Simple Feedback Driven Accuracy Based Reputition Mechanism for IoV

Rohan Dahiya, Frank Jiang, *DontKnow, IEEE,* and Robin Doss, *Dont Know, IEEE*

***Abstract*—The abstract goes here.**

***Index Terms*—IoV, IoT, Reputition System, Trust Management, VANET.**

I. Introduction

THIS demo file is intended to serve as a “starter file” for IEEE journal papers produced under LATEX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

mds
August 26, 2015

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Litrature Survey

III. Proposed Methods

We Describe our proposed system for Reputation Score generation in this section as well as propose methods of sharing the Feedback to the network and usage of the scores. We divide the section into three subsections (’Feedback Generation and Collection’, ’Score Generation’ and ’Usage’). The first subsection details how the feedback is created and propagated and collected in the network, in the second subsection methods of score generation are described and the third sub-section details the proposed method for the distribution/application of the scores generated. It is assumed that the RSUs can communicate with each other and therefore can maintain shared storage in the from of IPFS etc. Processing tasks can also be distributed among the RSUs (TODO find citations), however the details of it are beyond the scope of this article. For simplicity the network of RSUs is referred to as a single unit henceforth.

A. Feedback Generation and Collection

As mentioned earlier nodes in an IoV may be able to classify some of the information they receive as true or false either by means of self observation or context aware deduction. If a message is determined to be true or false by a receiving node, it can share feedback on it with the other nodes of the network. The usage of this feedback is detailed in subsequent subsections. In the proposed system feedback is shared primarily in the form of a report comprising of 4 elements: the reportee’s id (or address), the reporter’s id (or address), the message id (or a unique message identifier), and a boolean value used to indicate if the message being reported on was found to be true or false by the reporter.

1) Feedback Generation by Nodes: The feedback exchange in the network is triggered by the following events:

- **Message Classified:** Whenever the truthfulness of a received message is determined, a report is created and shared on the network.
- **Message received from a sender for the first time:** In order to bring new nodes joining a network up to par on the feedback on nodes with the rest of the network, whenever a node a receives a message from a node b on which it has no feedback information, it (a) sends a ”Request for Reports” message containing only b’s id.
- **Request for Reports Recieved:** When a Request for Reports on any node b is receieved by some node a, it sends a ”Reports Dump” message containing all the reports it has sent on b that it contains in storage.

2) Feedback Collection at RSU: Three lists of messages are maintained: Current Scope, Staged Scope, Archived Scope. All reports that are recieved are put into one of two baskets:

1) **Current Basket**: if the message that the report is on is not in Staged Scope or Archived Scope, the message is added to the Current Scope and the report is collected in the Current Basket.
2) **Staged Basket**: Only reports on messages that are listed in the staged scope are collected in the staged basket.

If a report is recieved on a message that is in the Archived Scope, it is ignored. Each basket stores the reports that are inserted in it and it’s implementation can include variables to maintain some metadata on the reports to facilitate the score generation process which is detailed later. At regular intervals of time, which roughly equals the average amount of time between a node sending a message a node receiving a report on that message, a ”Stage Shift” event is triggered. When Stage shift occurs the elements of the staged scope are added to archived scope. Secondary Scores and Message Truthvalues are calculated based on the contents of the Staged Basket before it is emptied, this process is elaborated in following subsections. These Truthvalues are then inserted into Logs. After this the contents of the Current Basket are shifted to the Staged Basket before it is emptied. Corresponding operations

R. Dahiya is a visiting researcher at the School of Info Technology, Deakin University, Geelong, VIC 3220, Australia, and a student at Vellore Institute of Technology, Vellore, TN 632014, India. e-mail: rohandahiya@outlook.in

F. Jiang and J. Doe are with School of Info Technology, Deakin University, Geelong, VIC 3220, Australia

Manuscript received Month 19, 2020; revised Month 26, 2020.

are performed on the Staged and Current Scope lists. The logs for each vehicle contain the calculated truthvalues of messages sent by it, and other counters that are used to calculate the Primary Scores for it with optimal complexity. It can be summarised that the Archived Scope is the list of messages for which the truthvalues have been calculated and inserted into the logs, the Staged Scope is the list of messages for which the reports are still being collected and that will be processed and inserted into logs at the next stage shift event, and the current scope is the expanding list of messages that will form the fixed message list of staged scope at the next stage shift event.

3) RSU Results Broadcasting and incorporation at Nodes: As mentioned previously, the Logs at the RSU contain the calculated truth-values of messages which are used to generate the Primary Scores, the calculation of these truth-values and Primary Scores is detailed in the subsequent subsection. In order to share it's results the RSU calculates the Primary Score of each node for various window sizes i.e [10,50,250,1250] or such and sends these (with other metadata) along with the list of blacklisted nodes in a message to all nodes in the network. Whenever a message from an RSU is received the node can make a copy of the blacklist to ignore reports from the blacklisted nodes, untill the blacklist is updated again and it can insert dummy messages/reports from various nodes into it's own storage by reverse calculating from the received primary scores and metadata in order to emulate the same knowledge.

B. Score Generation

We describe the method of score-generation in this sub-section. All data pertaining to reports and counters in this subsection and in Fig. 1-4 refers to the data collected in the staged basket up until the stage shift event occurs and the various score generation methods are called, unless otherwise specified to refer to data in Logs. In order to generate an estimate of the overall accuracy of all the information shared by a node, the system calculates a truth-value for each message sent by that node based on all the feedback on that message, then the mean of the truth-values of a set of messages is taken. This estimate of the overall accuracy of a node is referred to as the Primary Score or the Reputation Score interchangeably in this article. Since the feedback may be deliberately falsified as well, it is first necessary to exclude the feedback from nodes that may be malicious. In order to distinguish nodes with false feedback another parameter is used which is referred to as the ”Secondary Score” for a node. The following sub-sub-sections describe the methods for ultimately generating the Primary Score for each node.

Fig. 1. Calculation of Secondary Scores

Algorithm 1 Calculate Median Implied Scores

INPUT:

1) N: Set of all nodes on which at least one report was received
2) $Count^{Total}_{i,j}$: Count of Total Reports on node i by node j $\forall i, j \in N$
3) $Count^{Positive}_{i,j}$: Count of Positive Reports on node i by node j $\forall i, j \in N$

OUTPUT: $Score^{MIS}_{n}$: Median Implied Score of node n $\forall n \in N$

```
for all $i \in N$ do
    $impliedScore_j \leftarrow Count^{Positive}_{i,j} / Count^{Total}_{i,j}$ $\forall j \in N - \{i\}$
    $Score^{MIS}_{i} \leftarrow$ median($impliedScore_j$ $\forall$ $j \in N - \{i\}$)
end for
return $Score^{MIS}_{i}$ $\forall i \in N$
```

Complexity: $\mathcal{O}(n^2)$.

Algorithm 2 Calculate Secondary Scores

INPUT:

1) N: Set of all nodes from whom at-least 1 report was received
2) $Count^{Total}_{i,j}$: Count of Total Reports on node n by node j $\forall i, j \in N$
3) $Count^{Positive}_{i,j}$: Count of Positive Reports on node n by node j $\forall i, j \in N$
4) $Score^{MI}_{n}$: MI Score of node n $\forall n \in N$

OUTPUT: $Score^{Secondary}_{n}$: Secondary Score of node n $\forall n \in N$

```
for all $i \in N$ do
    $TSE \leftarrow 0$
    $TotalReports \leftarrow 0$
    for all $j \in N \mid j \neq i$ do
        $impliedScore \leftarrow Count^{Positive}_{j,i} / Count^{Total}_{j,i}$
        $error \leftarrow Score^{MI}_{j} - impliedScore$
        $TSE \leftarrow TSE + (error^2 \times Count^{Total}_{j,i})$
        $TotalReports \leftarrow TotalReports + Count^{Total}_{j,i}$
    end for
    $Score^{Secondary}_{i} \leftarrow \frac{TSE}{TotalReports}$
end for
return $Score^{Secondary}_{i}$ $\forall i \in N$
```

Complexity: $\mathcal{O}(n^2)$.

1) Secondary Score Generation: When ever a new report is received by the RSU, counters corresponding to the number of positive and negative reports on a node and the number of positive and negative reports from a node on another node can be incremented. These counters can then be utilised to quickly calculate each node's median implied reputation score (MI Score), which is defined as the median of all the implied reputation scores for that node, where an implied reputation score of a node is the fraction of positive reports to total reports by another node. See Fig. 1, Alg. 2. By assuming the MI score for each node as a benchmark, the deviation of the scores implied by a node's reports from the respective MI scores is utilised as a measure of the utility of it's reports. This deviation is measured in terms of Mean Squared Deviation per report and is termed as the Secondary Score of the Node. It

Fig. 2. Generation of Blacklist

Algorithm 3 Generate Blacklist

INPUT:

1) N: Set of all nodes for which a secondary score exists
2) $Score_n^{Secondary}$: Secondary Score of node $n \ \forall \ n \in N$

OUTPUT: $Blacklist$: Set of all blacklisted nodes

```
$m \leftarrow$ median($\{Score_i^{Secondary} \ | i \in N\}$)
$ADm \leftarrow \emptyset$
for all $i \ \in \ N$ do
    $ADm \leftarrow ADm \cup \{|Score_i^{Secondary} - m|\}$
end for
$MADm \leftarrow$ median($ADm$)
$threshold \leftarrow m + 2 \times MADm$
$Blacklist \leftarrow \emptyset$
for all $i \ \in \ N$ do
    if $Score_i^{Secondary} > threshold$ then
        $Blacklist \leftarrow Blacklist \cup \{i\}$
    end if
end for
return $Blacklist$
```

Complexity: $\mathcal{O}(n)$.

Fig. 3. RAW Score Calculation

Algorithm 4 Calculate RAW Scores

INPUT:

1) N: Set of all nodes on whom atleast 1 report exists
2) $Count_i^{Total}$: Count of Total Reports on node $i \ \forall i \in N$ in Staged Basket as well as Logs.
3) $Count_i^{Positive}$: Count of Positive Reports on node $i \ \forall i \in N$ in Staged Basket as well as Logs.

OUTPUT: $Score_n^{RAW}$: RAW Score of node $n \ \forall n \ \in N$

```
for all $i \ \in \ N$ do
    $Score_i^{RAW} \leftarrow (Count_i^{Positive}/Count_i^{Total})$
end for
return $Score_i^{RAW} \ \forall i \in N$
```

Complexity: $\mathcal{O}(n)$.

is calculated for each node as described in Fig. 1 Alg. 2.

2) Blacklist Generation: The Secondary Scores are a measure of abnormality on the reports of a node. This measure may be used to scale down the effect of it's reports during the Primary Score generation, or more simply it may be used to blacklist nodes who's Secondary Score crosses a certain threshold. The latter approach was pursued by us during experimentation where a simple heuristic method of determining a dynamic threshold based on the distribution of the secondary scores was adopted. With the assumption of honest majority the median of secondary scores must not be more than that of a dishonest node with abnormally lower utility of reports. Therefore the threshold value was defined as a function of the median of all Secondary Scores and the median absolute deviation around that median. The process of the generation of the blacklist is as described in Fig. 2 Alg. 3

3) Primary Score Generation: The counters utilised during secondary score calculation may also be used to calculate estimates of accuracy for each node in constant time as described in Fig. 3 - Alg. 4. This estimation is referred to as RAW Score and serves as a baseline during experiments, in itself the RAW score is highly inaccurate however it is quick to calculate.

With all the reports on each message indicating it to be either true or false, the mean of these reports can be calculated and treated as a fuzzy truth-value of it. As mentioned earlier the primary score is an estimation of the accuracy of all the information shared by a node, this can be trivially calculated as the mean of the calculated truth-value of a set of messages by that node. During this calculation the reports by blacklisted nodes are ignored. The process of calculating these truth-values is as described in Fig. 4 - Alg. 5. These calculated truth-values for messages are then inserted into the logs. From these logs we generate the Primary Score which is the mean of the truth-values of a set of messages. Calculating the Primary Score over the all the messages would give a close estimate of a nodes actual accuracy assuming it remains consistent however it is not likely to reflect a change in the accuracy of a node quickly. On the other hand the Primary Score of a node over a small window of message truth-values is likely to fluctuate more on every stage shift and not be an accurate estimate of the behaviour of a node however it would reflect a sudden change in th accuracy of a node quickly. The process of generating Primary Scores over a given window size for all nodes is described in Fig.4 - Alg.6

C. Usage

Trends in the Primary Scores of each nodes, over different window sizes, can be tracked to detect malicious behaviour such as perpetual inconsistency, drastic changes etc. besides the trivial case of being low. Likewise Secondary Scores can be monitored as well to identify and remove nodes from the network. In the process of calculating the MI Scores prior to Secondary Score Calculation (See Fig. 1 Alg. 1) the distribution of the implied scores can be analysed by using statistical methods of multimodality detection such as TODO add DIPTest cite, where a high degree of bimodality can be an indicator of collusion to manipulate the reputation system. These options of deeper analysis have not been further explored in this article however have been mentioned here as an indicator of possible extension of the methods. In the scope of this article the calculated truth values are used to calculate the Primary Scores at various sizes and share them with the Nodes where there utilisaton can be done in a number of alternative ways.

1) Usage at RSU: As previously mentioned in Section III-A, at the RSU the Primary Scores are calculated at certain

Fig. 4. Primary Score Calculation

Algorithm 5 Calculate Truth-values

INPUT:

1) N: Set of all nodes
2) M_i: Set of all messages (ids) from node $i \ \forall \ i \ \in \ N$
3) $Report_j^{i,m} \ \in \ \{0,1\}$: Alleged Validity of message m by node i in a Report by node $j \quad \forall m \ \in \ M_i | Report_j^{i,m}$exists $\forall \ i, j \ \in \ N | i \neq \ j$
4) B: Set of all blacklisted nodes

OUTPUT:

1) M'_i: Set of all messages (ids) from node i for which a truth-value is calculated $\ \forall \ i \ \in \ N$
2) tv_m^n: Truth-value of message m from node n $\forall m \in M'_n \ \forall n \in N$

```
for all $n \in N$ do
    $M'_n \leftarrow \emptyset$
    for all $m \in M_n$ do
        $R \leftarrow \{Report_j^{n,m} | \ j \in N{-}B$ and $Report_j^{n,m}$exists$\}$
        if $|R| \neq 0$ then
            $tv_m^n \leftarrow \frac{\sum_{r \in R} r}{|R|}$
            $M'_n \leftarrow M'_n \cup m$
        end if
    end for
end for
return $\ tv_m^n \forall m \in M'_n \ \forall n \in N$
```

Complexity: $\mathcal{O}(n^2 m)$.

Algorithm 6 Calculate Primary Scores

INPUT:

1) N: Set of all nodes in Logs
2) M_i: List of all messages (ids) in Logs from node i in reverse order of arrival $\ \forall \ i \ \in \ N$
3) tv_m^n: Truth-value of message m from node n $\forall m \in M_n \ \forall n \in N$
4) w: Size of the window for which average truth-value needs to be calculated

OUTPUT: $Score_n^{Primary,w}$: Primary Score of Node n calculated over a window of size w $\forall n \ \in \ N$

```
for all $n \in N$ do
    $M' \leftarrow M_n[1$ to $w]$
    $sum \leftarrow \sum_{m \in M'} tv_m^n$
    $Score_n^{Primary,w} \leftarrow sum/w$
end for
return $\ Score_n^{Primary,w} \ \forall n \in N$
```

Complexity: $\mathcal{O}(nm)$.

Complexity *for optimized implementation of logs*: $\mathcal{O}(n)$.

window sizes for all nodes and this data along with some meta data such as the id or timestamp of the last message considered from each node is sent along with the blacklist as a message to the network. At the nodes this data of primary scores along with the metadata is used to create dummy message truth-values or reports (depending on the usage) to emulate the data that would produce these Primary Scores, This is done to reduce the amount of information that needs to be sent over the network.

2) Usage at Nodes - Disabled Node: The nodes can depend on the RSU entirely to provide the Primary Scores. Pessimistically minimum of the primary scores of a node (over different window sizes) can be taken.

Advantages: Low Computation at nodes

Disadvantages:

- Less adaptable to change in behaviour of nodes, scores remain static between RSU Stage Shifts
- System is Entirely dependant on RSU

3) Usage at Nodes - Adapted RAW Score Algorithm: Reports can be used by nodes to calculate Reputation Scores by using an adaptation of the RAW Score Algorithm at the RSU (See Alg. 4) where the average of the last w' reports on a node is taken as a Primary score over a window size of w'. When a RSU Broadcast is received, for each window size w for which a primary score exists on nodes, w messages with truth-value $tv = Score^{Primary,w}$ are emulated. For each message to be emulated x number of dummy reports, whose average ≈ desired truth-value, are inserted into the logs, where x = average number of reports per message (this can be included as part of the metadata in the RSU broadcast). The window sizes to be used when calculating Primary Scores locally can be calculated by scaling the window sizes used at the RSU by a factor of x.

Advantages:

- More adaptable to change in behaviour of nodes, scores are dynamic between RSU Stage Shifts
- Moderate Computation at nodes

Disadvantages:

- System is Largely dependant on RSU, Estimates will become highly inaccurate over time without RSU updates

4) Usage at Nodes - Adapted Primary Score Algorithm: Reports can be used by nodes to calculate Reputation Scores by using an adaptation of the Primary Score Algorithm at the RSU (See Alg. 5 and 6) where the average of the truth-values of the last w messages on a node is taken as a Primary score over a window size of w. Messages with less than a certain number of reports on it can be kept in a buffer while more reports are received. Whenever a new report on a message is received it's truth-value is accordingly updated and so is the Primary Score over windows what include the message. When a RSU Broadcast is received, for each window size w for which a primary score exists on nodes, w dummy messages with truth-value $tv = Score^{Primary,w}$ are inserted. Reports from nodes that are blacklisted are purged and so are the reports on messages that have been evaluated by the RSU for Score Genaration. This data can be provided in the metadata of the RSU Broadcast

Advantages:

- More adaptable to change in behaviour of nodes than disabled node, scores are dynamic between RSU Stage Shifts
- System is Less dependant on RSU, Estimates will remain accurate to a high degree over time without RSU updates in the absence of malicious nodes that share false feedback.

Disadvantages:

- System would be adversely affected in the absence of RSU if malicious nodes send false feedback.
- Higher amount of computation and storage at node

IV. Experiments

The system was tested in software simulation in various scenarions. Details on the Simulation Set-up and parameters can be found in Appendix A. Three different cases in two separate environments were simulated to give a total of six scenarios. The Environments were as follows:

A. Environments

1) City: klorem

2) Highway: klorem

B. Situations

1) Situation 0: klorem

2) Situation 1: klorem

3) Situation 2: klorem

C. Scenarios

- Scenario 0: Situation 0 in City
- Scenario 1: Situation 1 in City
- Scenario 2: Situation 2 in City
- Scenario 10: Situation 0 on Highway
- Scenario 11: Situation 1 on Highway
- Scenario 12: Situation 2 on Highway

V. Results

VI. Conclusion

T

Appendix A
Simulation Set-Up

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

PLACE
PHOTO
HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.