Minimum Cost Reconfigurable Network Template Design with Guaranteed QoS

Xiaoli Xu, Darryl Veitch, *Fellow, IEEE*, Yonghui Li, *Fellow, IEEE,* and Branka Vucetic, *Fellow, IEEE*

***Abstract*—Conventional networks are based on layered protocols with intensive cross-layer interactions and complex signal processing at every node, making it difficult to meet the ultra-low latency requirement of mission critical applications in future communication systems. In this paper, we address this issue by proposing the concept of *network template*, which allows data to flow through it at the transmission symbol level, with minimal node processing. This is achieved by carefully calibrating the inter-connecting links among the nodes and pre-calculating the routing/network coding actions for each node, according to a set of preconfigured flows. In this paper, we focus on the minimum cost network template design to minimize the connections within the template, while ensuring that all the pre-defined configurations are feasible with the guaranteed throughput, latency and reliability. We show that the minimum cost network template design problem is difficult to solve optimally in general. We thus propose an efficient greedy algorithm to find a close-to-optimal solution. Simulation results show that the construction cost of the templates obtained by the proposed algorithm is very close to a lower bound. Furthermore, the construction cost increases only slightly with the number of pre-defined configurations, which confirms the flexibility of the network template design.**

***Index Terms*—Ultra-low latency, reconfigurable network template, network coding, linear optimization, greedy algorithm, quality-of-service (QoS).**

I. INTRODUCTION

As one of the key 5G communication services defined by the International Telecommunication Union (ITU) [1], ultra-reliable low-latency communications (URLLC) find a wide range of applications, such as tele-surgery [2], intelligent transportation [3], and industrial automation [4]. However, existing communication networks have been designed mainly to maximize network capacity, with reasonable reliability and latency that are suited to human perception, but far from the requirements of the above-mentioned critical applications. Specifically, the existing Internet is based on layered protocols, with extensive interactions between different layers and protocols that usually lead to significant communication delay [5]. Furthermore, the complex physical-layer signal processing, e.g., encoding and decoding, which is usually compulsory for each node in the network, adds another source of delay [5]. In the latest 3GPP specifications, the targets for the end-to-end delay of various URLLC applications are set within the range from 5 to 60 milliseconds (ms) [6]. Note that the 5G targets do not really meet the requirements of some critical applications, such as robotic motion control, which requires end-to-end latency of 1 ms or less [7].

In this paper we propose a new communication paradigm for supporting URLLC applications based on the novel concept of a *network template*. A network template is a wired network building-block with a small number of nodes and a carefully calibrated set of inter-connecting links, capable of efficiently transporting pre-configured sets of data flows across the template at the transmission symbol level, using corresponding pre-calculated routing and network coding actions installed in each node. The flows are defined by sets of throughput, latency and reliability requirements, from specified source and destination nodes within the template. Any node can be a source and destination for one or more flows, and an intermediate node for other flows. The scale of a template is intermediate to that of local and wide area networks. It can be thought as a sparse backbone network spanning a small city, which can act as a building block for a wired wide-area inter-connection network, ultimately connecting edge devices (mainly wireless) across a wide area.

Unlike conventional networks, the tight design of the template allows data to flow through it at the transmission symbol level, with node processing limited to per-symbol routing and network coding actions, which can be executed with minimal delay in hardware as they are *deterministic*, i.e., they do not depend on symbol contents. Essentially, the data enters and exits template nodes with no processing delay, with decoding performed only at the destination nodes. In a sense, the aim of the template is to allow the enviable features of the cross-bar switch of circuit switched networks, namely minimal and deterministic latency from any input to any output, natural support of one-to-many multicasting, and easy reconfigurability, to be reinvented in a more powerful form operating at the scale of a Local Area Network and beyond. Specifically, templates have the potential to allow the benefits over more general topologies than a bipartite graph, to support inter-penetration of data rather than being limited to circuit switching, and to support variable requirements in each of delay, bandwidth and redundancy.

Templates can be flexibly (re)configured to transport any one of a number of alternative flow-sets. The required actions for each configuration are pre-calculated and stored in each node. Hence, once activated a template-wide switch can be effected rapidly, in principle of the order of the largest latency

The work of Branka Vucetic was supported in part by the Australian Research Council Laureate Fellowship grant number FL160100032. *(Corresponding author: Yonghui Li.)*

X. Xu is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (email: xiaolixu@seu.edu.cn)

D. Veitch is with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney, Australia (email: darryl.veitch@uts.edu.au).

Y. Li and B. Vucetic are with the School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia (email: {xiaoli.xu, yonghui.li, branka.vucetic}@sydney.edu.au).

across the template so that in-transit flows can clear, with some guard band. It is envisaged that configuration switching occurs on long timescales, perhaps hours or even weeks.

Whereas each configuration exploits the given (fixed) topology, the topology itself will be chosen depending on the overall flow requirements. Topology design in this framework is in fact the key object of this paper, using link construction cost as a metric, as this would be the dominant cost at the envisaged template scale. Although we focus on the capabilities of a single generic template in this paper, the longer term goal is to show how larger networks may be constructed by inter-connecting multiple templates which overlap geographically, leading ultimately to Wide Area Network end-to-end symbol transport.

Our work is fundamentally different from the concepts of software-defined networking (SDN) and network function virtualization (NFV) [8], as these still operate within the paradigm of packet based switching, with packet headers and all the latency their transport and processing implies. Our proposed network template is an attempt to create a *fundamentally different approach to networking*, combining the low latency of circuit switching, with the flexibility and fine time granularity of packet switching.

To highlight how significant this difference is, Fig. 1 contrasts the operations required by a packet switching node (left) compared to a template node (right). The top row of the figure covers the case of routing: comparing required node operations during packet forwarding against the template's symbol stream forwarding. The bottom row considers the case of network coding, which requires additional signal processing and packet processing to effect packet content modification in the packet switched case, compared to relatively simple operations in the template case, as they reduce to preconfigured deterministic operations at the stream level.

The minimal latency design of the template precludes the possibility of per-node error correction, and so symbol errors may accumulate across the template.To meet the end-to-end reliability requirement, the bit-error-rate (BER) per link needs to be sufficiently small and the number of hops should be limited. Therefore, in this paper, we focus on small-scale local area networks with wired connections, which can be engineered to very high reliability. Furthermore, when the achievable data rate is higher than the target rate requirement, redundant information bits can be sent to enhance reliability, a natural benefit of the network coding capability of the template.

There are many new challenges to be addressed in order to bring the template concept into reality, in particular the design of switches powering each node, which is fundamentally different from the conventional network node. However, we note that the node processing in the proposed template is limited to per-symbol routing and deterministic network coding actions, which is basically symbol-independent linear combinations. Hence, the complexity for the network node design is actually greatly reduced. One possible choice for the new network device is a generalised cross connect-based switch augmented with network coding functionality. Another challenge for network template design is how the

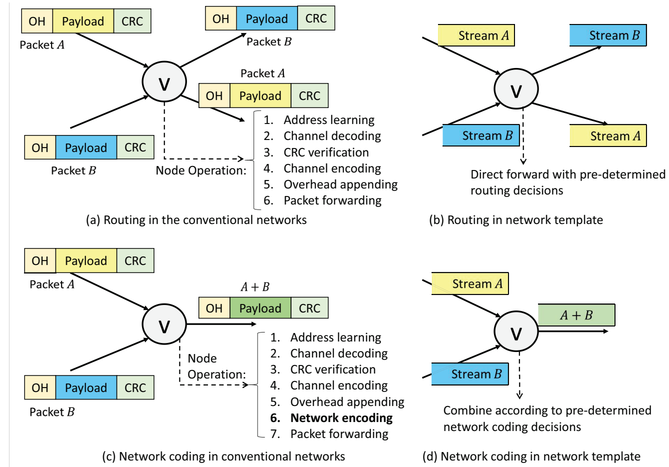


Fig. 1: Comparison of the operations of a node V in a conventional packet switched network (left) and in the proposed template (right). Top row: example routing action. Bottom: example network coding action.

tight symbol-level synchronization can be achieved so that the network encoding and decoding can be performed exactly as designed across the template. In this paper, we assume that all the nodes are well synchronized and each of them is able to copy arriving symbols, forward them to multiple ports and/or perform simple linear network coding[1]. We focus on designing the minimum set of network template connections for supporting the pre-defined configurations with guaranteed quality-of-service (QoS), i.e., the throughput, latency and reliability requirements. Each configuration specifies a set of concurrent flows over the network and the associated QoS requirements of each flow. We need to find the respective routing/network-coding operations at the intermediate nodes for each configuration, so that all the flows can be realized with their desired QoS. Intuitively, the more connections we build between nodes, the more and shorter communication paths we can create, leading in general to higher throughput and lower latency. However, adding a wired link between a pair of distant nodes can be very costly. Therefore, it is important to identify the cost-effective constructions for network template, based on the pre-defined configurations.

Note that the minimal cost network template design problem is challenging in general, since we need to consider different configurations, multiple concurrent flows per configuration and multiple QoS requirements. In fact, even for a given network topology, i.e., a network template with established connections, determining whether a single multicast flow with a target rate and latency can be supported is proven to be NP-hard [10]. Significant effort has been devoted to analyzing the performance of a network with given topology. Specifically, if there is no latency or reliability requirement, the maximal throughput of a unicast flow, i.e., sending from a single source to a single receiver, can be achieved with pure routing and the optimal routing scheme can be obtained

[1]The assumption of tight synchronization can be relaxed in practical implementation if only routing is adopted. In the presence of network coding, this assumption can also be relaxed by introducing pilots for stream alignment [9].

in polynomial time [11]. For a multicast flow, i.e., sending from a single source to multiple receivers, allowing replication as needed at branch points, finding the optimal routing is usually reduced to the minimum Steiner tree problem in graph theory, which is proven to be NP-complete. When network coding is allowed, the maximal multicast rate can be achieved with simple random linear network coding (RLNC) over a directed graph [12]. It is shown in [13] that the maximal rate for network coded multicast over an undirected graph (i.e., where communication can occur in both directions, subject to a capacity limit in each direction) can be solved with linear optimization. When certain delays are introduced for the network links and a maximal tolerable delay is specified for the flow, the optimal achievable rate becomes more difficult to obtain. Specifically, the optimal multicast routing with delay constraint is shown to be NP hard [14]. While it is well known that network coding provides no throughput gain over routing for the conventional unicast networks, recent research has shown that it outperforms pure routing in delay-constrained unicast networks [15]. However, it is not known how to calculate the optimal network coding rate in general delay-constrained networks yet, even for the simple unicast flow [15] [16].

For network template design, we assume that the locations of network nodes are fixed and a connection between any pair of nodes can be established with a given cost. Hence, there is no given network topology as in the above-mentioned works [10]–[15]. On the other hand, network template design is equivalent to finding the minimum cost subgraph of a fully-connected network, such that this subgraph can support different flows in different configurations with their corresponding QoS requirements. The minimal cost multicast subgraph selection with network coding that achieves the optimal multicast rate over a directed graph can be formulated into a linear programming problem and solved in polynomial time [17]. However, when the target data rate is lower than the optimal multicast rate, the algorithms presented in [17] do not guarantee to return the minimum cost subgraph. A linear programming algorithm that yields an acceptable solution for minimal cost multicast subgraph selection, with required data rate less than the capacity, was presented in [18]. When the cost of each link is associated with its congestion probability, a linear programming problem is formulated in [19] to maximize the throughput subject to the cost constraint. However, the above-mentioned works [17]–[19] assume a single network configuration with a single multicast flow, and no latency or reliability requirement, and hence are not applicable for the network template design problem considered in this paper.

In this paper, we propose a greedy algorithm for solving the minimal cost network template design problem. We start from a fully-meshed network, and sequentially trim the most costly link, as long as all the pre-defined configurations are still feasible. A feasible configuration over a network means a routing/network-coding protocol such that all the flows defined by the configuration can be realized with the target throughput, latency and reliability. To tackle this difficult problem, we first propose a sufficient condition to achieve the target throughput with tolerable delay, by considering only those paths that meet the delay requirement. Then, we formulate a linear optimization problem to find the most reliable subgraph that meets this condition. If the returned objective value meets the reliability constraint, we conclude that this configuration is feasible. This linear optimization problem is solved for all the pre-defined configurations before trimming a link from the network template. The efficiency of the proposed greedy algorithm is verified by comparing the obtained results with the cost lower bound, which is obtained by assuming that the cost of a link is proportional to its load and the total construction cost is paid only when the link is fully loaded. The main contributions of this paper are summarized as follows.

- First, we propose a new communication paradigm based on reconfigurable network templates, which aims to support ultra-low latency and high data rate communication by avoiding information packetization and complex signal processing at intermediate nodes.
- Second, we formulate the problem for constructing the minimum cost network template and propose an efficient greedy approach to find a high-quality approximate solution to this problem. To evaluate the performance of the proposed greedy algorithm, we propose a linear optimization problem to obtain a tight lower bound for the construction cost.

The rest of the paper is organized as follows. Section II presents the system model and problem formulation of network template design problem to minimize the construction cost. In Section III, a greedy solution is proposed in the special case of exactly one multicast flow per configuration. Furthermore, a lower bound of the template construction cost is also presented in this section. The proposed solution is extended to the general case with multiple concurrent flows per configuration in Section IV. Section V provides the numerical results, and finally we conclude this paper in Section VI.

II. Network Template

A. System Model

Consider a set of network nodes, denoted by $\mathcal{V} = \{v_1, ..., v_n\}$ and a set of pre-defined configurations $\mathcal{F} = \{F_1, F_2, ..., F_k\}$ to be implemented over $\mathcal{V}$. Only one configuration will be in place at any given time, depending on the use case of the template. Each configuration is represented by a set of *concurrent* unicast/multicast flows that share the network resource, i.e., $F_i = \{f_1^{(i)}, f_2^{(i)}, ...\}$. The flow $f_j^{(i)} = (s_j^{(i)} \to T_j^{(i)})$ is defined by a source node $s_j^{(i)} \in V$ and a set of receivers $T_j^{(i)} \subset V$. When $|T_j^{(i)}| = 1$, the flow $f_j^{(i)}$ is known as a unicast flow. Otherwise, it is a multicast flow. Each flow is associated with a throughput, latency and reliability requirement, denoted by a QoS requirement tuple $QoS_j^{(i)} = \{R_j^{(i)}, D_j^{(i)}, \eta_j^{(i)}\}$. The QoS of configuration F_i is guaranteed if the flows $\{f_j^{(i)}, \forall j\}$ can be *simultaneously* realized over the network with the desired throughput, latency and reliability. In other words, for the configuration F_i, there must exist a routing/network-coding protocol such that for all $f_j(i) \in F_i$, the information can be simultaneously sent from

$s_j^{(i)}$ to $T_j^{(i)}$ at the rate greater than or equal to $R_j^{(i)}$, with delay no greater than $D_j^{(i)}$, and with BER less than or equal to $\eta_j^{(i)}$.

In practical applications, there is in general a tradeoff between the *rate* and *reliability*, i.e., reliability can be enhanced by sending more redundant information via proper coding techniques, and this is achieved at the cost of lower rate. In this paper, we assume a fixed coding at the source node and the effect of coding has been considered when specifying the QoS requirement of each configuration. We aim at finding the minimum cost network template and the associated network coding/routing operations to support multiple configurations with their required QoS.

When the template is set with a new configuration, the routing/coding operations at each node are adjusted. We assume that all the nodes in the network can operate synchronously at the symbol level, and that the node operations are aware of symbol offset delays for each incoming link arising from the choices of routing and coding across the template. We also assume that any resulting symbol level buffering that may be needed (a small number of symbols at most) can be implemented in hardware at line rate without presenting a bottleneck to the network coding and routing processing. Once the configuration is set, the symbol streams can flow over the template without any packetization, or channel encoding/decoding.

B. Communication Link Model

To construct the network template, we need to add proper connections among $\mathcal{V}$. We assume that a pair of nodes $u, v \in \mathcal{V}$ can be connected with some standard fiber, denoted as an undirected link $e = (u, v) = (v, u)$. The communication link e is associated with four parameters, namely, the construction cost λ_e, the maximal transmission rate c_e, the communication delay d_e and the induced small bit error rate (BER) p_e. In practice, the maximum transmission rate may be determined by the fiber optic cable type. The construction cost and the induced BER usually depend on the fiber type as well as the distance between the nodes to be connected. The delay associated with link e usually composes of two portions, one fixed portion caused by signal processing (e.g., node synchronization, stream alignment and network coding operations), denoted by $\hat{d}$, and one variable portion induced by propagation delay, denoted by $\tilde{d}(e)$, i.e., $d_e = \hat{d} + \tilde{d}(e)$. In the conventional network as shown in Fig. 1(a), the data processing time can be as large as a few milliseconds (ms) [5], which usually overwhelms the propagation delay. Hence, many existing works have assumed fixed delay associated with each link [15] [16]. On the other hand, with the proposed network template as shown in Fig. 1(b), the processing delay is significantly reduced, which may be comparable to the propagation delay. The template design algorithms proposed in this paper are applicable for general communication link model, with the assumption that the parameters $\{\lambda_e, c_e, d_e, p_e\}$ are given for each potential link e between any pair of nodes.

C. Minimal Cost Template Construction

Our goal is to find the minimum-cost template construction for a given set of nodes $\mathcal{V}$ that can support all the pre-defined configurations over $\mathcal{V}$ with guaranteed QoS (i.e., throughput, latency and reliability). The returned network topology should be a subgraph of the fully-connected network, denoted as $\mathcal{G}_F = (\mathcal{V}, \mathcal{E}_F)$. For the fully-connected network $\mathcal{G}_F$, there exists a link between each pair of nodes, and the collection of all links is denoted as $\mathcal{E}_F \triangleq \{(v_i, v_j), 1 \leq i \leq j \leq n\}$. The construction cost of the fully-connected template is $\sum_{e \in \mathcal{E}_F} \lambda_e$. The hope is that the template construction cost may be significantly reduced by designing the proper routing/network-coding protocols only utilizing a subset of links $\mathcal{E}$. Hence, the template design problem can be formulated as

$$\textbf{P1:} \quad \min_{\mathcal{E} \subseteq \mathcal{E}_F} \sum_{e \in \mathcal{E}} \lambda_e \tag{1a}$$

$$\text{s.t., } \mathcal{G} = (\mathcal{V}, \mathcal{E}) \text{ can support } F_i \text{ with guaranteed } QoS^{(i)}, \quad \forall F_i \in \mathcal{F}. \tag{1b}$$

Example 1. *Consider the toy example shown in Fig. 2. There are 5 nodes located within the area of* $100m \times 100m$*, with 4 at the corners and 1 at the center, as shown in Fig. 2(a). We aim to construct a network template to support two configurations,* F_1 *and* F_2*. Each configuration contains a single multicast flow, namely* $f_1^{(1)} = (v_1 \to \{v_4, v_5\})$*, and* $f_1^{(2)} = (v_2 \to \{v_4, v_5\})$*. We assume each link to have unit capacity and zero BER. As propagation delays will be sub-microsecond for such a network, for simplicity we assume that each link incurs the same delay, equal to one unit. On the other hand cost for connecting a pair of nodes is given as one unit per 100 meters.*

With exhaustive search, we can find the minimal cost given the rate and delay requirement[2]*. Specifically, if each multicast flow wants to achieve unit rate and unit delay, the optimal connection is shown in Fig. 2(b), for a total construction cost of* $4 + 4\sqrt{2}$*. If each flow wants to achieve unit rate and can tolerate a delay of 2, the optimal connection is shown in Fig. 2(c) with cost* $4\sqrt{2}$*. If each flow wants to achieve rate 2 with delay 2, the optimal connection is shown in Fig. 2(d) with cost* $6 + 4\sqrt{2}$*. Note that the optimal template design may not be unique. Consider Fig. 2(d), another template with the same cost can be constructed by directly connecting* (v_1, v_5) *and* (v_2, v_4)*, and removing all the connections with* v_3*.*

In general, given a network topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, checking the condition (1b), i.e., determining whether a flow can be realized with specific rate, delay and reliability requirement, is very challenging. Furthermore, the number of potential network topologies that need to be checked grows exponentially with the number of nodes, which makes exhaustive search infeasible. In this paper, we will propose an efficient greedy algorithm that provides a high-quality approximate solution

[2] Although routing is sufficient in this simple network, for general networks with more nodes, network coding may achieve better performance in terms of throughput, reliability and latency. Furthermore, network coding can also significantly simplify the subgraph selection for multicast flows (which is usually formulated as linear optimizations) as compared with routing, where the problem is NP-hard. Hence, for general template design, all the nodes are assumed to be equipped with network coding functionality.

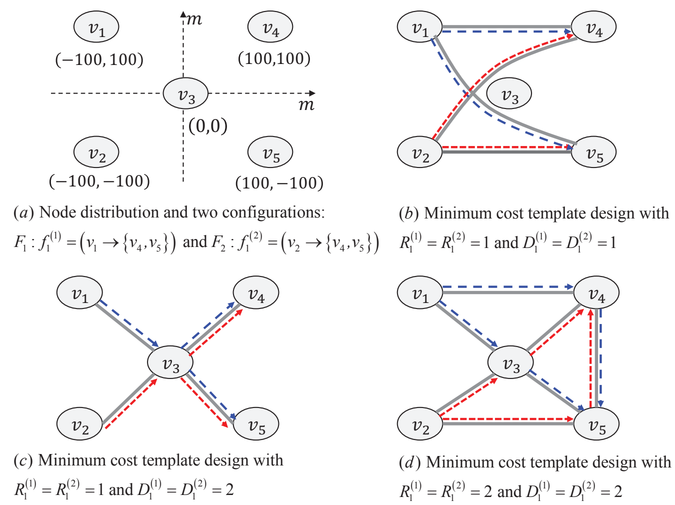


Fig. 2: A toy example to illustrate the network template design. The grey lines represent the physical links to be constructed. The blue and red dashed lines represent the routing protocols to realize $f_1^{(1)}$ and $f_1^{(2)}$, respectively.

to **P1**. The proposed algorithm is firstly presented with the scenario where each configuration has a single multicast flow, and then extended to the general case where each configuration contains multiple concurrent multicast flows.

III. Network Template Design for One Multicast Flow Per Configuration

In this section, we consider the scenario where each configuration contains only one single multicast flow. Hence, we drop the flow index label j. Furthermore, as Sections III-A to III-B develop results for any given configuration, for the moment we also drop the configuration label i. Thus the general flow under consideration is denoted simply as $f = (s \to T)$, with corresponding QoS tuple $QoS = \{R, D, \eta\}$. Multiple configurations are considered in Section III-C and Section III-D.

To find the minimum-cost network template, our main strategy is to start by checking, for each configuration, whether the given requirement can be supported by a given network topology, which initially is a full mesh. Then, a greedy algorithm is proposed to sequentially trim the most costly link, as long as the flow constraints remain valid for each configuration.

A. Throughput for Delay-Constrained Multicast

In this subsection, we start by checking whether a multicast flow can be supported under the throughput and delay requirements only, while ignoring the reliability constraint. For a given undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the capacity of network coded multicast can be found with efficient linear optimization [13]. However, when we impose a delay constraint, characterizing the achievable throughput becomes extremely challenging, even for the simple unicast network [15]. It was shown in [15] that network coding can achieve higher throughput than pure routing in delay-constrained *unicast* networks and the throughput gain can be unbounded for *multicast* networks with increasing network size. However, constructing the optimal network code for delay-constrained networks remains unknown. For the conventional network without delay constraint, it is well known that simple RLNC can achieve network capacity with probability approaching 1 when the field size is sufficiently large [12]. However, the penalty for applying simple RLNC on delay constrained networks can be unlimited [15].

To tackle this problem, we consider a sub-optimal network coding protocol by enabling coding only over those paths that meet the delay requirement. A path is defined as a set of directed links that connect the source and the receiver. For example, consider the network shown in Fig. 2(d), there are two paths from v_1 to v_5, a direct path $P_1 = \{(v_1, v_5)\}$ and a two-hop path $P_2 = \{(v_1, v_4), (v_4, v_5)\}$. The delay associated with a path is defined as

$$d(P) = \sum_{e \in P} d_e, \tag{2}$$

where d_e is the delay associated with link e. r As discussed in Section II-B, the delay associated with each link depends only on its length, which can be calculated according to the node locations. Hence, the latency of all the paths can be calculated from (2) before the rate allocation.

Since the size of network template is usually small, we can list all the possible paths from a source node s to a specific receiver t with maximum delay D, i.e.,

$$\mathcal{P}^{(s,t)}(D) \triangleq \{P = \{(s, \cdot), ..., (\cdot, t)\} : d(P) \leq D\}. \tag{3}$$

For the multicast flow $f = (s \to T)$, we can define the set of paths that meet the delay constraint D over the network $\mathcal{G}$ for each receiver, i.e., $\mathcal{P}^{(s,t)}(D), \forall t \in T$. Let $x(P)$ denote the flow rate assigned over path P. To ensure that the multicast rate R is satisfied, we must have

$$\sum_{P \in \mathcal{P}^{(s,t)}(D)} x(P) \geq R, \quad \forall t \in T. \tag{4}$$

The flow rate assigned to all paths should not violate the capacity constraint of each link. Consider a *directed* link (u, v) where $u, v \in \mathcal{V}$, let $\delta_{(u,v)}(P)$ be the function that indicates whether (u, v) is used for path P, i.e.,

$$\delta_{(u,v)}(P) = \begin{cases} 1, & (u, v) \in P \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

With routing, different information conveyed to different receivers over the same link need to share the channel resource. However, with network coding, the information delivered to different receivers can be combined together, even though they are not identical.

Example 2. *A classic example of network coding is shown in Fig. 3. Consider a single multicast session from s to two receivers t_1 and t_2, i.e., $f = (s \to \{t_1, t_2\})$. Assume that each link has unit capacity and the desired throughput is 2. For each receiver, we assume that only the identified paths listed below meet the delay requirement. Different information need to be sent along the selected paths for each receiver. It is noted that* *information delivered to different receivers can be combined together at each link with network coding, e.g., the information sent over link (v_3, v_4) in Fig. 3, and the receiver can recover the original information with simple linear operations.*

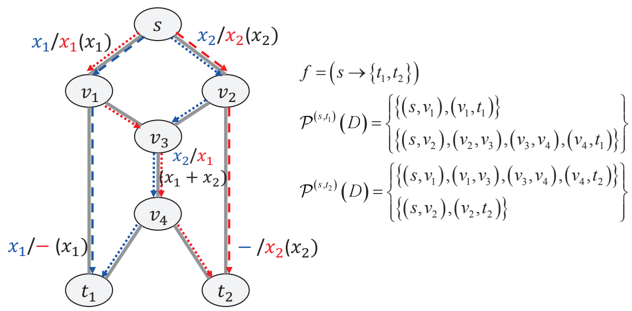


Fig. 3: An example to illustrate the transmission rate after network coding. The blue and red dashed/dotted lines represent the selected paths for t_1 and t_2, respectively. The blue/red labels of the link represent the information to be conveyed to t_1 and t_2, respectively, and the information in the bracket represents the real information sent over the link.

In general, with linear network coding, the transmission rate over directed link (u, v), denoted as $z_{(u,v)}$, is determined by the maximum flow rate for all the receivers [17], i.e.,

$$z_{(u,v)} = \max_{t\in T} \sum_{P\in\mathcal{P}^{(s,t)}(D)} \delta_{(u,v)}(P)x(P). \tag{6}$$

Following the convention in [13], we assume that the capacity of a undirected link $e = (u, v) = (v, u)$ can be arbitrarily allocated to the two directions. Then, the capacity constraint of each link can be expressed as

$$z_{(u,v)} + z_{(v,u)} \le c_e, \quad \forall e = (u, v) \in \mathcal{E}. \tag{7}$$

In summary, the flow $f = (s \rightarrow T)$ can be supported by the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with rate no less than R if we can find the set of flow rates $\{x(P) : P \in \{\mathcal{P}^{(s,t)}(D), \forall t \in T\}\}$ such that (4)-(7) are satisfied. Furthermore, since only those paths that satisfy the delay constraint are utilized, all the coded information that allow the original information to be recovered will arrive at the receivers before the deadline.

Note that when the multicast flow reduces to a unicast flow, i.e., $|T| = 1$, the solution presented above reduces to simple routing, which is shown to be suboptimal for those examples artificially constructed in [15]. We lose the optimality here because the information is restricted to flow along those paths satisfying the delay constraint. If we want to achieve the optimal rate with a delay constraint, we must carefully handle the temporal interference induced by network coding, which will significantly boost the complexity of the network code design [15]. Hence, equations (4)-(7) represent a sufficient but not necessary condition for supporting the multicast flow f with the specific rate and delay requirements, which can be efficiently verified.

B. Reliable Subgraph Selection

To incorporate the reliability constraint into our flow-based network coding solution presented in the preceding subsection, we consider the problem of finding the most reliable subgraph of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that meets the rate requirement, i.e., that satisfies equation (4)-(7).

To minimize the delay, we have assumed that the information is sent through the template without any processing at intermediate nodes, e.g., the channel encoding/decoding. Therefore, erroneous bits will accumulate with communication hops. Since the BER p_e is usually very small for the wired link $e \in \mathcal{E}$ and the admitted paths are usually short due to the delay constraint, the probability that a single bit is flipped more than once along a path is neglected. Hence, a bit remains unaltered over that path P is given by

$$q(P) = \prod_{e\in P}(1 - p_e) \overset{(a)}{\approx} 1 - \sum_{e\in P} p_e, \tag{8}$$

where (a) follows from the fact that $p_e \ll 1$. In other words, the BER of path P is given by

$$\rho(P) \triangleq 1 - q(P) = \sum_{e\in P} p_e. \tag{9}$$

Since the path $P \in \mathcal{P}^{(s,t)}(D)$ will carry flow rate $x(P)$ for receiver t, the expected BER observed by receiver t is given by

$$\rho_t = \frac{\sum_{P\in\mathcal{P}^{(s,t)}(D)} x(P)\rho(P)}{\sum_{P\in\mathcal{P}^{(s,t)}(D)} x(P)} \overset{(b)}{=} \frac{1}{R}\sum_{P\in\mathcal{P}^{(s,t)}(D)} x(P)\rho(P)\ , \tag{10}$$

where (b) follows from the total rate constraint. If the rate constraint in (4) does not hold with equality, we can always reduce the flow rate assigned to the most unreliable path to enhance the reliability, without violating any other constraint.

Next, we just need to find the most reliable subgraph that admits the required data rate R for the multicast flow f, and check whether the achievable reliability meets our requirement. Let κ be the ratio of achievable BER over the desired BER η. Then, the optimal value of κ can be obtained by solving the following linear optimization problem

$$\textbf{P2:} \quad \min_{\{x(P):P\in\{\mathcal{P}^{(s,t)}(D),\forall t\in T\}\}} \kappa \tag{11a}$$

$$\text{s.t.,} \quad \sum_{P\in\mathcal{P}^{(s,t)}(D)} x(P) \ge R, \quad \forall t \in T, \tag{11b}$$

$$z_{(u,v)} + z_{(v,u)} \le c_e, \quad \forall e = (u, v) \in \mathcal{E}, \tag{11c}$$

$$\frac{1}{R}\sum_{P\in\mathcal{P}^{(s,t)}(D)} x(P)\rho(P) \le \kappa\eta, \quad \forall t \in T, \tag{11d}$$

where $z_{(u,v)}$ is the flow rate along the directed link (u, v) after network coding compression. Hence, (11c) represents the capacity constraints for all the edges in the network. Constraint (11b) specifies the rate requirement for all the receiving nodes $t \in T$. In summary, (11b) and (11c) form sufficient conditions to meet the rate requirement with the delay constraint. Equation (11d) corresponds to the reliability constraint. If the optimal value κ^* to problem **P2** is no greater than 1, we can conclude that the multicast flow $f = (s \rightarrow T)$ can be supported over the network $\mathcal{G}$ with the guaranteed QoS.

C. A Greedy Algorithm for Template Construction

In the preceding subsection, we have formulated a linear optimization problem that determines the feasibility of a single multicast flow f over a given undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Based on the formulated problem **P2**, the template design problem **P1** can be further elaborated. Consider the complete graph $\mathcal{G}_F = (\mathcal{V}, \mathcal{E}_F)$ given the set of nodes $\mathcal{V}$, where $\mathcal{E}_F$ includes all the potential links that may be constructed. A link needs to be built only if it is necessary for supporting any flows, i.e., it belongs to a path with non-zero flow rate. Let μ_e be the indicating function for the undirected link $e = (u, v) = (v, u)$ that have to be constructed, i.e.,

$$\mu_e = \begin{cases} 0, & \text{if } \sum_i \sum_{t \in T_i} \sum_{P \in \mathcal{P}_F^{(s^{(i)},t)}(D^{(i)})} \left(\delta_{(u,v)}(P) + \delta_{(v,u)}(P)\right) x(P) = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

where $\mathcal{P}_F^{(s^{(i)},t)}(D^{(i)})$ denotes the set of paths from $s^{(i)}$ to t, with $t \in T^{(i)}$, that satisfy the delay requirement of the ith configuration.

Then, the template design problem **P1** can be re-formulated as

$$\textbf{P3:} \ \min \sum_{e \in \mathcal{E}_F} \mu_e \lambda_e \tag{13a}$$

$$\text{s.t.,} \quad \sum_{P \in \mathcal{P}_F^{(s^{(i)},t)}(D^{(i)})} x(P) \geq R^{(i)}, \quad \forall t \in T^{(i)}, \ \forall \{i : F_i \in \mathcal{F}\}, \tag{13b}$$

$$z_{(u,v)}^{(i)} + z_{(v,u)}^{(i)} \leq c_e, \quad \forall e = (u,v) \in \mathcal{E}, \ \forall \{i : F_i \in \mathcal{F}\} \tag{13c}$$

$$\frac{1}{R^{(i)}} \sum_{P \in \mathcal{P}_F^{(s^{(i)},t)}(D^{(i)})} x(P)\rho(P) \leq \eta^{(i)}, \quad \forall t \in T^{(i)}, \ \forall \{i : F_i \in \mathcal{F}\}, \tag{13d}$$

where (13b) is the rate constraints for all the flows in the set of predefined configurations, (13c) reflects the link capacity constraints, and (13d) is the reliability constraint for the flows. The delay constraints are guaranteed via the path selection process.

Note that the link indicating function μ_e in (13a) is a highly non-convex function with the design variable $\{x(P) : P \in \{\mathcal{P}^{(s,t)}(D), \forall t \in T\}\}$. Therefore, solving **P3** directly is very challenging.

In this section, we introduce a heuristic greedy algorithm for solving **P3**. With the proposed greedy approach, we start from a fully-connected graph with $\mathcal{E} = \mathcal{E}_F$, and then sequentially trim the unnecessary links. For notational convenience, we denote the set of links that are potentially available for trimming from $\mathcal{G}$ by $\mathcal{E}_c$, where $\mathcal{E}_c \subseteq \mathcal{E}$. At each step, we look at the links in $\mathcal{E}_c$ and find the most costly link that can be removed without violating the flow constraints. Note that if a link cannot be trimmed from the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, it cannot be trimmed from any subgraphs of $\mathcal{G}$. Therefore, each link in $\mathcal{E}_F$ is at most evaluated once. The detailed procedures of the proposed greedy algorithm are described as follows:

1) Starting from the fully-connected graph with $\mathcal{E} = \mathcal{E}_F$, for all the predefined configurations $F_i \in \mathcal{F}$, we check the feasibility of flow $f^{(i)}$ by solving the linear optimization problem **P2**. If **P2** is feasible with optimal value no greater than 1 for all configurations, we proceed to the next step. Otherwise, the set of pre-defined configurations are infeasible. We should either relax the requirements or improve the capacity and reliability of each link.
2) Sort all the links in $\mathcal{E}$ according to their cost such that $\lambda_{e_1} \geq \lambda_{e_2} \geq \cdots$. Then, the set of candidate links that are available for trimming can be denoted as a sorted set $\mathcal{E}_c = \mathcal{E} = \{e_1, e_2, ...\}$.
3) Starting from the most costly link in $\mathcal{E}_c$, denoted as e^*, and defining the subgraph $\mathcal{G}_s = (\mathcal{V}, \mathcal{E} \setminus e^*)$, for all predefined configurations, we check the feasibility of $f^{(i)}$ over $\mathcal{G}_s$ by solving problem **P2** and update $\mathcal{E}_c = \mathcal{E}_c \setminus e^*$. If the returned objective value is no greater than 1 for all configurations, we trim link e^* from $\mathcal{E}$, i.e., set $\mathcal{E} = \mathcal{E} \setminus e^*$, and then repeat this step. If e^* cannot be trimmed, we simply repeat this step with $\mathcal{E}$ unchanged.
4) The iterative process in the above step terminates when $\mathcal{E}_c = \emptyset$.

The overall greedy algorithm is summarized in Algorithm 1. From line 3 to line 8, we check the feasibility of the pre-defined configurations in the complete network topology. The iterative process of link trimming is implemented from line 9 to line 18. Note that the dimension of $\mathcal{E}_c$ is reduced by 1 for each iteration. Hence, the maximum number of subgraphs to be considered is no greater than $|\mathcal{E}_F|$. For each subgraph, we need to solve the linear programming problem **P2** at most $|\mathcal{F}|$ times. Therefore, Algorithm 1 requires solving the linear programming problem at most $|\mathcal{E}_F||\mathcal{F}|$ times.

Algorithm 1 Greedy Algorithm for Network Template Design

1: **Input:** $\mathcal{V}, \mathcal{F}, \{\mathbf{QoS}_i, \forall F_i \in \mathcal{F}\}, \{c_e, d_e, p_e, \lambda_e, \forall e \in \mathcal{E}_F\}$
2: **Initialize:** $\mathcal{E} = \mathcal{E}_F$, $\mathcal{E}_c = \{e_1, ..., e_{|\mathcal{E}|}\}$ with links sort in descending order of cost.
3: **for** $F_i \in \mathcal{F}$ **do**
4: Solve **P2** for flow $f^{(i)}$ and network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
5: **if P2** is infeasible or the optimal value $\kappa > 1$ **then**
6: $\mathcal{E}_c = \emptyset$ and output "*The predefined configurations are not feasible*".
7: **end if**
8: **end for**
9: **while** $\mathcal{E}_c \neq \emptyset$ **do**
10: $e^* = \mathcal{E}_c(1)$ and $\mathcal{E}_c = \mathcal{E}_c \setminus e^*$
11: **for** $F_i \in \mathcal{F}$ **do**
12: Solve **P2** for flow $f^{(i)}$ and network $\mathcal{G}_s = (\mathcal{V}, \mathcal{E} \setminus e^*)$
13: **if P2** is infeasible or it has objective value $\kappa > 1$ **then**
14: Go to line 9.
15: **end if**
16: **end for**
17: Trim link e^* by updating $\mathcal{E} = \mathcal{E} \setminus e^*$.
18: **end while**
19: **Output**: $\mathcal{E}$

D. A Lower Bound for the Construction Cost

To evaluate the performance of the proposed greedy algorithm, we present a lower bound of the template construction cost in this section, by relaxing the non-convex objective function in (13a). Specifically, instead of paying the fixed cost for every connection, we assume that the cost is paid per unit rate place on each link, and the total cost λ_e is paid only if the link e is fully loaded.

For notational convenience, denote by $\beta_e^{(i)}$ the load on link $e = (u, v)$ when the network is configured for flow $f^{(i)}$, i.e.,

$$\begin{aligned}\beta_e^{(i)} &= z_{(u,v)}^{(i)} + z_{(v,u)}^{(i)} \\ &= \max_{t \in T^{(i)}} \sum_{P \in \mathcal{P}_F^{(s^{(i)},t)}(D^{(i)})} \delta_{(u,v)} x(P) + \max_{t \in T^{(i)}} \sum_{P \in \mathcal{P}_F^{(s^{(i)},t)}(D^{(i)})} \delta_{(v,u)} x(P)\end{aligned} \tag{14}$$

By comparing $\beta_e^{(i)}$ with the binary indication variable η_e given in (12), it is clear that $\beta_e^i = \eta_e = 0, \forall i$, if link e is not used by any flow. Furthermore, according to the link capacity constraint, we have $\max_i \{\beta_e^{(i)}\} \leq c_e$ if the link has a nonzero load. Hence, the construction cost of the template in (13a) is lower bounded by

$$\sum_{e \in \mathcal{E}_F} \mu_e \lambda_e \geq \sum_{e \in \mathcal{E}_F} (\max_i \beta_e^{(i)}) \frac{\lambda_e}{c_e} = \sum_{e \in \mathcal{E}_F} \beta_e^{\max} \frac{\lambda_e}{c_e}. \tag{15}$$

where $\beta_e^{\max} = \max_i \{\beta_e^{(i)}\}$ is the maximum load on link e over all the configurations. The lower bound can be obtained by solving the following linear optimization problem

$$\textbf{P4:} \ \min \sum_{e \in \mathcal{E}_F} \beta_e^{\max} \frac{\lambda_e}{c_e} \tag{16a}$$

$$\text{s.t.,} \ z_{(u,v)}^{(i)} + z_{(v,u)}^{(i)} \leq \beta_e^{\max}, \ \forall e = (u,v) \in \mathcal{E}, \ \forall \{i : F_i \in \mathcal{F}\} \tag{16b}$$

(13b)(13d).

Note that (16b), (13b) and (13d) together form a sufficient condition to support a flow with desired QoS. For some special network topologies, it may not be the necessary condition [15]. Therefore, the cost value returned by **P4** is not necessarily the theoretical lower bound for all distributions of nodes. However, provided that only those paths meeting the delay requirement can be used, **P4** is a valid lower bound, and hence it provides an efficient way to evaluate the performance of the proposed greedy algorithm.

The lower bound of the template construction cost obtained through **P4** may be tight for most practical networks. However, the routing/network-coding solution obtained from **P4** has no intention of making full use of the link capacity since the link cost is assumed to be proportional to link load. Therefore, the routing/network-coding solution obtained via **P4** cannot be directly applied in the original template design problem in **P3**.

Example 3. *Consider the network shown in Fig. 2(a). When* $\{R^{(1)} = R^{(2)} = 1, D^{(1)} = D^{(2)} = 1\}$, *both Algorithm 1 and* ***P4*** *return the same template construction as shown in Fig. 2(b). When* $\{R^{(1)} = R^{(2)} = 1, D^{(1)} = D^{(2)} = 2\}$ *and* $\{R^{(1)} = R^{(2)} = 2, D^{(1)} = D^{(2)} = 2\}$, *Algorithm 1 still returns the same template construction as shown in Fig. 2(c) and Fig. 2(d), respectively. However, the templates returned by* ***P4*** *are different, which are shown in Fig. 4. If we assume that the cost is proportional to link load, the template designs shown in Fig. 4(a) and Fig. 4(b) have construction cost* $\frac{10+4\sqrt{2}}{3}$ *and* $5 + 4\sqrt{2}$, *which are slightly lower than that for Fig. 2(c) and Fig. 2(d), respectively. This example validates that* ***P4*** *returns a lower bound of the optimal template construction cost. However, consider the original problem in* ***P3*** *where the link cost is a fixed value, the template designs shown in Fig. 4(a) and Fig. 4(b) require much higher construction cost than that returned by Algorithm 1.*

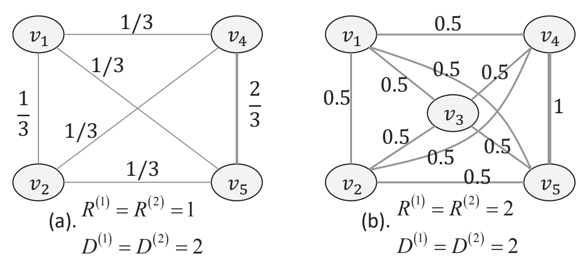


Fig. 4: The resulting template design by solving **P4**, with the links labels representing the corresponding loads.

IV. Network Template Design for Multiple Multicast Flows Per Configuration

In this section, we consider the general case with multiple multicast flows per configuration, i.e., with $|F_i| \geq 1$. In other words, there are multiple flows sharing the network resource at the same time. Note that the information generated by different sources (i.e., belonging to different flows) may interfere each other. For such cases, inter-session network coding is necessary to achieve the optimal capacity region of the network, by maximizing the desired information rate and minimizing the interference. However, the construction of the optimal inter-session network coding for general networks topologies remains unknown. There are some strategies for designing efficient, though sub-optimal, inter-session network code. For instance, a "poison-antidote" approach was proposed in [20] by searching for the butterfly structures in the network using linear optimization method. Besides, Wang et al. [21] introduced a pairwise inter-session network coding approach, where the coding operations are restricted to two sessions to reduce the inter-session interference. In [22], the authors proposed to apply RLNC at the intermediate node, with proper precoding at the source node for minimizing the inter-session interference. However, it is not clear how to extend the above-mentioned strategies [20]–[22] to delay-constrained networks. Specifically, if inter-session network coding is adopted, not only the desired information, but also the "antidote" that help to cancel the inter-session interference must be available at the receivers before the deadline, which is difficult to be guaranteed in the above-mentioned strategies [20]–[22].

To tackle this problem, we restrict to the intra-session network coding, which is shown to be very effective, though not optimal, for conventional multi-session networks without delay constraint [23]. Specifically, we divide the network into a few subnetworks each supporting one flow, where network coding is only applied within the flow. Since the information belonging to different flows is conveyed using separate network resource, there is no inter-session interference observed at the receivers. Following similar procedure as in Section III, we will formulate a linear optimization problem to check whether a configuration can be supported with a given network topology. Then, the greedy algorithm presented in Alg. 1 can be used to find the most cost effective template design that can support all the configurations. For brevity, the superscript i that represents the configuration index will be omitted.

Note that each configuration contains multiple flows in this scenario, i.e., $F = \{f_1, f_2, ...\}$. For the jth flow in configuration F, denote by $\mathcal{P}^{(s_j,t_j)}(D_j)$ the set of paths from node s_j to receiver t_j with delay no greater than D_j, for $t_j \in T_j$. Further denote by $z_j(u,v)$ the transmission rate allocated to the flow f_j over the directed link (u,v). Since intra-session network coding is applied, the coding rate is equivalent to the maximum flow rate over different receivers per multicast flow, i.e.,

$$z_j(u,v) = \max_{t_j \in T_j} \sum_{P \in \mathcal{P}^{(s_j,t_j)}(D_j)} \delta_{(u,v)}(P) x(P), \quad \forall f_j \in F, \tag{17}$$

where $\delta_{(u,v)}(P)$ is a binary indicating function defined in (5) and $x(P)$ is flow rate on the path P.

Without inter-session network coding, there is no further compression across different flows within each configuration. Therefore, the total transmission rate over the directed link (u,v) to support configuration F is given by

$$z_{(u,v)} = \sum_j z_j(u,v). \tag{18}$$

Then, the target throughput $\{R_j : f_j \in F\}$ can be achieved within the corresponding delay constraint $\{D_j : f_j \in F\}$ if we can find the proper flow rate tuple $\{x(P) : P \in \mathcal{P}^{(s_j,t_j)}(D_j), \forall f_j \in F\}$ such that the following constraints can be satisfied simultaneously,

$$\sum_{P \in \mathcal{P}^{(s_j,t_j)}(D_j)} x(P) \geq R_j, \quad \forall t_j^{(i)} \in T_j, \forall f_j \in F \tag{19}$$

$$z_{(u,v)} + z_{(v,u)} \leq c_e, \quad \forall e = (u,v) \in \mathcal{E}, \tag{20}$$

where $z_{(u,v)}$ is the total transmission rate over the directed link (u,v) by all the flows in the configuration.

Next, we need to ensure that the reliability constraints are also satisfied. By following similar procedure as the preceding subsection, we can formulate the reliable subgraph selection as a linear optimization problem. Since different flows within the same configuration have different BER targets, we define κ as the maximal ratio between the achievable BER over the target BER over all the flows within the configuration, which is to be minimized via

$$\textbf{P5:} \quad \min_{\{x(P): P \in \{\mathcal{P}^{(s_j,t_j)}(D_j), \forall f_j \in F\}\}} \kappa \tag{21a}$$

$$\text{s.t.,} \quad \frac{1}{R_j} \sum_{P \in \mathcal{P}^{(s_j,t_j)}(D_j)} x(P)\rho(P) \leq \kappa\eta_j, \quad \forall t_j \in T_j, \forall f_j \in F, \tag{21b}$$

$$(19), (20), \tag{21c}$$

where $\rho(P)$ is the expected BER of path P and η_j is the target BER for flow f_j. We conclude that the configuration $F = \{f_1, f_2, ...\}$ can be supported by the network $\mathcal{G}$ with guaranteed QoS if the problem **P5** is feasible with optimal value no greater than 1.

As a result, the greedy algorithm presented in Section III-C can be applied directly with only a small change, i.e., instead of solving problem **P2** in line 15 of Algorithm 1, we need to solve a more general problem **P5**.

Example 4. *Consider the network shown in Fig. 2(a). We assume that the two flows, $(v_1 \to \{v_4, v_5\})$ and $(v_2 \to \{v_4, v_5\})$ belong to the same configuration, and hence they need to be realized simultaneously. Although the template design presented in Fig. 2(c) can support both flows at rate 1 and delay 2 at different time, it is not sufficient to support both flows simultaneously at the same rate. By executing the modified Algorithm 1, we can find the new template design as shown in Fig. 5.*

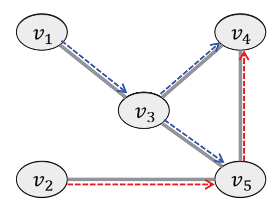


Fig. 5: The template designed by Algorithm 1 for multiple concurrent flows, where the blue and red dashed lines represent the routing paths for realizing two flows simultaneously.

V. Simulation Results

We consider a set of n nodes uniformly distributed within the area of $1000m \times 1000m$, i.e., $|\mathcal{V}| = n$. There are k predefined configurations denoted by $\mathcal{F} = \{F_i, i = 1, ..., k\}$. Each configuration contains m flows, i.e., $F_i = \{f_j^{(i)}, j = 1, ..., m\}$. For flow $f_j^{(i)}$, the source node $s_j^{(i)}$ is randomly selected from $\mathcal{V}$ and the set of receivers $T_j^{(i)}$ are randomly selected from $\mathcal{V} \setminus s_j^{(i)}$. One illustrative network is shown in Fig. 6.

We assume that the cost for constructing a link between a pair of nodes is proportional to the distance between them. For example, consider a link e with distance $l(e)$, the cost for constructing the link is set as $\lambda_e = l(e)/100$, i.e., unit cost per 100 meters. Furthermore, each link is assumed to have unit capacity, with bit flip probability 10^{-9}, i.e., $c_e = 1$ and $p_e = 10^{-9}$, $\forall e \in \mathcal{E}_F$. The processing delay is common to all the links and set as $\hat{d} = 0.01$ ms. The propagation delay is determined by the length of the link, which is set as $\tilde{d}(e) = \frac{l(e)}{3\times10^5}$ ms. The total delay associated with link e is given by $d_e = \hat{d} + \tilde{d}(e)$.

First, we consider the case where each configuration contains a single multicast flow, i.e., $m = 1$. All the flows are assumed to have the same QoS requirement, i.e., with rate constraints $R^{(i)} = R$, delay constraints $D^{(i)} = D$ and reliability constraints $\eta^{(i)} = \eta$, for $i = 1, ..., k$. The simulation results shown in Fig. 7 are generated by 100

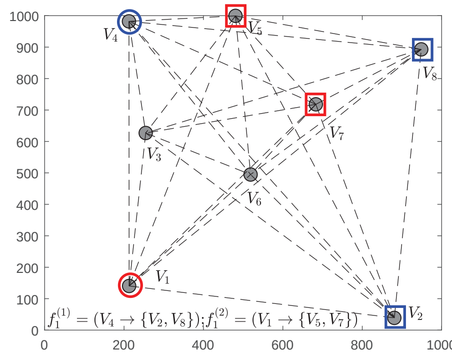
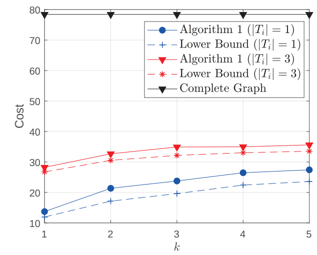
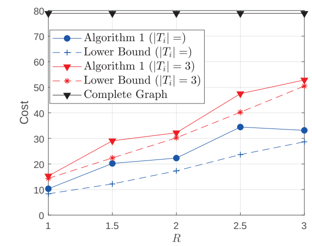


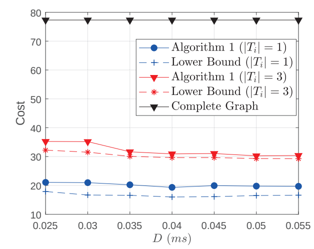
Fig. 6: An illustrative network with $n = 8$, $m = 1$, $k = 2$ and $|T_j^i| = 2$. The dashed lines denote the potential links that can be built. The source and receivers of different configurations are labelled with circles and squares of different colours, respectively.



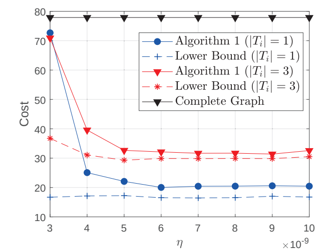
(a) Cost versus the number of configurations with $D = 0.035$ ms, $R = 2$ and $\eta = 10^{-8}$.



(b) Cost versus the throughput requirement with $k = 2$, $D = 0.035$ ms and $\eta = 10^{-8}$.



(c) Cost versus the tolerable delay with $k = 2$, $R = 2$ and $\eta = 10^{-8}$.



(d) Cost versus the tolerable BER with $k = 2$, $R = 2$ and $D = 0.035$ ms.

Fig. 7: Comparing the cost of the proposed template design with the cost lower bound and the complete graph, where the network size is $n = 8$ and the number of flows per configuration is 1.

random realizations of the node distributions. It is observed that the network template designed with the proposed greedy algorithm achieves the cost that is close to the lower bound obtained from **P4**. Furthermore, with the proposed design, the resultant templates usually have much lower construction cost than the complete graph, since the unnecessary links are trimmed. This thus validates the efficiency of the proposed algorithm.

Fig. 7(a) plots the template construction cost versus the number of configurations k with $R = 2$, $D = 0.035$ ms and $\eta = 10^{-8}$. It is observed that the total cost increases slowly with the number of configurations and the number of receivers in each multicast flow. This shows the flexibility for supporting different configurations with one single template design. Fig. 7(b) plots the template construction cost versus the throughput requirement R with $k = 2$, $D = 0.035$ ms and $\eta = 10^{-8}$. Note that the cost increases quickly with the throughput requirement since more edge-disjoint paths are required to support higher data rate. Furthermore, it is observed that the lower bound becomes loose for fractional rate requirement. This is consistent with expectation. Since each link is assumed to have unit capacity, the probability of partially loading a link is higher when the target rate is a fractional value. In the original problem **P3**, we need to pay the whole cost even if the link is not fully loaded. On the other hand, for the lower bound computation in **P4**, the cost is paid proportionally to the rate placed on a link. Hence, the gap between the cost values returned by Algorithm 1 with the lower bound gets larger when the target throughput is fractional. Fig. 7(c) plots the template construction cost versus the latency requirement D with $k = 2$, $R = 2$ and $\eta = 10^{-8}$. Note that it is infeasible to achieve rate 2 when $D \leq 0.02$ ms, in which case only direct communication is allowed. Hence, we start from $D = 2$. As expected, the cost generally decreases with the tolerable latency. When the tolerable latency increases from 0.025 to 0.03 ms, some reduction of cost is observed since there are more feasible paths to be selected. However, when the tolerable latency is further increased, the gain becomes negligible. This is in accordance with our expectation, since the longer path usually incurs larger cost and hence has lower probability to be selected. Fig. 7(d) plots the template construction cost versus the reliability requirement η with $k = 2$, $R = 2$ and $D = 0.035$ ms. Similar as the latency, substantial cost reduction is observed when the reliability requirement is relaxed from a stringent value. However, the performance gain reduces with further relaxation of the requirement.

Next, we consider the complex case where each configuration contains multiple flows to be supported simultaneously. For illustration purpose, we assume that there is a single configuration F_1 with m multicast flows, i.e., $F_1 = \{f_1^{(1)}, ..., f_m^{(1)}\}$. The simulation results shown in Fig. 8 are generated by 100 random realizations of the node distributions and multicast flows. The template construction costs are plotted versus the number of concurrent flows m in Fig. 8. Note that only those realizations where the flows are feasible on the complete graph are counted. It is observed that the cost of the network template grows much faster with the number of concurrent flows, as compared with the number of configurations, as shown in Fig. 7(a).

VI. Conclusion

We have introduced the concept of a network template, an attempt to move beyond the limitations of packet based networks, whose layering and complexity inherently create high levels of latency incompatible with the demands of URLLC applications. The network template is based on the idea of wide-area symbol transport, where template nodes perform pre-configured, deterministic networking coding operations on input symbol streams. Templates have the notable

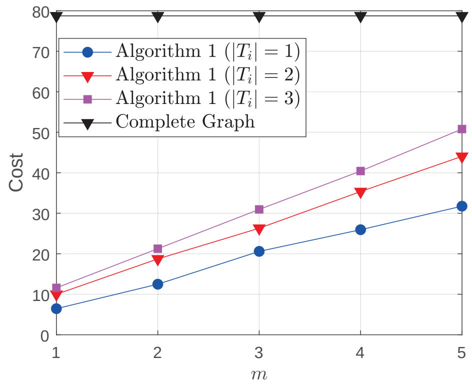


Fig. 8: Cost versus the number of concurrent flows with $n = 8$, $D = 0.035$ ms, $R = 1$ and $\eta = 10^{-8}$.

characteristic that they are small in terms of the number of nodes, making their network coding design feasible, yet significant in terms of geographic extent, implying that link construction is likely to be major cost. This then points to the importance of the design of the topology linking template nodes.

We have investigated the minimum cost network template design problem, for supporting a set of pre-defined configurations with guaranteed QoS. While it is difficult to solve the problem in general, a high-quality approximate solution has been proposed by using an efficient greedy algorithm. A key step was to identify a sufficient but not necessary condition for determining whether a flow is feasible over a given network with guaranteed QoS, which is believed to be of considerable practical interest in itself.

This paper is our first attempt to explore the template concept, with the focus on design and tradeoffs within a single template. Our next step is to examine the template inter-connection problem. That is, to investigate the feasibility of constructing a Wide Area Network by inter-connecting multiple templates which overlap geographically, and to study efficient algorithms for choosing coordinated template configurations to optimize performance across the larger network. We will then compare the performance (latency, data rate and reliability) of the network constructed based on network templates with a conventional packet-based network.

References

[1] ITU-R Rec. M.2083-0, "IMT vision-framework and overall objectives of the future development of IMT for 2020 and beyond," tech. rep., ITU, Sep. 2015.

[2] 5GPPP Association, "5G and e-health," tech. rep., Oct. 2015. available online at https://5gppp.eu/wp-content/uploads/2016/02/5G-PPP-White-Paper-on-eHealth-Vertical-Sector.pdf.

[3] 5GPPP Association, "5G automotive vision," tech. rep., Oct. 2015. available online at https://5gppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf.

[4] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: Challenges and directions," *IEEE Trans. on Industrial Informatics*, vol. 13, pp. 1448–1459, Jun. 2017.

[5] 3GPP TR 36.881, "Study on latency reduction techniques for LTE (release 14)," tech. rep., Jul. 2016. available online at http://www.3gpp.org/ftp//Specs/archive/36_series/36.881/.

[6] 5G Americas Whitepaper, "New services & applications with 5G ultra-reliable low latency communications," tech. rep., Nov. 2018. available online at http://www.5gamericas.org/files/5115/4169/8314/5G_Americas_URLLLC_White_Paper_Final_11.8.pdf.

[7] Gabriel Brown, Principal Analyst, Heavy Reading, "Ultra-reliable low-latency 5G for industrial automation," tech. rep., Qualcomm. available online at https://www.qualcomm.com/media/documents/files/read-the-white-paper-by-heavy-reading.pdf.

[8] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys & Tutorials*, vol. 17, pp. 358–380, Firstquarter 2015.

[9] S. C. L. L. Lu, T. Wang and S. Zhang, "Implementation of physical-layer network coding," in *IEEE Intern. Conf. Commun.*, Jun. 2012.

[10] Q. Liu, L. Deng, H. Zeng, and M. Chen, "On the min-max-delay problem: NP completeness, algorithm, and integer gap," Jun. 2018. available online at https://arxiv.org/abs/1707.02650v4.

[11] J. Schroeder, "Computing the minimum cut and maximum flow of undirected graphs," tech. rep., 2004. available online at http://www.inf.ufpr.br/pos/techreport/RT_DINF003_2004.pdf.

[12] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4413–4430, Oct. 2006.

[13] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 2467–2485, Jun. 2006.

[14] W. Lee, M. Hluchyi, and P. Humblet, "Routing subject to quality of service constraints in integrated communication networks," *IEEE Networks*, vol. 9, pp. 46–55, Jun. 1995.

[15] C.-C. Wang and M. Chen, "Sending perishable information: Coding improves delay-constrained throughput even for single unicast," *IEEE Trans. on Inform. Theory*, vol. 63, pp. 252–279, Jan. 2017.

[16] C. Chekuri, S. Kamath, S. Kannan, and P. Viswanath, "Delay-constrained unicast and the triangle-cast problem," in *IEEE Inter. Symp. on Inform. Theory*, (Hong Kong, China), Jun. 2015.

[17] D. Lun, N. Ratnakar, M. Medard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 2467–2485, Jun. 2006.

[18] T. H. Szymanski, "Max-flow min-cost routing in a future-internet with improved QoS guarantees," *IEEE Trans. on Commun.*, vol. 61, pp. 1485–1497, Apr. 2013.

[19] Y. Xuan and C.-T. Lea, "Network-coding multicast networks with QoS guarantees," *IEEE/ACM Trans. on Networking*, vol. 19, pp. 265–274, Feb. 2011.

[20] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, "Newtork coding for multiple unicasts: An approach based on linear optimization," in *IEEE Inter. Symp. on Inform. Theory*, (Seattle, WA), pp. 1758–1762, Jul. 2006.

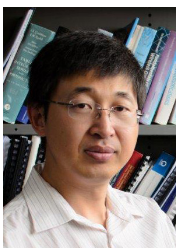
[21] C.-C. Wang and N. B. Shroff, "Pairwise intersession network coding on directed networks," *IEEE Trans. on Inform. Theory*, vol. 56, pp. 3879–3900, Aug. 2010.

[22] X. Xu, Y. Zeng, Y. L. Guan, and T. Ho, "An achievable region for double-unicast networks with linear network coding," *IEEE Trans. on Commun.*, vol. 62, pp. 3621–3630, Oct. 2014.

[23] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *IEEE Trans. on Inform. Theory*, vol. 55, pp. 797–815, Feb. 2009.

Xiaoli Xu is with the School of Information Science and Engineering, Southeast University, China. She received the Bachelor of Engineering (First-Class Honours) and Ph.D. degrees from Nanyang Technological University, Singapore, in 2009 and 2015, respectively. From 2015 to 2018, she was a Research Fellow at the Nanyang Technological University. From 2018 to 2019, she was a Post-doctoral Research Associate at the University of Sydney, Australia. Her research interests include network coding, information theory, vehicular ad-hoc network and UAV communications. She received the Best Paper Award for the 10th IEEE International Conference on Information, Communications and Signal Processing.

Darryl Veitch (F'01) completed a BSc. Hons. at Monash University, Australia (1985) and a mathematics Ph.D. from DAMPT, Cambridge (1990). He worked at TRL (Telstra, Melbourne), CNET (France Telecom, Paris), KTH (Stockholm), INRIA (Sophia Antipolis and Paris, France), Bellcore (New Jersey), RMIT (Melbourne), Technicolor (Paris) and EMUlab and CUBIN at The University of Melbourne, where he was a Professorial Research Fellow until end 2014. He is now a Professor in the School of Electrical and Data Engineering at the University of Technology Sydney. His research interests are centered around computer networking and inference and include traffic modelling, parameter estimation, the theory and practice of active measurement, traffic sampling and sketching, information theoretic security, and clock synchronisation over networks.



Yonghui Li (S'12-M'14) Yonghui Li (M04-SM09-F19) received his PhD degree in November 2002 from Beijing University of Aeronautics and Astronautics. From 1999 C 2003, he was affiliated with Linkair Communication Inc, where he held a position of project manager with responsibility for the design of physical layer solutions for the LAS-CDMA system. Since 2003, he has been with the Centre of Excellence in Telecommunications, the University of Sydney, Australia. He is now a Professor in School of Electrical and Information Engineering, University of Sydney. He is the recipient of the Australian Queen Elizabeth II Fellowship in 2008 and the Australian Future Fellowship in 2012.

His current research interests are in the area of wireless communications, with a particular focus on MIMO, millimeter wave communications, machine to machine communications, coding techniques and cooperative communications. He holds a number of patents granted and pending in these fields. He is now an editor for IEEE transactions on communications and IEEE transactions on vehicular technology. He also served as a guest editor for several special issues of IEEE journals, such as IEEE JSAC special issue on Millimeter Wave Communications. He received the best paper awards from IEEE International Conference on Communications (ICC) 2014, IEEE PIMRC 2017 and IEEE Wireless Days Conferences (WD) 2014. He is Fellow of IEEE.

Branka Vucetic is an ARC Laureate Fellow and Director of the Centre of Excellence for IoT and Telecommunications at the University of Sydney. Her current research work is in wireless networks and the Internet of Things. In the area of wireless networks, she works on communication system design for millimetre wave frequency bands. In the area of the Internet of Things, Vucetic works on providing wireless connectivity for mission critical applications.

Branka Vucetic is a Fellow of IEEE, the Australian Academy of Technological Sciences and Engineering and the Australian Academy of Science.