A Functional Complexity Framework for Dynamic Resource Allocation in VANETs

Kunal Pattanayak*, Aritra Chatterjee*, Merim Dzaferagic**, Suvera Sekhar Das* and Nicola Marchetti**

*G. S. Sanyal School of Telecommunications, Indian Institute of Technology, Kharagpur, India
**CONNECT, Trinity College Dublin, Ireland

Abstract—In this work we present a complex systems science analysis of the Self Organised Time Division Multiple Access (SOTDMA) algorithm. We translate the interaction among member nodes into a functional topology graph in order to measure the effect of each individual node’s adaptability on the global performance. The functional complexity metric corresponding to the functional topology is shown to have substantial correlation with important Key Performance Indicators (KPIs), namely probability of collision and probability of correct packet detection. We further use the functional complexity metric to analyze the trade-off between the two aforementioned KPIs in terms of system parameters. We finally show that the results obtained using this approach satisfy the predefined KPI constraints imposed on the algorithm and thus is successful in capturing the system behaviour.

Index Terms—Complex systems science, functional complexity, VANET, IoT, SOTDMA, probability of collision, probability of packet detection.

I. INTRODUCTION

The blending of wireless devices into the fabric of daily lives of people over the past decade has made wireless communication an indispensable component in today’s living world. The continual success of emerging wireless technologies, coupled with ever increasing demands of higher data rates, better connectivity and mobility by users worldwide, have led to a spike in the usage of wireless devices and services. These factors have led to the revolution of fifth generation wireless systems (5G) [1] which are expected to support much higher data rates, meet the demands of connectivity and mobility, and facilitate the growth of frontiers like Internet of Things (IoT) [2], Machine-to-Machine communication (M2M) [3], Cyber Physical Systems (CPS) [4] etc. These new technology drivers have resulted in the development of many useful applications like Connected Vehicles (e.g. Unmanned Aerial Vehicles) [5], Traffic Safety, Smart Cities [3], Autonomous Driving [6] etc. These applications involve a very large number of densely deployed devices and are expected to use distributed MAC which should enable self-organisation within the network [7–9]. In such scenarios, one of the major problems encountered is the evaluation of key performance indicators (KPIs) which do not possess a closed form expression. They can thus be evaluated only through simulation [10–12] and this tends to become cumbersome with increasing node population due to high computational overhead. As a result, there is a need for a more scalable, reliable and efficient measure for such KPIs.

In this work, we present a method for analysing one such application protocol related to vehicular traffic, namely Self Organised Time Division Multiple Access (SOTDMA) [13]. Although the SOTDMA algorithm has traditionally found its use in applications pertaining to the maritime and avionics industry [14–17], it has been analysed in a vehicular environment as well, alongside existing channel access techniques like CSMA and CSMA/CA in [10]. Owing to comparatively much lower channel delay leading to a decrease in packet drops, SOTDMA was concluded to be a potential candidate for use in over-utilised environments such as dense vehicular traffic scenarios [18].

In this work, we adopt a Complex Systems Science (CSS) based approach in analysing the SOTDMA algorithm. CSS provides us with a measure of cooperative intelligence in a system of individual entities to achieve a common objective. CSS studies the local interactions between sub-parts of a system and the effect they have on the system’s global behaviour, thus making it a potential tool for analysis of distributed systems. According to CSS, a group of unrelated nodes results in zero complexity [19]. Counterintuitively, a completely connected set of nodes (e.g. a network dependent on full connection of nodes) also results in zero complexity [19]. There are several complexity metrics for quantifying the complexity of a system, such as Mutual Information [19], Excess Entropy [20], Statistical Complexity [21] etc. A compilation of such complexity metrics can be found in [22]. We use the complexity metric based on the progression of the works [19, 20, 23], namely Functional Complexity. In this work, we analyze the interaction between nodes which use the SOTDMA algorithm for potential future vehicular communication through the lens of CSS.

In SOTDMA, the individual nodes listen to transmissions from their neighbours and individually decide the transmission slot they could occupy within the SOTDMA frame, to achieve the global objective of minimising packet collisions. Some of the measurable KPIs (objectives) of such networks as mentioned in [11] are Probability of Collision and Probability of Correct Packet Detection. As highlighted in [11], KPIs that cannot be expressed as closed form functions of system parameters require extensive simulations for the performance evaluation. Even though authors of [24, 25] have obtained closed form expressions of probability
of collision, the expressions are subject to strict assumptions that restrict the scope of their application. We predict, using a CSS approach, the variation of aforementioned KPIs using functional complexity. We also propose a method to comprehend the tradeoff between probability of collision and probability of correct packet detection, using functional complexity.

II. OVERVIEW OF THE SOTDMA ALGORITHM

In SOTDMA, each node gathers information about the surrounding nodes in order to decide which transmission slot it could occupy in the SOTDMA frame, to achieve the global objective of minimising concurrent packet transmissions in the system. The nodes in SOTDMA “listen” to the channel during one frame and then select free slots for transmitting their data. If there are no free slots, a node chooses to transmit in an occupied slot used by the farthest node from itself. When a node is switched on in the network, it goes through four different phases, as follows: (1) Initialization phase, (2) Network Entry phase, (3) First Frame phase and (4) Continuous Operation phase. The SOTDMA algorithm relies on GPS for timestamp synchronisation. During the Initialization phase, the node listens to its active neighbours’ broadcasts for one full frame. In the Network Entry phase, the new node prepares for its first transmission, by selecting the Nominal Start Slot (NSS) in the SOTDMA frame. In the First Frame phase, the node begins transmitting in the frame as decided upon during its previous transmission. The end of the first frame marks the start of the Continuous Operation phase where the node transmits periodically over the slots decided in the First Frame phase, while decrementing the slot timeout value every time it transmits on the pre-assigned slot. When the slot timeout reaches the value zero, from the candidate slots around the current transmission slot (Selection interval (SI)) available to that node, a new non-colliding slot is selected for subsequent transmissions. In the case where such a slot is unavailable, the new transmission slot becomes the slot occupied by the node farthest from the current node. The details of the algorithm can be found in [17].

III. SYSTEM MODEL

The typical SOTDMA vehicular network scenario comprises mobile nodes scattered in a 2-D or 3-D space, with each node having its own maximum power for transmission. Each node typically has a circle (for 2-D) or sphere (for 3-D) respectively, over which its broadcast can be picked up by its neighbours. This region also dictates where a surrounding node must be present for its broadcast to be received. As input, the SOTDMA algorithm requires the distance between each pair of nodes in the network, and a threshold distance for each node within which the neighbouring nodes must be present for their broadcasts to be received. We assume that this distance threshold is the same for all nodes. We represent the “reachability” of the nodes by a distance matrix $D$, with each element $d_{ij}$ denoting the distance between node $i$ and node $j$. In this work, we study a 1-D system, consisting of an array of equally spaced static nodes, as depicted in Fig. 1a.

Based on the 1-D system’s description above, we model the physical topology of the SOTDMA network under analysis as a function of the system parameters that are described as follows. The number of nodes present in the SOTDMA network is denoted by $N$, with the nodes numbered from the left as 1 to $N$. The window length of a node is denoted by $W$, which is a measure of the threshold distance of a node. It is equal to the number of nodes present on either side of a node whose broadcast messages can be received by the present node. $W$ for a node can also be seen as a measure of its transmit power. The window length, $W$, is varied from 1 node (minimal connectivity in network) to $(n-1)$ nodes (complete connectivity in network).

In the SOTDMA algorithm, if a node is not transmitting at a particular instant of time, then it is in the receiving mode (half-duplex configuration), “listening” to its neighbors’ transmissions. If a node is receiving message packets from multiple sources, we assume that the intended message is from the node nearest to it while the rest are interfering messages. We further assume that a node will be able to detect the intended message properly only if there are no interfering nodes transmitting during that particular time slot. The relevant KPIs of the SOTDMA algorithm
are defined as follows:

1) **Probability of Collision** \( (P_c) \): The probability that a time slot having non-zero number of nodes transmitting on that slot, has more than 1 node transmitting on it.

2) **Probability of Correct Packet Detection** \( (P_d) \): Probability that a packet intended for a node is properly detected i.e., given that a node receives at least one transmission, the probability that it has only one transmitting node within its hearing range during a single time slot.

IV. FUNCTIONAL FRAMEWORK FOR COMPLEXITY ANALYSIS

In this section, we map the SOTDMA algorithm into a functional topology and define the functional complexity metric [23].

A. **Functional Topology of SOTDMA**

The functional topology is a graph theoretical representation of any algorithm, with the “functional” units of the algorithm as the vertices in the graph, and the edges representing a measure of interaction between the functional units. For SOTDMA, the corresponding functional topology be defined by a graph \( G_{\text{func}} \), with vertices denoted by the set \( V_{\text{func}} \) and edges denoted by the set \( E_{\text{func}} \). Each element in \( V_{\text{func}} \) denotes a functional unit of the SOTDMA algorithm. We propose each physical node in the SOTDMA network to represent a functional unit, as each node contributes individually to the overall execution of SOTDMA by mutual interaction. Hence, \( |V_{\text{func}}| \) equals the node size \( (N) \). For each element \( d_{ij} \in D \) (as defined in Sec. III) satisfying the condition \( d_{ij} \leq W \), (i.e. the pair of nodes \( i \) and \( j \) lies within each other’s hearing range), there exists a corresponding edge in the set \( E_{\text{func}} \), which is denoted by the unordered pair \((i, j)\) having edge weight unity.

B. **Definition of Functional Complexity** \( (C_F) \)

In a graph, the scale size is defined as the maximum number of hops a vertex is allowed to take to reach another vertex. A graph is said to be connected if for every pair of vertices belonging to the graph, there exists at least one path joining them. For such a connected graph with number of vertices \( N_g \) and scale size \( r \), we define a Bernoulli random variable \( x_n \) for each vertex \( n \) having the following probability mass function:

\[
P_r(x_n = 1) = \frac{N_{\text{reach},n}}{N_g},
\]

\[
P_r(x_n = 0) = \frac{N_g - N_{\text{reach},n}}{N_g},
\]

where \( N_{\text{reach},n} \) is the number of nodes \( n \) can reach in the graph including itself, in at most \( r \) hops. The entropy of the random variable \( x_n \) can be defined as:

\[
H_r(x_n) = -\sum_{i=0}^{1} P_r(x_n = i) \log_2(P_r(x_n = i))
\]

Let \( \Lambda^j \) be a connected subgraph of \( G_{\text{func}} \) having \( j \) nodes. For a scale size \( r \), the integration of \( \Lambda^j \) is defined as:

\[
I_r(\Lambda^j) = \sum_{n \in \Lambda^j_r} H_r(x_n),
\]

The average integration over all possible connected subgraphs of size \( j \) in \( G_{\text{func}} \) can be expressed as:

\[
\langle I_r(\Lambda^j) \rangle = \frac{1}{K} \sum_{k=1}^{K} I_r(\Lambda^j_k),
\]

where \( K \) is the total number of connected subgraphs \( (\Lambda^j_k) \) of size \( j \) in \( G_{\text{func}} \). The functional complexity \( (C_F) \) for \( G_{\text{func}} \) is expressed as follows (as defined in [26]):

\[
C_F = \frac{1}{R - 1} \sum_{r=1}^{R-1} \sum_{j=1+r}^{N} \|I_r(\Lambda^j)\| - \left( \frac{r + 1 - j}{r + 1 - N} \right) I_r(\Lambda^N),
\]

where the maximum scale size of \( G_{\text{func}} \) is denoted by \( R \). Fig. 2 depicts the variation of the functional complexity \( (C_F) \) of the functional topology corresponding to SOTDMA obtained by varying the node size \( (N) \) and window length \( (W) \).

**TABLE I. SIMULATION PARAMETERS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node size ( (N) )</td>
<td>10 to 15</td>
</tr>
<tr>
<td>Window length ((W))</td>
<td>1 to ((N-1))</td>
</tr>
<tr>
<td>Average turning on rate of nodes</td>
<td>0.5s^-1</td>
</tr>
<tr>
<td>SOTDMA frame duration</td>
<td>1 s</td>
</tr>
<tr>
<td>Report rate (RR) of nodes</td>
<td>Once every 15 slots</td>
</tr>
<tr>
<td>Nominal increment (NI)</td>
<td>20</td>
</tr>
<tr>
<td>Selection interval (SI) length</td>
<td>0.2NI + 1 = 5</td>
</tr>
<tr>
<td>Nominal Start Slot Vector (for node 1 to node 15)</td>
<td>[11, 15, 9, 2, 15, 7, 18, 13, 8, 3, 5, 10, 12, 4]</td>
</tr>
</tbody>
</table>
combination of $N$ to $15$. The window length ($W$) is measured for varied $N$ and $W$. The system parameters used in the simulation are listed in Table I. For computational convenience, we calculate the KPIs over a pre-defined set of Nominal Start Slots (NSS) for each node. The KPI values were then evaluated after the algorithm was simulated for 1000 SOTDMA frames. Once a node is turned on, it will remain on till the end of simulation.

In this work, we simulate the time slot allocation of the SOTDMA algorithm by varying the number of nodes ($N$) from 10 to 15. The window length ($W$) varies from 1 to ($N - 1$). We measure the mentioned KPIs, namely $P_c$ and $P_d$, for each combination of $N$ and $W$. The system parameters used in the simulation are listed in Table I. For computational convenience, we calculate the KPIs over a pre-defined set of Nominal Start Slots (NSS) for each node. The KPI values were then evaluated after the algorithm was simulated for 1000 SOTDMA frames. Once a node is turned on, it will remain on till the end of simulation.

The probability of collision ($P_c$) of a node for varied $N$ and $W$ is shown in Fig. 3. The observed trend of $P_c$ can be justified as follows. For fixed $N$, with increase in $W$, the probability of collision is expected to decrease, since each node can listen to more and more nodes in its vicinity, thereby decreasing the chance of a collision by taking a better decision about the slot it should transmit on. Increasing the number of nodes ($N$) (and keeping $W$ constant) increases the probability of collision due to a larger number of slots occupied by contending nodes, around each node’s transmission slot, which in turn forces a node to transmit on an already occupied time slot.

In Fig. 4, the probability of correct packet detection ($P_d$) has been shown for varied $N$ and $W$. For a fixed $N$, increase in $W$ results in increase in the number of interferers due to increase in the transmit power and hearing range of each node. That results in decremental trend of $P_d$. Similarly, increasing $N$ for a fixed $W$ results in an increase in average number of nodes neighboring each node, which also decreases $P_d$. For instance, for a fixed $W = 2$, increase in $N$ from 10 to 15 results in increase in the average number of neighbours surrounding a node from 3.4 to 3.6.

In order to quantify the similarity in their variation resulting from varying $W$ while keeping $N$ constant. In order to calculate $\rho$, we construct two vectors, $V_{CF}$ and $V_{KPI}$ which denote the functional complexity vector and

$$\rho = \frac{E[C_F,KPI] - E[C_F]E[KPI]}{\sqrt{(E[C_F^2] - E^2[C_F])(E[KPI^2] - E^2[KPI])}} = \frac{\frac{1}{N-1} < V_{CF},V_{KPI} > - \frac{1}{N-1} < V_{CF},V_1 > < V_{CF},V_1^N >}{(\frac{1}{N-1} < V_{CF},V_{KPI} > - \frac{1}{N-1} < V_{CF},V_1^N >^2)(\frac{1}{N-1} < V_{KPI},V_{KPI} > - \frac{1}{N-1} < V_{KPI},V_1^N >^2)}$$

(6)

V. SIMULATION AND RESULTS

![Fig. 3. Probability of Collision ($P_c$) for the physical topology depicted in Fig. 1a.](image)

![Fig. 4. Probability of Correct Packet Detection ($P_d$) for the physical topology depicted in Fig. 1a.](image)

![Fig. 5. Correlation of each of the KPIs with the functional complexity for different node sizes](image)
KPI vector respectively. They are formed by taking the functional complexity and KPI value repectively for each $W$ from 1 to $N-1$ for a fixed $N$. Clearly, the number of elements in both $V_{CF}$ and $V_{KPI}$ equals to $N-1$. The Pearson correlation coefficient ($\rho$) between these two vectors is defined in (6), where $V^N_i$ denotes a vector of length $N-1$ with each element equal to 1. It is shown in Fig. 5 that the complexity metric exhibits moderate to high correlation with both the KPIs for the considered range of $N$ (10 to 15). This observation leads to the motivation of designing a framework which uses $C_F$ as a controlling variable, in order to control the system parameters while satisfying certain constraints on the KPIs, which shall be elaborated upon in the next section.

VI. COMPLEXITY AS A BRIDGE BETWEEN SYSTEM PARAMETERS AND KPIs

In this section we illustrate the application of functional complexity as an indirect measure in order to achieve an efficient trade-off between two important KPIs, namely $P_c$ and $P_d$. For example, let us consider that for node size ($N$) equal to 15, suitable window length range is to be identified which satisfies the following two constraints simultaneously:

$$P_c \leq \alpha,$$
$$P_d \geq \beta.$$  

(7a)  

(7b)

Let $\alpha_1$ and $\beta_1$ denote the functional complexities corresponding to a Probability of Collision equal to $\alpha$ and a Probability of Correct Packet Detection equal to $\beta$ respectively. From Fig. 5, we see that there is high correlation ($\rho$) observed between the KPI and $C_F$ at higher values of $N$. From Fig. 2, we see that for a fixed value of $N$, $C_F$ varies almost monotonically with $W$. Based on these observations, we claim that $C_F$ lies below $\alpha_1$ for almost all values of $W$ for which $P_c$ lies below $\alpha$. Similarly, $C_F$ lies above $\beta_1$ for almost all values of $W$ which result in $P_d$ greater than $\beta$. Thus, the inequalities in (7) result in:

$$\beta_1 \leq C_F \leq \alpha_1.$$  

(8)

Our claim implies that through the mapping between functional complexity and KPIs, one can estimate the system parameter (in our case, the Window Length ($W$)) using the $C_F$ for the corresponding KPI. Through this exercise, one can infer that the functional complexity acts like an intermediary between the system parameters and KPIs. For example, taking a value of $\alpha = 0.2778$ and $\beta = 0.807$ yields $\alpha_1$ and $\beta_1$ equal to 5.05 and 1.09 respectively, as depicted in Fig. 6. The values of $W$ which satisfy Eq. (7) are $\{5, 6, 7, 8\}$. As per our claim, the values of $W$ which satisfy Eq. (8) are $\{5, 6, 7, 8\}$ as shown in Fig. 7. Here we see there is a $100\%$ match between the actual values satisfying the constraints and the values obtained from our claim involving functional complexity.

In Fig. 8 we compare the time taken for evaluating the KPIs (through extensive simulation) as well as the complexity metric.
for the described SOTDMA algorithm in a DELL® Precision Tower 3620 Desktop PC with 4 GB RAM and Intel® Core i5-4570, 3.2 GHz processor running Matlab® on Ubuntu 14.04 LTS-64 bit with kernel 3.13.0-70. The evaluation times were calculated using tic and toc functions of Matlab®. It can be observed that the evaluation of functional complexity takes significantly less time compared to the evaluation of KPIs, mainly due to the absence of any closed form expressions of $P_C$ and $P_D$ for SOTDMA. For example, it is noted that for node size = 15, evaluation of KPIs took approx. 400 sec, whereas the evaluation of $C_F$ took approx. 109 sec which is approximately 2.4 times faster. Thus, the framework involving complexity metric presented in this work presents a numerically efficient alternative to capture the variation in KPIs upon variation of system parameters.

However, as can be seen from Fig. 2, this method works well for those KPI values whose corresponding W value is greater than 3, due to the fact that the variation of complexity metric around $W = 3$ is not monotonic in nature. Thus, for KPIs whose corresponding W is less than 3, the proposed method shall give rise to outliers. Or in other words, the resultant set of supportable system parameters obtained from Eq. (7) and Eq. (8) may not be fully matched in such a situation. Development of better application-specific complexity metrics can be seen as a potential future work to alleviate the issue.

VII. CONCLUSION

In this work, we presented a CSS based analysis of SOTDMA algorithm for capturing the effects of interaction between involved nodes. We obtained high values of correlation between the relevant KPIs (Probability of Correct Packet Detection and Probability of Collision) of SOTDMA and functional complexity of the functional topology. We observe that evaluation of the KPIs corresponding to different system parameters involves time consuming simulation. In this work, we exploit the correlation between the KPIs and the functional complexity so as to estimate the KPIs with the help of functional complexity, thus capitalising on the substantially shorter time it takes to compute the functional complexity. We see that increasing complexity improves one KPI ($P_D$) while worsening the other ($P_C$). We then show that a tradeoff between the two KPIs can be achieved using the functional complexity as a controlling metric. As a result, we are able to see that the functional complexity metric is able to capture the system behaviour when subject to changes in system parameters, which may prove to be useful in scenarios involving a very large number of nodes, where simulation based analysis is not feasible.

REFERENCES