MITHRIL: Multi-Objective Topology Synthesis with Reinforcement Learning for Critical Networks

Oliver Wandschneider*, Anum Talpur*, Mathias Fischer*, Doğanalp Ergenç†
Universität Hamburg * Technische Universität Berlin†
name.surname@uni-hamburg.de* name.surname@tu-berlin.de†

***Abstract*—Mission-critical systems (MCSs) have evolving latency and reliability requirements, even under challenging conditions such as node and link failures and cyberattacks. To fulfill these requirements, emerging networking technologies like the IEEE 802.1 Time-Sensitive Networking standards provide several protocols for deterministic communication on top of off-the-shelf Ethernet equipment. While Ethernet-based networks offer better configurability than legacy fieldbus systems, they still require the design of adequate topologies for MCSs that fulfill various design objectives such as optimal quality of service and increased resilience against challenges. In this paper, we propose MITHRIL, a multi-objective topology synthesis model with reinforcement learning. It leverages deep reinforcement learning to optimize Ethernet-based topologies in terms of resilience and effectiveness, while adhering to realistic MCS constraints. Our evaluation indicates that MITHRIL enhances the failure and attack tolerance of network topologies while reducing the associated costs, compared to well-connected topologies and other heuristics from the literature.**

***Index Terms*—topology synthesis, resilience, Ethernet**

I. Introduction

Modern mission-critical systems (MCSs), e.g., cars, avionics, and industrial automation, comprise complex networks of several interconnected components with strict latency and reliability requirements. While legacy field buses cannot cope with the increasing complexity of MCSs, proprietary networking equipment and protocols cannot be used across different domains and cause vendor lock-in. Accordingly, different Ethernet-based standards such as Time-triggered Ethernet (TTEthernet) and IEEE 802.1 Time Sensitive Networking (TSN) have been designed to address these challenges [1]. Their deployment on top of Ethernet provides more flexibility in network configuration, lowers design costs, and fosters innovation across different domains [2].

When these new protocols are deployed in MCSs, the underlying network topologies must satisfy additional requirements beyond those of typical Ethernet networks. Fig. 1 illustrates a network example with several endpoints (v_1–v_6) transmitting time-sensitive streams (s_1–s_4) through Ethernet switches (n_1–n_3), highlighting these requirements and associated challenges (①–④). First, MCSs must be resilient to failures and attacks. In the figure, ① highlights n_2 as a critical node whose compromise can disconnect the entire network. Likewise, ② marks e_2 as an overloaded link and a potential single point of failure (SPoF). Second, satisfying quality of service (QoS) requirements for mixed-criticality streams, such as time-sensitive and best-effort traffic, demands careful network provisioning. A long network diameter complicates configuration and increases the likelihood of unexpected delays. For example, ③ shows a longer end-to-end path for s_1, which may lead to both latency issues and capacity bottlenecks. Lastly, the overall cost of MCS network design is a major concern for enterprises. As the network scales, underutilized components (such as e_1 indicated by ④) can significantly increase both monetary costs and maintenance burdens.

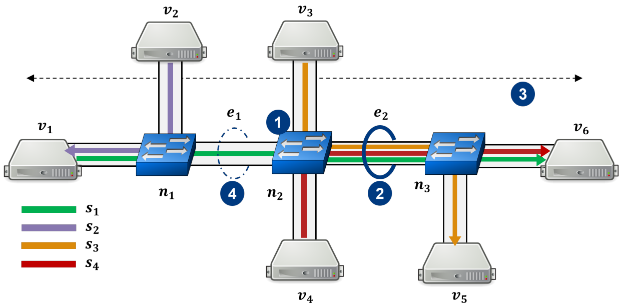


Fig. 1: An example Ethernet-based network.

Considering these design dimensions simultaneously for critical Ethernet-based topologies is challenging. Resilient topologies are not necessarily QoS-optimized nor cost-efficient; thus, several trade-offs occur. Although related work encompasses optimal scheduling and configuration of critical streams, the design of Ethernet-based topologies with realistic constraints is so far overlooked. Accordingly, the main contribution of this paper is the **M**ult**i**-objective **T**opology Synt**h**esis with **R**e**i**nforcement **L**earning (MITHRIL) model to design Ethernet-based topologies that are resilient against failures and attacks, satisfy QoS constraints and minimize costs in MCSs. In total, we make the following contributions:

- We characterize *resilience* and *effectiveness* as design objectives composing several robustness metrics, QoS, and cost metrics aligned with the MCS requirements.
- We propose MITHRIL, a Deep Reinforcement Learning (DRL)-based topology synthesis model, to build resilient topologies optimizing these characteristics.
- We evaluate MITHRIL and compare it with another approach from the literature that considers fault tolerance and QoS optimization in topology design [3].

Table I: Network elements and their properties.

Component	**Property**	**Description**
Switch (n)	$\rho_n \in \mathbb{Z}$	Number of ports
	$c_n \in \mathbb{R}$	Monetary cost
Link (e)	$l_e \in \mathbb{R}$	Propagation delay
	$c_e \in \mathbb{R}$	Monetary cost
	$h_e \in \mathbb{R}$	Bandwidth capacity
Stream (s)	$l_s \in \mathbb{R}$	Tolerable latency
	$h_s \in \mathbb{R}$	Data rate
	$\alpha_s \in \mathbb{Z}$	Criticality level

II. Related Work

Several works propose topology synthesis models focusing on Ethernet-based TSN networks and their latency and fault tolerance requirements. Lee et al. [4] use Demand-driven Interconnecting-Tree to design TSN-based industrial networks. Their objective is to minimize the network cost while ensuring connectivity in case of a link failure, under additional geographical constraints subject to industrial systems. However, their resulting topologies are limited to a tree structure. In [5], the authors propose Joint Topology, Routing, and Schedule Synthesis (JTRSS) to synthesize fault-tolerant topologies, ensuring the feasible routing and scheduling of TSN streams. It takes a fully connected topology and eliminates the redundant components to obtain the network with minimum cost. The resilience countermeasure of JTRSS is limited to finding disjoint redundant paths against (only single) link failures. Gavrilut et al. [3] propose two heuristics and an optimization model to synthesize fault-tolerant topologies with minimum design cost. While their heuristics provide sub-optimal solutions within minutes, the optimization model can achieve optimal results only for small problem instances in a significantly longer time.

Similar to our approach, some studies leverage DRL for topology synthesis. Li et al. [6] present a DRL- and Graph Neural Network (GNN)-based approach to modify a network topology regarding policy-based QoS, design cost, and link utilization, excluding any resilience countermeasure. In [7], the authors propose a DRL-based approach to optimize optical networks regarding the number and capacity of fibre links, transponders, and routers. Their key objective is to ensure network reliability under different failure scenarios.

Although they address cost- and QoS-related objectives in-depth, these works consider limited resilience measures that are merely effective against random link failures. Besides, they do not reflect several specific network constraints, such as using Ethernet-based TSN equipment, optimizing switch and link utilization, etc. Compared to combinatorial optimization techniques typically used for topology building [3], [4], MITHRIL leverages DRL to capture the complexity of multi-dimensional topology synthesis including Ethernet-based design constraints and objectives. Our design objectives include different topological metrics that increase resilience against more complex failures and attacks than random ones. Finally, we explore the trade-offs between those design objectives.

Table II: Different types of switches and links.

Component	**Property**	**Value**			
Switch	type	3	2	1	0
	ρ_n	32	20	12	4
	c_n	20	12	5	3
Link	type	1		0	
	h_e	1 Gbit		10 Gbit	
	c_e	0.25		0.4	
	l_e	1.5 ms		1 ms	

III. System Model

In our system model, a network comprises two interconnected parts: a physical topology and a communication model. Here, we consider the architectural aspects of TSN, since it is the most recent Ethernet-based technology, supporting traffic scheduling and redundancy, required by MCSs. However, our model can be adapted to any Ethernet-based topology with varying protocol stacks as well. Table I shows the components and their properties in the system model.

A. Topology Model

A network topology G consists of endpoints, switches (e.g., TSN bridges), and links, s.t., $G = (V, N, E)$. An endpoint $v \in V$ produces or consumes data, e.g., a sensor or computing unit. An Ethernet switch $n \in N$ interconnects endpoints and other switches. It has a fixed number of ports ρ_n and a monetary cost c_n according to its port count. Note that an off-the-shelf switch does not usually have an arbitrary size but is standardized with a certain number of ports. To make our topology design problem more realistic, we consider four tiers of switches as shown in Table II. We define their monetary cost as an abstract value aligned with our market research for TSN bridges.

Lastly, a link $e \in E$ is an Ethernet cable that connects endpoints and switches. Links are characterized by their function, monetary cost, bandwidth capacity, and induced propagation delay. An access-link $e \in E_a$ is deployed between an endpoint and a switch, and a trunk-link $e \in E_t$ connects a switch to another switch. Like Ethernet switches, off-the-shelf Ethernet cables are manufactured in standard bandwidth capacity h_e. Accordingly, we consider $1\,\mathrm{Gbit}$ and $10\,\mathrm{Gbit}$ links as shown in Table II rather than assigning arbitrary link capacity. We assign a fixed latency l_e to each link that does not only reflect the propagation delay but also the potential processing delay on the egress port that e is connected to. A link with lower capacity is associated with higher latency since it can cause more congestion compared to high-bandwidth links, thus causing higher queueing and processing delays on a switch. Links also have varying monetary costs c_e aligned with their actual market value. We approximate the ratio of switch and link prices according to their respective market values.

B. Communication Model

The communication model consists of several data streams $s \in S$. Each stream has specific QoS and reliability requirements similar to the TSN-based networks. The QoS requirements of s comprise its maximum allowed end-to-end

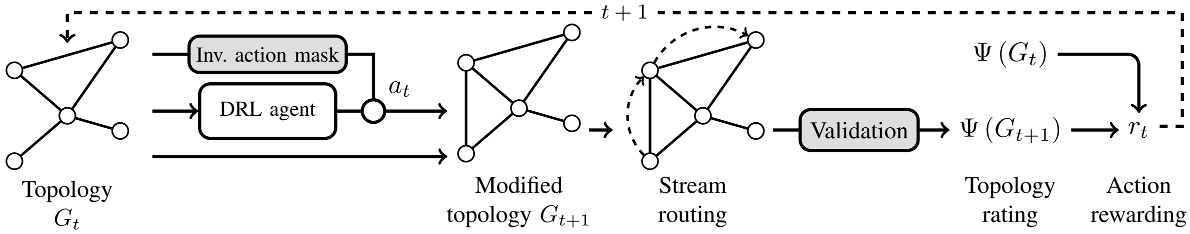


Fig. 2: MITHRIL framework.

latency l_s and its data rate h_s. Besides, each stream has a criticality level α_s that specifies the required number of redundant disjoint paths. In TSN context, redundant paths can be configured against node and link failures [8]. Eventually, it is possible to set these parameters per stream to specify various communication models with different proportions of critical and non-critical streams.

C. Problem Definition and Constraints

Given the topology and communication model, we can define the overall topology synthesis problem based on the example in Fig. 1. In the figure, six endpoints v_1-v_6 are connected via the switches n_1, n_2, n_3 and communicate via the streams s_1-s_4. This imposes n_1 and n_3 having at least three, and n_2 having at least 4 available ports. While the streams are assigned to end-to-end paths, the resources of each physical link on the respective paths are allocated, e.g., the capacity of the link between endpoint v_1 and switch n_1 is reduced by the combined data rate $h_{s_1} + h_{s_2}$, such as $200\,\mathrm{Mbit/s}$ and $150\,\mathrm{Mbit/s}$ for s_1 and s_2, respectively. Additionally, all switches and links induce (fixed) processing, queueing, and propagation delays, e.g., $1\,\mathrm{ms}$ and $0.1\,\mathrm{ms}$, respectively, as parts of an end-to-end path. Besides, critical streams, i.e., streams where $\alpha_s \geq 2$, require redundant paths for seamless fault tolerance against link failures. For instance, if we change $\alpha_{s_1} = 2$ at least one additional switch must be added to the network to route the required disjoint path.

Following this scenario, our main problem is to synthesize resilient and effective Ethernet topologies that meet the given communication requirements. As objectives, resilience mainly targets failure and attack tolerance, and effectiveness includes improving QoS and reducing design costs. This problem is further subject to several structural (i-iii) and communication (iv-vi) constraints such as:

i) The topology must be connected, where each endpoint is connected to strictly one switch via an access link.
ii) All switches and links must be off-the-shelf equipment, i.e., not arbitrary size or capacity.
iii) The number of links connected to a switch must not be greater than the number of ports of the switch.
iv) The capacity of a link (h_e) must not be exceeded by the total data volume of its associated data streams (h_s).
v) For every stream s, the end-to-end delay between their corresponding endpoints (i.e., induced by all links (l_e) in-between) must be less than its tolerable (l_s).
vi) For every stream s, the number of disjoint paths between their corresponding endpoints must be greater or equal to their criticality level (α_s).

Overall, we approach topology synthesis as an augmentation problem, in which we optimize a given network towards the specified resilience and effectiveness metrics by adding, removing, or altering links and switches. This problem quickly becomes intractable, as the search space grows exponentially with network size. As proven in [9], even optimizing a graph towards a single objective by adding a specified number of edges to the graph is NP-hard. This is additionally challenging when multiple and potentially conflicting objectives are introduced. We propose MITHRIL for tackling this challenge.

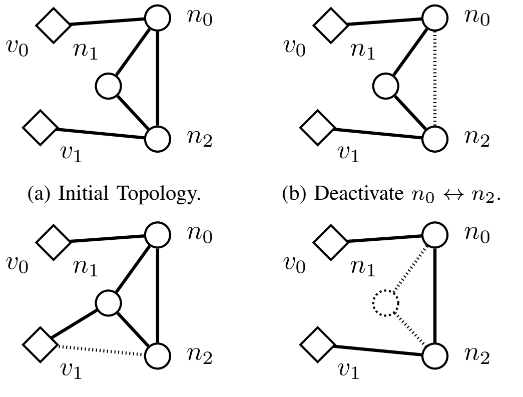
IV. Multi-objective Topology Synthesis Model

MITHRIL leverages DRL to improve a given topology towards different design objectives targeting resilience and effectiveness, comprised of cost-efficiency and high QoS. This section describes its overall design artifacts.

A. General Overview

MITHRIL performs consecutive *topology building actions* on an initial topology, evaluates the modified topology using its *topology rating function*, and learns taking the best actions by positive and negative *rewarding* as illustrated in Fig. 2.

In the figure, the initial state, i.e., topology G_t with $t = 0$ is an arbitrarily connected network (Section III-A) represented as a heterogeneous graph [10]. For each topology building action, MITHRIL first ensures their feasibility by masking the invalid actions that can violate structural constraints ii)-iii) in Section III-C. Then, it modifies G_t to G_{t+1} by performing a valid action a_t such as adding a link or reconnecting an endpoint. After each action, it also verifies the validity of the communication constraints (see iv)-vi) in Section III-C) for the streams potentially impacted by this action. In case of a violation, MITHRIL *re-routes* the respective streams to different end-to-end paths. The next phase calculates the topology rating, $\Psi(G_{t+1}, S)$, using several objective metrics. If it observes a better rating s.t. $\Psi(G_{t+1}, S) > \Psi(G_t, S)$, MITHRIL gets a positive reward, r_t. Otherwise, it is penalized because of a diminishing action. Employing a DRL agent, MITHRIL learns taking the best actions by making consecutive topology modifications a_t for $t = 0..T$ that achieve a continuously increasing topology rating. The following sections introduce these steps in more detail.



(a) Initial Topology.

(b) Deactivate $n_0 \leftrightarrow n_2$.

(c) Swap $v_1 \leftrightarrow n_1$.

(d) Deactivate n_1.

Fig. 3: Illustration of different actions on a given topology.

B. Topology Building Actions

MITHRIL can take five different topology building actions illustrated in Fig. 3. These actions manipulate the network elements and their connectivity for improving the overall topology toward the desired objectives.

i) **Deactivate trunk link:** The target trunk-link $e \in E_t$ is excluded from the topology (Fig. 3b).
ii) **Activate trunk link:** The target trunk-link $e \in E_t$ is added or (re)included to the topology if deactivated.
iii) **Swap access link:** This action activates the target access-link $e \in E_a$ while deactivating all other access-links originating from the corresponding endpoint (Fig. 3c). It simply rewires an endpoint to a different switch.
iv) **Downgrade switch:** The type of a switch is decreased to the lowest possible tier so that the number of ports still satisfies the connected links. This affects its number of ports and cost. For instance, a 32-port switch with 10 active links can be downgraded to a 12-port switch. If the switch type is of the lowest tier (see Table II), it can be completely deactivated, removing all its trunk-links as long as no endpoint is attached to it (Fig. 3d).
v) **Pass:** This action does not modify the topology and the current timestep is skipped without changes.

C. Stream Routing

MITHRIL allocates all streams on the available shortest paths between their respective endpoints. It also finds disjoints paths for critical streams. The model performs rerouting and/or finds new disjoint paths after every topology modification.

We calculate the suitable paths between endpoints using the Bellman-Ford shortest path algorithm [11]. The weight of a link is defined by the delay it induces. The streams are sequentially allocated, ordered by the lowest l_s first, to ensure that streams with more stringent requirements are routed with reduced possibility of bandwidth restrictions. For every stream s, MITHRIL selects a shortest path p, which induces less end-to-end delay than l_s. This path should have sufficient capacity on all links e that it is composed of s.t. $h_s + h_e^+ \leq h_e$, where h_e^+ is reserved capacity on e for other streams.

Table III: Metrics for the design objectives.

Objective	**Metric**	**Description**
Resilience (Ψ_{res})	d_n β_n β_e	Active ports of switch n Betweenness centrality of switch n Betweenness centrality of link e
Effectiveness (Ψ_{eff})	c_n, c_e $u_h(e)$ σ_s	Monetary cost of switches and links Bandwidth utilization rating for link e Deviation in experienced latency

Redundant disjoint paths are calculated by the Edmonds-Karp algorithm [12] for the streams with criticality level $\alpha_s \geq 2$. Redundant disjoint paths must satisfy the same latency and capacity constraints as the shortest paths. If MITHRIL cannot find at least α_s routes that satisfy the delay and bandwidth requirements of $\forall s \in S$, the modified topology is rendered invalid, and the respective action is penalized.

D. Topology Rating

As its primary topology synthesis principle, MITHRIL aims to increase the rating of a given topology in terms of two design objectives: resilience and effectiveness, which is a combined measure of the network cost and QoS of the routed streams. These objectives induce several trade-offs that eventually lead to a Pareto-optimal topology. For instance, increasing resilience against failures via redundant paths induces additional link cost. Similarly, distributing traffic homogeneously over the network reduces the risk of a SPoF but could result in QoS degradation compared to using the shortest paths.

We formulate the topology rating function $\Psi(G, S)$ as a combination of two distinct objectives $\Psi_{\text{res}}(G)$ and $\Psi_{\text{eff}}(G, S)$:

$$\Psi = w_{\text{res}} \;\; \Psi_{\text{res}} + w_{\text{eff}} \;\; \Psi_{\text{eff}} \tag{1}$$

In Eq. (1), w_{res} and w_{eff} are set to adjust the relative importance of the individual objective dimensions s.t. $w_{\text{res}} + w_{\text{eff}} = 1.0$. When $w_{\text{res}} > w_{\text{eff}}$, MITHRIL is expected to build more resilient topologies disregarding the cost and QoS characteristics related to effectiveness. Each objective is measured by several metrics summarized in Table III and further described below.

1) Resilience: MITHRIL considers two aspects for the resilience objective: redundancy and network homogeneity. Firstly, redundancy over disjoint paths provides fault tolerance against node and link failures, as also the primary resilience measure in TSN [8]. MITHRIL ensures the existence of at least α_s disjoint paths for $\forall s \in S$ as a hard constraint (Section III-C).

Secondly, MITHRIL aims to increase network homogeneity to reduce the risk of SPoF in case of attacks and failures on critical components. We utilize two graph metrics to quantify homogeneity: node degree and betweenness centrality. A component with a high degree is an attractive target for attackers since it interconnects many other components. Besides, failures on such components usually have a destructive impact [13], [14]. Similarly, the betweenness centrality of a network measures the tendency of a component, i.e., switch or link, to be more central than others [15]. Any failed or attacked component with high centrality could induce high damage [16]. Accordingly, MITHRIL flattens attack and failure surfaces by homogenizing switches and links regarding these metrics to avoid centralized hubs and bottleneck links.

We calculate the switch degree d_n as the number of active ports of a switch, i.e., utilized links connected to a switch. Then, MITHRIL regards node degree in topology rating as:

$$f_d(N) = 1 - g\left(\left\{ d_n \;\middle|\; n \in N \right\}\right) \tag{2}$$

In this function, $g(\cdot) \in [0, 1]$ represents the Gini coefficient [17]. It measures the similarity between the degrees of all switches, i.e., $d_n \; \forall n \in N$ and $N \in G$. A lower coefficient indicates less dispersion, which means no switch stands out as a more appealing target. Thus, with Eq. (2), MITHRIL assesses a higher topology rating for decreasing $g(\cdot)$.

Betweenness centrality of a switch β_n and link β_e is the ratio of the number of shortest paths passing over those components to all the shortest paths in the network. MITHRIL assesses the centrality of the whole network and obtains a higher rating for a decreasing value, which indicates fewer central components. Accordingly, we formulate Eqs. (3) and (4) following the definition in [15] to compute *reverse* centrality of a graph regarding switches and links:

$$f_\beta(N) = 1 - \frac{\sum_{n \in N} (\beta_n^* - \beta_n)}{|N| - 1} \tag{3}$$

$$f_\beta(E) = 1 - \frac{\sum_{e \in E} (\beta_e^* - \beta_e)}{|E| - 1} \tag{4}$$

where β_n^* and β_e^* are the highest switch and link centrality in the network, respectively.

$$\Psi_{\text{res}} = w_1 \, f_d(N^+) + w_2 \, f_\beta(N^+) + w_3 \, f_\beta(E^+) \tag{5}$$

Finally, the objective Ψ_{res} is scalarized by weighting the aforementioned metrics (s.t., $\sum_{i=1} w_i = 1.0$) as shown above. Note that N^+ and E^+ represent the set of active switches and links, i.e., included in the modified topology by MITHRIL.

2) Effectiveness: The effectiveness objective entails design cost, resource utilization, and QoS. Firstly, the monetary cost of switches and links is dependent on their respective types shown in Table II. We define f_c to compute the total cost of a given set of components, e.g., $f_c(N)$ returns the total switch cost as follows:

$$f_c(N) = \sum_{n \in N} c_n \tag{6}$$

for the topology $G = (V, N, E)$. Then, we compute the monetary gain $\overline{f_c}(N)$ as a normalized value:

$$\overline{f_c}(N) = \frac{f_c(N) - f_c(N^+)}{f_c(N)} \tag{7}$$

where, N and N^+ represent the respective switches in the initial topology G and only the active components in the (final) modified topology. Accordingly, MITHRIL aims to maximize monetary gains for switches by $\overline{f_c}(N)$ (as formulated above) and links by $f_c(E)$, which is formulated similarly to Eq. (7).

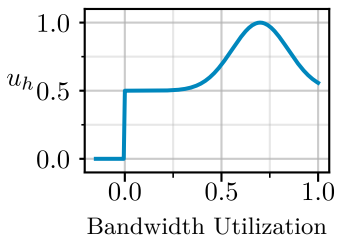


Fig. 4: Rating function for link utilization u_h.

Secondly, MITHRIL aims to keep links under moderate utilization to minimize the expenses induced by underutilized resources but still ensures available resources to reconfigure the network in case of any incidents. We formulate u_h to compute a high rating for links with moderate utilization:

$$u_h(e) = \begin{cases} \frac{1}{2} \mathrm{e}^{-24 \left(\frac{h_e^+}{h_e} - \delta \right)^2} + 0.5, & \text{if } h_e^+ > 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

In Eq. (8), h_e^+ is the active bandwidth use of link e, calculated by the total amount of data traffic for all streams s forwarded through e. The targeted link utilization can be adjusted through the parameter δ. We set $\delta = 0.7$ so that the function gives the highest topology rating for a link utilized around 70%, as also shown in Fig. 4. This is set based on the observation that a higher link utilization than 70% can worsen the end-to-end latency under realistic queueing models [18]. Note that Eq. (8) does not harshly penalize the underutilized links but only the non-utilized links to deactivate the empty ones more effectively. Finally, MITHRIL uses the rating function $f_h(E^+)$ to calculate the average utilization for links. It is defined as the average rating $u_h(e)$ of all active links $e \in E^+$.

Thirdly, to improve QoS, MITHRIL activates the *best* access links to connect endpoints to the edge switches that minimize end-to-end latency for the streams initiated by the respective endpoints. MITHRIL already guarantees that each stream is assigned to a path fulfilling their latency requirement, s.t. $l_p^s \leq l_s$, where l_p^s is the end-to-end latency on path p that stream s is forwarded through. We further measure σ_p^s, the normalized difference between experienced (l_p^s) and tolerable (l_s) latency, so that MITHRIL can find an available path p with the highest deviation, i.e., achieving the lowest end-to-end delay as follows:

$$\sigma_p^s = \frac{l_s - l_p^s}{l_s} \tag{9}$$

Finally, we formulate $f_s(P, S)$ to rate the overall topology for all streams by averaging σ_p^s as follows:

$$f_s(P, S) = \frac{1}{|S|} \sum_{s \in S} \sigma_p^s \qquad \exists p \in P \tag{10}$$

By weighting the metrics for monetary cost, resource efficiency and QoS (s.t., $\sum_{i=4} w_i = 1.0$), the effectiveness rating Ψ_{eff} of a topology is calculated as follows:

$$\Psi_{\text{eff}} = w_4 \overline{f_c}(N) + w_5 \overline{f_c}(E) + w_6 f_h(E^+) + w_7 f_s(P, S) \tag{11}$$

E. Action Rewarding

MITHRIL aims to gradually increase the topology rating for modified topologies G_t after each topology building action a_t. An increasing topology rating $\overline{\Psi}_t = \Psi(G_{t+1}, S) - \Psi(G_t, S) > 0$ indicates an improvement in the topology towards the desired design objectives, according to their priority set by the weights defined in Eq. (1). Contrarily, a decrease in topology rating means that MITHRIL has taken an action that diminishes the topology. Accordingly, the action rewarding function r_t rewards the DRL agent of MITHRIL for an improving action and penalizes it for a diminishing one at step t as follows:

$$r_t = \begin{cases} 1.00 & \text{if } \overline{\Psi}_t \geq 0.025 \\ 0.75 & \text{if } 0.025 > \overline{\Psi}_t \geq 0.010 \\ 0.50 & \text{if } 0.010 > \overline{\Psi}_t \geq 0.001 \\ 0.10 & \text{if } 0.001 > \overline{\Psi}_t > 0 \\ 0.00 & \text{if } \overline{\Psi}_t = 0 \\ -1.00 & \text{otherwise} \end{cases} \tag{12}$$

For $\overline{\Psi}_t > 0$, MITHRIL is rewarded proportional to the degree of improvement in topology rating. Otherwise, it is penalized by a fixed value of -1.00. Note that we have experimented with different reward functions, such as simply rewarding MITHRIL based on the topology rating, i.e., $r_t = \Psi(G_{t+1}, S)$. However, we observe that the step function (12) accelerates the learning significantly and a fixed high penalty helps to avoid diminishing actions.

V. Implementation

We implemented MITHRIL using TensorFlow [19] and the TensorFlow GNN library [20]. It is trained with the Advantage Actor-Critic algorithm, derived from the Asynchronous Advantage-Actor-Critic algorithm [21]. MITHRIL employs a GNN with multi-head-attention [22] to encode a topology G to a learned vector embedding $\vec{G}$, and two multi-layer perceptrons as the actor and the critic. GNNs are used, since they can utilize the structure of graphs in order to learn the feature representations of the node and edge features, and have shown outstanding performance in graph datasets across different domains [22]. The actor represents the policy π and returns the logarithmic probabilities $\mathbb{P}$ for sampling the topology building actions in Section IV-B. The critic $\mathbb{V}^{\pi}$ is an approximation of the value function and estimates the expected returns when following policy π. The structure of actor and critic is: Input, ReLU (64), Dropout (0.5), ReLU (64), ReLU (64), followed by Linear (5) and (1), respectively.

Starting with the initial topology G_0, MITHRIL sequentially performs actions, until $t = T$ or an invalid state is reached due to a violated constraint. For each step t, MITHRIL samples an action a_t from $\mathbb{P}_t$ with the invalid actions masked as described in Section IV. It then performs a_t to modify G_t into G_{t+1} and calculates the reward r_t based on the topology rating change. We calculate the discounted reward R with discount factor $\gamma = 0.99$. If G_t is valid, the final value $R_t = v_t$ and otherwise $R_t = 0$. Finally, the loss functions ℓ_π and $\ell_{\mathbb{V}}$ denoted in Eqs. (13) and (14), are calculated. They are adopted from the loss functions defined by [21], but we use the Generalized Advantage Estimation (GAE) [23] $\hat{A}^{GAE(\gamma,\lambda)}$ with the GAE-lambda $\lambda = 0.97$. To improve exploration and discourage premature convergence to a suboptimal policy, the entropy is added in ℓ_π [21], with H_t defined as the entropy of sampling a_t from the logarithmic probabilities $\mathbb{P}_t$, i.e., the average level of uncertainty when selecting this particular action. Since the GNN is shared by the actor π and the critic $\mathbb{V}^{\pi}$, its loss is defined as $\ell_\pi + \delta \ell_{\mathbb{V}}$ with a weighting factor $\delta = 0.5$.

$$\ell_\pi = \frac{-1}{n} \sum_{i=1}^{n} -\mathbb{P}_i \times \hat{A}_i^{GAE(\gamma,\lambda)} + \beta \frac{\sum_{i=1}^{n} H_i}{n} \tag{13}$$

$$\ell_{\mathbb{V}} = \frac{1}{n} \sum_{i=1}^{n} (R_i - \nu_i)^2 \tag{14}$$

We trained MITHRIL for 2000 episodes and $T = 75$ actions per episode. We used a learning rate of 0.0003 for the actor and GNN, and 0.001 for the critic.

VI. Evaluation

In this section, we present our experimental setup, metrics, and the numerical results of our evaluation. Our primary goal is to demonstrate both the benefits and potential limitations of MITHRIL to flexibly optimize Ethernet-based topologies with configurable priorities on resilience and effectiveness.

A. Experimental Setup

For training and evaluation, we generated several Watts-Strogatz networks [24] with $|N| = 16$ switches (all type 3) and $|V| = 16$ endpoints, and links attached with 60% probability. 75% of all links are randomly selected as type 1 and the rest is set as type 0 (see Table II). We input these graphs as initial topology G_0 to MITHRIL. They show small-world properties reflecting typical characteristics of TSN networks that are not highly connected but ensure network-wide reachability with low latency. Note that MITHRIL is topology-agnostic and an arbitrary initial topology can also be given.

We generated several instances for the described communication model with $|S| = 32$, assigning two streams per endpoint. We set 20% of these streams as critical, requiring at least two redundant paths ($\alpha_s \geq 2$). Their latency and data rate demands are assigned randomly between $4\,\mathrm{ms}$ to $10\,\mathrm{ms}$ and $250\,\mathrm{Mbit/s}$ to $750\,\mathrm{Mbit/s}$, respectively. We repeated the experiments for 100 randomly generated topologies per evaluation scenario. For the experimental results we only consider the valid synthesized topologies (on average 70). Table IV summarizes these parameters.

Lastly, we selected eleven weight sets $w = (w_{\text{res}}, w_{\text{eff}})$ s.t. $w = (w_{\text{res}}, 1 - w_{\text{res}})$ for every w_{res} in $\{0.0, 0.1, \ldots, 1.0\}$. This helps to evaluate the impact of objective prioritization, e.g., $w_{\text{res}} = 1.0$ and $w_{\text{eff}} = 0.0$ focus purely on resilience without cost or QoS concerns. In the remainder of this section, the individual weight sets are denoted using only their respective value of w_{res}.

Table IV: Experiment parameters.

Parameter	**Value**
Number of switches	16, all type 3
Number of endpoints	16
Link attachment probability	60%
Link type	75% type 1 and 25% type 0
Number of streams	32
Proportion of critical streams	20% with $\alpha_s \geq 2$
Stream latency (l_s)	4 ms to 10 ms
Stream data rate (h_s)	250 Mbit/s to 750 Mbit/s
Repetitions per weight set	100

For benchmarking, we compute the evaluation metrics for the initial topology G_0 (**Initial**), and also compare our results to the Topology Routing Heuristic (**TRH**) introduced in [3]. TRH starts with a fully connected network, iteratively routes all streams sorted by their timing properties, and finally changes the switch types to be of minimal cost with the required ports. Unused links and switches are removed from the topology. We modified TRH to utilize a shortest-path algorithm instead of breadth-first-search as the links in our topology have different weights, and we initialize with Watts-Strogatz networks analogously to MITHRIL, instead of fully connected networks. All experiments are conducted in parallel on a server with 96 CPU cores and 128 GB of memory.

B. Evaluation Metrics

We use the following metrics for our evaluation:

i) **Ratio of surviving streams** measures the percentage of surviving streams in case of multiple attacks or failures, to evaluate the impact of Ψ_{res} for the resilience objective.

ii) **Relative cost** is the ratio of the total cost of switches and links in the final topology to the cost of the initial topology. This measures the impact of $\overline{f_c}(N)$ and $\overline{f_c}(E)$ for the effectiveness objective. We also consider the **deducted cost**, where we additionally remove all unused links and switches, and assign the switch types so that they are of minimal cost with the required number of ports. We further elaborate on this metric later.

iii) **Average end-to-end latency** is the mean latency of the shortest paths forwarding at least one stream. It evaluates the impact of the $f_s(P, S)$ on the effectiveness objective.

C. Experimental Results

In the following figures, we compare the topology synthesis results of MITHRIL trained with different weight sets denoted as $w_{\text{res}} = 0.0, 0.1, \ldots, 1.0$ (on the x-axis of the following figures), with the respective evaluation metrics. We measure these metrics for initial topologies at $t = 0$ (Initial, dashed line), and the TRH results (TRH, solid line) to set a baseline for the evaluation. Training MITHRIL takes on average 23 s per episode (12.75 h for 2000 episodes). Inference with a trained model takes on average 99 ms for one action (7.4 s for $T = 75$ actions). The results are categorized according to two design objectives, resilience and effectiveness.

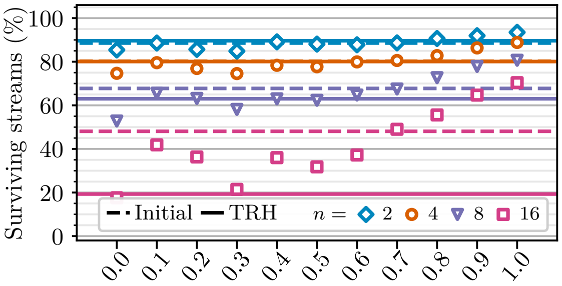


Fig. 5: Surviving streams after n link failures/attacks.

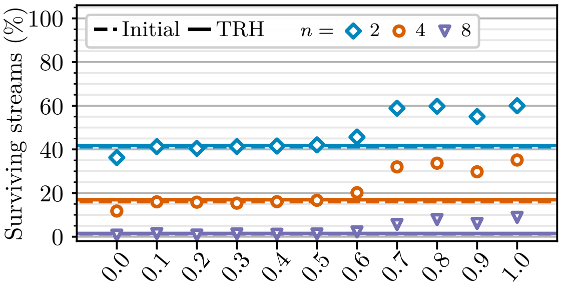


Fig. 6: Surviving streams after n switch failures/attacks.

1) Resilience: Firstly, we simulate calibrated attacks (or worst-case failures) targeting critical links and switches in the network to evaluate resilience. Fig. 5 illustrates the ratio of the surviving streams after removing 2 (blue, diamond), 4 (orange, circle), 8 (purple, triangle) and 16 (pink, square) links with the *highest centrality* ($\max\{\beta_e \mid \forall e \in E\}$). *Initial* and *TRH* lines with different colors also show the results for the respective scenarios. The results show that training MITHRIL with a focus on resilience ($w_{\text{res}} > 0.6$) improves the survivability of streams, compared to the baselines and the remaining models ($w_{\text{res}} < 0.6$). The reason is that MITHRIL reduces the number of high-centrality links by homogenizing the network, and eventually the impact of individual failures/attacks. Here, the difference between MITHRIL and TRH goes up to 20% and 50% for 8 and 16 critical link removals, respectively.

Switch failures or compromises typically have a more severe impact than individual link failures, as they can disconnect multiple components simultaneously. In Fig. 6, we simulate the removal of 2 to 8 switches with the highest active degree (d_n) and measure the resulting ratio of surviving streams. Topologies generated by MITHRIL ($w_{\text{res}} > 0.6$) are able to sustain more streams (up to 20%) following 2 to 4 switch failures (blue and orange), compared to the initial topology and the TRH baseline, where most streams fail to be delivered. Notably, the optimized topologies retain partial functionality even after half of the switches (8 out of 16) are removed, whereas baseline approaches result in total stream failure.

2) Effectiveness: For effectiveness objective, we focus on the design cost and the resulting QoS for the synthesized topologies. Note that, compared to the results in Fig. 5 and Fig. 6, the MITHRIL models with $w_{\text{res}} < 0.6$ provides better results as they prioritize effectiveness over resilience.

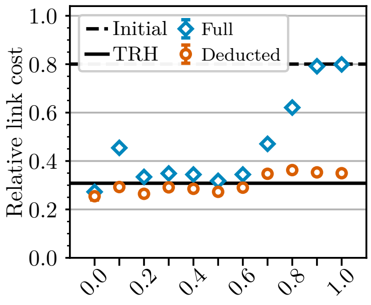


Fig. 7: Link cost.

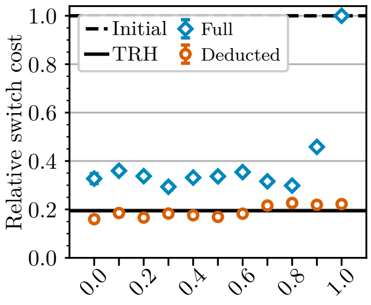


Fig. 8: Switch cost.

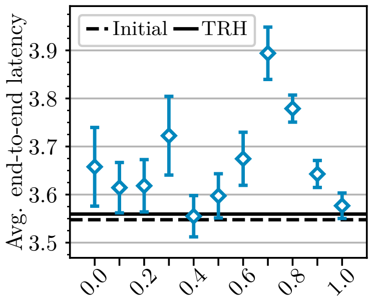


Fig. 9: Average end-to-end latency.

Fig. 7 presents the relative link cost ($\overline{f_c}(E)$) of the synthesized topologies. We observe that MITHRIL (blue, diamond) can reduce link costs by up to 65% compared to the initial topology for configurations where $w_{\text{res}} < 0.6$. In contrast, models emphasizing resilience (e.g., $w_{\text{res}} > 0.8$) tend to maintain a cost level similar to the initial topology, as they retain additional (and sometimes unused) links to promote structural homogeneity. TRH achieves lower costs by aggressively removing all unused links, regardless of their potential utility. MITHRIL's *deducted* topologies (orange, circle), which remove redundant links and switches similar to TRH, yield comparable cost reductions. In this context, the gap between the *Full* and *Deducted* costs for $w_{\text{res}} > 0.6$ can be interpreted as the cost of resilience, representing the overhead MITHRIL incurs to enhance network homogeneity.

Similarly, Fig. 8 shows the relative switch cost ($\overline{f_c}(N)$) for MITHRIL-generated topologies. Here, MITHRIL consistently incurs higher switch costs than TRH in the original configuration (blue, diamond). Although these costs can be reduced through further switch removals or downgrades (deducted cost), this highlights a limitation: MITHRIL does not necessarily minimize cost, even in scenarios where $w_{\text{res}} = 0.0$ and $w_{\text{eff}} = 1.0$. This behavior stems from MITHRIL's objective to balance cost with QoS metrics under the effectiveness goal, rather than exclusively optimizing for minimal design cost.

Fig. 9 measures QoS regarding the average end-to-end latency. Except $w_{\text{res}} = 0.4$, MITHRIL usually induces more latency than the initial topology and TRH. In these results, there is no clear correlation between w_{res} and latency. There are two potential reasons for this. First, the influence of QoS metric ($f_s(P, S)$) may remain limited in the compound rating function quantifying the overall effectiveness. This can be further adjusted by its respective weight, i.e., w_7 in Ψ_{eff}. Counterintuitively, $w_{\text{res}} = 1.0$ results in the lowest latency because it keeps many links active for homogenizing the network (see Fig. 7) and provides shorter alternative paths. Second, MITHRIL utilizes a simplified routing approach, which fulfills all QoS requirements without further optimization. It only has an indirect control over routing by manipulating links, which may remain limited for QoS optimization.

VII. Discussion

MITHRIL utilizes a range of metrics for topology evaluation and optimization. While all metrics are normalized to enable fair comparison, each varies differently after topology modifications. As a result, *their influence on design objectives can vary significantly*. To account for this, MITHRIL incorporates configurable weighting factors ($w_1 \dots w_7$, see Section IV-D) that allow users to adjust the relative impact of each metric. Although our results demonstrate that MITHRIL can successfully synthesize topologies aligned with specific objectives, the choice of weights remains a key factor. With careful tuning, the model's performance can be further improved.

Moreover, assessing the individual impact of each metric is challenging due to their overlapping influence on multiple objectives. For instance, reducing path lengths can enhance both resilience and QoS. Conversely, using lower-tier links (e.g., 1 Gbit) may introduce higher latency but can lower costs and improve link utilization, potentially conflicting with the goals of the effectiveness objective. Therefore, *a deliberate analysis is required to select orthogonal metrics that primarily affect their respective design objectives, whether positively or negatively*. While we have carefully evaluated several metrics based on their relevance, further study is needed to fully understand their independence and interactions.

Lastly, we introduce MITHRIL as a framework for flexibly optimizing topologies toward multiple objectives. Our goal is not to prescribe a fixed set of objective weights for an optimal network design. Nevertheless, as a final takeaway, we show that setting $0.6 \leq w_{\text{res}} \leq 0.8$ achieves a balance between resilience, cost, and QoS, outperforming even some configurations that prioritize effectiveness alone.

VIII. Conclusion

The emerging IEEE 802.1 TSN standards enable the use of Ethernet in MCSs, which require resilient, low-latency communication. However, this necessitates the design of suitable Ethernet-based network topologies that meet the stringent requirements and constraints of MCSs. In this paper, we introduce MITHRIL, a multi-objective topology synthesis framework based on reinforcement learning. It employs DRL to generate Ethernet networks optimized for both resilience and effectiveness, under realistic topological constraints. Our evaluation demonstrates that MITHRIL can substantially reduce network design costs and enhance resilience against multiple worst-case failures and attack scenarios, all while satisfying the QoS requirements of the target systems.

References

[1] L. Zhao, F. He, E. Li, and J. Lu, "Comparison of Time Sensitive Networking (TSN) and TTEthernet," in *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pp. 1–7, 9 2018.

[2] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.

[3] V. Gavrilut, B. Zarrin, P. Pop, and S. Samii, "Fault-Tolerant Topology and Routing Synthesis for IEEE Time-Sensitive Networking," in *25th International Conference on Real-Time Networks and Systems (RTNS)*, pp. 267–276, 2017.

[4] Y.-R. Lee, P.-C. Hou, and T.-C. Hou, "Topology Design of Time Sensitive Networks in A Smart Factory," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 938–943, 2022.

[5] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Fault-Resilient Topology Planning and Traffic Configuration for IEEE 802.1Qbv TSN Networks," in *IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*, pp. 151–156, 2018.

[6] Z. Li, X. Wang, L. Pan, L. Zhu, Z. Wang, J. Feng, C. Deng, and L. Huang, "Network Topology Optimization via Deep Reinforcement Learning," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2847–2859, 2023.

[7] H. Zhu, V. Gupta, S. S. Ahuja, Y. Tian, Y. Zhang, and X. Jin, "Network planning with deep reinforcement learning," in *ACM SIGCOMM Conference*, pp. 258–271, 2021.

[8] D. Ergenç and M. Fischer, "On the Reliability of IEEE 802.1CB FRER," in *IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–10, 2021.

[9] D. Mosk-Aoyama, "Maximum algebraic connectivity augmentation is np-hard," *Operations Research Letters*, vol. 36, no. 6, pp. 677–679, 2008.

[10] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous Graph Attention Network," in *The World Wide Web Conference (WWW)*, pp. 2022–2032, 2019.

[11] R. Bellman, "On a Routing Problem," *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[12] J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM*, vol. 19, p. 248–264, apr 1972.

[13] M. J. F. Alenazi and J. P. Sterbenz, "Evaluation and Comparison of Several Graph Robustness Metrics to Improve Network Resilience," in *7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pp. 7–13, 2015.

[14] J. Paillisse, S. Bergillos, M. Madrenys, and E. Calle, "Supervised Machine Learning Techniques to Calculate the Robustness of Networks," in *13th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 1–6, 2023.

[15] S. K. Raghavan Unnithan, B. Kannan, and M. Jathavedan, "Betweenness Centrality in Some Classes of Graphs," *International Journal of Combinatorics*, vol. 2014, no. 1, p. 241723, 2014.

[16] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, "Attack robustness and centrality of complex networks," *Public Library of Science (PLoS) One*, vol. 8, no. 4, 2013.

[17] C. Gini, "On the measure of concentration with special reference to income and statistics," *General Series*, vol. 208, no. 1, 1936.

[18] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier, 2004.

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. Software available from tensorflow.org.

[20] O. Ferludin, A. Eigenwillig, M. Blais, D. Zelle, J. Pfeifer, A. Sanchez-Gonzalez, W. L. S. Li, S. Abu-El-Haija, P. Battaglia, N. Bulut, J. Halcrow, F. M. G. de Almeida, P. Gonnet, L. Jiang, P. Kothari, S. Lattanzi, A. Linhares, B. Mayer, V. Mirrokni, J. Palowitch, M. Paradkar, J. She, A. Tsitsulin, K. Villela, L. Wang, D. Wong, and B. Perozzi, "TF-GNN: Graph Neural Networks in TensorFlow," *CoRR*, vol. abs/2207.03522, 2023.

[21] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, pp. 1928–1937, PMLR, 2016.

[22] V. P. Dwivedi and X. Bresson, "A Generalization of Transformer Networks to Graphs," in *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[23] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," in *International Conference on Learning Representations (ICLR)*, 2016.

[24] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.