Q-MiNT: Quantum Multicast via hybrid Multipartite eNTanglement

Suman Kumar*, Ved Dharkar*, Awinash Kumar†, Reshmi Mitra‡
*Department of Computer Science, Troy University, Troy, AL, USA
†Department of Mathematics, SS College Jehanabad, Magadh University Bodh Gaya, Bihar, India
‡Department of Computer Science, Southeast Missouri State University, Cape Girardeau, MO, USA
Email: {skumar, vdharkar}@troy.edu, awinash@sscollegejehanabad.org, rmitra@semo.edu

***Abstract*—Group-oriented quantum applications, such as conference key agreement, distributed sensing, and one-to-many teleportation require an efficient multicast primitive. We present Q-MiNT, a planner that realizes quantum multicast with a hybrid design: GHZ/cluster states are created across a switch backbone and delivered to users over short edge Bell pairs. Q-MiNT models multicast demands as hyperedges mapped to switch-layer resources (memory, coherence, and attempt capacity) and uses precomputed success probabilities that capture path loss and fusion depth. The planner solves two coupled optimization stages: (i) a MILP for group admission that enforces user exclusivity and per-switch memory/coherence, and (ii) an LP/MILP for throughput allocation to maximize expected GHZ/cluster delivery. Evaluations on metro and regional topologies show a stable memory–throughput knee around 10–15 qubits per switch; beyond this, throughput plateaus while coverage continues to rise. The results provide concrete provisioning guidance and identify when physical-layer improvements (higher fusion success, shorter paths, purification) are more beneficial than additional memory. Q-MiNT is protocol-agnostic and supports concurrent sessions in the emerging quantum internet.**

***Index Terms*—Quantum networks; Quantum Internet; Multipartite entanglement; Quantum multicast; Hypergraph models; Throughput maximization; Quantum memory.**

I. Introduction and Motivation

Quantum networks are a transformative substrate for secure and efficient quantum information exchange among spatially distributed processors. By exploiting superposition and entanglement, they enable tasks unattainable classically, with applications in quantum key distribution, distributed quantum computing, sensing, and machine learning [1]. Early designs and most routing formulations focus on pairwise (bipartite) entanglement, proposing path-based protocols under resource constraints such as limited qubits, finite generation rates, coherence windows, and fidelity requirements [2], [3].

Bipartite-only designs scale poorly for group communication. Serving many users requires numerous pairwise paths and swaps, inflating resource consumption and exposure to decoherence. Even concurrent multipath scheduling primarily improves pair throughput and typically abstracts away per-switch memory and coherence budgets [3]. Many networked applications are inherently group-oriented (e.g., QCKA, distributed sensing/computation), motivating the use of multipartite entanglement (GHZ/cluster) to distribute correlations in parallel, reduce swapping overhead, and lower latency [4]. Recent work shows multipartite resources are practical GHZ extraction from graph states [5], routing via local complementation [6], and distribution under noise with memory limits [7] and multi-user GHZ has been demonstrated over tens of kilometers of fiber [8].

Quantum multicast arises naturally in group-oriented applications such as conference key agreement, distributed sensing, and one-to-many teleportation. Classical networks realize multicast via branching trees; the quantum analogue prepares a single multipartite resource (e.g., GHZ or cluster state) and distributes its shares to all group members. Bipartite-only designs emulate multicast by stitching many Bell pairs and performing repeated swaps along multiple paths, inflating memory usage, latency, and decoherence exposure. However, recent advances are largely protocol-centric and stop short of a network-layer planner that (i) enforces per-switch memory and coherence budgets, consistent with link-layer systemization [9], and (ii) jointly decides which multipartite groups to admit and how to allocate entanglement-generation capacity.

Our pososed solution, Q-MiNT treats multicast as a first-class network objective. It creates a GHZ/cluster state across a backbone of switches (a Steiner tree) and delivers shares to users via short edge Bell pairs. A switch-level hypergraph captures multicast groups (hyperedges) and their backbone embeddings, enabling a two-stage optimization: (i) group selection under per-switch memory/coherence budgets (MILP), and (ii) throughput allocation across admitted multicast groups (LP/MILP). In our hypergraph, each selected hyperedge corresponds to a multicast session, and its switch embedding realizes the multicast (Steiner) tree for GHZ/cluster distribution.

This paper makes reports following findings/contributions:

- A network-layer formulation of quantum multicast using a switch-level hypergraph with user attachments, enforcing per-switch memory and coherence budgets.
- Two-stage optimization one for multicast group admission, and second for allocating generation capacity to maximize expected GHZ/cluster throughput.
- Clean per-switch qubit accounting and precomputed success that captures path loss and fusion depth; scalable to metro, multi-metro, and Internet-like topologies.
- A stable memory–throughput knee at $\sim$10–15 qubits/switch. Large-group performance is physics-limited, motivating investment in fusion success, shorter

paths, and purification.

Rest of the paper is organized as follows. Section II presents background on multipartite resources and surveys related work. Section III specifies the optimization framework and the associated algorithms. The Results and Discussion are presented in IV. Section V concludes wich future directions.

II. Background and Related Work

A. Related Work

Multipartite entanglement for network routing remains relatively underexplored compared to bipartite approaches. Recent efforts focus on protocol-level primitives, such as GHZ extraction via graph-state measurements with small per-user memory [5] and routing using local complementation [6]. Under realistic noise and memory constraints, small GHZ states (e.g., GHZ_3) are often optimal [7], enabling applications like conference key agreement [10]. These methods have been experimentally validated over tens of kilometers [8], establishing key building blocks for multipartite services.

Architectural studies examine measurement-based repeaters under memory and fidelity constraints [11], central-node models [12], and tree-based topologies that outperform chains in scalability and robustness [13]. Decision-theoretic stopping rules tune rate–fidelity trade-offs during GHZ creation [14]. Related bipartite work includes multipath routing, path scheduling [2], [15], and link-layer design for heralding and buffering [9]. Protocol-level evaluations address QCKA efficiency in fixed groups [16], motivating a need for network-layer, topology- and resource-aware planning.

Prior work lacks a network-layer model of quantum multicast that enforces switch-level memory/coherence while jointly optimizing group admission and throughput; evaluations also skew to small or idealized topologies. We instead cast multicast as a two-stage hypergraph MILP: (i) admit groups under memory and exclusivity constraints, and (ii) allocate rates to maximize expected multicast yield. Proposed planner realizes a GHZ-on-backbone / Bell-on-edge hybrid and scales to concurrent sessions on realistic metro/backbone.

B. Background

Owing to page limitations, we direct readers to [17] for a comprehensive introduction to quantum phenomena such as superposition, entanglement, and teleportation, and quantum networking and Internet. Below, we briefly summarize the quantum networking aspects relevant to our proposed work.

Multipartite entanglement involves superposition of three or more qubits. A canonical example is the n-qubit GHZ state:

$$|\text{GHZ}_n\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\big).$$

Cluster states are generated by applying controlled-Z gates along the edges of a graph $G = (V, E)$ to $|+\rangle^{\otimes |V|}$:

$$|G\rangle = \Big(\prod_{(i,j)\in E} \text{CZ}_{ij}\Big) \bigotimes_{i\in V} |+\rangle_i.$$

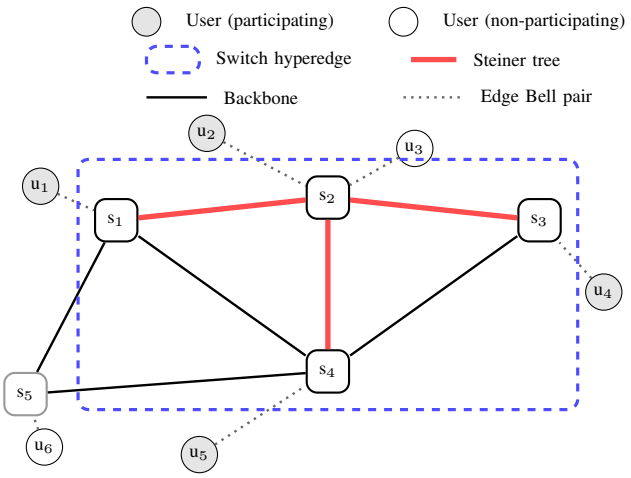

Fig. 1: An example hypergraph showing a group of size 4. The blue dashed region shows the *switch-layer hyperedge* $S_e = \{s_1, s_2, s_3, s_4\}$ preparing the backbone multipartite state. The red thick path is the *Steiner tree* where multipartite occurs. Participating users $U_e = \{u_1, u_2, u_4, u_5\}$ receive shares via dotted switch–user Bell pairs.

Fidelity F: closeness between a produced state ρ and the target $|\psi\rangle$= $\langle\psi|\rho|\psi\rangle$.

Coherence time T_c: time over which stored qubits remain usable (before decoherence).

Success probability P_s: probability of establishing the resource. For a link of length L, $P_s(L) = e^{-\alpha L}$.

III. Q-Mint: A Hybrid Entanglement Multicast

A. Quantum Network Model

An example multicast Quantum network is showing in Figure 1. We model user entanglement demands as a hypergraph $\mathcal{H} = (\mathcal{U}, \mathcal{E})$, where each hyperedge $e \in \mathcal{E}$ corresponds to a user subset $U_e \subseteq \mathcal{U}$ (the group to be served). Each hyperedge e is realized at the switch layer by a set of participating switches $S_e \subseteq \mathcal{S}$ and a set of backbone links $\mathcal{L}_e \subseteq \mathcal{L}$ (typically a Steiner tree) required to distribute a multipartite state to U_e.

a) Quantum Network Components:

1) Quantum users: $\mathcal{U} = \{u_1, \ldots, u_M\}$. Each user requests multipartite entanglement to communicate securely or collaborate across multiple nodes.
2) Quantum switches: $\mathcal{S} = \{s_1, \ldots, s_S\}$. Each $s \in \mathcal{S}$ has limited quantum memory M_s (qubits) and can perform entanglement swapping and multipartite-state generation (e.g., GHZ, W).
3) Quantum links: $\mathcal{L}$ connects switch pairs (s_i, s_j) with success probability $P(l) = e^{-\alpha L_{ij}}$, where L_{ij} is link length and α is the attenuation.

b) Assumptions.:

1) Precomputation: For each e, feasibility (its S_e, a Steiner tree $\mathcal{L}_e$, and auxiliary parameters) is precomputed.
2) Hyperedge realization: We first identify the participating switches S_e, then construct a Steiner tree over S_e to realize the multipartite entanglement.

3) Success probability (backbone):

$$\pi_e \approx \left(\prod_{l \in \mathcal{L}_e} e^{-\alpha L(l)} \right) q^{|S_e|-2},$$

where $L(l)$ is the length of link l and q is the success probability of a fusion/swap at an internal (switch) node.

4) Epoch model: Optimization is per epoch; constraints apply within a single epoch, and entanglement does not persist across epochs.

c) Switch and Link Constraints: Each switch $s \in \mathcal{S}$ must satisfy the following constraints:

1) Memory capacity (per switch):

$$\sum_{e \in \mathcal{E}:\, s \in S_e} m_{e,s}\, y_e \;\le\; M_s, \tag{1}$$

where $m_{e,s}$ is the *per-switch* memory footprint at s if group e is active, $y_e \in \{0,1\}$ indicates whether e is selected, and M_s is the memory capacity of s.

2) Switch-only Number of Qubits.
Let $N_e \equiv |S_e|$ be the number of participating switches and $K_e \equiv |U_e|$ the number of users in group e. Counting only qubits stored at *switches*, a Steiner tree on N_e switches has $N_e - 1$ backbone links and therefore $2(N_e - 1)$ stored Bell halves network-wide. The total switch memory for one group including exactly one switch-side edge buffer per user is

$$Q_{\text{switch}}(e) \;=\; 2\big(N_e - 1\big) \;+\; K_e. \tag{2}$$

3) Coherence

$$y_e\, T_e \;\le\; T_c^{(s)}, \qquad \forall\, s \in S_e, \tag{3}$$

where T_e is the storage time required to realize e and $T_c^{(s)}$ is the coherence limit at switch s.

4) User exclusivity

$$\sum_{e:\, u \in U_e} y_e \;\le\; 1, \qquad \forall\, u \in \mathcal{U}. \tag{4}$$

5) Incidence and overlaps. Let $A \in \{0,1\}^{|\mathcal{U}| \times |\mathcal{E}|}$ be the incidence matrix with $A_{u,e} = 1$ iff $u \in U_e$. Two hyperedges e, e' overlap if $U_e \cap U_{e'} \neq \varnothing$. Under the epoch model, exclusivity enforces a hypergraph matching constraint over $\mathcal{H}$.

6) Switch embedding and classical communication Each hyperedge e is implemented at the switch layer via $S_e \subseteq \mathcal{S}$ and a prescribed entanglement-generation along a Steiner tree (T_e over S_e). classical channels provide coordination, heralding, and scheduling. They are assumed fast and reliable and are not modeled explicitly.

B. Optimization Formulation

a) List of Variables:

1) S_e: the set of switches used in group e.
2) $m_{e,s}$: qubits reserved at switch $s \in S_e$ in group e.
3) T_e: generation time for group e.
4) $\pi_e \in (0,1]$: success probability for a generation attempt of group e.
5) $a_{e,s}$: generation load incurred at switch s per unit allocation rate to group e.
6) M_s: memory capacity of switch s.
7) C_s: generation/attempt capacity of switch s.
8) $T_c^{(s)}$: coherence time for stored qubits at switch s.
9) $y_e \in \{0,1\}$: binary decision variable, equals 1 if hyperedge/group e is selected.
10) $r_e \ge 0$: attempt rate (allocation rate) to group e.

Items 1–5 are precomputed and attached to each e (hyperedge embedding and parameters), allowing optimization at the user layer while enforcing physical constraints at the switch layer. Items 6–8 are network properties.

b) Objective Function I: Group Selection : We select a set of non-overlapping user groups subject to switch memory and coherence constraints (described above).

$$\max \sum_{e \in \mathcal{E}} w_e\, y_e$$

Subject to:

$$y_e \in \{0,1\}, \quad \forall e \in \mathcal{E}. \tag{5}$$

Here w_e is a weight that steers the objective toward different goals (e.g., $w_e = |U_e|$ to maximize served users, or $w_e = \pi_e |U_e|$ to favor reliable groups).

c) Objective Function II: Throughput Allocation: Given the selected groups, allocate generation attempts to maximize expected entanglement throughput. Let $R_{\max}$ be a large bound on feasible allocation per group in the epoch.

$$\max \sum_{e \in \mathcal{E}} \pi_e\, r_e$$

Subject to:

$$\sum_{e \in \mathcal{E}} a_{e,s}\, r_e \;\le\; C_s, \quad \forall s \in \mathcal{S}, \tag{6}$$

$$0 \;\le\; r_e \;\le\; R_{\max}\, y_e, \quad \forall e \in \mathcal{E}, \tag{7}$$

$$r_e \in \mathbb{Z}_{\ge 0} \text{ (or } \mathbb{R}_{\ge 0}). \tag{8}$$

The expected number of successful multipartite states for group e in the epoch is $\pi_e\, r_e$. Constraint (6) enforces per-switch generation capacity; (7) ensures no allocation flows to unselected groups. The hypergraph $\mathcal{H} = (\mathcal{U}, \mathcal{E})$ captures user-side concurrency (via user exclusivity), while the attached per-edge vectors $\{m_{e,s}\}_{s \in S_e}$ and $\{a_{e,s}\}_{s \in S_e}$ map groups to switch-layer resources. Coherence is enforced per participating switch . Precomputing π_e keeps Objective II linear while reflecting path loss and fusion success.

C. Algorithms

The multipartite group selection is NP-complete in its decision form, and throughput allocation is a linear program. If integer attempt rates are required, it becomes NP-hard. Given the NP-hardness of the selection stage, we develop efficient approximation and heuristic algorithms for objective

function I, while solving for objective function II optimally via LP (See [18]). This preserves scalability without sacrificing optimality in the continuous allocation stage.

1) COIN-OR Branch-and-Cut Multipartite Group Selection: The proposed algorithm utilizes the CBC (COIN-OR Branch-and-Cut) solver[1]. CBC solves the LP relaxation and then branches on fractional variables to build a search tree. It adds cutting planes to tighten relaxations and prune the tree. CBC maintains a best feasible (incumbent) and a global bound, stopping when the MIP gap target or time limit is met.

Algorithm 1: Q-MINT Multipartite Group Selection

Input: $\mathcal{E}$; $\mathcal{U}$; $\mathcal{S}$; $U_e \subseteq \mathcal{U}$, S_e, T_e, π_e; $m_{e,s}$; M_s, $T_c^{(s)}$; Weights $(w_e)_{e\in\mathcal{E}}$

Output: Selected group set $\mathcal{E}^\star$ and objective value $Z^\star$.

1 Initialize MILP model $\mathcal{M}$ with CBC backend
2 **foreach** $e \in \mathcal{E}$ **do**
3 create binary variable $y_e \in \{0,1\}$
4 Set objective $Z \leftarrow \sum_{e\in\mathcal{E}} w_e\, y_e$ and maximize Z
5 **foreach** $s \in \mathcal{S}$ **do**
6 add memory constraint $\sum_{e\in\mathcal{E}:\ s\in S_e} m_{e,s}\, y_e \le M_s$
7 **foreach** $e \in \mathcal{E}$ **do**
8 **foreach** $s \in S_e$ **do**
9 add coherence constraint $y_e\, T_e \le T_c^{(s)}$
10 **foreach** $u \in \mathcal{U}$ **do**
11 add user exclusivity constraint $\sum_{e:\ u\in U_e} y_e \le 1$
12 Set CBC parameters: `time_limit` $\leftarrow$ 120 s, `mip_gap` $\leftarrow 10^{-3}$
13 Solve $\mathcal{M}$ and obtain $(y_e^\star)_{e\in\mathcal{E}}$ and $Z^\star$
14 $\mathcal{E}^\star \leftarrow \{\, e \in \mathcal{E} \mid y_e^\star = 1 \,\}$

Result: $(\mathcal{E}^\star,\ Z^\star)$

Algorithm 1 first instantiates a MILP with the CBC backend (Line 1). For each candidate group (hyperedge) it introduces a *binary* selection variable $y_e \in \{0,1\}$ (Lines 2–3). The objective maximizes a weighted utility with the default $w_e = \pi_e|U_e|$ this biases the solver toward selecting larger, more reliable groups (Line 4). It then enforces per-switch memory and per-switch coherence constraints (Lines 7–8), and user exclusivity (Lines 9-10). Finally, the solver runs with a specified time limit and MIP gap (Line 11) and returns the selected set together with the attained objective.

2) CBC/CLP-Based Throughput Optimization with Resource Allocation: Algorithm 2 utilizes COIN-OR LP (CLP)[2], a high-performance LP solver that finds a globally optimal solution to the continuous problem with polynomial-time methods in practice. CBC uses CLP internally for LP relaxations. Because the problem is an LP, the solvers typically return provably optimal solutions very quickly.

Algorithm 2 takes the selected groups $\mathcal{E}^\star$ from Step I and builds a linear program with the CLP/CBC backend (Line 1).

[1]https://github.com/coin-or/Cbc
[2]https://github.com/coin-or/Clp

Algorithm 2: Q-MINT — Throughput Allocation

Input: Selected groups $\mathcal{E}^\star$ from Algorithm 1; $\mathcal{S}$; $a_{e,s}$; $(C_s)_{s\in\mathcal{S}}$; $(\pi_e)_{e\in\mathcal{E}^\star}$;

Output: Optimal attempt rates $\{r_e^\star\}_{e\in\mathcal{E}^\star}$ and $Z^\star$.

1 Initialize LP model $\mathcal{P}$ (CBC/CLP backend)
2 **foreach** $e \in \mathcal{E}^\star$ **do**
3 create continuous decision variable $r_e \ge 0$
4 Set objective $Z \leftarrow \sum_{e\in\mathcal{E}^\star} \pi_e\, r_e$ and maximize Z
5 **foreach** $s \in \mathcal{S}$ **do**
6 add capacity constraint $\sum_{e\in\mathcal{E}^\star} a_{e,s}\, r_e \le C_s$
7 **if** $R_{\max}$ *provided* **then**
8 **foreach** $e \in \mathcal{E}^\star$ **do**
9 add bound $r_e \le R_{\max}$
10 Set solver parameters: `time_limit` $\leftarrow$ 60 s, `optimality_tolerance` $\leftarrow 10^{-8}$
11 Solve $\mathcal{P}$ to obtain $(r_e^\star)_{e\in\mathcal{E}^\star}$ and $Z^\star$

Result: $(\{r_e^\star\}_{e\in\mathcal{E}^\star},\ Z^\star)$

For each $e \in \mathcal{E}^\star$ it introduces a nonnegative continuous attempt-rate variable r_e (Lines 2–3). It then enforces per-switch generation-capacity constraints (Lines 5–6), and, per-group upper bounds (Lines 7–8). Finally, the solver is configured with a time limit and numerical tolerance (Line 9) and solved to obtain the optimal rates and objective (Lines 10–11). The LP allocates rate to groups with the highest gain per unit capacity while balancing loads across switches.

IV. Result and Discussion

This section details the simulation setup, network parameters, evaluation metrics, and performance analysis.

TABLE I: Quantum Network Configuration Parameters

Symbol	Value	Symbol	Value
$T_c^{(s)}$	0.1 s	L_{ij}	[0,25] km
α	0.046 (1/km)	q	0.5
T_e	$2\,d_{\max}/v_{\text{fiber}} + 0.01$	v_{fiber}	2×10^8 m/s
T_{ctrl}	0.01 s	C_s	100

A. Network Parameters and Evaluation Metrics

1) Network Topology: We synthesize topologies using a Waxman model on a $10k \times 10k$ square (1 unit $= 1\,\text{km}$). We place S switches and U users i.i.d. uniformly; users never connect to other users and attach only to switches (e.g., nearest switch). Switch–switch edges are sampled by the Waxman rule (parameters chosen to target an average switch degree of 10) and we cap link length by $\delta = 50/\sqrt{|V|}$ km to avoid long edges. The link attempt capacity is assumed sufficient.

2) Service Mix: The load ratio is fixed at $\lambda = U/S = 1.5$, reflecting *shared* access switches, which is typical of a metro access layer that aggregates a few nearby endpoints. This naturally induces groups in which multiple users share the same access switch.

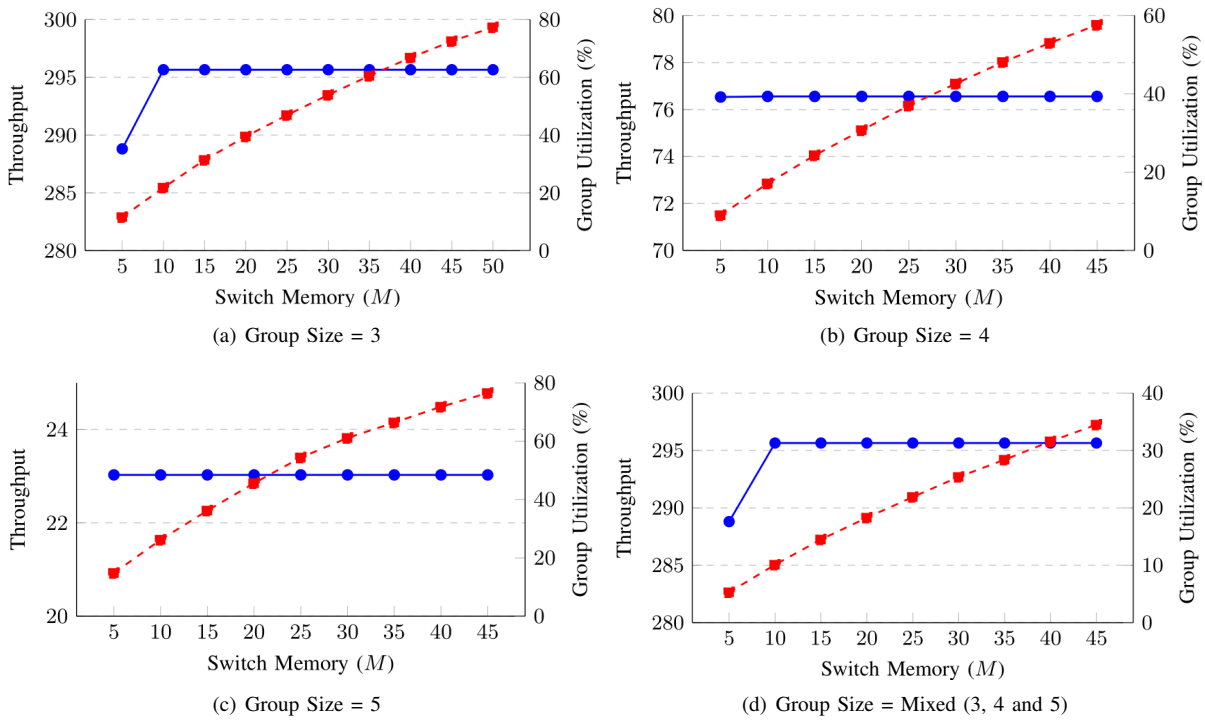

Fig. 2: Throughput and Group Utilization vs Switch Memory, for Number of Switch (S) = 50 and Number of Users (U)= 75.

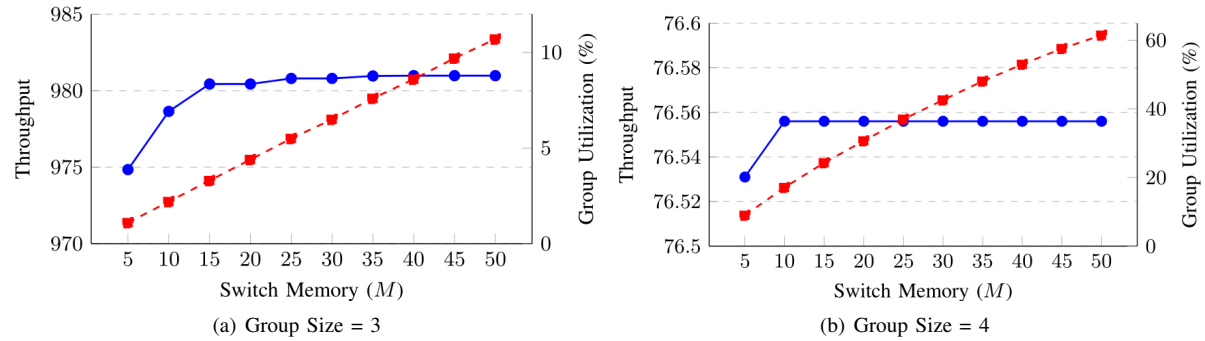

Fig. 3: Throughput and Group Utilization vs Switch Memory, for Number of Switch (S) = 100 and Number of Users (U)=150.

3) Deployment Tiers: Two deployment tiers are considered:

a) Metro/City-scale ($S = 50$, $U = 75$).: We emulate a single metropolitan quantum backbone spanning a 40–80 km urban area. Switches form a degree-5–7 mesh using an RGG.

b) Regional / Multi-metro ($S = 100$, $U = 150$): We model 3–4 metro clusters interconnected by a regional backbone. This setting produces deeper Steiner trees and lowers the group success parameter π_e for larger K.

4) Evaluation Metrics and Network Parameters: We evaluate the planner as a function of the per–switch memory budget M_s (qubits). See Table I for network configuration parameters. Two metrics are reported for different group sizes:

a) Groups Utilization: Let $\mathcal{E}_{\text{cand}}$ be the set of candidate groups, Group Utilization is $100 \times \frac{|\mathcal{E}_{\text{sel}}(M_s)|}{|\mathcal{E}_{\text{cand}}|}$ [%].

b) Expected Throughput: For each admitted group $e \in \mathcal{E}_{\text{sel}}(M_s)$, let P_e denote its per-epoch expected success, the total throughput is $\sum_{e \in \mathcal{E}_{\text{sel}}(M_s)} P_e$.

B. Result

The planner design is prsented by using the per-switch memory M_s for GHZ sizes $K \in \{3, 4, 5\}$ and a mixed workload for Metro and but only GHZ sizes $K \in \{3, 4\}$ for Regional Deployments, owing to page limitations.

1) Metro/City-Scale: Figure 2(a) shows results for $K = 3$ throughput rises steeply for small $M_s \approx 5$ and stabilizes near ~ 296 at $M_s \approx 10$–12 qubits per switch. Coverage continues to grow (from $\sim 10\%$ to $\sim 78\%$) even beyond the throughput knee. At low M_s, the planner is memory-limited. Once buffers and ancillas suffice, more groups are selected and throughput saturates. For Four-Party (Figure 2(b)) shows coverage rising from $\sim 8\%$ to $\sim 58\%$. Moderate multipartite groups benefit from increased memory, but the higher fusion depth ($\propto q^2$)

and longer creation trees limit throughput gains relative to $K = 3$. Coverage improves significantly (from $\sim 18\%$ to $\sim 75\%$), but throughput remains flat around ~ 23 for Five-Party (Figure 2(c)). The system becomes physics-limited for new groups, resulting in negligible gain to the objective.

Figure 2(d) shows a mixed load consisting of parti sizes $K \in \{3, 4, 5\}$. Throughput rises to a plateau ($\approx$76.5) by 10–12 qubits/switch; coverage increases roughly linearly. The MILP admits high-reliability groups first (mostly $K = 3$ and short-path $K = 4$); additional memory mainly expands coverage. As memory increases, additional groups (including longer $K = 4$ and $K = 5$) are admitted, which raises coverage but the flat throughput. Even under heterogeneous demand, the dimensioning point remains at $M_s \approx$ 10–12 qubits/switch. Beyond this, gains appear primarily as increased service coverage, not higher throughput.

2) Regional/Multi-Metro: Figure 3(a) shows result for Tripartite Groups ($K = 3$). It shows rapid rise to a knee at 12–15 qubits/switch, then a tight plateau around $\approx$981. Coverage increases $\approx$1%$\rightarrow\approx$11%. At low memory budgets, the system is memory-limited. The planner prioritizes high-π_e tripartite groups, producing the steep initial rise. Once $M_s \approx$ 12–15 is reached, most "good" groups have been admitted. The regime then becomes physics-limited, where additional groups contribute marginally to throughput while increasing coverage. In the case of four-party group (Figure 3(a)), throughput again rises with increasing M_s, plateauing by $M_s \approx 10$. The qualitative pattern mirrors $K = 3$ but with a lower ceiling and a slightly delayed knee, due to deeper fusion operations ($\propto q^2$) and longer average Steiner trees.

C. Discussion

Above results demonstrate a clear memory–throughput "knee" at roughly the same place across network sizes. In practice, sizing switches to about a dozen usable qubits captures most of the achievable throughput. Bigger networks deliver proportionally more total throughput after the knee, but per-switch/per-user efficiency stays about the same. The percentage of groups selected falls as the network grows (the candidate set explodes), even though the actual number admitted rises. Small–to–mid groups benefit from added memory up to the knee; beyond that, gains taper off. Large groups are limited by physics (loss/fusion success), so more memory boosts availability and not rate. The optimizer naturally prioritizes reliable groups; small selection "wiggles" near the knee are expected. Second stage can be used to funnel attempts toward the most reliable groups. After memory knee is hit, it is better to focus on the physical layer achieve better fusion success, shorter paths/topology choices, or purification.

V. Conclusion

This work introduces a practical, optimization-driven framework for quantum multicast that elevates multipartite resources to a first-class network objective. By combining a hypergraph admission MILP with a throughput-allocation LP/MILP, Q-MiNT enforces per-switch memory and coherence while maximizing expected delivery of GHZ/cluster states. Experiments on realistic topologies reveal a consistent provisioning rule about a dozen usable qubits per switch capture most attainable throughput for small-to-mid group sizes after which the regime becomes physics-limited. Q-MiNT offers a practical and scalable path to support conference key agreement, distributed sensing, and one-to-many teleportation in the quantum internet. Our future work is focussed on time-coupled scheduling with decoherence, integrated purification/coding, a better hardware heterogeneity and user fairness, and larger-scale evaluations on realistic carrier backbones.

References

[1] V. Kumar, C. Cicconetti, M. Conti, and A. Passarella, "Quantum internet: Technologies, protocols, and research challenges," *International Journal of Networked and Distributed Computing*, vol. 13, no. 2, p. 22, 2025.

[2] E. Sutcliffe and A. Beghelli, "Multiuser entanglement distribution in quantum networks using multipath routing," *IEEE Transactions on Quantum Engineering*, vol. 4, pp. 1–15, 2023.

[3] J. Halder, E. Matus, and G. Fettweis, "On the concurrent multipath entanglement distribution in quantum networks," in *GLOBECOM 2024*. IEEE, 2024, pp. 2791–2796.

[4] M. Walter, D. Gross, and J. Eisert, "Multipartite entanglement," *Quantum Information: From Foundations to Quantum Technology Applications*, pp. 293–330, 2016.

[5] V. Mannalath and A. Pathak, "Multiparty entanglement routing in quantum networks," *Physical Review A*, vol. 108, no. 6, p. 062614, 2023.

[6] F. Hahn, A. Pappa, and J. Eisert, "Quantum network routing and local complementation," *npj Quantum Information*, vol. 5, no. 1, p. 76, 2019.

[7] L. Bugalho, B. C. Coutinho, F. A. Monteiro, and Y. Omar, "Distributing multipartite entanglement over noisy quantum networks," *Quantum*, vol. 7, p. 920, 2023.

[8] J. W. Webb, J. Ho, F. Grasselli, G. Murta, A. Pickston, A. Ulibarrena, and A. Fedrizzi, "Experimental anonymous quantum conferencing," *Optica*, vol. 11, no. 6, pp. 872–875, 2024.

[9] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpedek, M. Pompili, A. Stolk, P. Pawelczak, R. Knegjens, J. de Oliveira Filho *et al.*, "A link layer protocol for quantum networks," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19)*, 2019, pp. 159–173.

[10] A. Pickston, F. Grafitti, A. Fedrizzi *et al.*, "Conference key agreement in a quantum network," *npj Quantum Inf.*, vol. 9, no. 1, p. 82, 2023.

[11] J. Wallnöfer, A. Pirker, M. Zwerger, and W. Dür, "Multipartite state generation in quantum networks with optimal scaling," *Scientific reports*, vol. 9, no. 1, p. 314, 2019.

[12] G. Avis, F. Rozpedek, and S. Wehner, "Analysis of multipartite entanglement distribution using a central quantum-network node," *Physical Review A*, vol. 107, no. 1, p. 012609, 2023.

[13] N. Basak *et al.*, "Improved routing of multiparty entanglement in quantum networks," arXiv preprint arXiv:2409.14694, 2024.

[14] A. S. Cacciapuoti, J. Illiano, M. Viscardi, and M. Caleffi, "Multipartite entanglement distribution in the quantum internet: Knowing when to stop!" *IEEE Transactions on Network and Service Management*, 2024.

[15] Y. Zeng, J. Zhang, X. Shang, J. Liu, Z. Liu, and Y. Yang, "Multi-user entanglement routing design over quantum internets," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2024, pp. 266–276.

[16] S. Oslovich, M. Z. Hossain, T. Thomas, B. Wang, W. O. Krawec, and K. Goodenough, "Efficient quantum conference key agreement over quantum networks," in *2025 QCNC*. IEEE, 2025, pp. 315–322.

[17] K. Azuma, S. E. Economou, D. Elkouss, P. Hilaire, L. Jiang, H.-K. Lo, and I. Tzitrin, "Quantum repeaters: From quantum networks to the quantum internet," *Reviews of Modern Physics*, vol. 95, no. 4, 2023.

[18] R. Lougee-Heimer, "The common optimization interface for operations research: Promoting open-source software in the operations research community," *IBM Journal of Research and Development*, vol. 47, no. 1, pp. 57–66, 2003.