the data processing operations are executed inside the network, yielding benefits from both the application (less data to analyse) and the network (less data to transport) perspective. In this context, we note that some data processing.

Today, big data is strongly associated with Map Reduce. Its simplicity along with its scaling capabilities have rendered Map Reduce the de facto standard for data processing and analytics implementation at large scale. Several frame-works have been based on it, ranging from integrated platforms, such as which will perform the necessary operations. Such techniques should take constraints into account which are typical of WSNs such as the strict limitations in terms of memory and processing capabilities as which will perform the necessary operations.

Such techniques should take constraints into account which are typical of WSNs such as the strict limitations in terms of memory and processing capabilities as the Hadoop file system, to proprietary applications spanning in These applications assume that information is available on the infrastructure (typically a data centre) where Map Reduce is executed several domains requiring intensive computations, such as machine learning, multimedia processing and more.

However, in the case of WSNs, it may not always be efficient, or even possible, to transfer all data to the cloud before processing it, due to connectivity, bandwidth or delay limitations. On the other hand, the use of proprietary, WSN-specific frameworks is not sustainable, as it increases the implementation complex-ity. Therefore, there is a need to support in-network execution of Map Reduce-based applications, so as to ensure a homogeneous implementation paradigm with location transparency provision.

However, supporting Map Reduce inside WSNs requires several research issues to be addressed. First of all, an appropriate architecture should be designed in which the support of Map Reduce is consistent with the available technologies. Then, techniques should be developed for the selection of the sensor nodes which will perform the necessary operations. Such techniques should take constraints into account which are typical of WSNs such as the strict limitations in terms of memory and processing capabilities as well as of communication resources (which are extremely critical because of the multi hop nature of the communication paradigm in WSNs). Finally, there is the need of adapting the routing of packets in the WSN to the selection of the nodes executing the Map Reduce functions
In this paper, we introduce a comprehensive solution to support Map Reduce inside WSNs. In our framework, sensor nodes run a specialized Contiki implementation in which some new features have been introduced to enable the Software Defined Networking (SDN) paradigm in WSNs according to the SDN-WISE approach. In fact, thanks to

this approach, the routing of packets can be easily modified according to the current deployment of the Map-Reduce functions, implemented by the big data application developers, inside the network.

Furthermore, according to the SDN approach, all network management functions are centralized in a server, called the Controller, which has a global view of the network and can interact closely with the application. It follows that the Controller is in the best position to decide where to deploy the Map Reduce functions and change the routing accordingly. In this course, we also develop an analytical framework which can be employed by the Controller to select the nodes, which will execute these functions. More specifically, three significant cases are considered:
(i) the case in which the deployment is not affected by the resource limitations of nodes of the WSN,
(ii) the case in which the limitations on the processing and memory resources of the sensor nodes are the cause of con-straints in the deployment of the MapReduce functions, and the case in which the strictest constraints are due to the bandwidth limitation.

The proposed system is evaluated under three scenarios considering different processing locations and network topologies. In the first scenario, all sensor nodes have to transmit their data to an external cloud service in order to be processed. In the second scenario, data processing takes place entirely inside the network, and only the results are sent to the cloud. Finally, in the third scenario, a hybrid approach is followed, where one part of the data is processed in the network and the rest in the cloud. Results show that the possibility of executing Map Reduce function inside the network can lead to significant reduction of communication cost. This is crucial as communication is the primary source of energy consumption in current sensor nodes.

The rest of this paper is organized as follows: Section 2 provides related work and background information both on Map Reduce and big data applications that are based on information they receive from WSNs and on the use of soft-ware defined networking principles in WSNs. Then, the proposed architecture is presented in Section 3, where it is explained how SDN-WISE has been extended in order to enable the execution of Map Reduce-based big data applications. The analytical model, which is used to determine the optimal deployment of the reducers, is given in Section 4. The performance of the proposed framework is evaluated in Section 5 by studying the behaviour of three widely used network topologies (bus, grid and ring), considering a diverse approaches (cloud-based, in-network and hybrid). Finally, we draw some conclusions in Section 6.

2. RELATED WORK

The proposed framework focuses on big data processing and software defined networking for WSNs. This section provides an overview of the technologies that will be leveraged in the rest work that has motivated us. More specifically, in Section 2.1 we will focus on big data processing in WSNs and provide the general principles of Map Reduce, we will present current SDN approaches for WSNs.

of the paper in both these aspects, as well as the related work that has motivated us. More specifically, in Section 2.1 we will focus on big data processing in WSNs and provide the general principles of Map Reduce, whereas in we will present current SDN approaches for WSNs.

Big Data Processing in WSNs: MapReduce is currently considered the de facto standard in big data processing, mainly due to its simplicity and its scaling capabilities. During its ten years of life, since its original publication, MapReduce has dominated the research and development in the area of big data. Apache Hadoop includes the most widely used implementation of Map Reduce, whereas several other systems, such as NoSQL data-bases, have been based on its paradigm to enhance their scaling out performance.

MapReduce is built upon the assumption that the input data set can be split into key-value pairs that will be passed to the analytics functions. The extraction of the key-value pairs from the original input is implemented in the map function. In principle, this operation is fully distributed, since input can be split in several smaller pieces, which are given to different processes executing an instance of the map function. Processing of the generated key-value pairs is performed by the reduce function.

In particular, all key-value pairs with the same key are processed by the same reduce function instance. In this way, there can be several processes (up to the number of different keys) executing a different instance of the reduce function, enabling parallel processing of input data. Even though more functions and operations may take place in a MapReduce-based application, here we are considering the two fundamental ones (map and reduce), since we are focusing on their in-network execution in WSNs, where the resources are anyway limited and therefore, operations should remain as simple as possible. In the rest of the paper, we will be referring to a network node executing a map or a reduce function as mapper or reducer, respectively.

Implementation of map and reduce functions is up to the big data application developer, since it is use-case spe-cific. MapReduce only provides the specification of the functions, so that the underlying infrastructure can assign them to several processes, thereby parallelizing the overall big data analysis operation.

Despite the original data centre oriented design of Map-Reduce, there have been efforts to support its operations in mobile devices. In particular, the Misco system enables smartphones to perform big data processing with Map Reduce. Apart from the system level support, a scheduling algorithm for real-time processing using Misco has been proposed.

However, the proposed approach focuses on the efficient handling of several different applications, treating the devices more like low-end servers of a data center rather than information sources. The latter is mostly addressed by works studying the use of MapReduce in per-forming analytics on information coming from sensors WSNs. They are focusing on complex spatio-temporal sce-narios where advanced analytics and massive amounts of data are required, such as global-warming research, and they derive data model for efficiently applying MapReduce operations in the cloud.

Another interesting and common application of WSN data is traffic analysis in urban environ-ments; in fact, authors in devise a framework for model-ing and predicting traffic phenomena using sensor data, while they are also considering in-network distributed exe-cution of their algorithm. However, even though this approach can be supported in principle by MapReduce, no concrete design is provided, it is not discussed how the net-work would support such operations and furthermore, the model is tailor-made for the particular scenario studied throughout that paper.

The SDN paradigm is attracting the increasing interest of both the industrial and academic research communities as it significantly simplifies network control and management. Its characteristics can therefore reduce management costs for network operators and allow the introduction of new solutions in a network as easily as installing a software program is, so fostering innovation. The enthusiasm for SDN has resulted in a large number of research contribu-tions for both wired and wireless cellular networks.

Less attention has been paid to the extension of the SDN approach to WSNs so far. Indeed, early attempts to make net-working protocols applied by WSNs programmable have been presented in. Later on, the advantages of the SDN approach in WSNs have been discussed.

Recently SDN-WISE has been introduced, which advances the state of the art by adopting a stateful approach that is useful to reduce the amount of information exchanged between the sensor nodes and the Controller. Further-more, SDN-WISE introduces the WISE-Visor which is a layer enabling the creation and management of several virtual WSNs based on the same physical sensor nodes. SDN-WISE has been prototyped and experimentally assessed.

The proposed framework builds on top of SDN-WISE and therefore, in the following we provide some back-ground information on SDN-WISE which is useful to under-stand the rest of this paper.

According to the SDN approach, SDN-WISE clearly sepa-rates the data plane run by the sensor nodes and the control plane executed by a software program, called Controller, running on a server.

The behavior of sensor nodes is mostly determined by the content of the so-called WISE Flow Table which is filled with information coming from the Controller. Like in Open-Flow, Entries of the WISE Flow Table can be divided in three sections: Rules, Action, and Statistics. In the Rules sec-tion up to three rules can be defined. Each rule can look into any byte (specified by the Address field in the entry) of the packet. If such byte is equal[2] to the value specified in the Value field, then the rule is satisfied.

If a packet satisfies all the rules, then it is treated by the node as specified in the Action section. The type of action to execute on the packet is specified in the Type field. Possible types are drop, forward, modify, etc. In case the action is for-ward, then the node the packet should be sent to, is specified in the Value field.

Finally, in the Statistics section information is stored about the usage of such entry. Other fields are defined in the WISE Flow Table entries which however, are beyond the scope of this paper. Furthermore, sensor nodes execute a simple protocol which enables them to learn their best next hop towards the sink (or the closest sink, if several sinks are deployed). Finally, SDN-WISE sensor nodes collect information about their neighbours and report such information to the Control-ler periodically.

Like in the Open Flow case, when a SDN-WISE node receives a packet for which none of its entries applies, it sends this packet to the Controller inside a Rule request message. Thanks to the periodic reports received by the sensor nodes, the Controller has a complete view of the network topology and conditions, which it uses to decide how to respond to the received Rule requests. Note that in this way, the Controller can give rules to sensor nodes in such a way that packets are routed in the network depending on any byte they contain, that is, not considering the destination address only. In the fol-lowing section we will see that the proposed framework uses this feature to execute routing based on the key of the key-value pair contained in the packet, in such a way that packets are always delivered to the responsible reducer.

System Architecture: In this section we present a solution for supporting MapRe-duce operations in wireless sensor networks exploiting the SDN paradigm. Following the SDN paradigm, in the proposed framework the Data and Control planes are clearly separated. More spe-cifically, the Data plane is run by sensor nodes and is respon-sible for packet forwarding. Whereas, the Control plane (which requires some basic functions to be deployed in the sensor nodes for the neighbor discovery) runs on the Control-ler that, in principle, is outside the WSN as shown.

Sensor nodes are equipped with sensors which generate data. The map function in the sensor nodes transforms such data in key-value pairs. For example, in Fig. 1 the data gen-erated by the sensors in nodes n_1 and n_2

are transformed in the pairs hk_1; v_1i and hk_2; v_2i, respectively. Such pairs must be delivered to the nodes running the corresponding reduce functions.

Let us note that, in accordance to the MapReduce approach, there is only one node running an instance of the reduce function for each key. For example, in Fig. 1 nodes n_3 and n_4 are responsible for keys k_1 and k_2 respectively. The sensor nodes inject packets containing key-value pairs in the WSN, which is responsible for forwarding them to the designated reducers. In this context, the Controller, after assigning the reduce functions to nodes, updates the flow tables in such a way that the generated key-value pairs are forwarded to the corresponding nodes.

So, in the example of Fig. 1, the packet generated by n_1 will be delivered to n_3, whereas the packet generated by node n_2 will be delivered to node n_4. According to the SDN-WISE approach, in the proposed framework packets are forwarded by each sensor node based on the content of its WISE Flow Table. Content of such table is decided by the Controller.

Since nodes must route packets based on the key, the Rules defined in the entries of the WISE Flow Table will consider the header of the packet in which the key iscontained. For example, two of the entries on the WISE Flow Table of node n_5 (which is represented in dark color in Fig. 1) will be as shown.

Both entries apply to packets that have been pro-duced by a MapReduce application as specified in Matching Rule 1 which considers the part of the packet where the application that produced the packet is specified.
However, while the first entry applies to packet carrying pairs with key k_1, the second applies to those carrying pairs with key k_2, as specified in the Matching Rule2.

Therefore, node n_5 will forward packets containing pairs with key k_1 towards n_6 and packets containing pairs with key k_2 towards n_4.

Finally, reducers will send the output of the reduce function towards the (closest) sink, which will make the data available to applications the functions of the proposed framework are deployed in both wireless sensor nodes and the Controller. The functions running in the wireless sensor nodes are integrated in the Contiki operating system as depicted. At the bottom of the architecture there is the Driver layer which is responsible for creating abstractions of the physical resources (e.g., radio, microcontroller, sensors, etc.) which are independent of the specific hardware platform. The proposed framework does not introduce any novelty with respect of standard Contiki at the Driver layer.

Abstractions of the physical resources offered by the Driver layer are utilized by the functions of the Core layer. At the Core layer, besides the functionality required to load new functions (i.e., the Loader) or to support multi-threads (i.e., the Protothreads), the Forwarding function runs. This is new in the Contiki operating system and is responsible for forwarding packets as dictated by the content of the WISE Flow Table which is constructed according to the indica-tions provided by the Controller.

Above the Core layer, the Mapper, the Reducer, and the Node Management functions run. More specifically, in any node, one (or several) map function(s), which function will be injected into the network which will forward it towards the reducer responsible for the specific key. Sensor nodes are equipped with sensors which generate data. The map function in the sensor nodes transforms such data in key-value pairs. Are responsi-ble for associating a key-value pair to the data measured by the sensor, will run. The output of the mapping function will be injected into the network which will forward it towards the reducer responsible for the specific key.

Furthermore, the sensor node may act as the reducer for one or several keys. In Section 4 we will show that the selec-tion of the reducer nodes depends on time-variant condi-tions regarding the network traffic and the network topology.

Implementation: Network Formation & electing Cluster Head
Data transmission and Aggregation
Recovering data using Signature
Data Prediction and Query Processing in Hadoop

Network Formation & electing Cluster Head: Three Completely different Clusters are formed in a Wireless Sensor Network named

- Homogenous Low Sensors Cluster.
- Homogenous High Sensors Cluster.
- Heterogeneous Sensor Cluster.

Cluster Heads are elected based on the battery, Memory and processing ability for each cluster and only the cluster head aggregate the received data from its cluster sensors and sends to the Base Station .The Homo High and Hete clusters have the high capability sensors and hence can do computations on aggregated data and send the corresponding result and also respond to Base Station (BS) request thereby reducing overhead drastically. So Homo Low Clusters can also be incorporated into WSN since it contains Low Sensors which just aggregate and sends the cipher text to BS. Each Node is equipped with 3 keys 1.Cluster Key 2.Private Key 3.public Key in which bit length varies depending on type of cluster.

Data transmission and Aggregation: Sensors send its own sensing data to its cluster head as each and every sensor knows its own cluster head and generates a shortest path to reach it and transmits through it. Each Sensed data is converted into a packet and is encrypted and the cipher is subjected to signature generation process. The Cluster Head receives the encrypted cipher text and signature is verified and the data is aggregated. Cluster Head recovers the data and in Homo High and Hete and generated the signature using elgamal. For Homogenous Low and Hete Clusters only Aggregation process takes place as Homogenous low cluster is memory constrained.

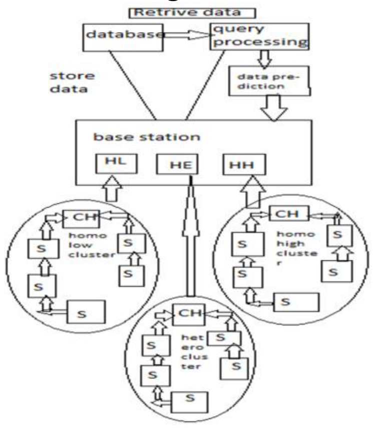

Figure.1. Architecture diagram

Recovering data using Signature: The Aggregated Data's are converted into a Single Packet when Aggregation Time out is triggered in Cluster Head. Now a cipher text is created by encryption and a signature is also generated for the same and the cipher is concealed in signature by compressing and converting the whole compressed content to binary. In Homo Low Cluster the Head Just verifies signature sends the data to Bs without aggregating packets. BS once again verifies each and every data by verifying signature thereby ensuring data Integrity and Authenticity. Base Station can also send request to Cluster Heads of High & Hete Clusters and can receive the recovered data from Cluster head by verifying signature.

Data Prediction and Query Processing in Hadoop: The Low Confidence data from the sensor nodes are dropped in cluster head by data prediction strategy on each cluster head. Hence the redundant data's are free from communication reducing overhead. The dropped packets are shown in a graph for redundancy evaluation by time vs. drop count. The Broadcasted compressed encrypted binary packets are being accumulated on Base Station and it is fed to a Database after Verification of Packets from various Clusters

The Historical thus formed by time are subjected to four kinds of Query Processing.

- Top-K Based Query Processing (Top Ranked Values on each Cluster).
- Necessary Set Based Query Processing (Values Should be Present).
- Sufficient Set Based Query Processing (Values that are more than enough).
- Boundary based Query Processing (Ranked values in a range)

3. CONCLUSION

Thus we design, develop and evaluate a energy efficient, light weighted, Secure and reliable system for Hybrid WSN to transmit and process Historical data on different types of clusters using Elgammal Signature Scheme, Aggregation methods and Data Prediction Strategies

REFERENCES

An SDN Assisted framework for optimal deployment of Map Reduce functions in WSNs Angelos-Christos G Anadiotis , member , IEEE ,Giocomo Morobito , member , IEEE and Sergio Palazzo, senior member , IEEE, 2016.

Bianchi G, Bonola M, Capone A and Cascone C, OpenState, Programming platform-independent stateful openflow applications inside the switch, SIGCOMM Comput. Commun. Rev, 44 (2), 204, 44–51.

Ganti R, Ye F and Lei H, Mobile crowdsensing, Current state and future challenges, IEEE Commun. Mag, 49 (11), 2012, 32–39.

Gregorczyk M, Pazurkiewicz T and Iwanicki K, on decentralized in-network aggregation in real-world scenarios with crowd mobility, in Proc. IEEE Int. Conf. Distrib. Comput. Sens. Syst, 2014, 83–91.

Luo T, Tan H.P and Quek T, Sensor open flow, Enabling soft-ware-defined wireless sensor networks, IEEE Commun. Lett, 16 (11), 2012, 1896–1899.

Zeng D, Miyazaki T, Guo S, Tsukahara T, Kitamichi J and Hayashi T, Evolution of software-defined sensor networks, in Proc. IEEE 9th Int. Conf. Mobile Ad-hoc Sens. Netw, 2013.