2017

Information Centric Strategies for Scalable Data Transport in Cyber Physical Systems (CPSs)

Ajay Krishna Teja Kavuri

Information Centric Strategies for Scalable Data Transport in Cyber Physical Systems (CPSs)

Ajay Krishna Teja Kavuri

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Vinod Kulathumani, Ph.D., Chair
Yanfang Ye, Ph.D.
Saiph Savage, Ph.D.

Lane Department of Computer Science and Electrical Engineering
Morgantown, West Virginia
2017

Keywords: Cyber Physical Systems, Data Transport, Information Centric Strategies, Regression, Time-series Segmentation

Abstract

Information Centric Strategies for Scalable Data Transport in Cyber Physical Systems (CPSs)

Ajay Krishna Teja Kavuri

Cyber-Physical Systems (CPSs) represent the next generation of computing that is ubiquitous, wireless and intelligent. These networked sensing systems are at the intersection of sensing, communication, control, and computing [16]. Such systems will have applications in numerous fields such as vehicular systems and transportation, medical and health care systems, smart homes and buildings, etc. The proliferation of such sensing systems will trigger an exponential increase in the computational devices that exchange data over existing network infrastructure.

Transporting data at scale in such systems is a challenge [21] mainly due to the underlying network infrastructure which is still resource constrained and bandwidth-limited. Efforts have been made to improve the network infrastructure [5] [2] [15]. The focus of this thesis is to put forward information-centric strategies that optimize the data transport over existing network infrastructure.

This thesis proposes four different information-centric strategies: (1) Strategy to minimize network congestion in a generic sensing system by estimating data with adaptive updates, (2) An adaptive information exchange strategy based on rate of change of state for static and mobile networks, (3) Spatio-temporal strategy that maintains spatial resolution by reducing redundant transmissions, (4) Proximity-dependent data transfer strategy to ensure most updated information in high-density regions. Each of these strategies is experimentally verified to optimize the data transport in their respective setting.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor, Dr. Vinod Kulathumani, for his constant support and mentorship. I sincerely thank him for believing in me and providing me an opportunity to work in his ardent research group. This thesis is possible due to his technical and editorial guidance throughout my graduate studies.

I am indebted to my committee members: Dr. Yanfang Ye, Dr. Saiph Savage. Their valuable insights, comments, and corrections have shaped this thesis.

I would like to thank Integrated Sensing Systems for Surface Mining Safety project. Also, I thank WVU Library Systems Department for providing an opportunity to work as a Graduate Service Assistant.

I extend my gratitude to Dr. Masahiro Nakagawa, Dr. Rahul Kavi, Mr. Venkata Raghava Siva Naga Shashank Sabniveesu, Mr. Rohit and Mr. Priyashraba Misra at the Distributed Computing and Networking Lab. The numerous technical discussions have fortified my understanding of the subject and contributed to my research.

Finally, I thank my parents Satyavathi Vemuri and Prabhakar Sarma Kavuri for their love and support. My parents have always been supportive of my decision to pursue graduate studies. Special thanks to my sister Bhavya Royyuru for her encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we present a brief overview and the challenges in building Cyber-Physical System(s). Also, the list of contributions and focus of this thesis is presented.

1.1 Motivation and Problem

Computational devices are increasingly occupying an import role in our daily lives. Thanks to the advancements in Wireless Networking, Wireless Sensor Networks, and Embedded Systems; this new generation of devices can communicate with one another and the physical world. These ubiquitous connected devices are finding applications in many domains including vehicular systems and transportation, medical and health care systems, smart homes and buildings, etc. Further proving Moore's law [18], according to Juniper research over 38 billion connected devices will be in use by 2020 [15]. Such integration of computation, networking, and physical processes form Cyber-Physical System(s).

Cyber-Physical System(s) enable a new generation of services that can sense the environment, communicate with one another and affect the physical process. Unlike traditional wireless sensor networks, the nodes are not just limited to resource constrained devices. Cyber-Physical System(s) can encompass energy-rich nodes such as workstations, mobile nodes such as vehicles and handheld devices such as smartphones, PDAs, tablets, etc. This connected ecosystem takes feedback from the physical world to perform critical actions.

At the core of the Cyber-Physical Systems is the information exchange between nodes. In many Cyber-Physical Systems, the central idea is to obtain data from distributed nodes to sense the physical world; usually, this data is queried for a fixed interval of time. Querying the data too often can introduce congestion in the network while querying for a larger interval of time might degrade the quality of information. Unlike traditional networked systems, the information flows through a variety of heterogeneous platforms and nodes. Transporting data at such scale is a major challenge. Hence a device oriented architecture is no longer valid [6]. A broader information-centric perspective is essential for building these complex ecosystems.

Through an information-centric perspective, our focus is on optimizing the information (or data) in Cyber-Physical System(s) without losing the quality of the information necessary for the application performance. We aim to reduce the amount of information being communicated through the network without degrading application performance so that a larger number of nodes can be accommodated in a Cyber-Physical System. In other words, our aim is to make Cyber-Physical System(s) scalable. Further, information-centric approaches have been surveyed and found to optimize the web services, provide better abstractions and improve security [3] [12].

In the past, such information-centric approaches are studied separately for Wireless Sensor Networks [7] [4]. Due to the nature of Cyber-Physical System(s), there is no one strategy that can achieve our goal for all the use cases. In this thesis, four different information-centric strategies are proposed for Cyber-Physical System(s). These strategies are inspired from challenges identified here at West Virginia University Distributed Computing and Networking Lab.

1.2 Thesis Contributions

1. We propose a strategy for optimizing a generic sensing system. A sensor model m_s and controller model m_c are designed such that the overall load on the network is optimized. Further, the strategy is validated for different network conditions. The strategy is inspired from Smart Refractory Systems project for overall network optimization.

2. A communication strategy applicable to both static and mobile networks based on the rate of change in sensing system state is developed. The optimization is implemented for MapMyTruck module within Integrated Sensing Systems for Surface Mining Safety project.

3. A scenario for mobile sensor networks such as MANET(s) contains nodes that are spatially distributed as a function of time t given by $D(t)$. A strategy is designed for such networks that ensures a spatial resolution while reducing redundant transmissions.

4. Strategy to ensure most recent information at dense regions is developed. The strategy is experimentally verified by simulating a VANET using NS3. We believe, such a strategy optimizes the Cooperative Collision Warning System developed as part of Integrated Sensing Systems for Surface Mining Safety project.

1.3 Thesis Outline

The rest of the document is organized as follows. In Chpater 2, we present a strategy to minimize network congestion in a generic sensing system by estimating data with adaptive updates. In Chpater 3, we propose a rate of change based information exchange strategy for static and mobile networks. In Chpater 4, we extend the information-centric approach to mobile nodes using a spatio-temporal strategy that ensures a spatial resolution. In Chpater 5, we propose a proximity-dependent data transfer strategy to ensure most updated information at high-density regions for Vehicular Ad hoc Network (VANET). We present conclusion and future work in Chpater 6.

Chapter 2

Strategy to minimize network congestion in a generic sensing system by estimating data with adaptive updates

2.1 Overview

In this case, our objective is to build a generic model that can optimize the sensor network infrastructure. Traditionally, networked sensors transmit data at a constant frequency. When the application requires most recent information, sensors communicate more frequently and vice versa. The goal is to curtail traffic in the network generated from sensors thereby reducing packet collisions in the network. This model achieves the objective using a linear model that intelligently makes a decision whether to communicate based on the generated data from the sensor.

The trade-off with this model is that it introduces a small error (up to 1%) in the value of the sensor data. This model applies to both static and mobile sensor networks. In the subsequent sections, the model, sensor communication algorithm, and experimental results are presented.

2.2 Methodology

For any given sensor s it generates data d_s such that at any given time t, sensor data d_s has a value v_t. For example, a temperature sensor s would generate data d_s that is a tuple of values $\{v_0, v_1, v_2, v_3......v_n\}$ corresponding to time $\{t_0, t_1.......t_n\}$. Ideally, the goal of a sensor network is to communicate this tuple d_s to a controller within an application at every interval controlled by frequency f.

This model aims to optimize the traffic irrespective of the underlying network protocol stack. The model runs synchronously on both ends of the communication; namely, sensor and controller within an application. In this model, in addition to the primary components, Sensor s consists a linear estimator, thresholding phase and network interface represented in Figure 2.1. Initially, sensor values corresponding to time t_0 and t_1 are communicated irrespective of the system state.

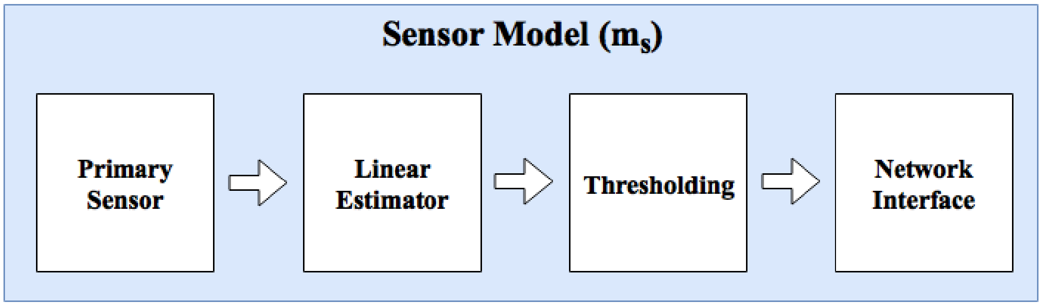


Figure 2.1: Sensor Model m_s block diagram

For the model m_s at the sensor s, m_s estimates the value e_t of the sensor s at time t using values at time $t-1$ and $t-2$. The model estimates the value e_t by using a linear equation represented by Eq. (3.1) over a sliding window with previous two time intervals. It achieves this by first solving m and c in Eq. (3.1) using the values $\{t-1, t-2\}$ and corresponding data values from sensor s: $\{v_{t-1}, v_{t-2}\}$.

$$e_t = m * t + c \tag{3.1}$$

Once it has an estimated value e_t at time t it is compared with the real value from sensor v_t; if the difference percentage δ_t calculated using Eq. (3.2) falls within the threshold percentage th, then the sensor does not communicate to the controller.

$$\delta_t = |e_t - v_t|/v_t * 100 \tag{3.2}$$

When the difference percentage δ_t doesn't compare with the th the sensor transmits the real value v_t and control message to update estimator. The algorithm for the model m_s can be summarized as follows:

Algorithm 1 Algorithm for sensor model m_s

```
Initialize:
    Transmit the values $(v_0, v_1)$ for time $(t_0, t_1)$
for each time $t$ do
    Solve for $m$, $c$ in Eq. (3.1) using $(v_{t-1}, v_{t-2})$ and $(t-1, t-2)$
    Estimate sensor data value $e_t$ at current time $t$
    Calculate difference percentage $\delta_t \leftarrow |e_t - v_t|/v_t * 100$
    if $\delta_t > th$ then                              ▷ If $\delta_t$ doesn't compare
        Transmit $v_t$
        Send control message
    end if
end for
```

On the other hand, model at controller m_c represented using Figure 2.2 also consists a linear estimator. Controller model m_c estimates the value e_t at time t and if no message is received from sensor s at the given interval, the controller reports estimated value e_t as the value from the sensor. It is important to note that e_t at both m_c and m_s are always identical at any given time t provided that there be no packet loss in the network. Further, we have studied the performance of model under packet loss in the Section 2.4.

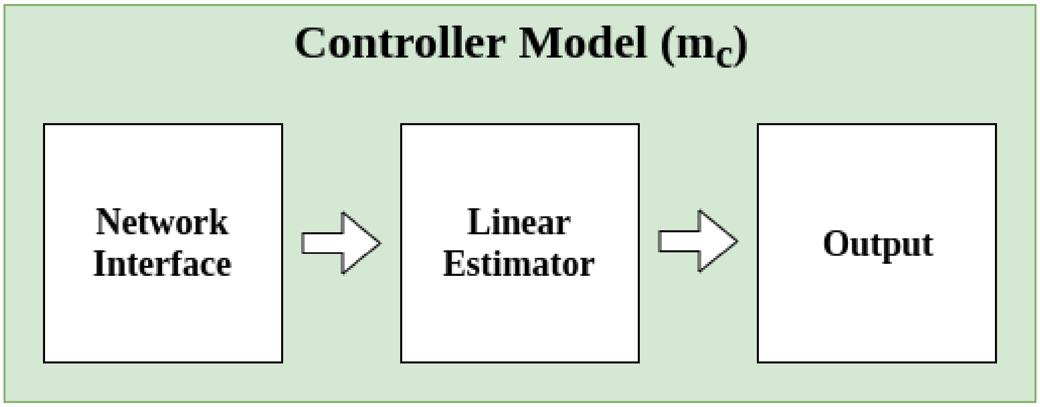


Figure 2.2: Controller Model m_c block diagram

The model m_c updates the parameters m, c in Eq. (3.1) on receiving the control messages and reports the received value v_t as the sensor s value. The model m_c can introduce a small error e_t but reduces traffic on the network. The strategy is represented with the following algorithm:

Algorithm 2 Algorithm for controller model m_c

```
Initialize:
    Receive the values $(v_0, v_1)$ for time $(t_0, t_1)$
    Solve for $m$, $c$ in Eq. (3.1) using $(v_0, v_1)$ and $(t_0, t_1)$
for each time $t$ do
    Estimate sensor data value $e_t$ at current time $t$
    if control message is heard then                ▷ If $\delta_t$ doesn't compare
        return $v_t$
    end if
    return $e_t$
end for
```

The above model has been validated using a sensor network simulated for different conditions. Further, this strategy has been inspired from Smart Refractory Systems project. In the following sections, the experiment and results are presented.

2.3 Experimental Setup

To validate the model, sensor network has been simulated using data from Mica2Dot sensors with weather boards deployed at Intel Berkeley Research Lab [13]. The data contains the time-stamped values of humidity, temperature, light, and voltage from 54 sensors. The data is queried once every 31 secs and collected between February 28th and April 5th, 2004. The entire simulation is implemented using a custom tool written in Python here at West Virginia University Distributed Systems and Networking Lab.

The model is validated for three different threshold percentage values i.e. $th = \{0.1, 0.5, 1\}$ for all 54 sensors. Further, packets are intentionally dropped to observe the behavior of models. The percentage of packets saved for each threshold and bar plots for the average of δ_t for all values t validate the strategy. In the following section, results are presented.

2.4 Results

2.4.1 Percentage of packets saved without packet loss

In the figure 2.3, the percentage of packets saved through the network for different threshold percentage (th) under no packet loss is presented. The Figure 2.3 shows the trend that when the threshold(th) is relaxed, the controller reports more data values that are estimated rather than queried from the real sensor. Further, it is important to note that even with the least allowed threshold percentage($th = 0.1$) the strategy saves about a median of 40% of the data packets.

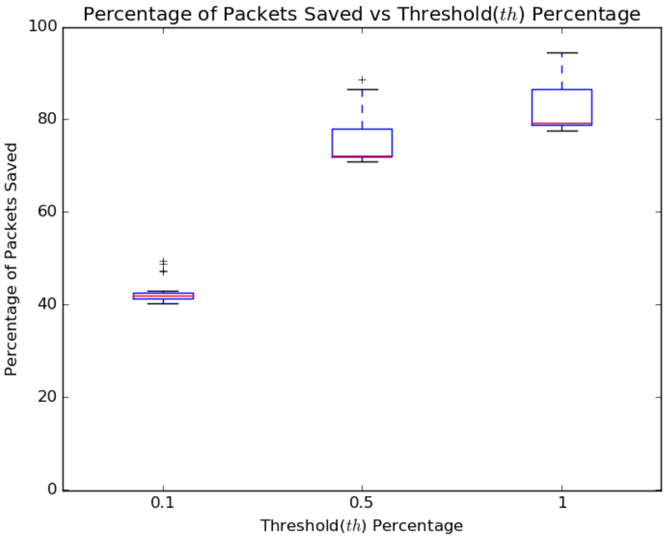
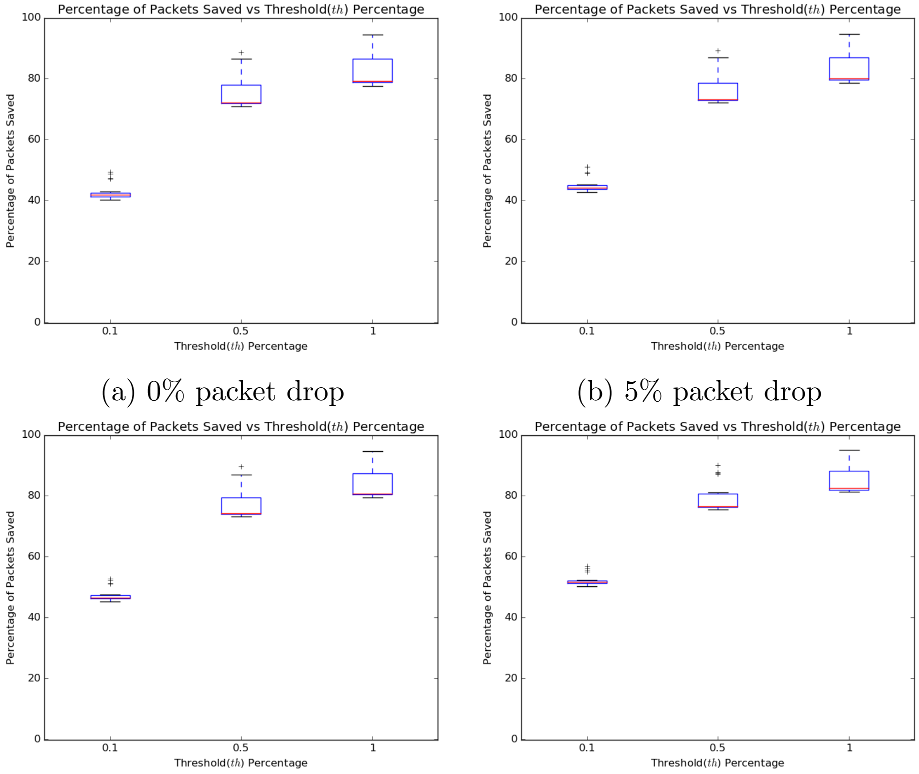


Figure 2.3: Percentage of Packets Saved vs Threshold Percentage (th)

2.4.2 Percentage of packets saved with packet loss

Further, the simulation is repeated for a list of packet drop percentages given by $\{5, 10, 20\}$. This is intentionally simulated to observe the behavior of the model at controller m_c when it doesn't receive the control messages to update the estimator. The Figure 2.4 presents all the plots for comparison. It is evident that although there is a packet drop the percentage of packets saved doesn't vary significantly.



(a) 0% packet drop

(b) 5% packet drop

(c) 10% packet drop

(d) 20% packet drop

Figure 2.4: Percentage of Packets Saved vs Threshold Percentage (th) under packet drop

Another important observation is that individual sensor values reported by the controller at any given time t should be close to the real sensor value v_t. This is validated by comparing the threshold percentage (th) with the average of actual difference percentage (δ_t) at all values of t calculated using Eq. (3.3).

$$\text{average of difference percentage} = \sum_{t=1}^{n} (|e_t - v_t|/v_t * 100)/n \tag{3.3}$$

2.4.3 Average difference percentage without packet loss

The box plot in Figure 2.5 suggests that although there is allowed threshold percentage(th); the majority of the reported sensor values has a lower difference percentage than (th).

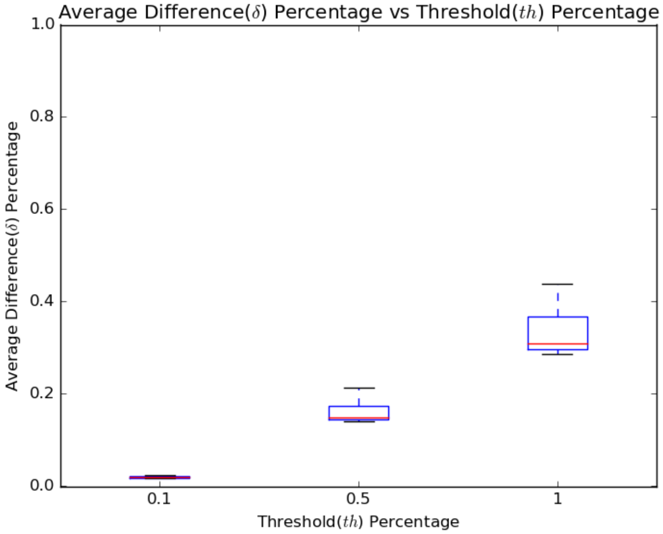
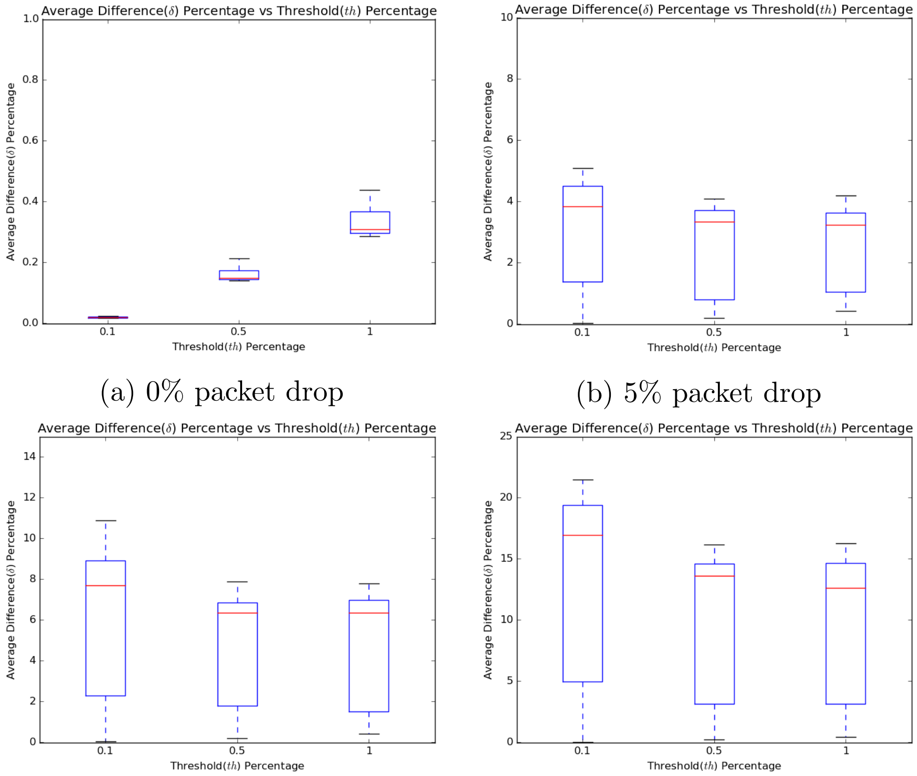


Figure 2.5: Average Difference Percentage (δ) vs Threshold Percentage(th)

2.4.4 Average difference percentage with packet loss

Further, Figure 2.6 contains the box plots for different packet drop values. Clearly, the spread of average difference percentage(δ) increases as the packet drop percentage increases. A counter-argument can be made that while this strategy ensures that there is less load on the network, one can expect a low packet drop.



(a) 0% packet drop

(b) 5% packet drop

(c) 10% packet drop

(d) 20% packet drop

Figure 2.6: Average Difference Percentage (δ) vs Threshold Percentage(th) under packet drop

2.5 Conclusion

This model presents a generic approach to optimize the sensor network infrastructure using the information generated from the sensor. Such information-centric approach has the flexibility to be applied to any underlying network infrastructure. The algorithms have been presented and experimentally validated. The initial results have been satisfactory and such models have a lot of potential to be investigated.

Chapter 3

An adaptive information exchange strategy based on rate of change of state for static and mobile networks

3.1 Overview

Similar to the strategy presented in Chapter 2, this strategy aims to minimize traffic in the network generated from sensors thereby reducing packet collisions in the network. The idea is to use rate of change of current sensor state as a key metric to dynamically vary communication interval of a sensor. Data from sensors change with respect to time, and a sensor state can be defined based on the value of the sensor. We use the rate of change of the defined state as key a metric; if it is identified that sensor state is changing at a greater rate, the strategy adjusts the interval such that sensors communicate more often and vice versa.

This strategy is inspired by a real-world use case. An important component of Integrated Sensing Systems for Surface Mining Safety project is to build MapMyTruck, a cloud-based logging system for GPS data from trucks for traffic analysis and route planing. The challenge is to build sensor cloud infrastructure while complying with bandwidth limitations of the mobile network. The rest of the chapter includes the methodology, experimental setup, results and MapMyTruck use case implementation.

3.2 Methodology

Consider a sensor s, at any given time t, sensor has a value v_t and a sensor state Ψ_t based on this value. We define the rate of change of state ROC_Ψ as the difference between sensor state values Ψ_t and Ψ_{t-1} over time interval t_r; the same is represented using Eq. (3.1). Unlike the strategy in Chapter 2, frequency of the sensor varies with time such that at given time t, frequency f_t is the value corresponding to rate of change of state ROC_Ψ. A discrete number of frequency values are mapped to the ROC_Ψ given by tuple $\{f_0, f_1, f_2, f_3 ...... f_n\}$.

$$\mathrm{ROC}_\Psi \leftarrow (\Psi_t - \Psi_{t-1})/t_r \tag{3.1}$$

Although this strategy applies to both static and mobile sensor networks, it is particularly useful if the sensor state is constantly varying and data is not required for every state. For example, if a given sensor is in an idle state (which can be identified using the sensor value); the corresponding interval could be set to a relatively higher value to reduce the congestion on the network. This strategy does not require any additional logic at the controller.

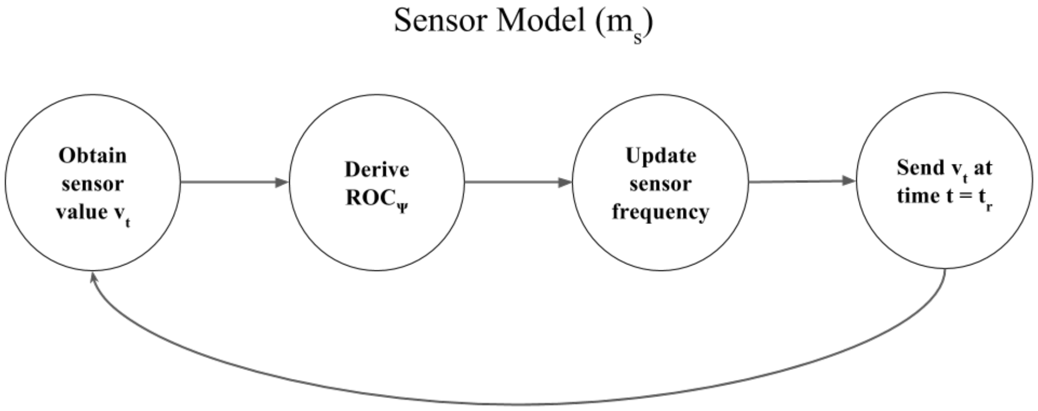


Figure 3.1: Sensor Model m_s block diagram

The sensor model m_s is represented in the Figure 3.1. The model m_s contains a set of rules that map different rate of change of sensor state ROC_Ψ and corresponding frequency. For each time interval $_rt$ sensor has a state Ψ_t, using Eq. (3.1) ROC_Ψ is obtained. The current frequency $f_{current}$ is updated based on the rules and if the interval duration is expired, then the value v_t is sent over the network. The algorithm for sensor model m_s is described as follows:

Algorithm 3 Algorithm for sensor model m_s

```
Initialize:
    Define rules R
for each time interval $t_r$ do
    Obtain rate of change of sensor state $\text{ROC}_\Psi$ using Eq. (3.1)
    Update $f_{current}$(current frequency) ← $f_t$(corresponding frequency)
end for
```

To evaluate the model, a sensor network is simulated, and a simplified version of the algorithm is implemented. The detailed algorithm is at the core of MapMyTruck module. The experiment and results are presented in the next section.

3.3 Experimental Setup

As described in the Section 2.3, the experiment consists of a sensor network simulated using the data from Mica2Dot sensors with weather boards deployed at Intel Berkeley Research Lab [13]. A custom tool written in Python is used to conduct the experiment here at West Virginia University Distributed Systems and Networking Lab.

The experiment consists of a simplified version of the Algorithm 3 described below:

Algorithm 4 Algorithm for sensor model m_s

```
Initialize:
    Transmit the value $v_0$ for time $t_0$
for Each time $t$ do
    Calculate rate of change of state $\text{ROC}_\Psi \leftarrow |v_t - v_{t-1}|/v_{t-1} * 100$
    if $\text{ROC}_\Psi > th$(threshold percentage) then
        $f_{current}$(current frequency) $\leftarrow f_{high}$
    else
        $f_{current}$(current frequency) $\leftarrow f_{low}$
    end if
end for
```

The Algorithm 4 described above has two major differences from the Algorithm 3. First, the algorithm uses different equation to obtain range of change of state ROC_Ψ given by Eq. (3.2) instead of Eq. (3.1). Second, instead of defining tuple of frequency values $\{f_0, f_1, f_2, f_3......f_n\}$, the algorithm has only two frequency values: $\{f_{low}, f_{high}\}$. The state is identified as active if $\text{ROC}_\Psi > th$(threshold percentage) is satisfied and the sensor s will transmit at frequency f_{high}. Otherwise, the sensor s is in a idle state and no transmission is recommended. Hence, a map of rules R is not required.

$$\text{ROC}_\Psi \leftarrow |v_t - v_{t-1}|/v_{t-1} * 100 \tag{3.2}$$

Further, the model is validated for five different threshold percentage values i.e. $th = \{1, 2, 3, 4, 5\}$ for all 54 sensors. The bar plot between the percentage of packets saved and threshold percentage (th) for all values of t validate the model. The following section contains the results.

3.4 Results

In the figure 3.2, the bar plot between the percentage of packets saved through the network for different threshold percentage (th) is presented. The figure 3.2 shows that there is a significant reduction in the packets sent through the network. A minimum threshold value ($th = 1\%$) saves about 60% of the packets. Also, there isn't a significant increase in the percentage of packets saved by increasing the threshold percentage (th).

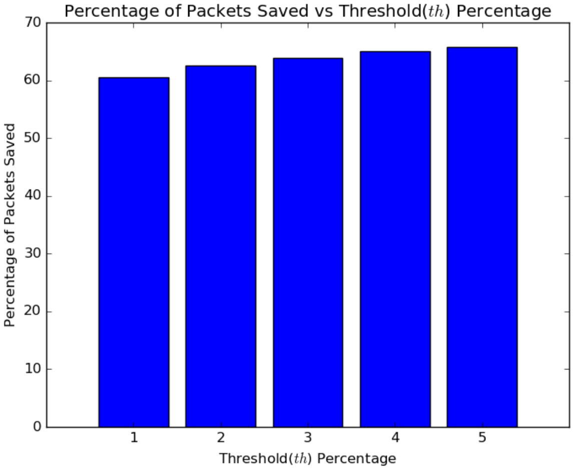


Figure 3.2: Percentage of Packets Saved vs Threshold Percentage (th)

3.5 Use Case: MapMyTruck

3.5.1 Overview

MapMyTruck is a sub-system of Integrated Sensing Systems for Surface Mine Safety Project. The goal of the tool is to log GPS traces for long term collision safety evaluation. MapMyTruck is available to download at: `http://bit.ly/mapmytruck`. The tool is the inspiration behind developing this strategy and has been tested in Red Hills Mine in Ackerman, Mississippi. During the initial testing, MapMyTruck saved up to 86.1% of data transmitted to cloud.

3.5.2 System Architecture

This section details the architecture designed to achieve the goal of the MapMyTruck described in section 3.5.1.

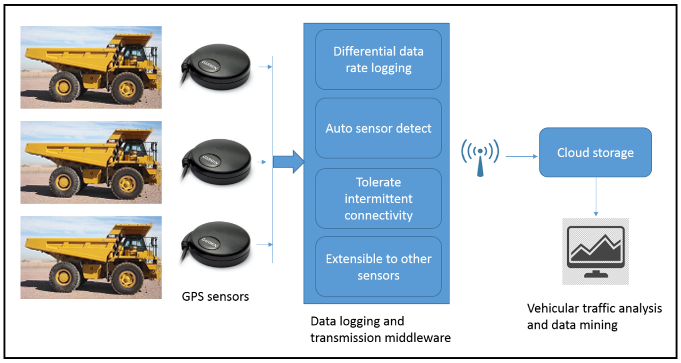

Differential data rate logging
Auto sensor detect
Tolerate intermittent connectivity
Extensible to other sensors
Cloud storage
GPS sensors
Data logging and transmission middleware
Vehicular traffic analysis and data mining

Figure 3.3: MapMyTruck system architecture

MapMyTruck architecture is described in Figure 3.3. The GPS information is logged from the CAT 769 haul trucks. A Garmin GPS device attached to a Windows Tablet running the .NET framework in each of the trucks. Upon installation, the GPS would log location coordinates locally as well as establish a remote connection to an FTP server and upload the data. Some of the features of this system are as follows:

- **Speed based logging rate:** Logging all the data periodically (even at a 10 second interval) would generate a huge amount of data which is not cost effective (Internet Service Providers and cloud services charge based on data transferred). For example, if each log is 100 bytes long, logging at 1 Hz would generate 360 Kbytes per truck per day. On the other hand, decreasing the logging frequency will prevent the data from being useful when vehicles come closer. Hence, rate of change based information exchange strategy has been designed. The idea is to use rate of change of position, i.e. speed to determine the logging frequency. Hence, it acts as a speed based differential logging service which logs and transmits data proportional to the speed. No data is transferred when the vehicle is idle and data rate is progressively increased with speed and is transferred at 1 Hz at speeds of 25mph and above. Further, Table 3.1 provides detailed information on logging frequency at different speeds.

Table 3.1: Logging rate as a function of vehicle speed and the corresponding precision in track accuracy; the precision denotes the maximum distance that can be travelled by a truck without logging any update

Speed (mph)	**Speed (mps)**	**Frequency (s)**	**Precision (m)**
<0.05	0.02572	600	15.432
0.05 - 2.3	1.0288	20	20.576
2.3 - 4.6	2.0576	10	20.576
4.6 - 9.2	4.1152	5	20.576
9.2 - 18.4	8.2304	2.5	20.576
18.4 - 36.8	16.4608	1	16.461
36.8 - 73.6	32.9216	0.5	16.461

- **Tolerates intermittent connection:** It is not feasible to assume that network connection is always available in a mine. So the service logs data locally, and whenever connection is available, it uploads the data.

- **Auto package installer and auto sensor detect:** The service can be installed with an easy to use package installer and starts itself automatically when a GPS device is detected.

3.5.3 Results

MapMyTruck has been validated in a real-world scenario with the rate of change based information exchange strategy at it's core. The Figure 3.4 shows the GPS trace plotted using a custom tool developed in Java here at West Virginia University Distributed Systems and Networking Lab. On a normal working day, MapMyTruck logs about 50 Kbytes of data per truck opposed to 360 Kbytes saving about 86.1% of data.

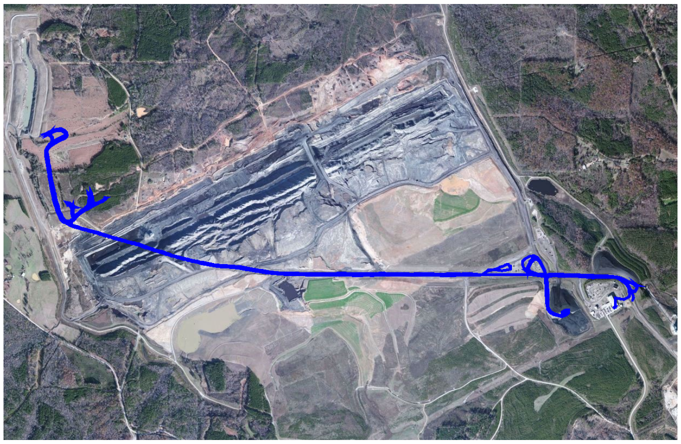


Figure 3.4: Sample track collected from the MapMyTruck for one of the trucks in Red Hills Mine on a working day

3.6 Conclusion

The model presented in this section optimize the sensor cloud infrastructure using a rate of change based information exchange strategy. The strategy can be generalized to any sensor cloud infrastructure to effectively reduce Internet Service Providers and Cloud Services costs. The results presented in Section 3.4 and Section 3.5.3 validate the model both experimentally and practically through MapMyTruck.

Chapter 4

Spatio-temporal strategy that maintains spatial resolution by reducing redundant transmissions

4.1 Overview

This section describes a spatio-temporal strategy for the mobile sensor networks. In a mobile sensor network, the nodes position changes with respect to time. The goal of this strategy is to obtain information from these spatially distributed nodes such that the information always maintains a spatial resolution over the time period. This strategy is particularly useful for monitoring and surveying applications [24] where sensor nodes are distributed, and the information is aggregated to obtain knowledge about environment parameters such as atmosphere humidity, water level, wind direction, etc.

In such monitoring and surveying applications, aggregating information from all the sensor nodes will have redundancies because the nodes that are in close proximity contain same knowledge about the environment parameter. Further, these spatially distributed nodes communicate in no particular order, i.e. it is not guaranteed that there is a spatial resolution of the region maintained in this information. A centralized control can be used to orchestrate among the nodes, but that would be computationally costly. To further illustrate the case, a snapshot of mobile sensor network scenario is presented in the Figure 4.1.

In the Figure 4.1, the sensor nodes are labeled and spatially distributed over the region. A virtual grid is overlayed through the region. Ideally, by maintaining a spatial resolution, the information from these mobile sensor nodes will detail about the whole region, i.e. data from all the squares in the virtual grid is communicated and maintained temporally. Also, multiple sensor nodes within close proximity (within the same square in the virtual grid) may introduce redundancies.

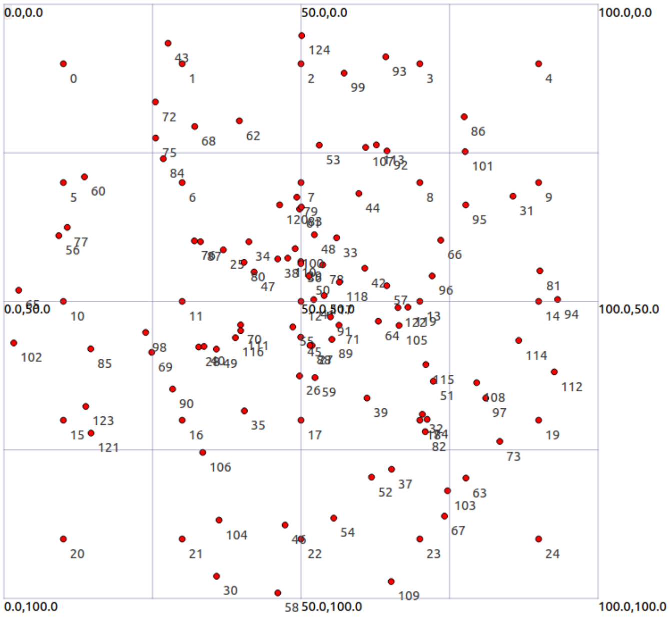


Figure 4.1: A snapshot of mobile sensor network scenario

The proposed spatio-temporal strategy uses a distributed asynchronous algorithm to achieve the goal of eliminating the redundancies while maintaining a spatial resolution without a need for centralized control. This model achieves the objective while transferring a minimum number of packets through the underlying network. The methodology, experimental setup, and results are presented in the following sections.

4.2 Methodology

Consider a mobile sensor network spatially distributed over a region with boundaries (x_{min}, y_{min}), (x_{min}, y_{max}), (x_{max}, y_{min}) and (x_{max}, y_{max}). Consider a mobile sensor node s in this network, the position $P_s =(x_s,y_s)$ varies according to time t as a function $D(t)$ such that the position P is always inside the boundaries. The same is represented using Eq. (4.1).

$$P_s = D(t) \mid x_{min} <= x_s <= x_{max} \text{ and } y_{min} <= y_s <= y_{max} \tag{4.1}$$

Consider a monitoring and surveying application that uses spatially distributed mobile wireless sensor network. Once deployed, a sensor node s has knowledge about an environment parameter such as atmosphere humidity, water level, wind direction, etc. at its location $P_s =(x_s,y_s)$. Each sensor node s listens to any transmission from neighbouring nodes before transmitting it's sensor value v_t at time t. Further, if all the nodes act synchronously, i.e. all the nodes listen and transmit at the same time the strategy does not have any effect. Hence, each node waits for a random amount of time t_w for sniffing any transmissions from neighboring nodes.

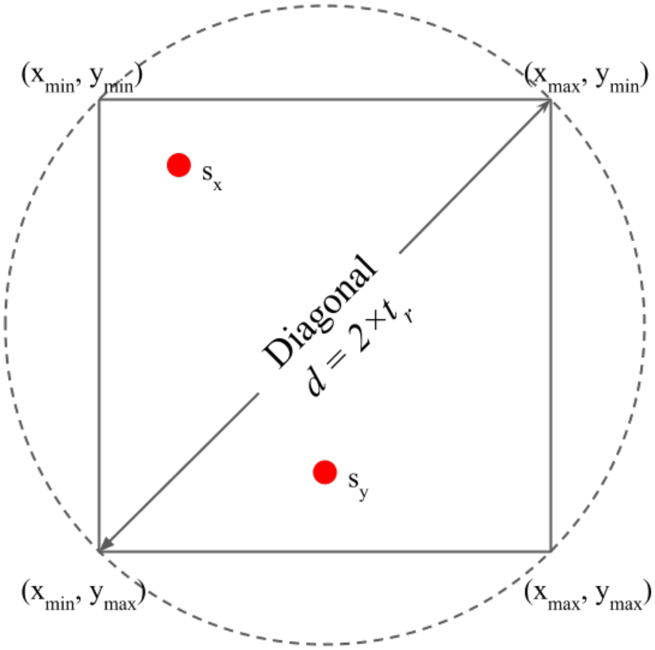

(x_{min}, y_{min})
(x_{max}, y_{min})
s_x
Diagonal
d = 2×t_r
s_y
(x_{min}, y_{max})
(x_{max}, y_{max})

Figure 4.2: A scenario with two sensor node s_x and s_y.

Consider two sensor nodes s_x and s_y such that their positions P_x and P_y are in communication range t_r as illustrated in Figure 4.2; for each time interval t_r, sensor nodes waits for a time $t_w = R(0.5 \times t_r, 0.8 \times t_r)$ during which sensor node s_x listens to any transmission from other sensor node s_y and vice-versa. The waiting time t_w is different for sensor nodes s_x and s_y. If either of the nodes listens to other node during the waiting time t_w it doesn't transmit the information, thereby redundant information is suppressed between s_x and s_y. The algorithm at each sensor node s is described below:

Algorithm 5 Distributed algorithm at each sensor node s

```
for each time interval $t_r$ do
    Wait for a time $t_w \leftarrow R(0.5 \times t_r, 0.8 \times t_r)$
    if no transmission is heard then
        Transmit $v_t$
    end if
end for
```

In addition to filtering redundant information, the strategy ensures that the sensor nodes only communicate once for the unit of region bounded by communication range t_r in a distributed manner. Consider the case represented in Figure 4.2, either s_x or s_y will transmit the information for every time t_r but not both of them for the unit region. This is true even when there are n sensor nodes $s_1, s_2, s_3....s_n$ in the region bounded by (x_{min}, y_{min}), (x_{min}, y_{max}), (x_{max}, y_{min}) and (x_{max}, y_{max}). In the following section, the experimental setup is detailed.

4.3 Experimental Setup

The proposed strategy is validated using a simulated mobile wireless sensor network in Network Simulator 3 (NS-3). The simulation consists of 100 mobile nodes spatially distributed in a 5×5 grid such that each unit satisfies the Eq. (4.2), similar to the case represented in Figure 4.2.

$$d \text{ (Diagonal of unit cell in grid)} = 2 \times t_r \tag{4.2}$$

The position of the nodes varies with time using a random walk mobility model. The random walk mobility model mimics an erratic movement of sensor nodes. The model retains no knowledge of the past position and speed. Using the model, each mobile sensor node position is determined by randomly choosing the direction and speed for a constant time interval or after covering a constant distance. If any sensor node reaches the boundaries, it retracts its position into the bounded region. The simulation is repeated for 20 runs with different seeding for each run.

The analysis for the simulation is done using a custom tool written in Python here at West Virginia University Distributed Systems and Networking Lab. The analysis is done for five different precision values, i.e. frequency precision $= \{20\text{ Hz}, 10\text{ Hz}, 2\text{ Hz}, 1\text{ Hz}, 0.2\text{ Hz}\}$; frequency precision is defined as a measure for how often data is required. The experiment is done with and without implementing distributed algorithm described in Section 4.2 for all 100 mobile sensor nodes. Finally, the percentage of packets saved for each frequency precision value is plotted. The results are presented in the following section.

4.4 Results

The Figure 4.3 shows the percentage of packets saved through the network for each frequency precision value using a bar plot for 20 runs. Further, a value frequency precision = 20 Hz implies that each sensor node transmits the data once every 0.05 Seconds.

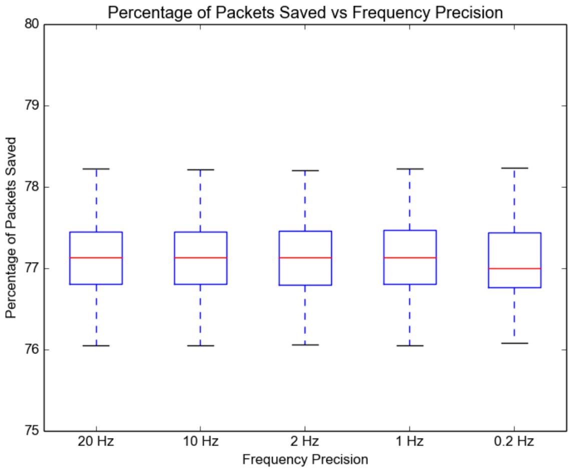


Figure 4.3: Percentage of Packets Saved vs Frequency Precision

The Figure 4.3 suggests that a median of 77% of the packets sent through the network can be saved using the strategy for frequency precision = 20 Hz and there is almost similar behavior for all frequency precision values. Hence, the percentage of packets saved through the network using the current strategy remains same as the frequency precision value decreases, i.e., when time interval after which each sensor node transmits the data is relaxed.

4.5 Conclusion

The presented model prescribes an asynchronous distributed strategy for mobile sensor nodes that maintains spatial resolution over a region while optimizing the number of packets sent through the network. Such a strategy eliminates the need for centralized control and agnostic to the mobility model. The algorithm presented has been experimentally validated.

Chapter 5

Proximity-dependent data transfer strategy to ensure most updated information in high-density regions

5.1 Overview

In the Chapter 2, Chapter 3 and Chapter 4, we have proposed strategies to optimize the packets transmitted in the network by communicating less often or not communicating at all when the current data is not contributing to the overall quality of information. Contrary to the previous strategies, this chapter presents a strategy that aims to communicate more often for effectively maintaining the quality of information.

We propose an information centric strategy for Vehicular Ad hoc Network (VANET). Vehicular Ad hoc Network (VANET) are a type of Mobile Ad hoc Networks (MANETs) that help in establishing a wireless ad-hoc network between vehicle to vehicle (V2V) and vehicle to infrastructure (V2I). VANET is an essential component for building Intelligent Transportation Systems (ITS); specially VANET is being investigated for safety applications such as Cooperative Collision Warning System, Collision Avoidance System, Path Tracking Modules, etc [17] [8] [14] [19].

Such an application requires that each node has the most updated information from the neighboring nodes which can be achieved by communicating more frequently. Unlike traditional wireless network, VANET comprises of varied network densities [23] and ensuring the most updated information at high density regions is a challenge. Also, broadcasting the information for each interval is an overhead on the network. This strategy proposes a distributed algorithm to achieve the objective.

The idea is to use the proximity of neighboring nodes as a key metric to dynamically adjust the frequency. This model is inspired from GPS based Cooperative Collision Warning System developed for Integrated Sensing Systems for Surface Mining Safety project [10]. The module uses 802.11 family of radios to establish an ad-hoc network between the haul trucks in surface mines. In the following sections, the methodology, experimental setup, and results are presented.

5.2 Methodology

In a Cooperative Collision Warning System consisting of n vehicles $\{\vee_0, \vee_1, \vee_2.......\vee_n\}$ moving on a highway, the position of the vehicles vary according to their respective velocities $\{\upsilon_0, \upsilon_1, \upsilon_2.......\upsilon_n\}$. Ideally, the vehicles broadcast their respective positions, and if the chance for collision is detected between two vehicles, warnings are triggered at each vehicle. This strategy aims to minimize the number of packets in the network using a distributed algorithm that runs at each vehicle in the Cooperative Collision Warning System.

Consider two vehicles $\vee_A$ and $\vee_B$ moving on a highway with respective velocities υ_A and υ_B. Each vehicle has the ability to communicate over a range R_c i.e. the packets advertised by a certain vehicle can be heard by another vehicle when the distance between them is less than the R_c. The vehicles are broadcasting periodically once for every time interval t_n determined by frequency f_n.

The vehicles are at a safe distance when the distance d between $\vee_A$ and $\vee_B$ is greater than the communication range R_c. The same is true for n vehicles on a highway, i.e. it is safe to assume that vehicles are at a safe distance if they cannot hear each other. Hence, the strategy prescribes to communicate less often by doubling the time interval t_n until t_{high} (highest time interval) when there is no transmission heard from neighboring vehicles.

Alternatively, the vehicles are in close proximity when distance d between $\vee_A$ and $\vee_B$ is less than the communication range R_c. The strategy identifies when a transmission is heard and prescribes to communicate more often by halving the time interval t_n until t_{low} (lowest time interval). Further, while the proposed scenarios consist of only two vehicles $\vee_A$ and $\vee_B$, the presented case for the chance of a collision hold true for n vehicles.

By identifying the chance of collision based on the proximity between two vehicles d will enable them to adjust the time interval t_n accordingly. Further, this adaptive mechanism will enable vehicles to communicate more often when there are more neighbors, i.e. the high-density regions will have the most updated information to enable safety critical applications such as Cooperative Collision Warning System. The distributed algorithm is summarized using the Algorithm 6. Further, the parameters t_{low} and t_{high} can be tuned to ensure that the t_{low} is high enough to avoid broadcast storm situation.

Algorithm 6 Proximity adaptive algorithm

```
Initialize:
    Set $t_{low}$ (lowest time interval), $t_{high}$ (highest time interval)
for each time $t$ do
    if transmission is heard and $t_n > t_{low}$ then
        $t_n \leftarrow t_n/2$
    else
        if $t_n < t_{high}$ then
            $t_n \leftarrow 2 \times t_n$
        end if
    end if
end for
```

The proposed strategy is tested using a simulated VANET. Further, the strategy helped in devising a VANET for Integrated Sensing Systems for Surface Mining Safety project. The detailed experimental setup and results are presented in the following sections.

5.3 Experimental Setup

The proposed distributed algorithm presented using Algorithm 6 is evaluated using a VANET simulated in Network Simulator 3 (NS-3). The simulation consists of a highway intersection as depicted in Figure 5.1. The vehicles V_0, V_1, V_2 and V_3 depicted in the Figure 5.1 move in four different directions; vehicles move from bottom to top, top to bottom, left to right and right to left respectively. However, the actual simulation consists of more one vehicle in each direction and an equal number of vehicles moving in each direction.

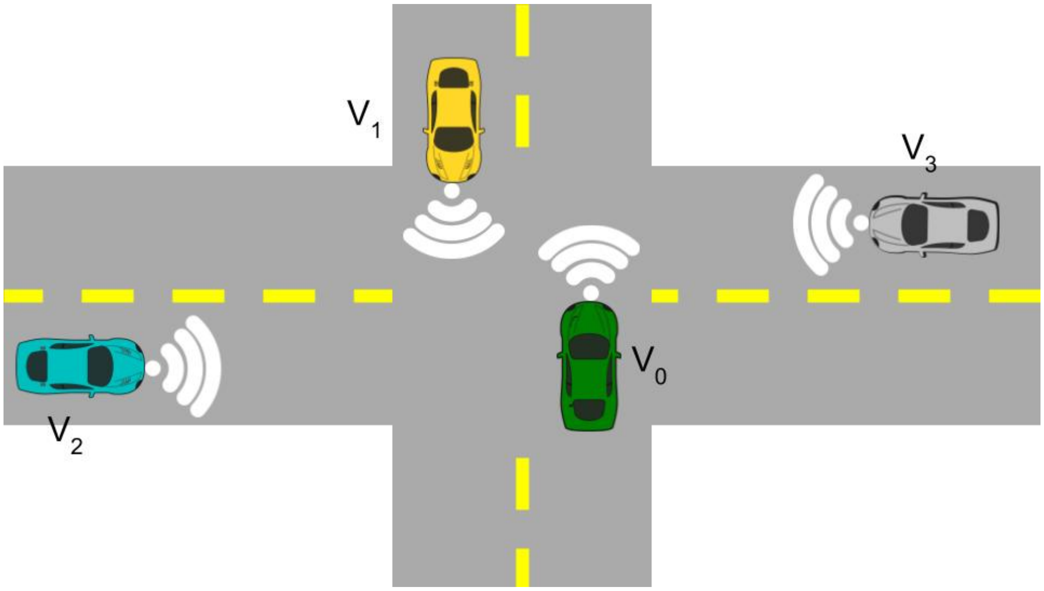


Figure 5.1: Schematic of a VANET for simulation in Network Simulator 3 (NS-3)

A different number of vehicles in tuple {40, 80, 120} are dispatched for an interval of 300 seconds over the entire simulation time. For example, from the starting time of the simulation t_0 to t_0 +300 s, there are 40 vehicles participating in the simulation, i.e., 10 vehicles moving in each direction and so on. The vehicles are initialized at a random position on the road using random rectangle position in NS-3. The nodes are assigned random velocities in the range of 25 mph - 70 mph and the position is determined using constant velocity mobility model in NS-3. The vehicles on each of four lanes start from one end of the road and travel in their respective direction. Node density is calculated as the number of vehicles traveling through a unit road segment area, and it is equal to the number of vehicles in the simulation.

The impact of variations in node density is observed. Since our focus is to observe the effects on the network for different node density values, we ignore the effects of the collision between vehicles. For each experiment setting, the simulation is repeated for 20 runs with random seeding. Similar to the previous strategies, a Python based tool is written here at West Virginia University Distributed Systems and Networking Lab for performing the analysis. The percentage of packets saved with and without using the strategy for *node density* = {40, 80, 120} is analyzed using this tool. The results are detailed in the following section.

5.4 Results

For each *node density*, the percentage of packets saved through the network with and without using the strategy is represented using a box plot for all 20 runs in Figure 5.2. As it is shown in the Figure 5.2, it is observed that a median value of 67.17% of packets sent through the network saved using this strategy for *node density* = 40. Unlike the strategy presented in the Chapter 4, the median value of the percentage of packets saved using the strategy decreases as node density increases.

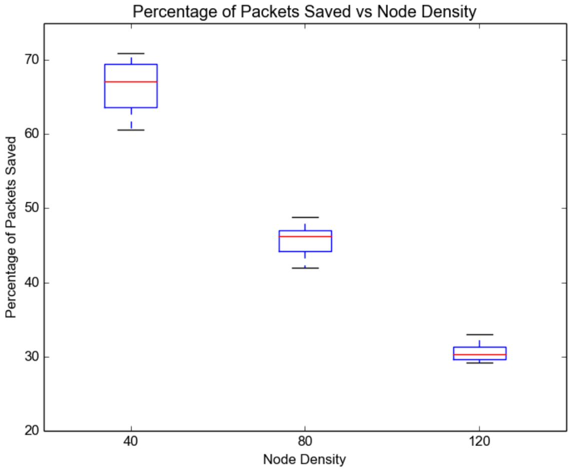


Figure 5.2: Percentage of Packets Saved vs Node Density (No. of vehicles in the simulation)

5.5 Conclusion

A distributed strategy that aims to ensure the most updated information at high density regions in a VANET is presented. The strategy achieves the objective while optimizing the number of packets in the network. The Algorithm 6 is validated using a simulated VANET. The strategy is observed to save packets broadcasted in the network for different values of *node density*.

Chapter 6

Conclusion and Future Work

6.1 Contributions

In this thesis, we have proposed four different information centric strategies for Cyber-Physical System(s). Although each strategy addresses a different challenge, they all share one common goal, i.e. optimizing the traffic without significant modifications to the underlying infrastructure. Further, the strategies are inspired by the challenges encountered while designing different systems here at West Virginia University Distributed Computing and Networking Lab. We list the proposed strategies below.

1. First, we propose a linear estimator based strategy for a generic networked sensing system. Using Eq. (3.1), the sensor value is estimated at sensor model m_s and data is only transmitted if the value is greater that certain threshold; otherwise, the controller model m_c uses estimated value as current sensor value. The strategy is validated using a simulated sensor network using data from weather boards deployed at Intel Berkeley Research Lab. We have observed that the number of packets sent through the network reduced by at least 40% calculated as a median over 54 sensors. Further, we have observed that the model introduces more error in the sensor value using the strategy under packet loss. Such a generic optimization can be coupled with other strategies

for better results.

2. Second, a data optimization strategy is proposed to address the challenges encountered while designing MapMyTruck as part of Integrated Sensing Systems for Surface Mine Safety Project. We propose an algorithm for sensor model m_s that identifies the sensor state Ψ_t and adjusts the current frequency $f_{current}$ based on a set of rules R. The strategy is experimentally validated and observed to save at least 60% of the packets sent through the network. Further, we have implemented the MapMyTruck application for haul trucks in Red Hills Mine in Ackerman, Mississippi and observed to save about 86.1% of data on a normal working day.

3. Third, for spatially distributed mobile sensor nodes, we have proposed a strategy to ensure a spatial resolution over a region while transmitting a minimum number of packets using a distributed algorithm presented using Algorithm 5. The proposed strategy is experimentally validated using a simulated mobile wireless sensor network in Network Simulator 3 (NS-3). We have seen that a median of 77% of packets can be saved using the strategy. Further, the strategy works very well when the data is needed more frequently.

4. Fourth, a distributed strategy for vehicles participating in a Vehicular Ad hoc Network (VANET) is proposed. Although we did not evaluate the strategy in the real world like MapMyTruck, it is designed for a GPS based Cooperative Collision Warning System developed for Integrated Sensing Systems for Surface Mining Safety project. We use the proximity of the vehicles to dynamically adjust the communication interval t_n. The strategy is found to save a median value of 67.17% of the packets when validated using a simulated VANET for $\textit{node density} = 40$ in Network Simulator 3 (NS-3).

Overall, the proposed strategies found to reduce the traffic in the network and there by reducing the packet collisions in the network. Specifically, applications like MapMyTruck have found to reduce the operational and carrier costs. We summarized the directions for future work in the following section.

6.2 Future Work

Cyber-Physical System(s) encompasses complex processes with communication at the core. A synergy between robust infrastructure and application strategies is very much needed to build such complex ecosystem(s). We acknowledge the fact that there are a great number of ecosystems each with different subsystems, components, and priorities; not all of them have been addressed in this thesis. With the proposed strategies, we would like to evaluate the performance when multiple strategies are combined. For example, generic strategy presented in Chapter 2 can be coupled with any of the other three strategies presented in Chapter 3, Chapter 4 and Chapter 5.

The strategy presented in Chapter 3 is evaluated using MapMyTruck in real world scenario, and we would like to implement other three strategies and observe the performance. Also, the strategies can be experimentally validated for larger simulation environments with more processes and sensor nodes. Further, we have observed satisfactory results using a linear model in Chapter 2 and we would like to investigate the performance of more complex machine learning techniques for information centric strategies.

References

[1] M. Mehdi Afsar and Mohammad-H. Tayarani-N. Clustering in sensor networks: A literature survey. *Journal of Network and Computer Applications*, 46:198 – 226, 2014.

[2] August Betzler, Carles Gomez, Ilker Demirkol, and Josep Paradells. Cocoa+: An advanced congestion control mechanism for coap. *Ad Hoc Networks*, 33:126 – 139, 2015.

[3] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, and M. Varvello. From content delivery today to information centric networking. *Computer Networks*, 57(16):3116 – 3127, 2013. Information Centric Networking.

[4] Chih-Yung Chang, Jang-Ping Sheu, Sheng-Wen Chang, and Yu-Chieh Chen. A frequency-aware data-centric mechanism for wireless sensor networks. *Wirel. Commun. Mob. Comput.*, 10(8):1078–1101, August 2010.

[5] Jiann-Liang Chen, Han-Chuan Hsieh, and Y. T. Larosa. Congestion control optimization of m2m in lte networks. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, pages 823–827, Jan 2013.

[6] Grit Denker, Nikil Dutt, Sharad Mehrotra, Mark-Oliver Stehr, Carolyn Talcott, and Nalini Venkatasubramanian. Resilient dependable cyber-physical systems: a middleware perspective. *Journal of Internet Services and Applications*, 3(1):41–49, 2012.

[7] Truong-Xuan Do and Younghan Kim. Information-centric wireless sensor and actor network in the industrial network. In *2013 International Conference on ICT Convergence (ICTC)*, pages 1095–1096, Oct 2013.

[8] Tamer ElBatt, Siddhartha K. Goel, Gavin Holland, Hariharan Krishnan, and Jayendra Parikh. Cooperative collision warning using dedicated short range wireless communications. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, VANET '06, pages 1–9, New York, NY, USA, 2006. ACM.

[9] Bronwyn Flores. The expanding reach of iot. *Dealerscope*, 57(11):8–9, 09 2015. Copyright - Copyright North American Publishing Company d/b/a NAPCO Media Sep 2015; Document feature - Photographs; Graphs; Tables; Last updated - 2015-10-10; SubjectsTermNotLitGenreText - United States–US.

[10] Ajay Kavuri, Bisleshana Prakash, Venkataraghavasivanagashashank Sabniveesu, Ashish Nimbarte, Vinod Kulathumani, and Vladislav Kecojevic. An adaptive, run-time navigation system for haul trucks in surface mines. *International Journal of Mining, Reclamation and Environment*, 0(0):1–11, 0.

[11] Ajay Kavuri, Bisleshana Prakash, Venkataraghavasivanagashashank Sabniveesu, Ashish Nimbarte, Vinod Kulathumani, and Vladislav Kecojevic. An adaptive, run-time navigation system for haul trucks in surface mines. *International Journal of Mining, Reclamation and Environment*, pages 1–11, 2016.

[12] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. *SIGCOMM Comput. Commun. Rev.*, 37(4):181–192, August 2007.

[13] Intel Berkeley Research lab. Intel Lab Data. `http://db.csail.mit.edu/labdata/labdata.html`. [Online; accessed 29-October-2016].

[14] Robert Lasowski and Florian Gschwandtner. A novel beaconing approach based on sender-centric movement predictions for collision avoidance in vanets. In *Proceedings of the Eighth ACM International Workshop on Vehicular Inter-networking*, VANET '11, pages 81–82, New York, NY, USA, 2011. ACM.

[15] Yang Liu, Zhikui Chen, Xiaoning Lv, and Feng Han. Multiple layer design for mass data transmission against channel congestion in iot. *International Journal of Communication Systems*, 27(8):1126–1146, 2014.

[16] Haining Mo, Neeti Sharad Wagle, and Michael Zuba. Cyber-physical systems. *XRDS*, 20(3):8–9, March 2014.

[17] Rola Naja. *Wireless Vehicular Networks for Car Collision Avoidance.* Springer Publishing Company, Incorporated, 2013.

[18] Robert R. Schaller. Moore's law: Past, present, and future. *IEEE Spectr.*, 34(6):52–59, June 1997.

[19] Zeeshan Shafiq, Latif Ullah Khan, Sahibzada Ali Mahmud, and Gul Muhammad Khan. Context-aware cooperative collision avoidance vehicle braking alert system for vanet. In *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications*, ICCASA '14, pages 92–96, ICST, Brussels, Belgium, Belgium, 2014. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[20] I. Stojmenovic. Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems. *IEEE Internet of Things Journal*, 1(2):122–128, April 2014.

[21] Chun-Wei Tsai, Chin-Feng Lai, and Athanasios V. Vasilakos. Future internet of things: open issues and challenges. *Wireless Networks*, 20(8):2201–2217, 2014.

[22] Sabniveesu Venkataraghavasivanagashashank, Kavuri Ajay, Kavi Rahul, Kulathumani Vinod, Kecojevic Vladislav, and Nimbarte Ashish. Use of wireless, ad-hoc networks for proximity warning and collision avoidance in surface mines. *International Journal of Mining, Reclamation and Environment*, 29(5):331–346, 2015.

[23] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz. Routing in sparse vehicular ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 25(8):1538–1556, Oct 2007.

[24] Dunfan Ye, Daoli Gong, and Wei Wang. Application of wireless sensor networks in environmental monitoring. In *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, volume 1, pages 205–208, Dec 2009.