Transcriptional Regulatory Network Topology with Applications to Bio-inspired Networking: A Survey

SATYAKI ROY, Missouri University of Science and Technology, USA
PREETAM GHOSH, Virginia Commonwealth University, USA
NIRNAY GHOSH, Indian Institute of Engineering Science and Technology, (IIEST), India
SAJAL K. DAS, Missouri University of Science and Technology, USA

The advent of the edge computing network paradigm places the computational and storage resources away from the data centers and closer to the edge of the network largely comprising the heterogeneous IoT devices collecting huge volumes of data. This paradigm has led to considerable improvement in network latency and bandwidth usage over the traditional cloud-centric paradigm. However, the next generation networks continue to be stymied by their inability to achieve adaptive, energy-efficient, timely data transfer in a dynamic and failure-prone environment—the very optimization challenges that are dealt with by biological networks as a consequence of millions of years of evolution. The transcriptional regulatory network (TRN) is a biological network whose innate topological robustness is a function of its underlying graph topology. In this article, we survey these properties of TRN and the metrics derived therefrom that lend themselves to the design of smart networking protocols and architectures. We then review a body of literature on bio-inspired networking solutions that leverage the stated properties of TRN. Finally, we present a vision for specific aspects of TRNs that may inspire future research directions in the fields of large-scale social and communication networks.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

Additional Key Words and Phrases: Robustness, motifs, gene interaction, energy efficiency, IoT

ACM Reference format:
Satyaki Roy, Preetam Ghosh, Nirnay Ghosh, and Sajal K. Das. 2021. Transcriptional Regulatory Network Topology with Applications to Bio-inspired Networking: A Survey. *ACM Comput. Surv.* 54, 8, Article 166 (October 2021), 36 pages.
https://doi.org//10.1145/3468266

1 INTRODUCTION

The present-day network architectures facilitate the communication among millions of users possessing heterogeneous smart devices. Such devices are equipped with sensing, computing, and

The work is partially supported by NSF grants OAC-1725755, OAC-2104078, CBET-1802588, and CBET-1609642.
Author's addresses: S. Roy, University of North Carolina, Chapel Hill, USA, 306 Estes Dr. Ext., Apt. J6, Carrboro, NC 27510; email: satyakir@unc.edu; P. Ghosh, Virginia Commonwealth University, Richmond, USA, College of Engineering, 601 West Main Street, Box 843068, Richmond, Virginia 23284-3068; email: pghosh@vcu.edu; N. Ghosh, Indian Institute of Engineering Science and Technology, Shibpur, India, Botanic Garden, Dist: Howrah, West Bengal, India - 711103; email: nirnay@cs.iiests.ac.in; S. K. Das, Missouri University of Science and Technology, USA, 500 W. 15th St. 325 Computer Science, Rolla, MO 65409; email: sdas@mst.edu.

0360-0300/2021/10-ART166 $15.00
https://doi.org//10.1145/3468266

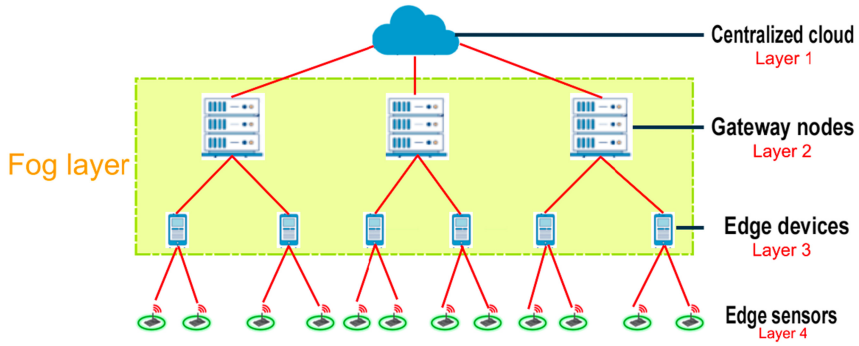


Fig. 1. Modern edge computing network paradigm.

actuating capabilities and employ a plethora of communication protocols, viz., Wi-Fi, **Bluetooth Low Energy (BLE)**, Zigbee, 3G/4G/5G LTE, and so on, to carry out data exchange over the Internet [1]. This network ecosystem, popularly termed the **Internet of Things (IoT)**, entails a high rate of message exchanges (which are both delay-intensive as well as privacy-invasive) between the smart devices and back-end servers, as elucidated by the following use-case from the domain of smart healthcare.

Consider a scenario in smart healthcare where a patient is equipped with smart device(s) monitoring his/her physiological parameters (e.g., temperature, blood pressure, glucose level, heart rate, etc.). Typically, the following generic steps are employed to push the patient data into the back-end cloud server and retrieve them on request from authorized users: (1) the caregiver uses a smartphone-based app. to control the monitoring device, (2) the app. uses Zigbee or BLE to send the request to a *gateway* device, which is sent to the cloud using Wi-Fi, (3) the remote security server hosted in the cloud authenticates the requester and the legitimacy of the request, (4) the server sends the response to the gateway device, which is forwarded to the device, and (5) the device actuates and sends back the response to the requester via the gateway. Evidently, such a multi-stage communication protocol, involving the edge sensors (i.e., healthcare IoT devices), edge devices (i.e., smartphone), gateway, and the cloud, is subject to delay as well as privacy breach.

To address such challenges, we see a shift from centralized cloud computing to a new paradigm of *edge computing*, which attempts to improve the task response time as well security by bringing data processing closer to the edge of the network [2]. Specifically, a new layer, called *fog layer*, has been added to connect the *centralized cloud* to *edge sensors*. The fog layer is an abstract layer that may not necessarily demand physical resources and can be loosely-coupled between the *edge* devices and *gateway* nodes (Figure 1). In practice, the capabilities of network devices, viz., access points, Wi-Fi routers, and so on, are enhanced so that they can host the fog application. The fog application maintains a snapshot of the global data (hosted in the cloud) and is capable of location-aware and low-latency computing. It is also equipped with data aggregation functions to reduce data replication. With respect to the smart healthcare example, as the patient's recent data are stored in the edge of the hospital network, the likelihood of security or privacy intrusion reduces considerably.

Let us consider how different types of wireless networks fit into the four-layered edge network paradigm. There are *wireless sensor networks* that transfer environmental data collected by sensors to the base station [3], while *disaster response networks* perform opportunistic event sensing and reporting in the aftermath of natural disaster [4]. **IoT networks (IoT-Nets)** sense and report data in urban spaces to inform decision-making in smart city settings [5]. There are *edge centric*

IoT networks that allow individuals in possession of mobile devices to collect and report events to the cloud platform via intermediate fog nodes [6]. It is interesting to note that the stated networks involve different layers of the modern network paradigm. Specifically, **wireless sensor networks (WSNs)** operate on layer 1, **disaster response network (DRNs)** involve mobile sensors (layer 1) and the **coordination center (CC)** (layer 3), and IoT-Nets entail communication between heterogeneous IoT sensors (layer 1) and the cloud (layer 4); finally, the edge computing platforms work with all the four layers by combining edge sensors, fog layer and the cloud. There are quite a few challenges that impede the design of the next generation networks [7].

(1) There is the *dynamic nature* of mobile ad hoc networks and cognitive radio networks w.r.t. connectivity, traffic and bandwidth.
(2) The DRN and edge-centric IoT network need to be *infrastructure-less and autonomous* to achieve high scalability.
(3) The communication networks need to provide seamless service despite *component failures* due to energy depletion or hardware faults.
(4) Network entities need to exhibit *self-organization* and *act collaboratively* to achieve complex system goals.
(5) The communication networks constitute resource-constrained sensing devices and must necessarily incorporate *energy efficiency* to continue communication over long periods [8].

One answer to such challenges lies with a field of computing inspired by nature—*bio-inspired computing*—based on the idea that the biological systems are characterized by certain key properties that are a direct consequence of evolution. They adapt to the changing environment, exhibit high resilience to failures and attacks, and collaboratively accomplish complex tasks, while making minimum use of available resources. Swarm intelligence-based algorithms [7], artificial immune systems [9], physarum-based transport network algorithms [10] are well-studied examples of bio-inspired computing. There is a special class of bio-inspired computing, called *bio-inspired networking*, defined as a class of strategies inspired by principles governing biological systems that enable efficient and scalable communication protocols under uncertain conditions [7].

Dressler et al. discussed that biological networks, as a consequence of millions of years of evolution, tackle the aforementioned challenges faced by large scale communication networks. Specifically, biological networks evolve and adapt to varying environmental factors, exhibit robustness against perturbation and failures, effectively manage constrained resources through global intelligence, and self-organize in a fully distributed fashion. For instance, *ant colony optimization* identifies optimized paths between food source and ants' nests based on the length and quantity of a chemical, called pheromone, left in the trail [11]. Artificial immune system mimics the immune system of mammals that detect, memorize and react to systemic changes [9], while systems replicating cellular signalling (of which transcriptional networks are a part) capture the complex patterns of biochemical interactions within living organisms [12]. Thus, it stands to reason that *large scale networks inspired from biological systems should naturally achieve the much-needed optimization of several conflicting system objectives, viz., adaptive, timely, energy-efficient information dissemination in a dynamic environment subject to component failures.*

The focus of this article is the application of a biological network, termed the **transcriptional regulatory network (TRN)** in the design of smart communication networking solutions. The robustness of TRNs in the face of mutation or noise has been a key area of interest in computational biology [13]. Studies show that the robustness of TRNs can be ascribed to its underlying network topology [14]. Consequently, there have been efforts to apply the innate robustness of TRNs in the design of *robust* and *energy-efficient* computer network architectures and protocols by exploiting the standard graph-theoretic attributes of TRNs. We define robustness as *the ability of the network*

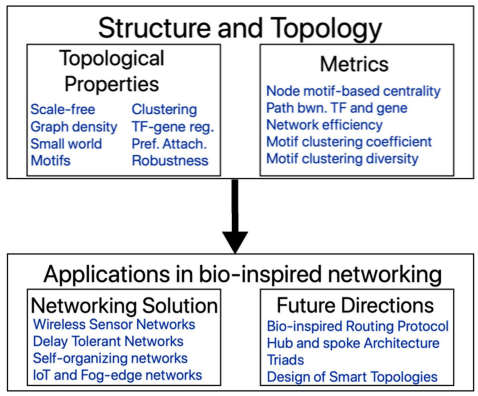


Fig. 2. Overview of contributions.

to carry out information flow despite component (i.e., node or link) failures. In this work, for the first time, we discuss the association between the graph properties of TRNs (such as motif abundance) and its observed robustness; this is followed by an investigation of the metrics to quantify TRN robustness and some networking solutions that leverage it. In Figure 2, we depict the four sections (graph properties, metrics, solutions and future directions) and sub-features thereof (in blue). We highlight the potential areas of exploration of TRN that may motivate new research directions. Note that unlike the topology control mechanisms that operate on the MAC layer by adjusting the sensor transmission power dynamically [15], most of the bio-inspired TRN-based solutions work on the network layer alone by exchanging control packets with neighbor devices [16].

The contents of the survey have been organized as follows. Section 2 briefly covers the biological background and modeling techniques of TRNs. Section 3 discusses the topological attributes of the TRN along with some pertinent results to illustrate them. Section 4 deals with the different metrics that either emerge from properties of TRNs or help explain its graph properties. Section 5 discusses the works that apply the properties of TRN in bio-inspired networking solutions, Section 6 motivates future research directions and Section 7 discusses the concluding remarks.

2 TRANSCRIPTIONAL REGULATORY NETWORKS

In this section, we discuss the rudimentary concepts that form the basis of transcriptional networks, followed by an overview of different ways to model them.

2.1 Components of a Transcriptional Regulation System

Each living cell possesses a *nucleus* that houses the genetic material of the cell. The nucleus has certain threadlike structures called *chromosomes* that carry vital information that cause the transfer of parental traits to the offspring. The chromosomes are made of molecules of **deoxyribonucleic acid (DNA)**. Segments or lengths of DNA, carrying certain instructions for protein synthesis, are called *genes*. The genes constitute the fundamental building blocks of cellular level intelligence.

A transcriptional regulation system consists of genes, cis-elements, and regulators [17]. The regulators are generally proteins, called **transcription factors (TFs)**, and other regulators such as RNAs and metabolites also take part in gene regulation. The process of gene expression works in two steps: transcription and translation (see Figure 3). During *transcription*, the regulating entities bind to the cis-elements lying in structural regions of the double-stranded DNA, called cis-region, to regulate transcription—the process of reading and copying the sequence of bases encoded by a gene to facilitate gene expression. The role of the cis-regions in turn is to aggregate all the

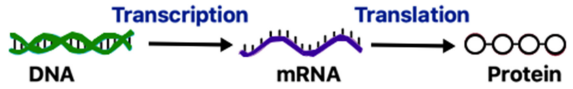
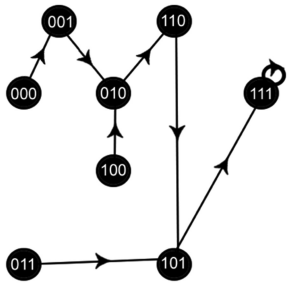


Fig. 3. Transcription and translation during gene expression.



A	B	C	f_1^1	f_1^2	f_1^3
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Fig. 4. Left: Boolean graph for a three-gene network. Right: Truth table for Boolean network.

input signals to facilitate the generation of a gene expression signal. During transcription, a DNA segment is copied into RNA, called **messenger RNA (mRNA)**. The mRNA molecules convey the information of specific amino acids to ribosome, leading to protein generation in a process called *translation*.

The ways to model transcriptional regulation systems can be broadly grouped into the following two classes: (i) dynamic and (ii) static. It has been shown that these network models can explain the dynamics of interactions in TRNs and its effects on the mechanism of diseases [18]. There are dynamic and static that can explain the dynamics of interactions in TRNs and its effects on the mechanism of diseases [18].

2.2 Dynamic Model

These models show the continuous change in gene expression over time. We discuss two categories of dynamic models.

2.2.1 Ordinary Differential Equation Model. This model captures how a gene is regulated by several other signals from regulatory entities, such as TFs [19]. Typically, each gene performs the task of aggregating the signals from its inputs to create an output signal as follows:

$$\frac{d(x_u(t))}{dt} = f_u(x_{u_1}(t), x_{u_2}(t), \ldots). \tag{1}$$

In the above equation, x_u represents the concentration of gene u at time t, while the variables $x_{u_1}, x_{u_2}, \ldots$ are concentrations of regulating molecules that influence the expression of x_u. Finally, f_u is the rate function specifying the manner in which the input signals affect x_u.

2.2.2 Boolean Network Model. A gene can assume two definite states: *ON* (denoted by 1) and *OFF* (denoted by 0), leading to a graphical representation [20, 21]. Since every target gene can be regulated by multiple regulator entities, there are logic operations like *AND*, *OR*, and *NOT* to define relationships between the regulating nodes that may activate (or deactivate) a gene.

Consider a set of N genes $x_1, x_2, \ldots, x_N$ and a set of Boolean functions F showing gene interaction. Since a gene can exist in binary states, a n-gene network is shown by a sequence of n bits, each bit corresponding to a gene state. Each node in the **Boolean network (BN)** is a bit-string (like 001)

(Figure 4 (left)), representing a single state of the network and the edges designate state transitions. Every gene can have l_i number of Boolean functions associated with it. The truth table defined in Figure 4 (right) has functions of the form $f_j^i : j = 1, 2, \ldots, l_i$, which shows the jth function controlling gene expression for the ith gene. For example, f_2^1 represents the first Boolean function associated with the second gene. BN has been extended to a probabilistic Boolean network model to incorporate the uncertainty of gene expression [22]. Here every function f_j^i is associated with a probability value p_j^i denoting the chance that f_j^i will be chosen from the Boolean functions set, F.

2.3 Static Model

Static models lack the time aspect. They represent a qualitative interaction between genes within the network, while capturing the underlying combinatorial interactions among them [17].

2.3.1 Bayesian Network Model. It combines the statistical and graph theoretic approaches of representing gene interactions [23, 24]. It uses an acyclic graph $G = (V, E)$, where $u \in V$ represents the gene expression level and the edges are indicative of the node dependencies.

The Bayesian network considers the probability of gene expression given its parent $P(u|Pa(u))$, where $Pa(u)$ is the parent of node u. If $P(u|Pa(u))$ is known, then one may calculate the joint probability distribution [25]. Analogous to the Boolean model (see Section 2.2.2), the nodes can have values ON and OFF implying activation and inhibition, respectively. The calculation of the joint probability for the N-node Bayesian network (where $|V| = N$) is given by

$$P(u_1, u_2, \ldots, u_N) = \prod_{i=1}^{N} P(u_i|Pa(u_i)). \tag{2}$$

This chain rule helps formalize the Bayesian network. The rule states that the joint probability of the expression or inhibition of multiple genes is a product of the probabilities of each gene given its parent (or predecessor). We compute $P(A = ON, B = ON, C = OFF, D = ON)$ in Figure 5,

$$\begin{aligned} &P(A = ON, B = ON, C = OFF, D = ON) = P(A = ON) \times P(B = ON|A = ON) \\ &\times P(C = OFF|A = ON) \times P(D = ON|B = ON, C = OFF) = 0.6 \times 0.7 \times 0.9 \times 0.1 = 0.0378. \end{aligned} \tag{3}$$

From Figure 5, it is clear that the Bayesian network is a directed acyclic graph. The absence of directed cycles makes it convenient to define a joint probability based on individual conditional probabilities. Also, if there are directed cycles in the Bayesian network, it would come under the ambit of an extended Bayesian model, called *dynamic Bayesian model*. Modeling a genetic network with a feedback (or a directed cycle) with the help of dynamic Bayesian network model requires the inclusion of the first-order Markov assumption into the definition of the Bayesian network. Given that a node $X_i(t)$ at time t has parent nodes $Pa(X_i)(t-1)$ at time $t-1$ [26], the joint probability is

$$P(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} P(X_i(t)|Pa(X_i)(t-1)). \tag{4}$$

2.3.2 Graph Theoretic Model. A graph is an ordered pair $G = (V, E)$, where V is a finite, non-empty set of objects called *vertices* (or nodes); and E is a (possibly empty) set of 2-subsets of V, called *edges* [27]. A directed graph is a graph in which edges have directions. A directed edge $(u, v) \in E$, where u and v are termed the *regulator* and *target* nodes, allows unidirectional information flow from vertex u to v and not necessarily from v to u.

The directed graph representation of TRN shows the interaction between *TFs* and genes, where V is a set of TFs/genes, E is a set of regulatory interactions between TF $\rightarrow$ TF or TF $\rightarrow$ gene, and edge signs $\mathbf{W} : \mathbf{W}((u, v)) \rightarrow \{+, -\}, \forall (u, v) \in E$. Here, *activation* $(+)$ implies that an increase in

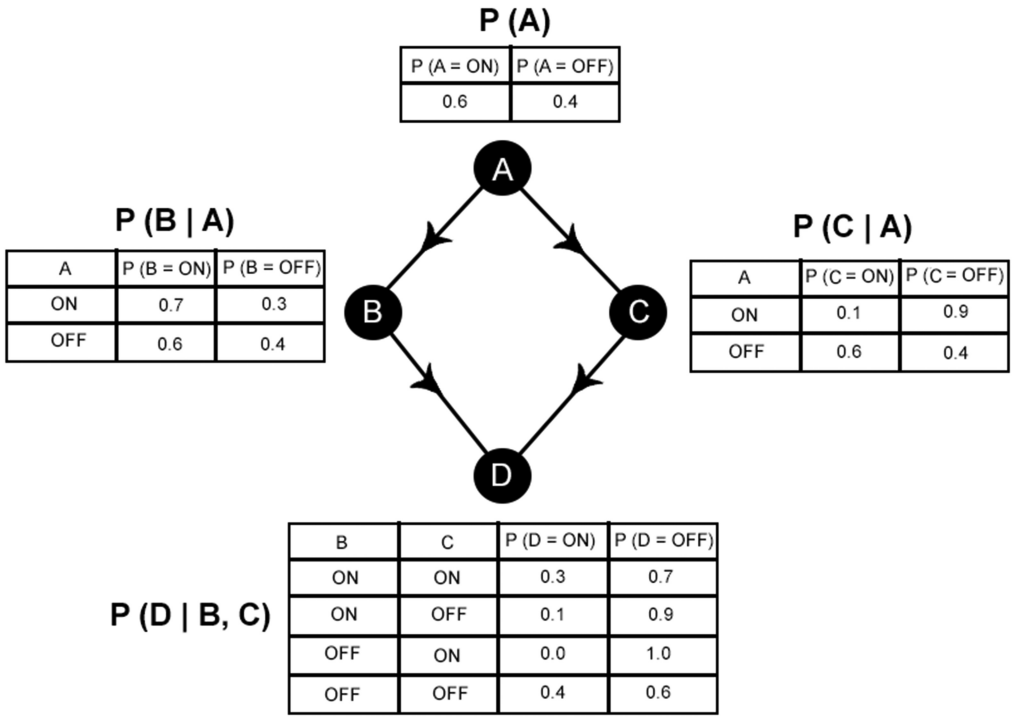


Fig. 5. The acyclic graph shows the probability distribution of each node given its parent node.

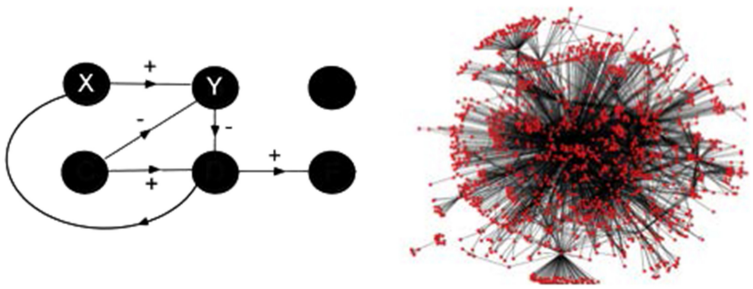


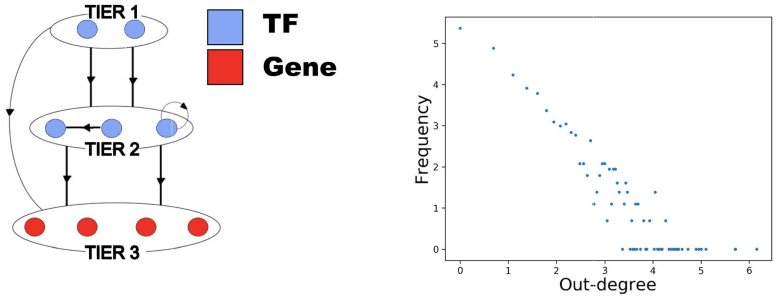
Fig. 6. Graph theoretic representation of TRN and snapshot of *E. coli* TRN (redrawn from Reference [28]).

the concentration of the regulator leads to an increase in the concentration of target. Analogously, *inhibition* (−) implies that an increase in the concentration of the regulator leads to a decrease in target's concentration. Figure 6 (left) shows a signed graph representation of TRN, where the node labeled X is a regulating gene/TF and Y is a target gene. (Often weights are assigned to edges to indicate the strength of regulation [29].) Figure 6 (right) depicts a snapshot of the TRN of a unicellular organism *Escherichia coli* [28].

Degree and path in directed graph: The number of edges leaving a node u is termed its *out-degree* (denoted by $deg^+(u)$) and number of edges entering a node is its *in-degree* (denoted by $deg^-(u)$). A *directed path* is a sequence of vertices such that there is a directed edge pointing from each vertex to its successor in the sequence, with no repeated edges. We represent a path as $p = \{u_j, u_{j+1}, \ldots, u_n\}$, where $(u_i, u_{i+1}) \in E$ $(j \leq i < n)$. A directed path is *simple* if it has no repeated node except the

Table 1. TRN Graphs

TRN type	*E. coli*	*S. cerevisiae*	human	Mouse
No. of nodes	1565	4441	2862	2456
No. of edges	3758	12873	8427	6490



(a) (b)

Fig. 7. Structure of TRN: (a) Three-tier topology: Directed edges indicate TF → TF and TF → gene links across and within tiers (redrawn from References [33, 34]); (b) out-degree distribution of human TRN on a log log scale.

starting and ending node. The length of a simple path is defined as the number of edges it contains. In the interest of consistency, the above graph notations have been used in Sections 3 and 4.

2.4 Dataset

The validated and nearly complete TRNs of *E. coli* and *Saccharomyces cerevisiae* were extracted from a tool called *GeneNetWeaver* [30] that creates meaningful network structures of functionally similar genes (i.e., genes that show a higher mutual interaction than expected by chance). The human and mouse TRNs were obtained from the TRRUST database that presents literature-curated human TF-target interactions [31, 32]; these TRNs catalogue the partially known interactions between TFs and genes. The orders and sizes of the four TRN topologies considered in this work are summarized in Table 1.

3 TOPOLOGICAL PROPERTIES OF TRN

Let us discuss the key topological properties of a TRN, which collectively make them effective templates for designing networking solutions. We visualize the graph properties of TRNs using a *three-tier topological characterization* [33, 34], which is a simplified depiction of the hierarchical structure of TRNs classifying TRN nodes into three tiers based on in- and out-degree distributions (see Figure 7(a)). It is noteworthy that Gerstein et al. studied the hierarchy of network interactions of TFs and mRNAs in humans on the basis of properties such as connectivity, motifs, and so on [35]. Similarly, Bhardwaj et al. employed breadth-first search to form a hierarchy of TFs based on regulating-regulated TF relationships to identify the master regulators in *E. coli* and *S.cerevisiae* TRNs [36]. Finally, Ma et al. [37] proposed the five-level hierarchy of TFs and operons in *E. coli*.

The three-tier topological characterization classifies the TRN nodes into the following three tiers based on in- and out-degree distributions:

- *Tier 1* consists of the set of nodes with only out-degree edges (i.e., $\{u \in V : deg^-(u) = 0\}$).
- *Tier 2* consists of the set of nodes with non-zero in and out-degree edges (i.e., $\{u \in V : deg^+(u) > 0 and deg^-(u) > 0\}$).
- *Tier 3* comprises the set of nodes with only in-degree edges (i.e., $\{u \in V : deg^+(u) = 0\}$).

Table 2. Percentage of Nodes in Each Tier of *E. coli*, *S. cerevisiae*, Human, and Mouse TRN

	E. coli	*S. cerevisiae*	Human	Mouse
Tier 1	4.1	0.7	12.9	14.8
Tier 2	6.2	2.8	14.8	18.8
Tier 3	89.7	96.4	72.2	66.3

Table 3. Percentage of Edges in Each Tier Pair in *E. coli*, *S. cerevisiae*, Human, and Mouse TRN

Tier pair	*E. coli*	*S. cerevisiae*	Human	Mouse
(1 → 2)	0.5	0.5	4.0	6.2
(1 → 3)	10.3	11.0	11.1	10.5
(2 → 2)	8.0	3.3	18.3	27.8
(2 → 3)	81.0	85.1	66.5	55.3

We discuss the node and edge distribution across the three tiers in *E. coli*, *S. cerevisiae*, human and mouse TRN. This topological characterization illustrates that information flow in the TRN takes place from the *hubs* (high degree TF nodes in tiers 1 and 2) to the *non-hubs* (tier 3 genes).

3.1 Node and Edge Distribution

We tabulate the distribution of nodes and edges within and across tiers in Tables 2 and 3.

3.1.1 Node Distribution. Table 2 shows that tiers 1 and 2 make up a small fraction of nodes in all TRNs. All nodes containing self-loops belong to tier 2 (see Figure 7(a)), and the tier 3 contains most of the nodes: 88.6% in *E. coli*, 96.4% in *S. cerevisiae*, 72.2% in human, and 66.3% in mouse.

3.1.2 Edge Distribution. The arrows in Figure 7(a) show possible edge direction within and across tiers, which are between tiers $1 \rightarrow 2$, $1 \rightarrow 3$, $2 \rightarrow 2$, and $2 \rightarrow 3$. We summarize the percentage of edges between and across tiers in Table 3. In all TRNs, over 50% of edges are between tiers 2 and 3.

3.2 Graph Properties

This section is dedicated to the following properties of TRN: (1) scale free out-degree distribution, (2) low graph density, (3) small world property, (4) motif abundance, (5) clustering tendency, (6) robustness to random node failures and vulnerability to hub failures, (7) TF-gene regulation, and (8) preferential attachment, the first six of which are discussed in light of the three-tier topology.

3.2.1 Scale Free Out-degree Distribution. A scale free network is one whose degree distribution follows a power law.[1] Such networks are characterized by the presence of a few well-connected nodes, called *hubs* that possess a high degree, while most of the nodes have a lower degree [38–40].

Table 2 shows that tier 1 and tier 2 nodes account for approximately less that 10% of total nodes in *E. coli* and *S. cerevisiae* and less than 35% nodes in human and mouse but possess all the out-degree edges; tier 3, containing most of the TRN nodes, have zero out-degree (see Figure 7(a)). Since a few nodes have a disproportionately high out-degree, TRN topologies are out-degree scale

[1] A power-law distribution has the functional form $P(k) = Ak^{-\gamma}$. Here, A is a constant that ensures that the $P(k)$ values add up to 1 and the degree exponent γ is usually in the range $2 < \gamma < 3$.

Table 4. Density of TRN Graphs

TRN type	*E. coli*	*S. cerevisiae*	Human	Mouse
D	0.0015	0.00065	0.0010	0.0010

Table 5. Diameter ($\mathcal{D}$) and Average Shortest Path (Z) of TRN

TRN type	*E. coli*	*S. cerevisiae*	Human	Mouse
$\mathcal{D}$	9	6	9	10
Z	2.6	4.6	4.3	4.8

free in nature. In Figure 7(b), we show the power-law degree distribution of human TRN on a log-log scale.

3.2.2 Low Graph Density. We define graph density D on a scale of 0 to 1, as

$$D = \frac{|E|}{|V| \times (|V| - 1)}. \tag{5}$$

In the above equation, $D = 1$ indicates a complete directed graph and $D = 0$ corresponds to an empty graph. We show in Table 4 that the TRN topologies exhibit a very low graph density [41].

Reason: Tier 3, which account for 60–90% nodes in TRN, have no edges within it. Since the majority of nodes are not directly connected, TRN topologies have an overall low graph density.

3.2.3 Small World Property. A small world network is one where it is possible to travel from one node to another in a limited number of hops [42]. Small world property is measured in terms of graph diameter, calculated as the largest shortest path between any pair of vertices [43].

The three-tier topological characterization of TRN depicts that information flows unidirectionally from tier 1 to tier 3 (see Figure 3 (left)), making TRNs weakly-connected graphs (i.e., there exist some nodes that are not reachable from other nodes). Since the diameter is not defined for weakly connected topologies, we use the following two metrics to demonstrate the small world property of TRN: (1) *diameter* of undirected TRN and (2) *average shortest path* from tier 1 to 3 nodes (defined below).

Given $V = t_1 \cup t_2 \cup t_3$, where t_i is the set of nodes belonging to the ith tier ($i = 1, 2, 3$) in the three-tier topology, the average shortest path is defined as

$$Z = \frac{1}{|P|} \sum_{u \in t_1} \sum_{v \in t_3} d(u, v). \tag{6}$$

Note that in the above equation P is the number of (u, v) node pairs such that $u \in t_1$, $v \in t_3$ and v is reachable from u.

From three-tier topology, we intuit that the expected number of hops from a tier 1 to a tier 3 node should be 2 (tier 1 $\rightarrow$ tier 2 $\rightarrow$ tier 3). From Table 5, we observe that the highest $\mathcal{D}$ is 10 and Z is 4.8 (using Equation (6)), for undirected TRN. This shows the small world property of TRNs [44, 45] and explains the low end-to-end communication delay in large-scale bio-inspired wireless networks. We discuss in Section 4.2 that TRNs exhibit shorter path length compared to their randomized counterparts.

3.2.4 Motif Abundance. Network motifs are subgraphs that repeat themselves in complex networks, such as social and technological networks [46]. They are considered to be the building blocks and play key functional roles in biological networks like TRNs [47–52]. For instance, motifs control the gene expression by moderating the responses to fluctuating external signals. They are considered to be input-output devices that require inputs like heat, nutrients and pressure and

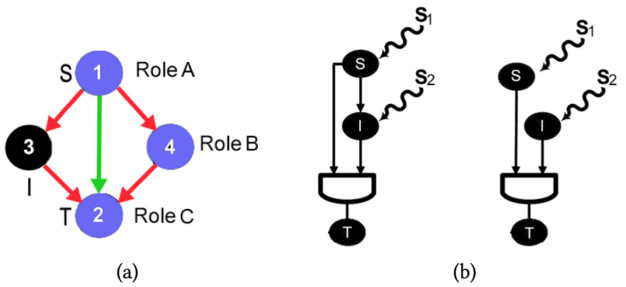


Fig. 8. (a) FFL motif with different FFL motif centrality roles (i.e., roles A, B, C) marked blue(b) left: TF S regulates TF I, and S and I jointly regulate T; right: S and I regulate T.

produce outputs like regulation signals that act upon the targets. Based on duration and intensity of regulation, the motifs could control some of the vital functions in the living organisms [53].

Motif detection tools: There exist a variety of tools to detect network motifs. Two popular tools are MFINDER [54] and MAVISTO [55]. While MFINDER is capable of detection of network motifs, MAVISTO is equipped with a visualization tool to capture the presence of a motif in a network by a force-directed graph layout algorithm. Wernicke et al. put forward a scalable and fast motif detection tool, called FANMOD [56], that is capable of handling colored vertices and edges to model different kinds of node interactions such as finding motifs in protein-gene interaction networks. In FANMOD, the subgraphs are grouped into isomorphic subgraph classes based on canonical graph-labeling algorithm NAUTY [57]. It then calculates the frequency of subgraph classes in a user-specified number of random graphs generated from the original network, by switching edges between vertices. More details about other existing motif detection tools can be found in Reference [58].

Let us now discuss the most abundant motifs and some of their functionalities in the TRN.

(1) **Feed forward loop (FFL)** is known to be one of the most abundant motifs in TRN [59, 60]. Figure 8(a) shows a FFL motif, where TFs S and I regulate the expression of gene T. S is the *general TF*, I is the *specific TF* and T is the *effector operon*; S regulates T directly and indirectly (via I). Since TRNs are signed networks (see Section 2.3.2), the sequence of +/− signs classifies a FFL into two categories of *coherent* or *incoherent* having different roles in the information flow.

Definition 1. A FFL is called *coherent* if the direct effect of the general TF S on the effector operon T, has the same sign as the indirect effect through the specific TF I. *Incoherent* FFLs have the opposite signs for the two different paths.

While calculating this *indirect effect* of activation (+) and inhibition (−), we apply the mathematics rule of product of two signs: The product of like-signs (+ and +; − and −) yields a net positive result, while the product of unlike signs (+ and −) yields a net negative output. This leads to 8 types of FFLs: four belonging to coherent and incoherent types each (see Figure 9).

The FFL has two input signals, the inducers, S_1 and S_2, which are molecules that activate or inhibit the activity of S and I (Figure 8(b) (left)). Coherent or incoherent FFL motifs have specific information-processing roles by regulating the activation of target gene T, defined in terms of the time (called *response time*) it takes a gene product (i.e., protein) to reach its steady-state level [50]. Incoherent FFLs act as accelerators, i.e., they provide a mechanism

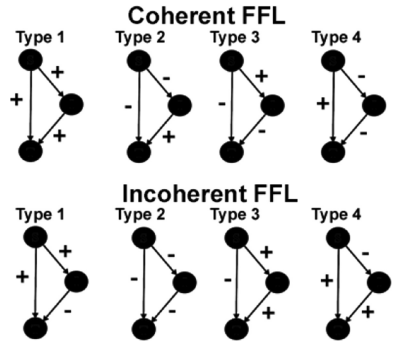


Fig. 9. Coherent and incoherent feed forward loop motifs.

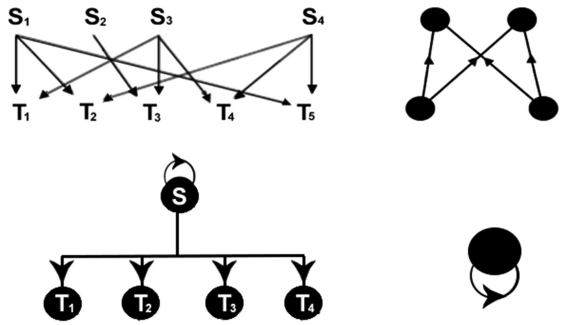


Fig. 10. Motifs in TRN. Top Left: Dense overlapping regulon; top right: bi fan; bottom left: simple input module; bottom right: auto-regulation.

for speeding up the responses of T, whereas the coherent FFLs lead to delay in the response of target T when compared against direct regulation (shown in Figure 8(b) (right)).

Abundance of FFL in TRN: As per three-tier topology, TRNs permit links between tiers $1 \rightarrow 2$, $1 \rightarrow 3$, $2 \rightarrow 2$, and $2 \rightarrow 3$ (see Figure 7(a)). Thus, FFLs exist among tiers $1 \rightarrow 2 \rightarrow 2$, $1 \rightarrow 2 \rightarrow 3$, $2 \rightarrow 2 \rightarrow 2$, and $2 \rightarrow 2 \rightarrow 3$. As per FANMOD, of 455, 152 3-node subgraphs enumerated in human TRN, approximately 5, 850 motifs are FFLs or motifs possessing FFLs as building blocks.

(2) **Dense overlapping regulons (DORs)** constitute a set of regulating genes S_i and target genes T_i set in the form of a bipartite graph. They are called dense because they are occur in cascades or layers as depicted in Figure 10 (top left). DORs are responsible for a number of biological functions like carbon utilization, growth and stress response [37]. We often consider a 4-node substructure of DOR, called *bi fan*, as a standalone motif (Figure 10 (top right)).

Abundance of bi fan in TRN: As per FANMOD, of 43, 995, 531 4-node subgraphs enumerated in human TRN, approximately 132, 481 motifs are bi fans or possess bi fans as building blocks.

(3) **Single Input Modules (SIMs)** has a single regulating TF S that regulates a number of genes T_i [61, 62]. Its key property is that all the target genes are either activated or all are repressed. The regulator TF S, called *hub* in Section 3.2.1, also regulates itself. As shown in Figure 10 (bottom left), regulator S has a self-loop. SIM causes the collective expression of multiple genes, although the regulator may have varying activation threshold for different targets.

(4) *Auto-regulation:* When a gene binds its own promoter (Figure 10 (bottom right)) and activates itself we call it positive auto-regulation, and when it represses itself it is called

Table 6. FFL Count and Average Clustering Coefficient of (Undirected) TRN and Corresponding E-R Random Graphs (R-) of the Same Graph Density

	E. coli	R-*E.coli*	*S. cerevisiae*	R-*S. cerevisiae*	*human*	R-*human*	*Mouse*	R-*Mouse*
FFL	4798	18	4115	30	5850	23	2714	29
ACC	0.2110	0.0033	0.0830	0.0015	0.1200	0.0017	0.0970	0.0026

negative auto-regulation [63]. Simulations on Boolean network models (see Section 2.2.2) show that robustness and stability of TRN correlates with frequency of auto-regulation in the network [64]. *E. coli*, *S. cerevisiae*, human, and mouse have 110, 0, 24, and 28 auto-regulation motifs, respectively.

3.2.5 High Clustering Tendency. We argue that the abundance of motifs in TRNs is a consequence of its tendency to form dense, tightly knit groups, called *clusters*. The clustering tendency of any node u in an undirected graph H is measured in terms of its clustering coefficient, given by

$$CC(H,u) = \begin{cases} 0, & \text{if } \delta(u) < 2 \\ \frac{2\times t(u)}{\delta(u)\times(\delta(u)-1)}, & \text{otherwise} \end{cases}. \tag{7}$$

In the above equation $t(u)$ is the number of triangles node u participates in and $\delta(u)$ is its degree. The **average clustering coefficient (*ACC*)** of H is given by

$$ACC(H) = \frac{1}{|V(H)|} \sum_{u \in V(H)} CC(G,u). \tag{8}$$

Equation (7) suggests that *ACC* is directly proportional to the number of triangles in an undirected graph.

Relationship between motif abundance and clustering: Table 6 shows that *ACC* of TRN is over 80 times that of **Erdös-Rényi (E-R)** random graphs[2] of same order and roughly same graph density. This high *ACC* of TRN is commensurate with its FFL motif abundance. Note that the motifs (primarily the FFLs and bi fans) do not appear in isolation; they form dense clusters [66–68]. Investigation on the *E. coli* TRN topology indicate that there are 42 FFLs, that form six FFL motif clusters. Similarly, 208 bi fan motifs participate into two clusters. Table 6 shows that in addition to *ACC*, the number of FFL motifs in TRNs are significantly higher than their random counterparts.

3.2.6 TF-gene Regulation. We have discussed in Section 2.3.2 that TRNs are signed networks, i.e., the edges carry positive or negative weights implying activation or inhibitory TF/TF or TF/target gene regulation. However, the information of the edge signs is not complete. The number of +, −, and unknown edge signs summarized in Table 7 play a key role in a bio-inspired networking solution (discussed in Section 5.4). Note that the edge signs for *S. cerevisiae* TRN is not available.

The signed regulation of the genes by TFs results in a dynamic and self-regulating system, where the expression of a single gene is determined by several regulating entities. We show in Section 5.4 that this property serves as a reference point for energy-efficient communication among IoT devices.

3.2.7 Preferential Attachment Growth Model. We have discussed in Section 3.2.1 that TRNs have a scale free out-degree distribution. One approach to construct such networks is to employ

[2]The Erdös- Rényi random graph is creating a set of isolated nodes and connecting each pair of nodes by an edge with a prespecified probability p [65].

Table 7. Percentage of Positive (P), Negative (N), and Unknown (U) Edges in TRN

TRN type	*P*	*N*	*U*
E. coli	53.20	41.10	5.50
S. cerevisiae	–	–	–
human	33.51	20.45	46.03
Mouse	40.28	19.18	40.52

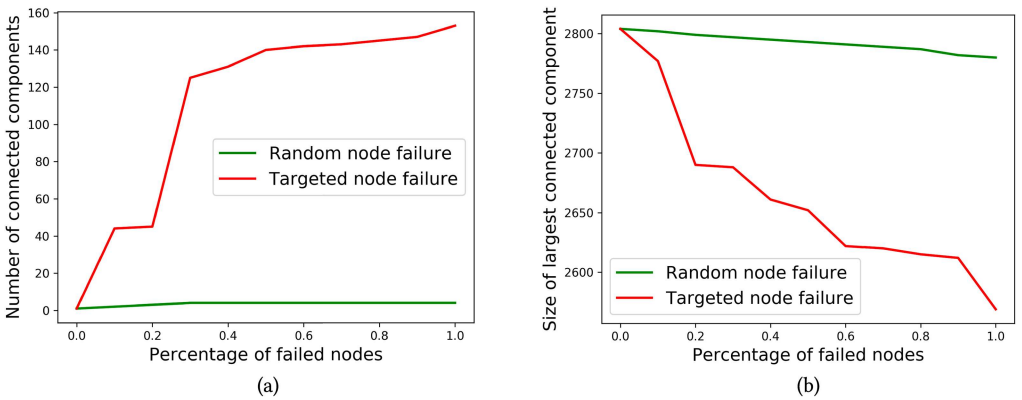


Fig. 11. (a) Number of connected components in random vs. targeted node failures. (b) Size of largest connected components in random vs. targeted node failures.

a *preferential attachment growth model*, wherein when a new node is inducted into a network, it prefers to get attached to a node that has high degree of connectivity [39].

As a consequence to preferential attachment the hub nodes, the preferred candidates of attachment for a new node, tend to acquire more and more links as the network grows. The probability of addition of an edge between a new node and an existing node of u of degree k_u could be *linear* to the degree of node (i.e., $p(u) = \frac{k_u}{\sum_{v \in V} k_v}$) or *nonlinear* (i.e., $p(u) = \frac{k_u^\gamma}{\sum_{v \in V} k_v^\gamma}$). In the past, there have been efforts to study the motif distribution of TRN with randomized networks generated by linear and nonlinear preferential attachment approaches [67, 69]. The FFL motif abundance of preferential attachment-based topologies has been shown to be comparable to that of *E. coli* TRN.

3.2.8 Robustness Against Random Node Failure. Robustness of a biological system is typically defined as the ability of the organism to retain its characteristic traits (called *phenotype*) in the face of genetic change (i.e., mutation) [70]. In our context, network *robustness* is defined in terms of its ability to withstand component failures (see Section 1). We discuss in Section 3.2.1 that the TRNs exhibit scale free out-degree distribution—a property that makes them inherently resilient to the failure of random nodes, yet particularly vulnerable to the failure of hub nodes [71, 72].

Reason: The targeted failure or removal of the hub nodes in tiers 1 and 2 of the three-tier topology is likely to knock off the majority of the poorly connected tier 3 nodes. Conversely, over 65% of TRN nodes reside in tier 3. Nodes randomly picked for removal are highly likely to belong to tier 3, and their removal should not affect the overall connectivity of the TRN [33]. To demonstrate this, we take a cue from the known measures of network robustness [73, 74] and carry out a simple experiment wherein we knock off 0.1–1% (1) *randomly chosen nodes* in human TRN and (2) *targeted nodes* chosen with likelihood equal to their degree. Figures 11(a) and (b) shows that

the targeted node failure results in the network fragmenting into higher number of components as well as lower size of largest connected component as compared to an E-R random network. This property motivates the design of bio-inspired wireless networks that are robust against random failures.

4 METRICS

In this section, we discuss the different metrics that are mostly derived from the graph properties of TRN. Specifically, the concepts of *node motif-based centrality* (Section 4.1), *motif clustering coefficient* (Section 4.4) and *motif clustering diversity* (Section 4.5) are based on the presence of and interconnections among the FFL motifs introduced earlier in Section 3.2.4. Similarly, the *average shortest path* (Section 4.2) is founded upon the TF-gene interactions in signed transcriptional networks discussed in Section 3.2.6.

4.1 Node Motif-based Centrality

We define FFL motifs in Section 1. A FFL motif is mathematically represented as an ordered triplet of three nodes u, v and w such that there exists a direct edge (u, w) and an indirect path $p = \{u, v, w\}$. For example, in Figure 8(a), nodes 1, 3, and 2 forms an FFL. The existence of an FFL motif between three nodes u, v and w can be denoted by an indicator variable $\mathbf{M}$ as follows:

$$\mathbf{M}(u, v, w) = \begin{cases} 1, & \text{if } (u, v), (v, w), (u, w) \in E(G) \\ 0, & \text{otherwise} \end{cases}. \tag{9}$$

In Figure 8(a), $\mathbf{M}(1, 3, 2) = 1$, whereas $\mathbf{M}(1, 3, 4) = 0$.

Definition 2. FFL motif centrality of any node u (or edge (u, v)), $\delta(u)$ (or $\delta((u, v))$), is the number of FFL motifs it participates in.

In Figure 8(a), $\delta(1) = 2$ and $\delta((1, 2)) = 2$.

Roles of nodes in motif centrality: Koschützki et al. showed that motif centrality of a node is classified into roles A, B, and C (marked blue in Figure 8(a)) [75]. Given nodes u, v and w, such that $\mathbf{M}(u, v, w) = 1$, the general TF u is role A, specific TF v is role B and regulated gene w is role C. In Figure 8(a), for the FFL motif (highlighted in blue), nodes 1, 4, and 2 play the roles of A, B, and C, respectively.

(1) *Role A motif centrality* is the number of FFLs where node u is the master regulator, i.e., $\delta_A(u) = |\{\mathbf{M}(u, v, w) : u, v, w \in V\}|$
(2) *Role B motif centrality* is the number of FFLs where u is the intermediate regulator, i.e., $\delta_B(u) = |\{\mathbf{M}(v, u, w) : u, v, w \in V\}|$
(3) *Role C motif centrality* is the number of FFLs where u is the regulated node, i.e., $\delta_C(u) = |\{\mathbf{M}(v, w, u) : u, v, w \in V\}|$

Finally, the *FFL motif centrality* of node u is the sum of its role A, B, and C motif centralities, i.e., $\delta(u) = \delta_A(u) + \delta_B(u) + \delta_C(u)$

Alternative communication pathways: Our experiments in Reference [76] reveal that FFL motifs play a significant role in the information flow across tier 1 to tier 3 in TRN. To better understand this role of FFL motif, it is imperative to discuss the concept of *simple* and *independent* paths.

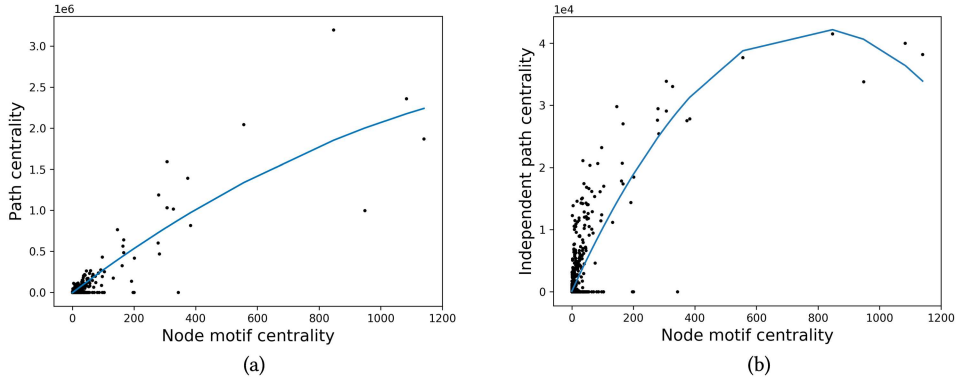


Fig. 12. Human TRN. (a) PC vs. node motif centrality δ. (b) IPC vs node motif centrality δ.

Definition 3. Simple path is one in which no node is visited more than once. Two paths between a node pair are called independent if they contain no common nodes except source and destination nodes.

In graph G in Figure 8(a), $p_1 = \{1, 3, 2\}$ and $p_2 = \{1, 4, 2\}$ are independent paths from node 1 to 2.

Definition 4. **Path centrality (PC)** of a node is the number of simple paths between all pair of nodes that the given node intercepts. Similarly, (vertex) **independent path centrality (IPC)** of a node is the number of independent paths between all node pairs that the given node intercepts.

The procedure for the calculation of IPC is discussed in Reference [76]. Each FFL possesses a direct path connecting node S to node T (marked in green in Figure 8(a)), and an indirect path via I (marked in red). The profusion of FFLs in a TRN has the following effects on its topology [76]:

(1) *Robustness due to independent paths:* According to Menger's theorem on vertex connectivity, the minimum number of vertices whose removal disconnects two nodes is equal to the maximum number of pairwise vertex-independent paths between them [77]. In Figure 8(a), at least two nodes (3 and 4) must be removed to disconnect 1 and 2. Since the FFL motif contains two independent paths connecting nodes S and T, the abundance of FFLs clusters (as shown in Table 6) ensures graph robustness in TRNs by offering multiple alternative pathways.
(2) *Increase in shortest path length:* Node or link failures may increase the shortest path length between pairs of existing nodes or render them unreachable from one another. In case of FFLs, the failure of direct links between source S and target T causes the shortest path length between S and T to increase only by a single hop. Thus, the abundance of FFLs makes TRNs resilient by minimizing the increase in shortest path length during node or link failures.

To demonstrate the role of FFL motifs in forming robust pathways of signal propagation between tier 1 to tier 3 in TRN, we plot *IPC* and *PC* against node motif centrality δ for all nodes from the human TRN. We apply nonlinear regression to obtain best fit lines from the scatter plot.

Figure 12(a) shows that PC and δ of TRN nodes are correlated, suggesting that FFLs participate in communication pathways in human TRN. Analogously, Figure 12(b) depicts the correlation between IPC and δ, implying that FFL motifs are responsible for rendering graph robustness to TRN, by creating vertex-independent paths (i.e., alternative pathways) between tiers 1 and 3. Wireless networks prone to failures where information may be time-sensitive (such as delay tolerant networks as discussed later in Section 5.3) attempt to preserve high **node motif-based centrality (NMC)** nodes. The abundance of FFL motifs translates to uninterrupted and timely information

exchange as a result of the multiplicity of communication pathways despite component failures as well as low latency offered by FFLs.

4.2 Average Shortest Path between TFs and Genes

Recall that $deg^+(u)$, $deg^-(u)$ denote the in- and out-degree of a node $u \in V$, and Equation (6) is employed to calculate the average shortest path between tier 1 to tier 3 nodes in TRN. Abdelzaher attempted to test the correctness of the statement: *The average shortest path for TRNs is among the smallest compared to networks with same in- and out-degree distributions and the same or less FFL count* [78].

Problem statement: Given any network G of an in-degree distribution $D^{in} = \{d_1^{in}, d_2^{in}, \ldots, d_n^{in}\}$, out-degree distribution $D^{out} = \{d_1^{out}, d_2^{out}, \ldots, d_n^{out}\}$, an embedded FFL motif abundance $M(G)$ and an average shortest path Z among pairs of P, obtain an optimal network G_0, such that $D_O^{in} = D^{in}$, $D_O^{out} = D^{out}$ and Z is minimum.

The problem of determining the optimal network topology led to the following optimization:

$$\textbf{Min} \sum_{l=1}^{f} \sum_{i \neq j}^{n} Y_{ij}^{l}, \tag{10}$$

$$\sum_{j=1}^{n} Y_{ij}^{l} - \sum_{j=1}^{n} Y_{ji}^{l} = \begin{cases} 1 & \text{if } i \in t_1 \\ -1 & \text{if } i \in t_3 \\ 0 & \text{Otherwise} \end{cases}, \tag{11}$$

$$x_{ii} = 0 \qquad \forall i \in V, \tag{12}$$

$$\sum_{j=1}^{n} x_{ij} = d_i^{out} \qquad \forall i \in V, \tag{13}$$

$$\sum_{j=1}^{n} x_{ji} = d_i^{in} \qquad \forall i \in V, \tag{14}$$

$$\sum_{i \neq j \neq k} y_{ijk} \leq M(G), \tag{15}$$

$$x_{ij} + x_{jk} + x_{ik} \leq 2 + y_{ijk}, \tag{16}$$

$$x_{ij}, y_{ijk}, Y_{ij}^{l} \in \{0, 1\}. \tag{17}$$

Description: Y_{ij}^{l} is an indicator variable suggesting that edge (i, j) is part of the shortest path between a source node s and a destination node t, defined as follows:

$$Y_{ij}^{l} = \begin{cases} 1 & \text{if } (i, j) \text{ belongs in the shortest path between nodes } u \text{ and } v \\ 0 & \text{otherwise} \end{cases}. \tag{18}$$

(1) The objective (Equation (10)) is to minimize the expression $\sum_{i \neq j}^{n} Y_{ij}^{l}$. The fact that the shortest path across all node pairs is considered, the objective is guaranteed by summing over all f possible node pairs ($\sum_{l=1}^{f}$).

(2) Constraint 11 encodes the shortest path between the tier 1 nodes (t_1) and tier 3 nodes (t_3). The expression $\sum_{j=1}^{n} Y_{ij}^{l} - \sum_{j=1}^{n} Y_{ji}^{l}$ (a) $= 1$ if node $i \in t_1$ and it has shortest path to genes j; (b) $= -1$ if node $i \in t_3$ and it has incoming edges from TFs j; (c) $= 0$ if node $i \in t_3$ is an intermediary between a TF and gene and has a shortest path edge entering and leaving it.
(3) In Constraint 12, variable $x_{ij} = 1$ if there exists a directed edge between nodes i and j. Constraint 12 eliminates self-loops in the optimized topology. Equalities (13) and (14) ensure that the in- and out-degree distributions of the original and optimized topologies are identical.
(4) Here variable $y_{ijk} = 1$ if there is a FFL motif between nodes i, j, and k. Constraint 15 ensures that the number of FFLs in the optimized graph does not exceed those in the original topology.
(5) There exists a FFL motif between nodes i, j and k only when there exists edges $(i, j), (j, k)$, and (i, k) (i.e., $x_{ij} = 1, x_{jk} = 1$ and $x_{ik} = 1$). This condition is enforced by Constraint 16.
(6) Constraint 17 stipulates that that the variables $x_{ij}, y_{ijk}, Y_{ij}^{l}$ are integers.

Randomized TRN: The authors generate randomized TRN by repeatedly selecting node pairs with uniform randomness from the set of possible pairwise combinations between distinct TRN nodes, and drawing a directed edge between them and excluding this node-pair from the set of available pairs, until the necessary graph density is reached. The goal is to compare the shortest path of TRN (Z_T), randomized TRN (Z_R) to that of the optimized topology (Z_O), using the following formula:

$$\delta = \frac{Z - Z_O}{Z}. \tag{19}$$

Here Z can be replaced by both Z_R and Z_T. Evidently, $\delta = 0$ indicates that the corresponding topology exhibits a perfectly optimal shortest path. Experimental results (not shown here) suggest that the subgraphs sampled from *E. coli* TRN topology possess a significantly smaller δ than their randomized counterparts [79], implying that it is naturally optimized for shortest paths.

4.3 Network Efficiency

We discuss in Section 3.2.2 TRNs possess low graph density, where a large number of node pairs are unreachable from one another. For any node pair $u, v \in V$, we use $d(u, v) = \infty$ to suggest that v is not reachable from u, which would make the average shortest path across all node pairs undefined. Literature on TRN uses an alternative measure of shortest path, based on harmonic mean of shortest paths, called network efficiency [66, 81]. Specifically, for any directed graph $G(V, E)$, it is given by

$$\mathcal{E} = \frac{1}{\rho} \sum_{u,v \in V} \frac{1}{d(u, v)}. \tag{20}$$

Here (i) $\rho = |V| \times (|V| - 1)$ and $d(u, v)$ is the shortest path length between any node pair $u, v \in V$. It is worth mentioning here that the abundance of FFL motifs result in short paths connecting the TFs to the genes (shown above in Section 4.2). We discuss in Section 5 that wireless sensor networks as well as IoT networks mimicking the TRN topology also possess naturally optimized (and significantly shorter) communication path lengths manifested as improved network efficiency.

4.4 Motif Clustering Coefficient

The motif clustering coefficient represents the amount of overlap in terms of shared nodes between all pairs of a particular set of (undirected) triangular motifs in a network [80]. Given that any pair of motifs can share 1 or 2 nodes, the motif clustering coefficient is then given by

$$M_c = \frac{S_c}{T_c}. \tag{21}$$

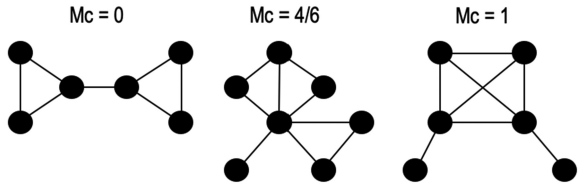


Fig. 13. Examples of three topologies with increasing motif clustering coefficient (redrawn from Reference [80]).

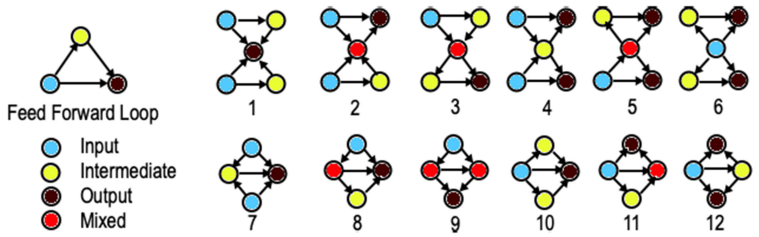


Fig. 14. Twelve types in MCD (redrawn from Reference [80]).

Here S_c is the total number of shared nodes between all pairs of motif and T_c is the total possible number of shared nodes for all triangular motif pairs. Figure 13 shows three graphs with increasing M_c.

4.5 Motif Clustering Diversity

For any node, one can extract all FFL motifs that contain the selected node as a member. For each pair of FFL motifs that share at least one node, it is possible to create 12 possible configurations (shown in Figure 14). ***Motif clustering diversity (MCD)*** *is defined as the number of distinct motif clustering types (i.e., configurations) that a node takes part in.* Its value ranges between 0 and 12 [80]. Gorochowski et al. report several nodes that exhibit high MCD act as global regulators controlling information flow resulting in the transcription of several target genes in TRNs. We discuss in Section 6.4 that metrics like motif clustering coefficient and diversity are being actively considered to quantify the efficient information dissemination achieved by TRNs. This may motivate the design of routing strategies that rely on directing bulk of the network traffic through nodes participating in motif structures.

5 APPLICATION OF TRN IN BIO-INSPIRED NETWORKING

The graph properties (see Section 3) and metrics (see Section 4) make TRN an ideal template for the design of the following smart network architectures: (1) WSN that monitor environmental conditions such as temperature, sound, winds, pressure, motion, pollutants, and so on, and pass the data via multiple hops to a repository for aggregation and analysis; (2) DRN comprising energy-constrained smartphones and laptops (possessed by mobile survivors) and a gateway fog node called the CC; (3) *IoT* IoT-Net, where IoT devices, capable of sensing and actuation, report time-sensitive data directly to a cloud platform; (4) *edge centric IoT network* that allow participants with smart devices to collect data in response to advertised events and report them to the server via fog devices. Table 8 summarizes the implications of the topological properties of TRNs on the aforementioned wireless networks. This section is dedicated to the approaches in existing literature that employ TRNs to design robust, energy-efficient, and distributed networking

Table 8. Network Implications of TRN Graph Properties (Discussed in Section 3)

Graph property	Network implications
Scale free distribution	Robust against random node failures
Low graph density	Energy efficiency
Small world property	Low communication latency
FFL motif abundance	Seamless network connectivity/rapid information spreader
TF-gene regulation	Distributed energy-level regulation
Preferential attachment	Existence of hubs

solutions. In Table 9, we categorize each work w.r.t. the TRN property, network type, and the performance measures used to analyze them. We close this section with a comment on the drawbacks and overheads of the schemes.

Performance measures: The works discussed hereafter consider the following performance measures:

- **Packet delivery ratio (PDR)** measured as the ratio of the number of unique data/event packets delivered at the sink/base station to the total number of packets generated/sensed by the sensing devices.
- *Latency* is the average delay (in seconds or number of hops) incurred in delivering the messages from the edge sensors to the sink/base station.
- *Energy efficiency* gauges the network lifetime in terms of the percentage of alive or active devices or the average residual energy of devices over a period of time.
- *Robustness to hub node failure* is gauged as the number of devices that are disconnected from a network topology by the failure of certain network components due to energy depletion or hardware faults.
- *Energy level* or *energy state* that dictates the energy consumption rate of a device that regulates its instantaneous data/event sampling rate. Inspired by the concept of the activator-inhibitor systems modeled using the nonlinear differential equations (where cells undergo state transition based on rules that factor in their proximity with neighboring cells [82]), the proposed methods allow nodes to determine their states based on the inputs from neighbors states. This results in distributed mechanisms to achieve uniform utilization of resources (like energy).
- **Quality and quantity of events sensed (qNq)** is a joint measure for *q*uality (q^t) and *q*uantity (n^t) of event data collected by an IoT device in time epoch t, calculated as $\sigma \cdot n^t + (1 - \sigma) \cdot q^t$, where quality of events is proportional to the instantaneous event sampling rate of the device and σ is the weighing factor lying between 0 and 1.

5.1 Wireless Sensor Networks

WSNs constitute devices that monitor physical and environmental conditions. The challenges of communication failures, storage and computation and energy limitations are traditionally built into the functioning of WSNs [103]. The paradigm of TRN-inspired WSNs is part of recent efforts to infuse computational intelligence from biological systems and make WSNs robust against failures, energy aware and adaptive to environmental vagaries [104, 105]. Let us consider a homogeneous WSN graph $G_w = (V_w, E_w)$ where V_w and E_w are the set of nodes and edges representing the sensors and the bidirectional links between the sensors, respectively. An edge (u, v) exists in the WSN if and only if sensors u and v are in communication range. The authors have drawn the following parallels between TRN and WSN:

Table 9. Classification of the Works on the Application of TRN-based Bio-Inspired Networking on the Basis of TRN, Properties of TRN, Network Type, and Performance Measures

	Networks				
Graph Properties	**WSN**	**DTN**	**Self-organized**	**IoT-Net**	**Edge Computing**
Small world shortest path	[33, 83]	[84, 85]			[86]
Low graph density	[33, 83]	[84, 85]		[87]	[86]
Preferential attachment	[33, 83, 88]				
TF-gene interaction	[89, 90]		[91–93]	[87, 94, 95]	
Motifs clustering	[33, 83, 96–101]	[84, 85]	[102]		[86]
Robustness	[33, 83, 96–100]	[84, 85]	[102]	[87, 95]	[86]
Performance measures	PDR, latency, robust	PDR, energy, robust	Neighbor regulation	Neighbor regulation, energy, qNq	energy efficiency, robustness

- The genes are capable of mutual interaction and regulation of their neighbours' inputs (refer to discussion on *activator-inhibitor systems* in section on performance measures). Similarly, the sensor nodes can communicate with each other and exchange vital information.
- The *robustness of TRN against random failures* (see Section 3.2.8) could be extrapolated to tackle possible component failures.
- TRN exhibit the *small world property* (see Section 3.2.3) that can lead to low latency in data transfer from prespecified source to sink nodes.

5.1.1 TRN-WSN Mapping Strategy. Nazi et al. proposed a mapping strategy that transfers the graph attributes of TRN graphs to an already deployed WSN topology [94, 96–100]. The idea behind this strategy is to generate a sparser (i.e., low graph density) yet robust WSN topology, called *mapped WSN*, that preserves the topological properties of TRN. The authors propose a mapping function (discussed below) to find a one-to-one correspondence between the TFs/genes and WSN nodes.

Mapping function: TRN-based mapping generates mapped WSN topology $G_w^m(V_w^m, E_w^m)$ based on a one-to-one mapping function $\rho : G_w^m \rightarrow G_g$, where $G_w^m(V_w^m, E_w^m)$ is a directed graph, such that $V_w^m \subset V_w$ and $E_w^m \subset E_w$. Directed edge $(u, v) \in E_w^m$ exists if and only if there exists a directed path between $\rho(u)$ and $\rho(v)$ in G_g. Note that as a pre-processing step to the mapping algorithm each edge in input WSN is assigned a weight ω representing the degree of interference, calculated as

$$\omega(u,v) = \begin{cases} 1.0 - \frac{A(u,v)}{2\pi r^2}, & \text{if } d < 2r \\ 1, & \text{otherwise} \end{cases}.$$

Here r is the sensing range, d and A represents the normalized area subject to interference between nodes u and v separated by a distance d, calculated as $A(u,v) = 2r^2 cos^{-1}\frac{d}{2r} - \frac{d}{2}\sqrt{4r^2 - d^2}$. Note that $\omega(u,v) = 1$ indicates maximum separation and therefore *minimum interference*.

Illustrative Example: As shown in Figure 15(a) and (b), the list of TRN node and IoT device labels are annotated by their normalized pageranks [106]. The algorithm processes the list of TRN nodes and sensor devices in the non-increasing order of their pageranks; it first maps the highest ranking sensor device 1 into TRN node c. Since 0 is the next highest ranked device having an edge with mapped device 1 is 0, it gets mapped to the next highest ranked TRN node g (neighbor of gene c). Similarly, device 2 is mapped to TRN node b. The next highest ranking device 4 shares edges with mapped devices 0 and 1. It is mapped to TRN node e, which has paths to corresponding mapped TRN nodes g and c. Note that the communication links inherit the direction of data forwarding from the direction of paths between corresponding mapped TRN nodes, resulting in

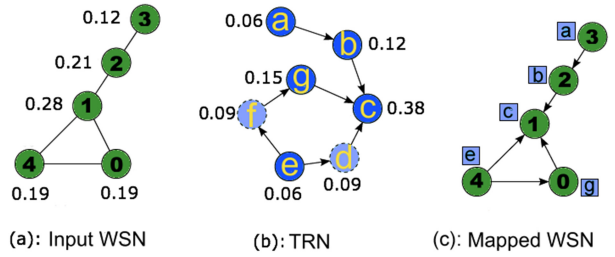


Fig. 15. Working of the mapping algorithm.

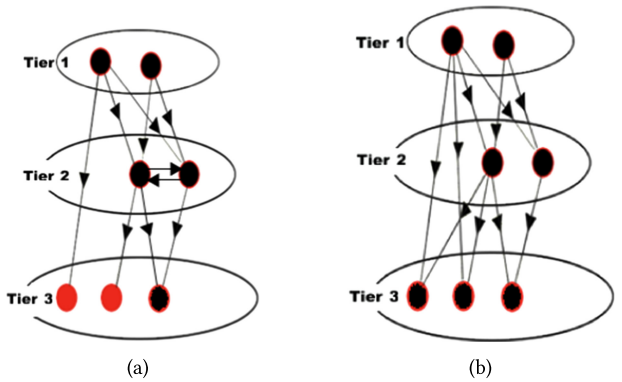


Fig. 16. Before edge rewiring. Nodes marked in red have in-degree 1 and are loosely connected, (b) after edge rewiring: All tier 2 and 3 nodes have in-degree 2.

a directed graph *mapped WSN*. The last sensor device 3 is mapped to TRN node a, because the latter interacts directly with b. The final *mapped WSN* topology is illustrated in Figure 15(c).

Experiments: The authors carried out simulation experiments on OMNET Castalia simulator [107] for the topology of *S. cerevisiae* TRN. The performance of bio-inspired (or mapped) WSNs were compared against k-connected and E-R random topologies. Bio-inspired WSNs not only preserved high number of FFL motifs (discussed in Section 4.1 to be responsible for robustness of TRNs), but also exhibited higher PDR and energy efficiency, and low packet loss due to interference, low latency.

5.1.2 Topological Enhancement by Edge Rewiring. Bradford discussed that the topological similarity between co-functional networks of genes and scale free topologies make them good fits for wireless sensor networks [88]. However, recall from our discussion in Section 4.1, TRNs, like scale free topologies, are vulnerable to failure of hubs. Roy et al. exploited the three-tier topology of TRN (introduced Section 3) to propose an edge rewiring mechanism to remedy this vulnerability, while preserving its other essential topological properties [33, 83]. Given original TRN (G_O), the proposed approach unfolds in two stages: (a) edge addition and (b) edge deletion.

Edge addition: This step ensures that every tier 2 and 3 node loosely connected to tier 1 and 2 nodes (marked in red in Figure 16(a)) connects with at least 2 nodes from tiers 1 and 2. This step is based on the preferential attachment growth model (see Section 3.2.7), where a directed edge (u, v) is added between a tier 1 or 2 node u (preferentially selected based on out-degree) and a tier 2 or 3

node v with degree less than 2. This process continues until all tier 2 and 3 nodes have in-degree at least 2 (i.e., *2-connectivity*). The TRN generated by edge addition is called augmented TRN G_A.

Edge deletion: In this step edges are removed from augmented TRN from G_A to ensure that the number of edges in original and rewired TRNs are the same, while the 2-connectivity property is intact in the final rewired TRN (Figure 16(b)). Edge deletion is formulated as a *nonlinear optimization problem* with the objective (Expression 22) of preserving the in-degree distribution of original TRN (G_O) by minimizing the squared error between the in-degree distribution of G_O (denoted by Δ_I^O) and G_M (denoted by Δ_I^M), with obtainability (discussed below) as one of the constraints,

$$\underset{\Delta_I^M(3:)}{argmin} \sum_{i=3}^{md} (f_i^M - f_i^O)^2, \tag{22}$$

$$s.t. \quad \sum_{i=0}^{md} f_i^M \times i = \sum_{i=0}^{md} f_i^O \times i, \tag{23}$$

$$f_2^A + \sum_{i=3}^{md} (f_i^A - f_i^M) = f_2^M, \tag{24}$$

$$f_i^M \leq f_i^A \qquad \forall i = 3, 4, \ldots, md. \tag{25}$$

- In objective function (expression 22) the indexing on the summation starts at 3, to ensure that all tier 2 and 3 nodes have in-degree of at least 2. This optimization returns $\Delta_I^M(3:) = f_3^M, f_4^M, \ldots, f_{md}^M$, which is later utilized to determine the exact number of edges with the lowest edge FFL centrality (see Section 4.1) to be removed off nodes with in-degree i.
- Given any directed graph $G(V, E)$, $|E| = \sum_{i=0}^{md} f_i \times i$, Constraint 23 ensures that the number of edges in original and rewired TRNs G_O and G_M are the same.
- Constraint 24 guarantees a property called *obtainability* introduced in this work, by virtue of which Δ_I^M returned by optimization is a valid in-degree distribution obtainable from G_A; Constraint 25 defines the bounds for the degree distribution curve of rewired TRNs.

Experiments: The authors attempt to preserve and enhance the topological robustness of *E. coli* and *S. cerevisiae* TRNs by retaining properties (like small world, scale free out-degree distribution, FFL motif abundance, and so on, discussed in Section 3) in rewired TRNs. To this end, the article presents *greedy* and *dynamic* edge deletion heuristics that follow the strategy to remove edges with the lowest FFL motif centrality. WSN topologies are realized by reversing the edge directions of rewired TRNs, such that the hubs nodes, possessing high out-degree (see Section 3.2.1), now have high in-degree and act as data sinks. The rewired TRN-based WSNs exhibit lower vulnerability to failure of hubs, high network efficiency and PDR as compared to original TRN- and E-R random graph-based WSNs.

5.1.3 Energy Balancing Node Scheduling. Byun et al. proposed a TRN-based mechanism for autonomous scheduling of sensor. The scheduling approach is based on the principle that TFs produce genes, which in turn dynamically regulate the production or suppression of TFs. The authors employ the following ordinary differential equation–based relationship (see Section 2.2.1) to show the above TF-gene interaction:

$$\frac{dg_i}{dt} = \lambda_g g_i + \alpha_g f(p_i), \tag{26}$$

$$\frac{dp_i}{dt} = \lambda_p p_i + \alpha_p g_i. \tag{27}$$

Here g_i is the expression level of gene i and p_i is the concentration of protein i. γ_g and γ_p are the decay rates of mRNA and protein concentration, respectively; α_g and α_p are the synthesis rates of mRNA and protein concentration, respectively, and $f(x)$ is a sigmoid function.

This work models the ON-OFF cycle of sensor devices on the basis of the principles of local TF-gene interaction (see Section 3.2.6). The scheduling is controlled by two parameters: (1) protein concentration P mapped to actuator variable, and (2) mRNA concentration M mapped to internal states. Specifically, each device produces M contingent upon its own P-level and diffuses M among its neighbor devices. Each neighbor device now regulates its P-level on the basis of M-levels of neighbors and a target P-level p^*. The instantaneous P-value determines the probability of the device operating in ON state. Simulation experiments performed on MATLAB show that this method attains adaptive, balanced energy consumption, while achieving application specific goals [89].

5.1.4 Sensor Network Deployment. Das et al. applied the dynamic interaction among the TFs/genes in TRNs in the deployment of sensor nodes, with the specific goal to maximize the coverage while minimizing energy overhead by reducing the number of active sensors [90]. This TRN-based solution to solve the sensor network coverage problem is a viable distributed solution alternative to the existing centralized multi-objective formulations such as NSGA-II [108] and FSGA [109]. Here, each sensor is modeled as a gene, which acquires active state if the corresponding gene expression value is high. The proposed TRN controller ensures that sensors covering larger sensing regions possess high expression values; also since any two adjacent sensors may have a high degree of overlap, the controller deactivates one of the sensors in the interest of energy efficiency.

5.1.5 Sink Selection Strategy. Kamapantula et al. argue that data transmission time in WSN relies greatly on the selection of sink nodes [101]. They compare two sink selection strategies: (i) a highest degree and (ii) FFL motif-centrality based (see Section 4.1) to show that both strategies pick the candidate as sink, because hub nodes, by virtue of their connectivity, participate in many FFLs.

5.2 Self-organizing Networks

Self-organizing networks (SONs) are networks where the entities coordinate with each other to form a system that adapts to achieve a goal more efficiently [110, 111]. In wireless network, self-organization may help achieve broad tasks (like sharing of processing and communication resources, adaptive behavior with information processing and dissemination), or specific tasks like target tracking and surveillance [112]. Analogous to genes that encode rules controlling the protein production, sensors can possess controllers that specify rules for activation of themselves as well as their neighbors. A key feature of SONs is the *emergence of new behaviors* as a consequence of self-organization.

5.2.1 Self Organized Sensing. El-Mawass et al. conceived a self-organized sensing system inspired from TRNs, capable of operating in a distributed manner without the need for manual configuration [91]. Each sensor node can have two distinct states *ON* and *OFF*. Its state is decided dynamically based on the environment and its ability to contribute to network functioning. A sensor node that is in *OFF* state wakes up intermittently to determine whether to stay *ON* or *OFF*. The authors represent a simplified controller as a genome (i.e., sequence of genes), where each gene is a system rule controlled by input and output protein signals generated by other genes.

Proteins: Modeled on TF-gene regulation (see Section 3.2.6), proteins serve as activation or inhibitory signals that regulate gene expression. Below is the functional role of the input and output proteins.

- *Sensory Proteins*: It indicates the overlap of transmission zone between a sensor and its neighbour and also, the total amount of energy that is remaining in the neighbour sensors.
- *Diffusion Protein*: It indicates the remaining energy in the sensor node.
- *Actioner Protein*: It determines whether the sensor should be ON or OFF. The sensor is kept ON only when its value of concentration exceeds a certain threshold.
- *Intern proteins*: It regulates the sensory-protein actuator relations.

Every gene has an output function that is a collective result of the input protein concentrations. Each sensor node has two associated properties: coverage range (within which the sensor is able to detect events) and radio transmission range. The system also takes into consideration the energy consumption for every sensor node due to sampling, processing and communication.

Experiments: The authors carry out simulation experiments on MATLAB to illustrate how the system achieves high energy efficiency by self-organization and mutual coordination among sensors. Initially, 50% nodes are active. The emergent behavior observed in this SON is as follow: over time, there is a collective drain of the energy in active sensors that soon become inactive; subsequently the inactive sensors are gradually activated to carry out the sensing activities.

5.2.2 Underwater Robot Controller. Taylor et al. [92] have utilized a TRN inspiration to design decentralized controllers for underwater robots (called *hydrons*), which would achieve tasks by mutual coordination. Additionally, the emergent property of this system is its robustness, self-repairing capability, adaptability and inclusion of the TRN properties in the autonomous controllers.

Hydron: Each hydron is a sphere with an impeller and nozzle for sucking and expelling water, syringe to vary the relative density of Hydron by altering the amount of water in the internal chamber and proteins for mutual regulation. There are interface sites on the surface of the Hydron, equipped with optical transmitter and receivers. Like in Reference [91], the controller for each hydron is modeled as a genome (for encoding information about genes) and a cytoplasm (possessing proteins located at interface sites). Each gene, when expressed, produces a specific type of protein.

Genome and Protein: The authors define a sequence of genes that encode genetic information; genes, when expressed, produce proteins, which may control the expression of other genes and serve as interfaces between the hydrons and the environment through diffusion.

Experiments: The article uses genetic algorithm to evolve the TRN controllers and guide them to accomplish a task of forming groups in coordination. The results show that the robots achieve high fitness in group formation, evaluated as negative of mean square distance from the group centroid.

5.2.3 Target Tracking Application. Markham and Trigoni [93] proposed a novel scalable and network of automated, self-organizing sensors. Their work is based on the premise that a sensor, like a living cell, maintains its own protein level (assumed to be discrete). Again, the proteins serve as means for inter-node communication. The nodes adjust their sampling rates, based on local parameters and information from neighbour nodes. The authors applied this system in the design of target tracking application, where each sensor node monitors the location of the target in its proximity and autonomously decides, based on certain user specified constraints, whether to track the target. Jin et al. presented a similar two-step multi-robot target tracking system [113]. For

each device, proteins represent internal parameters regulated by neighbor devices or the presence of target in the vicinity, guiding them to the target on a path free from obstacle.

5.2.4 Mobile Sensor Networks. Meng et al. designed self-organizing mobile sensor networks using transcriptional networks [102]. Unlike prior studies attempting to retain TRN topologies, Meng showed that evolutionary algorithms can be leveraged to organize the nodes in a manner that motif structures are preserved and the networks are demonstrably robust to environmental adversities.

5.3 Delay Tolerant Networks

Delay Tolerant Networks (DTNs) are networks where connectivity is intermittent and not end-to-end, resulting in higher communication delays [114]. We consider a class of DTNs, termed DRNs, consisting of smart handheld devices of survivors and rescue workers in the absence of a primary infrastructure in the aftermath of a natural disaster (see Section 1). The information flowing among the survivor nodes and coordination center in the setting of a disaster is context specific, failure prone and time sensitive [4, 115]. Thus, DRNs represent a class of DTNs that offer low delay as well as uninterrupted service despite component failures. Shah et al. conceived a TRN-inspired DRN, **bio-inspired disaster response network (Bio-DRN)**, that exhibits (1) high network lifetime (i.e., energy-efficient) and (2) robustness against component failures for seamless communication [84, 85].

Inspiration: Given any input DRN topology comprising survivors, responders and CCs, the authors formulate the Bio-DRN construction as an Integer Linear Programming optimization problem, with the objective to find a common subgraph of the DRN and TRN topologies, while preserving the graph properties of TRN, particularly *low graph density* (see Section 3.2.2) and *FFL motif abundance* (see Section 3.2.4). The energy consumption of a certain node in DTNs has been shown to be quantifiable by the number of forwarding links it shares in the network [85, 116]; hence low graph density ensures fewer communication links, which translates into fewer message replications and forwarding resulting in *energy-efficient* DRN. Also, high FFL motif abundance renders alternative communication pathways, thereby improving the *robustness* against node failures (see Section 4.1).

Approach: Shah et al. [84, 85] showed that the *Bio-DRN* construction problem is NP-Hard and proposed a heuristic that exploits the topological similarity between TRN and DRN. Specifically, analogous to the three-tier topological characterization of TRN (see Section 3), DRN nodes can be classified into three tiers based on functional roles: *tier 1* contains the unique CC; *tier 2* comprises the set of points of interest and volunteers; and *tier 3* contains the set of survivors (Figure 17(a) and (b)). The proposed heuristic generates a subgraph of TRN with the maximum FFL motifs, called *reference TRN*, that subsequently acts as a template for Bio-DRN construction. The proposed heuristic employs Blondel's node similarity [117] to perform one-to-one mapping between similar nodes of same tiers in reference TRN and DRN; the resultant subgraph of DRN is termed *Bio-DRN*.

Experiments: Experiments on a map of a real-disaster prone region, Bhaktapur, Nepal shows that *Bio-DRN* retains the topological properties of TRN, exhibiting a steady tradeoff between energy efficiency and network robustness against node failures, while ensuring high PDR and low latency.

5.4 IoT-Net

An IoT-Net consists of smart devices, deployed over urban spaces in smart cities, possessing the ability to sense events, communicate, process and store event data. Like in case of WSN, an

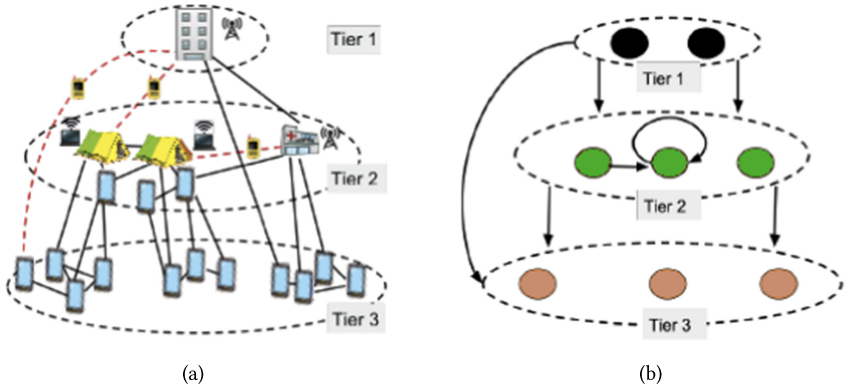


Fig. 17. Bio-DRN mapping heuristic. (a) DRN three-tier structure based on the functional role. (b) TRN three tier structure (redrawn from Reference [84]).

IoT-Net can be represented as a graph topology where a link exists between two devices if they are within the (wireless) communication range of one another. Like most communication networks, IoT-Nets demand uninterrupted sensing and data dissemination over long durations [118]. Given that IoT devices are energy constrained, Roy-Ghosh et al. proposed a TRN-based data collection framework, called *bioSmartSense*, that enables energy-efficient and qNq-aware data sensing and reporting for sustainable IoT-based smart city applications [87].

Inspiration: The authors draw a parallel between the *energy level* of IoT devices (which controls its data sampling rate) and the *expression level* of genes that dictates the quantity of proteins produced to meet cellular requirements and enable sustainability of organism [11, 119]. Similar to the TF-gene regulation, the article proposes that IoT devices shall regulate self and neighbor device *energy levels*. (Note that the self-regulation of a device's energy level is analogous to self-regulation motif as discussed in Section 3.2.4). Specifically, we know from Section 3.2.6 that the TF-gene interactions are directed edges that have positive or negative signs (see Figure 6 (left)). *bioSmartSense* mimics the signed TF-gene interaction TRN, and positive or negative directed link (u, v) implies that an IoT device u will increase or decrease the energy level of its neighbor device v. Higher energy level of a device guarantees higher higher sampling rate (and higher data accuracy) at the expense of energy.

Approach: bioSmartSense transfers the edge signs of TRN to IoT-Net by utilizing the TRN-based mapping (see Section 5.1.1). Subsequently, IoT devices conserve energy by regulating energy levels of one another based on TF-gene regulation. Furthermore, keeping in mind that several devices may sense the same event at a given time, *bioSmartSense* attempts to conserve energy by restricting redundancy while reporting event data to a centralized application platform called the base station. To this end, it generates a graph transformation of IoT-Net where two nodes share an undirected edge if they have sensed at least one common event over a prespecified time interval. From the transformed graph, redundancy in event reporting is minimized through a heuristic for maximum weighted independent set [120] that selects a subset of nodes exhibiting the highest *qNq* metric (see section on *performance measures*) and having no overlap in events sensed. This approach is extensible to incorporate heterogeneous device varying in sensing capacities and energy consumption rates, signal attenuation with distance, and event priority levels [95].

Experiments: Extensive simulation performed on customized simulator (implemented on Python SimPy library [121]) show that *bioSmartSense* framework exhibits higher energy efficiency

and qNq as compared with those of a state-of-the-art data collection approach for smart city applications.

5.5 Edge-centric IoT Platform

We discuss in Section 1 that edge-IoT networking is a result of the increasing computational capabilities at the edge [122]. This paradigm, characterized by decentralized cloud, more processing at the edge and reduced waiting time, is challenged by issues of reliable connectivity and energy [123]. It is clear that edge devices are still not as memory, processing and energy-efficient as the centralized count. In a traditional **mobile crowdsensing (MCS)** setting, mobile users in possession of handheld devices sense task of different events (of traffic, environment, etc.) and forward data via energy-constrained fog nodes to the base station in a multihop fashion. Roy et al. conceive a scalable TRN-based collaborative data transfer framework over fog computing platforms [86] that utilizes *agglomerative hierarchical clustering* [124] to identify regions of densely connected fog devices. It applies the TRN-based mapping strategy (discussed in Section 5.1.1) to each cluster to design sparse yet robust backbone fog topologies. *bioMCS* dynamically identifies fog nodes of high fitness as gateways (1) device residual energy and (2) connectivity and fit mobile nodes as owners of mobile user groups on the basis of (1) residual energy and (2) promptness in carrying out sensing and reporting tasks. Finally, the sensed event data is transferred by the mobile peers to base station via mobile group owners, fog and gateway fog nodes. *bioMCS* exhibits higher energy efficiency, robustness against fog node failure, task data load balancing across fog nodes over traditional MCS platforms.

Comparison, drawbacks and overheads. Approaches for the generation of sparse yet robust WSNs by employing mapping with TRN topologies [33, 83, 96–100] are centralized strategies, where the network administrator must have the global view of the WSN topology; moreover, they assume that the sensor nodes are static. Similarly, the design of *Bio-DRN* is based on the assumption that the input DRN topology largely remains steady over time [84]. The CC, the most well-connected entity in the original DRN topology, observes the topology over time and invokes the Bio-DRN topology construction algorithm. Knowledge of the entire topology may be a strong assumption and limitation in large dynamic network scenarios. These approaches outperform *k-connected* and E-R random topology-based networks in terms of PDR, latency and energy efficiency, under conditions of node and link failure.

Let us now turn to the semi-centralized strategies: (1) data collection in fog computing platform (*bioMCS*) [86]—where the urban space is partitioned into dense clusters of fog devices and the TRN-based mapping is employed on each cluster by time-varying cluster heads called gateway fogs and (2) IoT-based data collection framework (called *bioSmartSense*) for smart cities [87, 95], where the sensors regulate their internal parameters based on the messages received from the neighbor devices. These approaches still rely on TRN mapping to construct the network backbone, but allow the devices to regulate internal device parameters (such as energy levels) by communicating with neighbors. These approaches exhibit comparable data delivery rate as state-of-the-art smart city application [125] and a significant improvement in energy efficiency. In fact, they are similar to ant colony optimization (discussed in Section 1), where ants gradually learn the efficient routes to the food source. However, unlike the explore and exploit strategy of ant colony, they avoid performance uncertainty by leveraging the reliable optimized protein-gene connectivity of TRN.

Contrast the above techniques with the bio-inspired self-organized target tracking application [93] or sensor energy state scheduling [89]. Similarly, References [92, 113] follow a decentralized approach of modulating system parameters based on feedback from neighbor devices and location of target being tracked. Despite being distributed, these approaches entail energy through

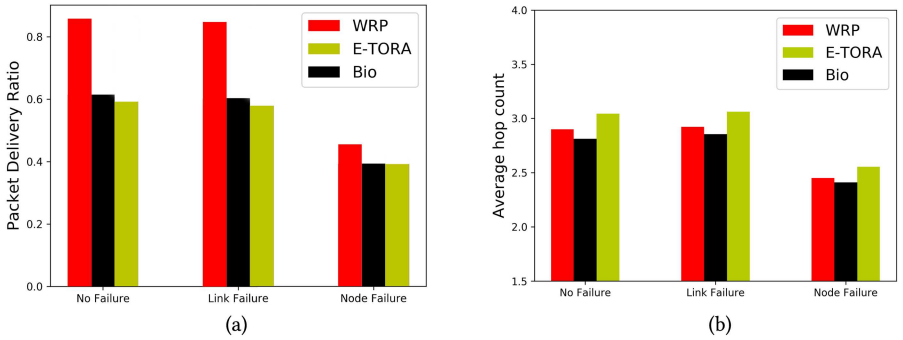


Fig. 18. Comparison of performance of WRP, E-TORA, and role *A* motif centrality-based routing.

the exchange of control messages for neighbor device regulation. This regulation strategy, inspired by the protein-gene regulation in TRN, is analogous to the triggering of an immune response by detector nodes to achieve event sensing and reporting in the artificial immune system-based computing application (mentioned in Section 1). Overall, the exchange of additional control messages help conserve overall network energy by eliminating redundancy in reported events as shown in Reference [87].

6 FUTURE DIRECTIONS

In this section, we go over the unexplored areas of TRNs that can motivate new research directions in the fields of bio-inspired network protocols, communication and social network analysis.

6.1 Bio-inspired Routing Protocol

In Section 5.1, we discuss that TRNs serve as an effective template for the design of static WSN topologies. TRN-based WSNs achieve topological robustness against node failures by maximizing FFL motifs. Interestingly, our recent findings reveal that FFL motif central nodes (i.e., nodes with high participation of FFL motifs) are the most effective information forwarders in TRNs [126]. Based on these findings, we plan to investigate the possibility of a dynamic bio-inspired routing protocols in WSN settings using *node motif centrality* (refer Section 4.1) as a metric for route selection. Since in any given FFL, role *A* motif centrality creates two independent paths between the general TF and target gene, we hypothesize that nodes with high role *A* FFL motif centrality will potentially present several pathways between any source and sink node pairs in WSNs.

To verify our intuition, we devise a simple *bio-inspired routing strategy* (Bio) wherein each node maintains information of the role *A* FFL motif centrality of the neighbor nodes and chooses the node with highest role *A* motif centrality as its next hop. We compare this strategy with two standard routing protocols, namely, (1) **Wireless Routing Protocol (WRP)** [127], which utilizes shortest path-based schemes to calculate the minimum cost routes and (2) **Energy-aware Temporally Ordered Routing Algorithm (E-TORA)** [128], which conserves energy by taking into consideration the level of power of each node and avoids using nodes with low residual energy. Our simulation experiment on a WSN of 50 nodes designed on a discrete event simulation environment of Python SimPy library [121] reveals that the PDR for WRP is the highest, followed by Bio. This is because WRP follows the optimal path from source to sink (Figure 18(a)); however, with respect to the average delay in packet transfer, Bio greatly outperforms other strategies in all three node failure conditions as it exhibits minimum hop count between source and sink (Figure 18(b)). Initial experiments showing this tradeoff (between PDR and delay) offered by the bio-inspired routing strategy motivate the design of *novel dynamic routing strategies*, particularly

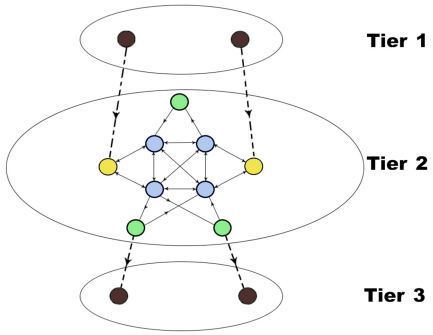


Fig. 19. Schematic representation of organization of motif central nodes within tier 2 of a TRN.

useful in wireless network operating under channel uncertainty due to interference, congestion and dynamic topology. It is possible to design a unified routing strategy—a combination of WRP, E-TORA and Bio—where weights can be tweaked to meet changing requirements, as seen in software defined networks [129]. For instance, if a low data delivery delay is preferred, role *A* nodes can be preferred as next hops; conversely, the weight for WRP can be increased to meet data delivery needs.

6.2 Hub and Spoke Architecture

The three-tier topology (see Section 3), which we have utilized to characterize the graph attributes of TRN, can also help identify significant patterns in the organization among high NMC nodes. As an experiment, we define high NMC TRN nodes as nodes possessing FFL motif centrality $\delta > 100$. In Figure 19, we show a schematic where a few high NMC nodes in tier 2 (marked in blue) form cliques among themselves while the other high NMC nodes (shown in green) are connected to some (but not all) of the blue nodes. Note that both green and blue nodes are connected through bidirectional edges leading to full duplex dataflow. There exists a third type of node (shown in yellow) that serve as intermediaries for information flow between the blue nodes.

This arrangement among the high NMC tier 2 nodes is similar to a *hub-and-spoke architecture*, where tier 2 NMC nodes form *motif hubs* (not to be confused with degree hubs mentioned in Section 3.2.1) with majority of the tier 1 and 3 nodes being directly connected to high NMC nodes in tier 2 [126]. We intuit that in such an architecture, role *A* motif central nodes act as the information spreaders, while the green nodes provide fault tolerance against failures of the blue nodes; our investigation reveals that the yellow nodes, possessing role *B* motif centrality (see Section 4.1), provide edge level fault tolerance by activating the indirect path of the FFL when the direct path is congested or error prone. We believe that the organization of high NMC nodes as well as the dynamic switching of traffic between role A and B nodes to achieve varying network goals can explain the robustness of TRNs and further motivate the design of fault-tolerant communication topologies.

6.3 Balanced and Unbalanced Triads

In Section 3.2.4, we discussed coherent and incoherent FFLs on the basis of the signs on directed edges that can cause acceleration or delay in information flow in TRN. It is noteworthy that this coherence or incoherence resembles the idea of balanced and unbalanced triads in social networks. *A triad is defined as a triplet of nodes connected by two edges (open triad) or three edges (closed triad).* In a signed network where positive and negative edges signify friendship (or trust) and

enmity (or distrust) between a node pair, three edges in a balanced triad works on the following principles: *the friend of my friend is my friend*, *the friend of my enemy is my enemy*, *the enemy of my friend is my enemy* and *the enemy of my enemy is my friend*. Leskovec et al. studied the signed interaction among the entities in social network to develop a theory that explains observed edge signs and the underlying social mechanisms [130]. We intuit that the use of influence diffusion mechanisms [131, 132] can help unravel specific influencial FFL motifs capable of amplifying or dampening spread of information in signed social and biological network topologies.

6.4 Design of Smart Topologies

The field of bio-inspired networking hinges upon the idea that biological networks are intrinsically capable of optimizing several disparate network goals such as seamless information flow, low delay, near-optimal resource utilization, and so on, in a dynamic noise- and error-prone environment. In Section 4.2, we elucidate that TRNs have been proven to be naturally optimized for average shortest path between TFs and genes. We intuit that exploring the average shortest path apropos to FFL motif centrality may yield new insights into the topological robustness of TRNs. This is because not only are FFL motifs closely linked to TRN robustness (as shown in Section 4.1), but Gorochowski et al. showed that the clustering tendency of FFL motifs, gauged using metrics such as motif clustering coefficient (refer Section 4.4) and motif clustering diversity (refer Section 4.5), are key to their information-theoretic and functional roles in TRNs [80]. Taking a cue from [80], network analysis on how smaller motifs such as feed forward loops organize themselves to form higher-order motifs can shed light on the exact role of motifs in achieving efficient information dissemination. Furthermore, one can design a joint optimization problem (similar to the formulation explained in Section 4.2) on several weighted versions of TRN subgraphs, by assigning a weight on each edge that equals its reciprocal FFL edge motif centrality. If this modified TRN exhibit a still lower δ than its randomized and original TRN counterpart, then it stands to reason that smart network topologies can be realized by pushing the bulk of the data packets along the high FFL motif central nodes and links.

7 CONCLUSION

In this article, we surveyed the structure and topology of the TRNs in light of its myriad graph properties such as *scale free out-degree distribution*, *low graph density*, *small world property*, *clustering tendency and motif abundance*, *TF-gene regulation*, *preferential attachment*, *robustness against random node failure*, and so on. This study is substantiated with some experimental results on human TRN. We then delineated the network science-based metrics, such as motif (1) centrality, (2) clustering coefficient, and (3) clustering diversity, protein-gene path length and network efficiency, that are either derived from or help explain the above graph properties. We mathematically demonstrate how these metrics enable TRN topologies in achieving optimized communication between the proteins and genes with low delay despite component failures or perturbation to gene coding sequences. Finally, we perform a comprehensive study of the applications of TRNs in the design of smart networking solutions in the areas of WSN, DTN, self-organizing networks, IoT-Nets and edge computing platforms. We concluded the article with a vision of the specific unexplored facets of TRNs that may further inspire new directions in the design of tangible social and communication network architectures and protocols, such as unified routing mechanisms capable of adapting to varying data delivery, delay, and energy requirements.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for insightful and constructive comments that helped improve the quality of the manuscript.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. 2013. Internet of things (IoT): A vision, architectural elements, and future directions. *Fut. Gener. Comput. Syst.* 29, 7 (2013), 1645–1660.

[2] W. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed. 2019. Edge computing: A survey. *Fut. Gener. Comput. Syst.* 97 (2019), 219–235.

[3] J. Yick, B. Mukherjee, and D. Ghosal. 2008. Wireless sensor network survey. *Comput. Netw.* 52, 12 (2008), 2292–2330.

[4] V. K. Shah, S. Roy, S. Silvestri, and S. K. Das. 2017. Ctr: Cluster based topological routing for disaster response networks. In *Proceedings of the 2017 IEEE International Conference on Communications (ICC'17)*. IEEE, 1–6.

[5] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo. 2017. Design of secure user authenticated key management protocol for generic iot networks. *IEEE IoT J.* 5, 1 (2017), 269–282.

[6] J. Barriga, J. Sulca, J. León, A. Ulloa, D. Portero, R. Andrade, and S. Yoo. 2019. Smart Parking: A literature review from the technological perspective. *Appl. Sci.* 9, 21 (2019), 4569.

[7] F. Dressler and O. B. Akan. 2010. A survey on bio-inspired networking. *Comput. Netw.* 54, 6 (2010), 881–900.

[8] C. Jones, K. Sivalingam, P. Agrawal, and J. Chen. 2001. A survey of energy efficient network protocols for wireless networks. *Wireless Netw.* 7, 4 (2001), 343–358.

[9] U. Aickelin, D. Dasgupta, and F. Gu. 2014. Artificial immune systems. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, Edmund K. Burke and Graham Kendall (Eds.).

[10] A. Fukushima, S. Kanaya, and K. Nishida. 2014. Integrated network analysis and effective tools in plant systems biology. *Front. Plant Sci.* 5 (2014), 598.

[11] H. Hernández, C. Blum, and G. Francès. 2008. Ant colony optimization for energy-efficient broadcasting in ad-hoc networks. In *Proceedings of the International Conference on Ant Colony Optimization and Swarm Intelligence*. Springer, 25–36.

[12] F. Dressler, I. Dietrich, R. German, and B. Krüger. 2009. A rule-based system for programming self-organized sensor and actor networks. *Comput. Netw.* 53, 10 (2009), 1737–1750.

[13] V. Sevim and P. A. Rikvold. 2008. Chaotic gene regulatory networks can be robust against mutations and noise. *J. Theor. Biol.* 253, 2 (2008), 323–332.

[14] N. Noman et al. 2015. Evolving robust gene regulatory networks. *PLoS ONE* 10, 1 (2015), e0116258.

[15] Y. Du, J. Gong, Z. Wang, and N. Xu. 2018. A distributed energy-balanced topology control algorithm based on a noncooperative game for wireless sensor networks. *Sensors* 18, 12 (2018), 4454.

[16] J. Sheu, S. Tu, and C. Hsu. 2008. Location-free topology control protocol in wireless ad hoc networks. *Comput. Commun.* 31, 14 (2008), 3410–3419.

[17] S. Aluru. 2005. *Handbook of Computational Molecular Biology*. Chapman & Hall/CRC.

[18] G. Karlebach and R. Shamir. 2008. Modelling and analysis of gene regulatory networks. *Nat. Rev. Molec. Cell Biol.* 9, 10 (2008), 770–780.

[19] H. De Jong. 2002. Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.* 9, 1 (2002), 67–103.

[20] P. Ghosh, M. Mayo, V. Chaitankar, T. Habib, E. Perkins, and S. K. Das. 2011. Principles of genomic robustness inspire fault-tolerant WSN topologies: A network science based case study. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops'11)*. IEEE, 160–165.

[21] Y. Xiao. 2009. A tutorial on analysis and simulation of boolean gene regulatory network models. *Curr. Genom.* 10, 7 (2009), 511–525.

[22] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. 2002. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18, 2 (2002), 261–274.

[23] Z. Ghahramani. 1997. Learning dynamic Bayesian networks. In *International School on Neural Networks, Initiated by IIASS and EMFCSC*. Springer, 168–197.

[24] B. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alche Buc. 2003. Gene networks inference using dynamic Bayesian networks. *Bioinformatics* 19, suppl. 2 (2003), ii138–ii148.

[25] V. Mihajlovic and M. Petkovic. 2001. Dynamic bayesian networks: A state of the art.

[26] A. V. Werhli and D. Husmeier. 2007. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat. Appl. Genet. Molec. Biol.* 6, 1 (2007).

[27] M. Newman. 2003. The structure and function of complex networks. *SIAM Rev.* 45, 2 (2003), 167–256.

[28] Y. Fu, L. R. Jarboe, and J. A. Dickerson. 2011. Reconstructing genome-wide regulatory network of E. coli using transcriptome data and predicted transcription factor activities. *BMC Bioinform.* 12, 1 (2011), 233.

[29] I. Pournara and L. Wernisch. 2007. Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC Bioinform.* 8, 1 (2007), 61.

[30] T. Schaffter, D. Marbach, and D. Floreano. 2011. GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* 27, 16 (2011), 2263–2270.

[31] H. Han, H. Shim, D. Shin, J. E. Shim, Y. Ko, J. Shin, H. Kim, A. Cho, E. Kim, T. Lee, et al. 2015. TRRUST: A reference database of human transcriptional regulatory interactions. *Sci. Rep.* 5 (2015), 11432.
[32] H. Han et al. 2017. TRRUST v2: An expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic Acids Res.* 46, D1 (2017), D380–D386.
[33] S. Roy, V. K. Shah, and S.K. Das. 2016. Characterization of E. coli gene regulatory network and its topological enhancement by edge rewiring. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*. 391–398.
[34] S. Roy, V. K. Shah, and S. K. Das. 2019. Design of robust and efficient topology using enhanced gene regulatory networks. *IEEE Trans. Molec. Biol. Multi-Scale Commun.* (2019).
[35] M. B. Gerstein, A. Kundaje, M. Hariharan, S. G. Landt, K. Yan, C. Cheng, X. J. Mu, E. Khurana, J. Rozowsky, R. Alexander, et al. 2012. Architecture of the human regulatory network derived from ENCODE data. *Nature* 489, 7414 (2012), 91.
[36] N. Bhardwaj, P. M. Kim, and M. B. Gerstein. 2010. Rewiring of transcriptional regulatory networks: Hierarchy, rather than connectivity, better reflects the importance of regulators. *Sci. Signal.* 3, 146 (2010), ra79–ra79.
[37] H. Ma, J. Buer, and A. Zeng. 2004. Hierarchical structure and modules in the Escherichia coli transcriptional regulatory network revealed by a new top-down approach. *BMC Bioinform.* 5, 1 (2004), 199.
[38] A. Barabási. 2009. Scale-free networks: A decade and beyond. *Science* 325, 5939 (2009), 412–413.
[39] R. Albert. 2005. Scale-free networks in cell biology. *J. Cell Sci.* 118, 21 (2005), 4947–4957.
[40] B. Kamapantula, A. Abdelzaher, M. Mayo, E. Perkins, S. Das, and P. Ghosh. 2017. Quantifying robustness in biological networks using NS-2. In *Modeling, Methodologies and Tools for Molecular and Nano-scale Communications*. Springer, 273–290.
[41] R. D. Leclerc. 2008. Survival of the sparsest: Robust gene networks are parsimonious. *Molec. Syst. Biol.* 4, 1 (2008), 213.
[42] Q. K. Telesford, K. E. Joyce, S. Hayasaka, J. H. Burdette, and P. J. Laurienti. 2011. The ubiquity of small-world networks. *Brain Connect.* 1, 5 (2011), 367–375.
[43] L. Gu, H. L. Huang, and X. D. Zhang. 2013. The clustering coefficient and the diameter of small-world networks. *Acta Math. Sin.* 29, 1 (2013), 199–208.
[44] Luıs A. Nunes Amaral, Antonio Scala, Marc Barthelemy, and H. Eugene Stanley. 2000. Classes of small-world networks. *Proc. Natl. Acad. Sci. U.S.A.* 97, 21 (2000), 11149–11152.
[45] T. Guo. 2014. *Design of Genetic Regulatory Networks*. Masters Thesis. Industrial Engineering, University of Illinois at Urbana-Champaign.
[46] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. Network motifs: simple building blocks of complex networks. *Science* 298, 5594 (2002), 824–827.
[47] E. Wong, B. Baur, S. Quader, and C. Huang. 2011. Biological network motif detection: Principles and practice. *Brief. Bioinform.* 13, 2 (2011), 202–215.
[48] U. Alon. 2007. Network motifs: Theory and experimental approaches. *Nat. Rev. Genet.* 8, 6 (2007), 450.
[49] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. 2002. Network motifs in the transcriptional regulation network of Escherichia coli. *Nat. Genet.* 31, 1 (2002), 64.
[50] S. Mangan and U. Alon. 2003. Structure and function of the feed-forward loop network motif. *Proc. Natl. Acad. Sci. U.S.A.* 100, 21 (2003), 11980–11985.
[51] E. Estrada. 2012. *The Structure of Complex Networks: Theory and Applications*. Oxford University Press.
[52] R. J. Prill, P. A. Iglesias, and A. Levchenko. 2005. Dynamic properties of network motifs contribute to biological network organization. *PLoS Biol.* 3, 11 (2005), e343.
[53] T. I. Lee, N. J Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, et al. 2002. Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science* 298, 5594 (2002), 799–804.
[54] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. 2004. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics* 20, 11 (2004), 1746–1758.
[55] F. Schreiber and H. Schwöbbermeyer. 2005. MAVisto: A tool for the exploration of network motifs. *Bioinformatics* 21, 17 (2005), 3572–3574.
[56] S. Wernicke and F. Rasche. 2006. FANMOD: A tool for fast network motif detection. *Bioinformatics* 22, 9 (2006), 1152–1153.
[57] B. D. McKay et al. *Practical Graph Isomorphism*.
[58] M. Zuba. 2009. A comparative study of network motif detection tools. UConn Bio-Grid, REU Summer.
[59] M. E. Wall, M. J. Dunlop, and W. S. Hlavacek. 2005. Multiple functions of a feed-forward-loop gene circuit. *J. Molec. Biol.* 349, 3 (2005), 501–514.
[60] E. Alm and A. P. Arkin. 2003. Biological networks. *Curr. Opin. Struct. Biol.* 13, 2 (2003), 193–202.

[61] M. Madan Babu, Nicholas M. Luscombe, L. Aravind, Mark Gerstein, and Sarah A. Teichmann. 2004. Structure and evolution of transcriptional regulatory networks. *Curr. Opin. Struct. Biol.* 14, 3 (2004), 283–291.
[62] O. Shoval and U. Alon. 2010. SnapShot: Network motifs. *Cell* 143, 2 (2010), 326–326.
[63] A. Martínez-Antonio, S. C. Janga, and D. Thieffry. 2008. Functional organisation of Escherichia coli transcriptional regulatory network. *J. Molec. Biol.* 381, 1 (2008), 238–247.
[64] R. Pinho, V. Garcia, M. Irimia, and M. W. Feldman. 2014. Stability depends on positive autoregulation in boolean gene regulatory networks. *PLoS Comput. Biol.* 10, 11 (2014), e1003916.
[65] J. Gómez-Gardeñes and Y. Moreno. 2006. From scale-free to Erdos-Rényi networks. *Phys. Rev. E* 73, 5 (2006), 056124.
[66] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. 2006. Complex networks: Structure and dynamics. *Phys. Rep.* 424, 4–5 (2006), 175–308.
[67] M. Mayo, A. Abdelzaher, E. J. Perkins, and P. Ghosh. 2012. Motif participation by genes in E. coli transcriptional networks. *Front. Physiol.* 3 (2012), 357.
[68] F. M. Camas and J. F. Poyatos. 2008. What determines the assembly of transcriptional network motifs in Escherichia coli? *PLoS One* 3, 11 (2008), e3657.
[69] A. F. Abdelzaher, A. F. Al-Musawi, P. Ghosh, M. L. Mayo, and E. J. Perkins. 2015. Transcriptional network growing models using Motif-Based preferential attachment. *Front. Bioeng. Biotechnol.* 3 (2015), 157.
[70] M. Aldana, E. Balleza, S. Kauffman, and O. Resendiz. 2007. Robustness and evolvability in genetic regulatory networks. *J. Theor. Biol.* 245, 3 (2007), 433–448.
[71] S. Mizutaka and K. Yakubo. 2013. Structural robustness of scale-free networks against overload failures. *Phys. Rev. E* 88, 1 (2013), 012803.
[72] G. Paul, T. Tanizawa, S. Havlin, and H. E. Stanley. 2004. Optimization of robustness of complex networks. *Eur. Phys. J. B* 38, 2 (2004), 187–191.
[73] C. M. Schneider, A. A Moreira, J. Andrade, S. Havlin, and H. J. Herrmann. 2011. Mitigation of malicious attacks on networks. *Proc. Natl Acad. Sci. U.S.A.* 108, 10 (2011), 3838–3841.
[74] H. Chan, L. Akoglu, and H. Tong. 2014. Make it or break it: Manipulating robustness in large networks. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 325–333.
[75] D. Koschützki, H. Schwöbbermeyer, and F. Schreiber. 2007. Ranking of network elements based on functional substructures. *J. Theor. Biol.* 248, 3 (2007), 471–479.
[76] S. Roy, M. Raj, P. Ghosh, and S. K. Das. 2017. Role of motifs in topological robustness of gene regulatory networks. In *Proceedings of the 2017 IEEE International Conference on Communications (ICC'17)*. IEEE, 1–6.
[77] M. Newman. 2010. *Networks: An Introduction*. Oxford University Press.
[78] A. F. Abdelzaher. 2016. *Identifying Parameters for Robust Network Growth using Attachment Kernels: A Case Study on Directed and Undirected Networks*. Ph.D. Dissertation. Virginia Commonwealth University.
[79] A. Abdelzaher, T. Dinh, M. Mayo, and P. Ghosh. 2016. Optimal topology of gene-regulatory networks: Role of the average shortest path. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*. ICST, 241–244.
[80] T. E. Gorochowski, C. S. Grierson, and M. di Bernardo. 2018. Organization of feed-forward loop motifs reveals architectural principles in natural and engineered networks. *Sci. Adv.* 4, 3 (2018), eaap9751.
[81] J. Goni, A. Avena-Koenigsberger, N. V. de Mendizabal, M. P van den Heuvel, R. F. Betzel, and O. Sporns. 2013. Exploring the morphospace of communication efficiency in complex networks. *PLoS One* 8, 3 (2013), e58070.
[82] S. Camazine. 2006. Self-Organizing Systems. *Encyclopedia of Cognitive Science* (2006).
[83] S. Roy, V. K Shah, and S. K. Das. 2018. Design of robust and efficient topology using enhanced gene regulatory networks. *IEEE Trans. Molec. Biol. Multi-Scale Commun.* 4, 2 (2018), 73–87.
[84] V. K. Shah, S. Roy, S. Silvestri, and S. K. Das. 2019. Bio-DRN: Robust and energy-efficient bio-inspired disaster response networks. In *Proceedings of the 16th IEEE International Conference on Mobile Ad-Hoc and Smart Systems (MASS'19)*.
[85] V. K. Shah, S. Roy, S. Silvestri, and S. K. Das. 2019. Towards energy-efficient and robust disaster response networks. In *Proceedings of the ACM Conference on Distributed Computing and Networking*. 397–400.
[86] S. Roy, N. Ghosh, P. Ghosh, and S. K. Das. 2020. bioMCS: A bio-inspired collaborative data transfer framework over fog computing platforms in mobile crowdsensing. In *Proceedings of the 21st International Conference on Distributed Computing and Networking*.
[87] S. Roy, N. Ghosh, and S. K. Das. 2019. biosmartsense: A bio-inspired data collection framework for energy-efficient, qoi-aware smart city applications. In *Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom'19)*. IEEE, 1–10.
[88] P. Bradford. 2018. Preliminary exploration of co-functional gene networks for wireless sensor networks. In *Proceedings of the IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC'18)*. IEEE, 1005–1008.
[89] H. Byun and S. Yang. 2017. Energy-balancing node scheduling inspired by gene regulatory networks for wireless sensor networks. *IET Commun.* 11, 17 (2017), 2650–2659.

[90] S. Das, P. Koduru, X. Cai, S. Welch, and V. Sarangan. 2008. The gene regulatory network: An application to optimal coverage in sensor networks. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. ACM, 1461–1468.

[91] N. El-Mawass, N. Chendeb, and N. Agoulmine. 2014. Robust self-organized wireless sensor network: A gene regulatory network bio-inspired approach. In *Genetic and Evolutionary Computing*. Springer, 105–114.

[92] T. Taylor. 2004. A genetic regulatory network-inspired real-time controller for a group of underwater robots. In *Intelligent Autonomous Systems, Vol. 8*. 403–412.

[93] A. Markham and N. Trigoni. 2010. Discrete Gene Regulatory Networks (dGRNs): A novel approach to configuring sensor networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'10)*. IEEE, 1–9.

[94] S. Roy and S. K. Das. 2019. A bio-inspired approach to design robust and energy-efficient communication network topologies. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'19)*. IEEE, 449–450.

[95] S. Roy, N. Ghosh, and S. K. Das. 2020. bioSmartSense+: A bio-inspired probabilistic data collection framework for priority-based event reporting in IoT environments. *Perv. Mobile Comput.* 67 (2020), 101179.

[96] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. 2015. Exploiting gene regulatory networks for robust wireless sensor networking. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'15)*. IEEE, 1–7.

[97] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. 2016. Efficient communications in wireless sensor networks based on biological robustness. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS'16)*. IEEE, 161–168.

[98] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. 2013. Robust deployment of wireless sensor networks using gene regulatory networks. In *Proceedings of the International Conference on Distributed Computing and Networking (ICDCN'13)*. Springer, 192–207.

[99] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. 2014. Deployment of robust wireless sensor networks using gene regulatory networks: An isomorphism-based approach. *Perv. Mobile Comput.* 13 (2014), 246–257.

[100] B. K. Kamapantula, A. Abdelzaher, P. Ghosh, M. Mayo, E. J. Perkins, and S. K. Das. 2014. Leveraging the robustness of genetic networks: A case study on bio-inspired wireless sensor network topologies. *J. Ambient Intell. Human. Comput.* 5, 3 (2014), 323–339.

[101] B. K. Kamapantula, A. Abdelzaher, P. Ghosh, M. Mayo, E. Perkins, and S. K. Das. 2012. Performance of wireless sensor topologies inspired by E. coli genetic networks. In *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 302–307.

[102] Y. Meng and H. Guo. 2012. A gene regulatory network based framework for self-organization in mobile sensor networks. In *Proceedings of the 2012 IEEE Congress on Evolutionary Computation*. IEEE, 1–7.

[103] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless sensor networks: A survey. *Comput. Netw.* 38, 4 (2002), 393–422.

[104] R. Kulkarni, A. Förster, and G. Venayagamoorthy. 2010. Computational intelligence in wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* 13, 1 (2010), 68–96.

[105] T. Kaur and D. Kumar. 2019. Computational intelligence-based energy efficient routing protocols with QoS assurance for wireless sensor networks: a survey. *Int. J. Wireless Mobile Comput.* 16, 2 (2019), 172–193.

[106] L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. *The Pagerank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.

[107] D. Pediaditakis, Y. Tselishchev, and A. Boulis. 2010. Performance and scalability evaluation of the Castalia wireless sensor network simulator. In *Proceedings of the 3rd international ICST conference on simulation tools and techniques*. ICST, 53.

[108] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 2 (2002), 182–197.

[109] P. Koduru, Z. Dong, S. Das, S. M Welch, J. L. Roe, and E. Charbit. 2008. A multiobjective evolutionary-simplex hybrid approach for the optimization of differential equation models of gene networks. *IEEE Trans. Evol. Comput.* 12, 5 (2008), 572–590.

[110] T. C. Collier and C. Taylor. 2004. Self-organization in sensor networks. *J. Parallel Distrib. Comput.* 64, 7 (2004), 866–873.

[111] K. L. Mills. 2007. A brief survey of self-organization in wireless sensor networks. *Wireless Commun. Mobile Comput.* 7, 7 (2007), 823–834.

[112] P. K. Biswas and S. Phoha. 2006. Self-organizing sensor networks for integrated target surveillance. *IEEE Trans. Comput.* 55, 8 (2006), 1033–1047.

[113] Y. Jin, H. Guo, and Y. Meng. 2012. A hierarchical gene regulatory network for adaptive multirobot pattern formation. *IEEE Trans. Syst. Man Cybernet. B* 42, 3 (2012), 805–816.
[114] F. Benhamida, A. Bouabdellah, and Y. Challal. 2017. Using delay tolerant network for the internet of things: Opportunities and challenges. In *Proceedings of the 2017 8th International Conference on Information and Communication Systems (ICICS'17)*. IEEE, 252–257.
[115] S. Saha, S. Nandi, P. Paul, V. Shah, A. Roy, and S. Das. 2015. Designing delay constrained hybrid ad hoc network infrastructure for post-disaster communication. *Ad Hoc Netw.* 25 (2015), 406–429.
[116] Md Y. S. Uddin, H. Ahmadi, T. Abdelzaher, and R. Kravets. 2013. Intercontact routing for energy constrained disaster response networks. *IEEE Trans. Mobile Comput.* 12, 10 (2013), 1986–1998.
[117] L. A. Zager and G. C. Verghese. 2008. Graph similarity scoring and matching. *Appl. Math. Lett.* 21, 1 (2008), 86–94.
[118] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo. 2017. Efficient energy management for the internet of things in smart cities. *IEEE Commun. Mag.* 55, 1 (2017), 84–91.
[119] J. Nielsen. 2007. Principles of optimal metabolic network operation. *Molec. Syst. Biol.* 3, 1 (2007), 126.
[120] S. Sakai, M. Togasaki, and K. Yamazaki. 2003. A note on greedy algorithms for the maximum weighted independent set problem. *Discr. Appl. Math.* 126, 2–3 (2003), 313–322.
[121] N. Matloff. 2008. *Introduction to Discrete-event Simulation and the Simpy Language*. Department of Computer Science. University of California at Davis.
[122] W. Yu, F. Liang, X. He, W. Grant Hatcher, C. Lu, J. Lin, and X. Yang. 2017. A survey on the edge computing for the Internet of Things. *IEEE Access* 6 (2017), 6900–6919.
[123] A. Ferrer, J. Marquès, and J. Jorba. 2019. Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing. *ACM Comput. Surv.* 51, 6 (2019), 1–36.
[124] W. Day and H. Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *J. Classif.* 1, 1 (1984), 7–24.
[125] A. Capponi, C. Fiandrino, D. Kliazovich, P. Bouvry, and S. Giordano. 2017. A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures. *IEEE Trans. Sust. Comput.* 2, 1 (2017), 3–16.
[126] S. Roy, P. Ghosh, D. Barua, and S. Das. 2020. Motifs enable communication efficiency and fault-tolerance in transcriptional networks. *Sci. Rep.* 10, 1 (2020), 1–15.
[127] S. Murthy and J. J. Garcia-Luna-Aceves. 1996. An efficient routing protocol for wireless networks. *Mobile Netw. Appl.* 1, 2 (1996), 183–197.
[128] F. Yu, Y. Li, F. Fang, and Q. Chen. 2007. A new TORA-based energy aware routing protocol in mobile ad hoc networks. In *Proceedings of the 2007 3rd IEEE/IFIP International Conference in Central Asia on Internet*. IEEE, 1–4.
[129] D. Bojic, E. Sasaki, N. Cvijetic, T. Wang, J. Kuno, J. Lessmann, S. Schmid, H. Ishii, and S. Nakamura. 2013. Advanced wireless and optical technologies for small-cell mobile backhaul with dynamic software-defined management. *IEEE Commun. Mag.* 51, 9 (2013), 86–93.
[130] J. Leskovec, D. Huttenlocher, and J. Kleinberg. 2010. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1361–1370.
[131] D. Kempe, J. Kleinberg, and É. Tardos. 2005. Influential nodes in a diffusion model for social networks. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*. Springer, 1127–1138.
[132] W. Chen, W. Lu, and N. Zhang. 2012. Time-critical influence maximization in social networks with time-delayed diffusion process. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*.

Received January 2020; revised March 2021; accepted May 2021