On the transmission capacity of quantum networks

Sandra König
Alpen-Adria Universität Klagenfurt
9020 Klagenfurt, Austria

Stefan Rass
System Security Research Group
Alpen-Adria Universität Klagenfurt
9020 Klagenfurt, Austria

***Abstract*—We provide various results about the transmission capacity of quantum networks. Our primary focus is on algorithmic methods to efficiently compute upper-bounds to the traffic that the network can handle at most, and to compute lower-bounds on the likelihood that a customer has to wait for service due to network congestion. This establishes analogous assertions as derived from Erlang B or Erlang C models for standard telecommunications. Our proposed methods, while specifically designed for quantum networks, do neither hinge on a particular quantum key distribution technology nor on any particular routing scheme. We demonstrate the feasibility of our approach using a worked example. Moreover, we explicitly consider two different architectures for quantum key management, one of which employs individual key-buffers for each relay connection, the other using a shared key-buffer for every transmission. We devise specific methods for analyzing the network performance depending on the chosen key-buffer architecture, and our experiments led to quite different results for the two variants.**

Keywords-Quantum network; Quantum cryptography; network transmission capacity; queuing network; system security.

I. INTRODUCTION

It took about two decades ever since quantum key distribution (QKD) has been proposed by [1] (the famous BB84 protocol) until the first experimental implementations of a quantum network were presented by the DARPA [2] and the European Union [3]. While the theory behind secure key-delivery between Alice and Bob is well-understood (see e.g. [4] for a proof regarding the security of BB84), the theory of network design and performance analysis has apparently seen rather limited attention over the last years. The works of [5] and [6] both considered the design of a network from the topological point of view, and in terms of optimal security and performance. In this work, we go the other way, asking for the best performance that we can get from a given quantum network infrastructure. In particular, we provide algorithmic means to answer two questions:

1. What is the maximal transmission capacity achievable in the network (using any classical routing scheme)?
2. What is the likelihood of local congestion that would temporarily disconnect the (logical) channel between any two peers in the network?

The second question can be rephrased into asking how likely a customer is to wait when asking the network for a secure delivery of payload from one point to another.

Our results are hence related to the field of communication theory, channel capacity and network coding. Particularly the latter has led to valuable insights (cf. e.g. [7], [8], [9]) regarding the rate at which information can be send through the network. Contrary to these (and many other related) approaches, we do not employ classical information theory to quantify the capacity, but rather work with the directly known performances of each link in the network. Similar to our work, network outage probabilities are as well discussed in [10], [11], [12] and [13], where most research effort, as it seems, has been put on wireless networks. So far, the problem appears hardly considered in (hard-wired) networks or quantum networks. In the quantum computing domain, the work of [14], [15], [16] and [17] is closely related. Contrary to ours, however, it strongly relies on quantum techniques and is less focused on algorithmic methods to analyze a given infrastructure. The work of [16] is particularly interesting as it employs percolation theory (which is rarely used in the related literature). The problem is studied elsewhere in [18], which comes up with proposals on how to enhance the existing capabilities once they are known. Here, we work out the limits similarly to the Erlang B and Erlang C models, so as to be able to improve them based on this related research. In the wireless domain, the interesting work of [19] deals with spread-spectrum techniques and uses Poissonian processes for determination of the network capacity, but is specific for this particular encoding technique. We explicitly avoid such restrictions here, but adopt some assumptions on the quantum key-management models (following the proposals of [20]; cf. also [21] for another discussion related to quantum key- and network-management). Finally, we mention the work of [22], who attempts to solve the problem of end-to-end quantum communication using a three-party protocol. This architecture is essentially different from what has been implemented nowadays, and thus subject of future considerations.

Among the sales arguments for quantum key distribution is its capability of running over existing fibre-optic lines. This claim has been substantiated in the demonstration of the SECOQC- and DARPA-Networks [2], [3]. Hence it is fair to assume that the topology of the network is fixed and that the individual key-generation rates on each link are known (cf. [3] for examples). Moreover, following the so-far proposed architectures of relay devices in a quantum network (see [2], [3],and [20]), it is reasonable to consider a quantum network as a system of interconnected buffers, where the secret message is repeatedly decrypted and re-encrypted before forwarding it to the next hop. The key-buffers in each relay node are constantly refilled by QKD protocols running in the background 24/7 and

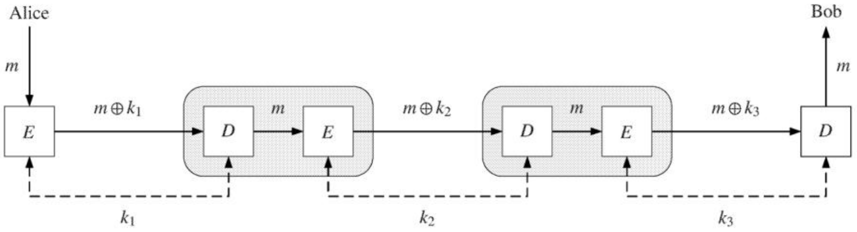


Figure 1 Trusted relay with re-encryption in each hop

endlessly generating key-material for later usage. This transmission regime, in its simplest form, is known as trusted relay, and is widely used in nowadays demonstration networks (cf. [2] and [3]). A transmission of a message m from Alice to Bob along a sequence of nodes that share keys $k_1, \ldots, k_3$ is displayed in Figure 1.

Organization of this work: in Section II, we describe the graph- and queuing model (Section II.A) used to analyze a quantum network. In particular, we will use maximal flows (briefly introduced in Section II.B) to compute bounds on the payload that the network can handle. Section III is divided into two main parts, giving algorithms for computing end-to-end transmission capacity (Section III.A) and waiting probability if a chosen path through the network is blocked due to congestion (Section III.B). In both cases, we show how to use standard maximum flow and shortest path algorithms to compute the desired quantities easily and efficiently from the known link capacities in the underlying graph model. The process is illustrated by an example (Section III.C), before conclusions follow in Section IV.

II. MATHEMATICAL MODEL

We will not burden ourselves with the details of any particular quantum key distribution (QKD) facility, but restrict our attention to the following model of a quantum network (QNet): let a QNet be modeled as an undirected graph $G = (V, E)$ with adjacent nodes sharing secret keys thanks to QKD. That is, on any line $u - v$ (with $u, v \in V$) maintain key-buffers on either side to store QKD key material for subsequent transmissions. These key-buffers are nothing else than queues, in which key-bits are inserted on a deterministic basis (we assume the QKD-devices to generate key-bits at constant rate). key-bits are used up on a random basis, depending on incoming payload for secret transmission. Based on this view, we can cast the QNet into a standard queuing network.

A. Quantum Networks as Queuing Networks

An open queuing network is a system in which a customer enters the network at some node, and moves onwards through the links, where he occasionally has to wait (queue) until he is served at the next node (i.e. he can enter the next node). Central questions in queuing theory regard the average time to wait until the customer reaches his destination point, or the average number of customers lining up in any given queue (link) in the network. For a quantum network, we can equally well set up such a model, based on the following correspondence:

1. incoming customers equal newly generated key-bits
2. leaving customers equal the (one-time) use of key-bits for Vernam-encryption of messages
3. a queue equals a key-buffer (storing bits for subsequent usage, or equivalently, hosting customers for subsequent service)

Observe that the generation of key-bits is deterministic, while the arrival of messages is non-deterministic. If we consider the QNet as a backbone network, then it is reasonable to consider the event of an incoming message bit as a Poisson-distributed random variable. In Kendall-notation, the link $u - v$ in a QNet therefore is nothing else than a $D/M/1$-queue, disregarding physical size limits of the key-buffers for now. The graph G modeling the QNet thus constitutes a network of queues, and we are interested in its stationary distribution (so it exists).

Remark: an alternative view yielding equivalent results is by associating incoming payload bits with customers, who get served (encrypted) based on how much key-material is available in the buffer. In this case, we would have to think of the message-queue as the physical incarnation of the queue model under consideration. For simplicity, and because real-life implementations work with a key-buffer too, we shall consider the first of these two views, keeping the second view in mind whenever needed.

Sufficient conditions for the existence of stationary distributions in queuing networks are well-known, such as Jackson's theorem for Jackson-networks or one of its generalizations, such as the BCMP-theorem. Openness of the network is assured, since the graph G is a mere transportation medium and messages necessarily leaving the network at some stage. Still, we cannot make any generally valid assertions on the routing algorithms implemented within the system. In lack of such information, we will try to find upper bounds to the transmission capacity by invoking maximum flow theory. This has the additional advantage of our results applying to conventional routing as well as network coding approaches for transmission.

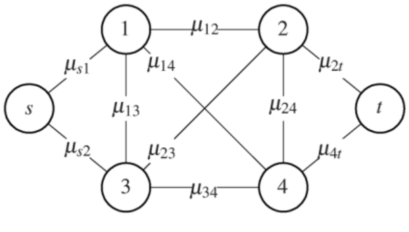


Figure 2 Network with link performances

B. Flows

To illustrate the approach, consider the example network topology shown in Figure 2. Assume that after start-up, all key-buffers are empty and the QKD-protocol on link $v_i - v_j$ start producing key-material at constant rate μ_{ij} per time unit. It follows that after one unit of time, the maximal transmission capacity of the network is determined by the maximum $s - t$-flow, constrained by the existing key-material on each link. As all links regenerate key-bits at constant rate, the minimum cut will not change over time. Let $C \subseteq E$ be the minimum edge cut, then the maximum flow has capacity $c(f_1) = \sum_{(v_i,v_j)\in C} \mu_{ij}$ after the first unit of time. After t units, we have the capacity $c(f_t) = \sum_{(v_i,v_j)\in C} t \cdot \mu_{ij} = t \cdot c(f_1)$. The consumption of key-bits happens upon arrival of payload to be transmitted secretly from the sender s to the receiver t. Hence, we can consider the entire network as one large queue, whose internal servicing is done by routing, network coding, or otherwise. From outside, we have $c(f_1)$ as the deterministic rate at which key-bits arrive for later consumption, and we are back at the $D/M/1$-queue. Considering multiple access-points to the network is trivial by switching to a multi-source-multi-sink flow. Unlike standard queueing disciplines, optimality in our context means the incoming amount of key-material outweighing the arrival of messages, i.e. an "unstable" queue whose expected length is infinite

C. Key-Buffer Architectures

It is easy to set up the devices so as to realize a single-path transmission as illustrated in Figure 1. However, it would not be reasonable to assume nodes to have only two ports, so the internal management of quantum keys is slightly more involved. Going back to Figure 1, we can instantly fix the problem of the message popping up in plaintext within each relay node by simply XOR-combining both, the incoming- and outgoing key into a single “relay-key” [20]. Figure 4 illustrates the idea: for the relay from node A to node B over node R, the latter would XOR-combine (⊕-operation) k_A and k_B into k_{AB}. Consequently, we would only store k_{AB} in an *individual buffer* for this link (see Figure 3; right). If the relay is trusted, then it may alternatively decrypt the incoming message and re-encrypt it before passing it onwards (as shown in Figure 1). Consequently, we would have to maintain *shared buffers* for each I/O-port, as displayed in Figure 3 (left) .

III. Results

We are now ready to present our main results.

A. End-to-End transmission capacity

Given the (constant) rate μ of key-bits generated on link $u - v$, we can ask for the maximal average rate λ of arriving messages that we are able to encrypt. Or stated differently: if we know the service rate, what is the highest rate of incoming customers that we can handle? Obviously, a necessary condition is the rate of arriving customers not exceeding the service rate. This is almost sufficient, as the following result shows:

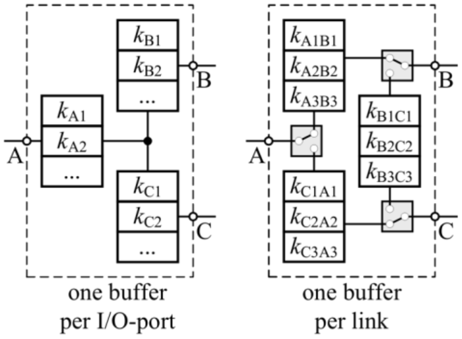


Figure 3 Shared vs. individual key-buffers

Proposition 1. Let a $M/D/1$-queue be given and denote the average arriving rate by λ. For a given constant service rate μ, any arrival rate $\lambda < \mu$ leads to a stable system.

So although the case $\lambda = \mu$ may be fine for a system with deterministic arrival, it is not appropriate for random systems.

Proof: Denote by N the number of customers in the system and by $p_n = \Pr\{N = n\}$ the probability of being in state n. In a stable situation, the total rate out of this state n equals the sum of the incoming rates from states $n-1$ and $n+1$, which yields for every $n \in \mathbb{N}$,

$$\lambda\, p_{n-1} + \mu\, p_{n+1} = (\lambda + \mu) p_n.$$

This recursive formula can be rewritten as

$$p_{n+1} = \left(\frac{\lambda}{\mu}\right)^n \cdot p_0,$$

and since probabilities sum up to one we get

$$1 = \sum_{n=0}^{\infty} p_n = \sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n \cdot p_0.$$

This is simply a geometric series, which converges if and only if $\lambda < \mu$, as stated. □

The above proof also shows that the probabilities p_n can be calculated via $p_n = \rho^n(1-\rho)$ for any $n \in \mathbb{N}$ (by using $1 = \frac{p_0}{1-\rho}$), where $\rho = \frac{\lambda}{\mu}$.

Putting this to practice within a quantum network is straightforward in two steps:

1. Upon given key-generation rates on each link, use

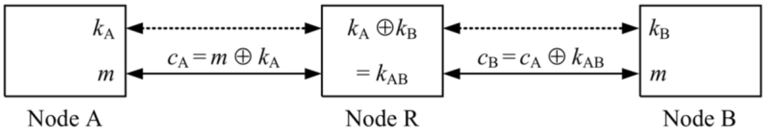


Figure 4 Forwarding with link-specific keys

these rates as edge-weights in an undirected graph and determine a maximal flow from the source node to the sink node. Call the value of this flow μ.

2. By Proposition 1, a payload of up to $\lambda < \mu$ bits per time unit can be handled by the quantum network in a perfectly secure manner (assuming trusted relay).

B. *Waiting probability*

Here, we need to distinguish two architectures in our theoretical considerations: let a node $v \in V$ be given, whose neighbors are $u_1, u_2, \ldots, u_{\deg(v)}$, where $\deg(v)$ denotes v's degree. Either each route passing through v is associated with its individual key-buffer (perhaps via logically partitioning the overall key-material somehow; cf. the right side of Figure 3), or all incoming and outgoing flow draws from the same key-buffer (Figure 3; left), in which case a very busy line can affect the capacities of other routes through v. However, short-term traffic peaks are easier to handle with this architecture. We consider both variants separately.

Individual key-buffers

To estimate the probability that one has to wait to get the key-bits needed for encryption anywhere along its way from the sender to the receiver, we first focus on the waiting probability in one particular node w. Observe that since the key-buffers are not shared, distinct links from a node w to any of its neighbors act independently. So we can restrict our considerations to any (arbitrary) key-buffer within w. Notice, however, that a link from w to its neighbor v has to be treated differently than the link from v to w, since we are concerned with *forwarding* packets.

Let $v \in V(G)$ be any node, and denote its neighbors by $N(v)$. Pick any key-buffer within v that refers to the connection $v - w$, where $w \in N(v)$. Let the incoming traffic per time unit on the route from v to w be $Poisson(\lambda)$ distributed, and assume the QKD protocol between v and w to reproduce an amount of v bits per time unit. Finally, assume the key-buffer to be full at the beginning. Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of i.i.d. random variables $X_i \sim Poisson(\lambda)$ where X_i is the traffic at time unit i. The corresponding filling level of the key-buffer at time i can never exceed the capacity L and is thus given by

$$Y_i = \min\left\{L, L + (i-1)\cdot\mu - \sum_{j=1}^{i} X_j\right\}, \qquad (1)$$

assuming that we start off with the full key-buffer and re-fill it at rate μ after the first time-unit (i.e. we do no refill within the first time-unit because the buffer is full already).

We are interested in the probability for the link being blocked, i.e. the likelihood of an empty key-buffer at time unit t. Hence, we ask for $p(t) := \Pr\{Y_t = 0\} = \Pr\{Y_t \le 0\}$. From $L > 0$ we deduce $\min\{L + (i-1)\cdot\mu - \sum_{j=1}^{i} X_j , L\} \le 0$ if and only if $L + (i-1)\cdot\mu - \sum_{j=1}^{i} X_j \le 0$. Hence,

$$p(t) = \Pr\left\{L + (t-1)\mu - \sum_{j=1}^{t} X_j \le 0\right\}$$
$$= \Pr\left\{\sum_{j=1}^{t} X_j \ge L + (t-1)\mu\right\}$$

If the traffic load over different time-units is independent, then $\sum_{j=1}^{t} X_j \sim Poisson(t\cdot\lambda)$ so that the above probability boils down to a mere evaluation of the Poisson distribution function $F(x|\lambda) = \sum_{i=0}^{\lceil x\rceil-1} \frac{\lambda^i}{i!} e^{-\lambda}$ and comes to

$$p(t) = 1 - F(L + (t-1)\mu | t\cdot\lambda).$$

It is legitimate to ask what happens if the refilling of the key-buffer happens on a random basis as well. Call X_i' the amount of fresh key-material in time-unit i. We can easily replace the term $(i-1)\cdot\mu$ in (1) by $\sum_{i=1}^{t-1} X_i'$ so as to take this randomness into account, but the distribution of $X_{i+1}' - X_i'$ is no longer Poissonian (mostly because the difference is not bounded from below). A straightforward way out of this dilemma is considering Gaussian approximations to the Poissonian densities, which takes us back to the wonderful world of distributions closed under convolution. In other words, if we approximate $X_i \sim Poisson(\lambda)$ by $\tilde{X}_i \sim \mathcal{N}(\lambda, \lambda^2)$, the above derivation and result becomes obvious. We leave this track aside here and go back to the deterministic refilling, giving us the following result:

Proposition 2. Let $w \in V(G)$ have neighbors $N(w)$, and consider the key-buffer shared with an arbitrary but fixed neighbor $v \in N(w)$ of w. Denote by L_v the size of the key-buffer associated with the link $w - v$ and assume that new key-bits are generated at a constant rate μ. The number of incoming message bits from w to v is assumed to be $Poisson(\lambda)$ -distributed. If $\lambda > \mu$, then the probability of waiting within w during a transmission is

$$\begin{aligned} p_{w\to v} &= \Pr\{\text{forwarding from } w \text{ to } v \text{ gets delayed}\} \\ &= 1 - F(L_v + (t_0 - 1)\mu | t_0\lambda) \\ &= 1 - \sum_{i=0}^{\lceil L_v + (t_0-1)\mu\rceil - 1} \frac{(t_0\lambda)^i}{i!} e^{-t_0\lambda}, \end{aligned}$$

where $t_0 = \frac{L_v}{\lambda - \mu}$.

Proof: The argument is merely a matter of noticing that it takes a period of $t_0 = \frac{L_v}{\lambda-\mu}$ time units to entirely empty the key-buffer, if λ bits are used up with μ bits growing back per time unit (the time for encryption is considered negligible). Hence, the average expenditure is $\lambda - \mu$. □

The alert reader might utter concerns about the stochastic independence of incoming traffic over different time-units. There are (at least) two ways to justify this assumption:

- Considering the transmission as a process resting on various routing protocols, we can reasonably assume the network's routing regime to rearrange, encode and decode, and to partition the messages in a way that stochastic correlations between packets are negligible. Notice that this in no way rules out the possibility of linking packets to each other via sequence numbers or matching delivery addresses. However, this "weaker" type of correlation does not necessarily imply dependencies among the payloads of different packets, e.g. when a long sequence of independent cryptographic key-material is transmitted.
- In case that the network is merely used for continuous shared key establishment (in fact, this is the way in which a quantum link is generally supposed to be used [3]), we can safely assume incoming traffic packets as stochastically independent, for otherwise we would have interdependence among key-bits. This is undesired for cryptographic keys, particularly for quantum keys (as it reduces the key's entropy).
- While Proposition 2 refers to only a single node, it is more interesting to find out how likely a blockage along a path from any node to any other node is. In the model of individual key-buffers, this problem boils down to identifying a path whose blockage probability is minimal. We can simply invoke any shortest-path algorithm for that matter, if we assume blockages to happen *independently* of each other. Consider a node $w \in V(G)$ having neighbors $N(v)$.

Observe that Proposition 2 is concerned with the likelihood of blockage when *forwarding* a message from w onwards. Hence, we need to cast the undirected network model graph into a directed graph by converting an undirected edge into two directed edges (with opposite directions).

Each link $w \to i$ for $i \in N(v)$ maintains its own key-buffer with blocking probability $p_{w\to i}$ as given by Proposition 2. We transform the undirected graph $G = (V, E)$ into a weighted directed graph $G' = (V', E', c)$ such that

1. Each link $\{u, v\} \in E(G)$ is carried over into two links $(u, v), (v, u) \in E'$ with cost $c(u, v) = c(v, u) = 0$.
2. Each node $v \in V(G)$ having a number $n = |N(v)|$ of neighbors is expanded into a complete graph with n nodes $v_1, \dots, v_n$, each of is connected by two directed edges in either direction. Each edge (v_i, v_j) is added to E' with the cost $c(v_i, v_j) = -\log(1 - p_{v_i\to v_j})$ according to Proposition 2. The set of edges joining v to its neighbors in G, i.e. the set $\{\{v, u\}: u \in N(v)\} = \{(v, u_1), (v, u_2), \dots, (v, u_n)\}$ is carried over to E' as $\{(v_1, u_1), (u_1, v_1), \dots, (v_n, u_n), (u_n, v_n)\} \subseteq E'$ with weights all zero.

Figure 5 illustrates this transformation.

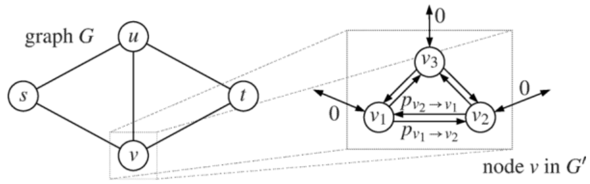


Figure 5 Transformation of a node with individual key-buffers

On G', we can run any shortest-path algorithm, to determine the minimum likelihood of blockage using single-path routing. For any given sender s and receiver t, let their most reliable interconnecting path have "weight" p in G'. Then, regardless of the routing, we have

$$\Pr\{\text{message will be blocked}\} \geq 1 - \exp(-p),$$

because p is the weight of the *shortest* path in G', i.e. the *most reliable* path in G. No matter what the routing actually does, it cannot do better than choosing the best path possible, hence the value

$$1 - \exp(-p) = \Pr\{\text{at least one node is blocked}\}$$

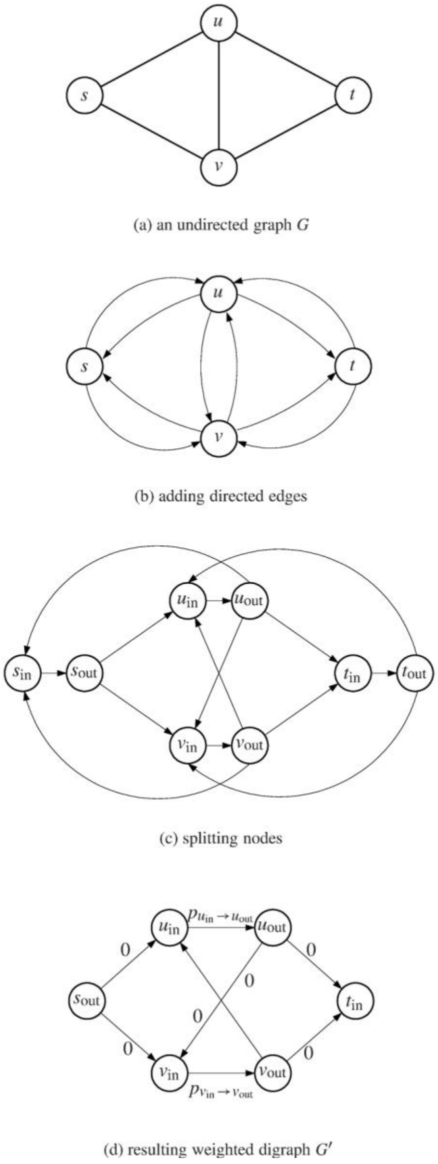
is a lower-bound to the actual likelihood.

Shared key-buffers

In the case of shared key-buffers we assume the incoming flows from different nodes to be independent. Then the distribution of the total flow trough $w \in V(G)$ at time t follows a Poisson distribution with parameter $\lambda = \sum_{i\in N(w)} \lambda_i$, where λ_i denotes the incoming traffic flow from node w to its neighbor i. Similarly, all neighboring links $i \in N(w)$ contribute μ_i bits of fresh key-material to the common key-buffer, giving a total refreshing rate of $\mu = \sum_{i\in N(w)} \mu_i$. With λ, μ we can invoke proposition Proposition 2 again to calculate the probability of a node being blocked in this case.

A little more care is to be taken when asking for the chance of blocking somewhere across the network as a whole. In this case, we use a transformation that is normally used to calculate maximal flows with vertex capacities. The transformation from the undirected graph $G = (V, E)$ (see Figure 6a for an example) to the directed weighted graph $G' = (V', E')$ is now specific for a sender s and receiver t, and proceeds as follows (cf. [23]):

1. Each node v including s and t is replaced by two nodes $v_{\text{in}}, v_{\text{out}} \in V'$, and a directed edge from v_{in} to v_{out} is placed to E'. This link $v_{\text{in}} \to v_{\text{out}}$ gets assigned the cost $-\log p$, where p is the blocking probability calculated as described above.
2. Each undirected edge $(u, v) \in E$ is replaced by two directed edges $u_{\text{out}} \to v_{\text{in}}$ and $v_{\text{out}} \to u_{\text{in}}$. See Figure 6b and Figure 6c for an illustration.
3. The nodes s_{in} and t_{out} are deleted, as well as all edges going into s_{in} and out of t_{out}.
4. Those nodes who remain to be assigned a cost receive zero cost. The resulting graph is shown in Figure 6d.



(a) an undirected graph G

(b) adding directed edges

(c) splitting nodes

(d) resulting weighted digraph G'

Figure 6 Transformation for shared key-buffer

Once having found the shortest path in G' between s and t, we can draw exactly the same conclusion as above: if p is the weight of this path in G', then the chance of this path being blocked for *any* path-based routing-scheme is lower-bounded by $1 - e^{-p}$.

C. Example

To get a more intuitive understanding of the above results, we give a simple example. Consider a modified version of the graph from Figure 2, with six nodes but with neither an edge between vertices 2 and 3 nor between 2 and 4. We call node 0 the sender and the receiver shall be node 5 (cf. Figure 7). We let L be 5000 kbits and choose the rates μ (at which new key-material is produced) randomly between 280 and 320 kbits and let the rate of the incoming message bits be $\lambda = \mu + a$ (see Table 1(a)), where a is a positive constant ($a = 5$ in this specific example). Under this setting, we get the average probabilities $p_{v \to w}$ shown in Table 1(a) for the incident of waiting between v and w, where the average was taken over $N = 100$ calculations.

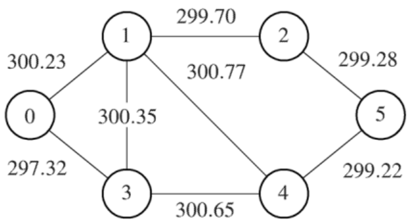


Figure 7 Example network (link performances $\boldsymbol{\mu}$ shown in kbit/sec)

Using the transformation described above we get that the probability of getting stuck is lower bounded by 0.9137 in the case of individual key-buffers and by 0.9929 in the case of shared key-buffers; again averaged over $N = 100$ trials. This means that the individual link performances are indeed sharp bounds to the true bandwidth, as even slightly overshooting (by $a = 5$ kbit/sec in out example) makes congestions highly likely. If we just look at a single evaluation of the two different methods mentioned above, we also see that the paths yielding the minimal value may differ: while working with the design of individual buffers the algorithm takes route $0 \to 1 \to 2 \to 5$ in the original graph, it does prefer $0 \to 3 \to 4 \to 5$ under the shared buffer design.

An illustration and interpretation of Proposition 1 is the following: with link performance values as given in Figure 7, a maximal flow is found at 607.54 kbit/sec. So this is the maximal traffic load that the network can handle.

Better performance is obtained when we double the size of the key-buffer in each link. Under the same set up as before, but with $L = 10$ Mbit of key-buffer and the λ-values as listed in Table 1(b), we get the blocking probabilities shown in the right column of Table 1(b). The blocking probability for an end-to-end communication in this case is $p \geq 0.8771$ when individual key-buffers are used, and $p \geq 0.9725$ when a shared buffer is employed. Finally, Proposition 1 tells that the overall end-to-end traffic from 0 to 5 is bounded above by 608.52 kbit/sec (which is the maximal flow under the respective capacities $\mu_{u \to v} = \lambda_{u \to v} - 5$ for each link).

It is important to observe that Proposition 2 explicitly is concerned with situations in which the traffic load *exceeds* the key-(re-)generation rate on the links. The converse case in which there is a positive surplus of key-material produced on each link is obviously not interesting in terms of congestion likelihoods (as the key-buffers cannot run empty in that case).

TABLE 1 BLOCKING PROBABILITIES EXAMPLES

Edge $e = u \to v$	Traffic λ [kbit/s]	Blocking prob. p_e
$\mathbf{0 \to 1}$	305.23	0.7062
$\mathbf{0 \to 3}$	302.32	0.7058
$\mathbf{1 \to 2}$	304.70	0.7063
$\mathbf{1 \to 3}$	305.35	0.7067
$\mathbf{1 \to 4}$	305.77	0.7066
$\mathbf{2 \to 5}$	304.28	0.7064
$\mathbf{3 \to 4}$	305.65	0.7068
$\mathbf{4 \to 5}$	304.22	0.7064

(a) Key-buffer size $\boldsymbol{L = 5}$ Mbit

Edge $e = u \to v$	Traffic λ [kbit/s]	Blocking prob. p_e
$\mathbf{0 \to 1}$	306.06	0.6497
$\mathbf{0 \to 3}$	304.53	0.6493
$\mathbf{1 \to 2}$	304.29	0.6492
$\mathbf{1 \to 3}$	305.29	0.6495
$\mathbf{1 \to 4}$	305.89	0.6499
$\mathbf{2 \to 5}$	304.34	0.6494
$\mathbf{3 \to 4}$	305.65	0.6497
$\mathbf{4 \to 5}$	304.23	0.6493

(b) Key-buffer size $\boldsymbol{L = 10}$ Mbit

IV. CONCLUSIONS

Given a quantum network, we have shown how to efficiently compute bounds to the transmission capacity and the likelihood of blocked paths due to local congestions.

A. *Future Work*

We focused on two specific architectures for key-buffers. Our approach and results are extensible towards more general architectures (as we considered only two “extreme” cases here) for the key-buffers as well as for the relay-regime as such (cf. [22], who propose a novel three-party quantum communication approach). It is well known that classical routing regimes face difficulties when trying to attain the upper bounds to the transmission capacity as implied by the max-flow approach (network coding is one way to resolve this dilemma). Consequently, our bounds are not necessarily tight. A closer investigation of this is subject of future work. Finally, since quantum networks have hardly reached a level of maturity beyond prototypes or lab demonstrators, reports on comparisons of our results to other competing approaches are part of future research.

B. *Summary*

Our analysis is entirely based on the physical topology of the network and the known key-generation rates on each link. In this work, we focused on single-path (classical) routing schemes, leaving analogous research in the field of multipath routing and network coding for future work. Our results are easy to implement with off-the-shelf algorithms, hence the proposed analysis technique is efficient in terms of computational, modeling and implementation efforts.

Despite quantum networks not having evolved beyond demonstrator prototypes yet, the possibility of setting up a high-security transmission network over existing fibre-optic lines is quite interesting. Our research here is meant as a starting point towards the construction of such infrastructures in an effective and appealing manner for the potential customer. Quality of service and service level agreements in quantum networks, unfortunately, have by now not seen the necessary attention to really bring the QKD technology to the market. Although ingenious solutions and brilliant theoretical achievements have been made, the "last mile" between lab implementation and large-scale practical business implementation needs more attention.

REFERENCES

[1] C. Bennett and G. Brassard, "Public key distribution and coin tossing," in Proceedings of IEEE International Conference on Computers Systems and Signal Processing , Bangalore, India, 1984.

[2] C. Elliott, "The DARPA Quantum Network," arXiv:quant-ph/0412029v1, 2004.

[3] M. Peev, C. Pacher, R. Alleaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. F. Dynes, S. Fasel, S. Fossier, M. Furst, J. D. Gautier, O. Gay, N. Gisin, P. Grangier, A. Happe, Y. Hasani, M. Hentschel, H. Hubel, G. Humer, T. Länger, M. Legre, R. Lieger, J. Lodewyck, T. Lorünser, N. Lütkenhaus, A. Marhold, T. Matyus, O. Maurhart, L. Monat, S. Nauerth, J. B. Page, A. Poppe, E. Querasser, G. Ribordy, S. Robyr, L. Salvail, A. W. Sharpe, A. J. Shields, D. Stucki, M. Suda, C. Tamas, T. Themel, R. T. Thew, Y. Thoma, A. Treiber, P. Trinkler, R. Tualle-Brouri, F. Vannel, N. Walenta, H. Weier, H. Weinfurter, I. Wimberger, Z. L. Yuan, H. Zbinden and A. Zeilinger, "The SECOQC quantum key distribution network in Vienna," New Journal of Physics, vol. 11, no. 7, p. 075001, 2009.

[4] P. Shor and J. Preskill, "Simple Proof of Security of the BB84 Quantum Key Distribution Protocol," Phys. Rev. Lett., vol. 85, pp. 441-444, 2000.

[5] R. Alleaume, F. Roueff, E. Diamanti and N. Lütkenhaus, "Topological optimization of quantum key distribution networks," New Journal of Physics, vol. 11, p. 075002, 2009.

[6] S. Rass, A. Wiegele and P. Schartner, "Building a Quantum Network: How to Optimize Security and Expenses," Springer Journal of Network and Systems Management, vol. 18, no. 3, pp. 283-299, 2010.

[7] S. Bhadra and S. Shakkottai, "Looking at Large Networks: Coding vs. Queueing," in Proceedings of the 25th IEEE International Conference on Computer Communications, Barcelona, Spain, 2006.

[8] S.-Y. R. Li, R. W. Yeung and N. Cai, "Linear Network Coding," IEEE Transactions on Information Theory, vol. 49, no. 2, pp. 371-381, 2003.

[9] R. Ahlswede, N. Cai, S.-Y. Li and R. Yeung, "Network information flow," IEEE Transactions on Information Theory, vol. 46, no. 4, pp. 1204-1216, 2000.

[10] D. H. Woldegebreal, S. Valentin and H. Karl, "Outage probability analysis of cooperative transmission protocols without and with network coding: inter-user channels based comparison," in Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, New York, USA, 2007.

[11] S. Weber, J. G. Andrews and N. Jindal, "An overview of the transmission capacity of wireless networks," IEEE Transactions on Communications, vol. 58, no. 12, pp. 3593-3604, 2008.

[12] M. Kaynia, F. Fabbri, R. Verdone and G. Øien, "Analytical study of the outage probability of ALOHA and CSMA in bounded ad hoc networks," in Proc. European Wireless (EW), 2010.

[13] R. Vaze and R. W. Heath, "Transmission Capacity of Ad-hoc Networks with Multiple Antennas using Transmit Stream Adaptation and Interference Cancelation," http://arxiv.org/abs/0912.2630, 2009.

[14] F. Caruso, S. F. Huelga and M. B. Plenio, "Noise-Enhanced Classical and Quantum Capacities in Communication Networks," Phys. Rev. Lett., vol. 105, p. 190501, 2010.

[15] S. Watanabe, Remarks on Private and Quantum Capacities of More Capable and Less Noisy Quantum Channels, arXiv:1110.5746v1 [quant-ph], 2011.

[16] C. Le Quoc, P. Bellot and A. Demaille, "On the security of quantum networks: a proposal framework and its capacity," in Proceedings of the

International Conference on New Technologies, Mobility and Security, 2007.

[17] G. Smith, J. A. Smolin and J. Yard, "Quantum Communication with Gaussian channels of zero quantum capacity," Nature Photonics, vol. 5, pp. 624-627, 2011.

[18] G. Zhang, S. Zhou, D. Wang, G. Yan and G. Zhang, "Enhancing network transmission capacity by efficiently allocating node capability," Physical A: Statistical Mechanics and its Applications, vol. 390, no. 2, pp. 387-391, 2009.

[19] S. Weber, X. Yang, G. d. Veciana and J. Andrews, "Transmission capacity of CDMA ad-hoc networks," in IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications, 2004.

[20] P. Schartner and S. Rass, "How to overcome the Trusted Node Model in Quantum Cryptography," in Proceedings of the 12th IEEE International Conference on Computational Science and Engineering, 2009.

[21] A. Mink, L. Ma, T. Nakassis, H. Xu, O. Slattery, B. Hershman and X. Tang, "A Quantum Network Manager That Supports A One-Time Pad Stream," in Proceedings of the second International Conference on Quantum, Nano and Micro Technologies, 2008.

[22] C. Le Quoc and P. Bellot, "A New Proposal for QKD Relaying Models," in Proceedings of 17th International Conference on Computer Communications and Networks, 2008.

[23] A. Abbas, "A Hybrid Protocol for Identification of a Maximal Set of Node Disjoint Paths," International Arab Journal Of Information Technology (IAJIT), vol. 6, no. 4, pp. 344-358, 2009.

AUTHORS PROFILE

Sandra König received her Bachelor and Master degree in Mathematics at the ETH Zurich, with a focus on statistics (subspecialty in regression analysis and time series analysis). Her research interests cover applications of statistics in electrical engineering as well as communication and information theory.

Stefan Rass graduated with a double master degree in mathematics (with a focus on statistics)and computer science from the Klagenfurt University in 2005, and gained a PhD degree in mathematics in 2009. His research interests include general system security, in particular applications of quantum cryptography and information-theoretic security. …