Secure Routerless Routing

Extended Abstract

Vince Grolmusz
Department of Computer Science
Eötvös University, Budapest
grolmusz@cs.elte.hu

Zoltán Király
Department of Computer Science
and Communication Networks Laboratory
Eötvös University, Budapest
kiraly@cs.elte.hu

ABSTRACT

Suppose that there are n Senders and r Receivers. Our goal is to design a communication network such that long messages can be sent from Sender i to Receiver $\pi(i)$ such that no other receiver can retrieve the message intended for Receiver $\pi(i)$. The task can easily be completed using some classical interconnection network and routers in the network. Alternatively, if every Receiver is directly connected to all n Senders, then the Senders can choose which channel to use for communication, without using any router. Fast optical networks are slowed down considerably if routers are inserted in their nodes. Moreover, handling queues or buffers at the routers is extremely hard in all-optical setting. An obvious routerless solution, connecting each possible Sender-Receiver pairs with direct channels seems to be infeasible in most cases. The main result of the present work is the mathematical model of two networks and corresponding network-protocols in which the Senders and the Receivers are connected with only $r^{o(1)}$ channels (in practice no more than 32 channels in both networks); there are no switching or routing-elements in the network, just linear combinations of the signals are computed. Such designs would be usable in fast all-optical networks. In the proof of the security of the networks we do not use *any* unproven cryptographical or complexity theoretical assumptions: the security is information-theoretically proved, and does not depend on cryptographical primitives.

Categories and Subject Descriptors: B.4.3 Hardware Input/output and data communications

General Terms: Design

Keywords: High speed optical networks, routerless routing, secure network protocols

SIGCOMM'04 Workshops, Aug. 30+Sept. 3, 2004, Portland, Oregon, USA.

1. INTRODUCTION

The extreme bandwidth of a single optical fiber (25 000 GHz) is 1000 times larger than the total radio bandwidth of planet Earth (25 Ghz) [2]. Using this bandwidth effectively requires novel network designs. We propose two network designs for high-speed optical communication.

Suppose that there are given n Senders $S_1, S_2, \ldots, S_n$ and r Receivers $R_1, R_2, \ldots, R_r$. Let π be a function from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, r\}$. Our goal is to send long messages from S_i to $R_{\pi(i)}$, for $i = 1, 2, \ldots, n$ such that

(a) $R_{\pi(i)}$ can easily retrieve the message of S_i, for $i = 1, 2, \ldots, n$

(b) $R_{\pi(i)}$ cannot retrieve the message of S_j if $\pi(i) \neq \pi(j)$.

An obvious method for doing this is connecting S_i with $R_{\pi(i)}$ with private channels, that is, we use n channels for the n Senders and the r Receivers. The advantage of this solution is that n bits can be sent in parallel, and the transmission is private, in the sense that $R_{\pi(i)}$ receives only the transmission of S_i, for $i = 1, 2, \ldots, n$. The privacy is satisfied only if others have not access to the private channels. The disadvantage of this solution is that the number of channels is equal to the number of communicating pairs, and this is infeasible in most cases. Another problem with this solution is that if next time S_i wants to send messages to $R_{\sigma(i)}$, for $i = 1, 2, \ldots, n$ for some other function σ, then the whole network has to be reconfigured. If every Sender is directly connected to all Receivers, this solves the reconfiguration problem, but then the number of channels becomes nr. Applying some classical interconnection networks (e.g., the butterfly, Benes network, CCC) needs routers with buffers (local memory). Due to the table-lookup features of routers and the need of optical memory, all-optical routers are hard to construct, expensive and still relatively slow components.

Another obvious solution is that all the Senders and Receivers use the same channel, and they transmit their messages one after the other. Transmitting n bits this way needs n steps. In this case either a router has to be used just before the messages get to the Receivers, or some sort of encryption is needed for maintaining the privacy of the transmission.

Using encryption has several drawbacks. Streamciphers, the most evident cryptographic tool which are fast and do not cause overhead in the communication have lots of recently proposed and successful attacks [6]. Block-ciphers are much slower, and may be infeasible in, say, in the 1000 Gbit/s range, and also, they causes non-negligible overhead in the communication.

Using routers and addressing in the messages will also slow down the communication, especially in all-optical environments: with, say, 1000 Gbit/s throughput, by the best of our knowledge, no routers exist.

The main result of the present work is a mathematical model of two networks, together with the associated network-protocols, in which

(i) The n Senders and the r Receivers are connected with only $r^{o(1)}$ channels in the first and $\log r$ channels in the second network [1] Note, that in practice at most 32 channels are enough in both networks. The parallel channels will not speed up the transmission relative to the 1-channel network: the goal of using them is to facilitate the privacy of the communication and the distribution of the messages between the recipients, without any encryption or routers.

(ii) The encoding and decoding is nothing else just linear combinations of the message-bits, and this linear combinations can be computed really fast.

(iii) There are no switching or routing-elements in the network with hard-to implement buffers and local memory, just linear combinations are computed, with fixed connections (channels or wires); moreover, the network components used are simple enough to implement in fast all-optical networks.

(iv) In the first network, $R_{\pi(i)}$ can learn only very little (to be specified later) about any bit of the message of S_j for any $\pi(j) \neq \pi(i)$, and only a negligible amount of information on longer messages of S_j; in the second network R_i cannot learn anything about any bit of the message of S_j for any $\pi(j) \neq \pi(i)$.

(v) In the proofs we do not use *any* unproven cryptographical or complexity theoretical or any other assumptions (e.g., no assumptions are used for the existence of one-way functions, or the computational hardness of specific computational tasks). The security is information-theoretical rather than cryptographical, in the sense that it does not depend on unproven cryptographical primitives.

For example, in perhaps the most appealing possible application, all the Senders are controlled by a Video on Demand (VoD) multicaster, who distributes – say – video or movie programming to several hundreds of thousands users.

Our first network uses MOD m addition gates for some small modulus m (say $m = 6$ or $m = 10$) (see [9] for all-optical realizations), the second network uses optically feasible MOD 2 addition (i.e., XOR) gates, (see [7] or [5]). The decoding at the user's side is done with very simple and optically feasible computations (counters or modular (XOR) gates).

We would like to add that unlike the recent advances (e.g., [1], [8], [3]) our networks will be fast because they do not contain routers, do not perform slow table-lookups, do not need buffers or network-traffic increasing deflections, only fast, optically feasible gates [5].

[1]Here $o(1)$ denotes a quantity which goes to 0 as r goes to the infinity.

1.1 The formal model

Let $S_1, S_2, \ldots, S_n$ denote the Senders, and let $R_1, R_2, \ldots, R_r$ denote the Receivers. Additionally, we have t data transmission channels, used for long-distance connection between Senders and Receivers. Each Sender is connected – possibly through some modular addition gates – to all of these t channels, while the Receivers may be connected – through modular addition gates – only to certain subsets of the channels.

On one channel one bit may be transmitted at a time. If one Sender sends several bits simultaneously to an ℓ element subset of the t long-distance channels, then these bits will travel synchronously on these ℓ channels: that means, that for any i, Receiver R_i will get those bits which were sent simultaneously, from all the long-distance channels, connected to R_i, at the same time. However, we do not suppose that different Receivers get these bits at the same time (it is allowed that farther situated Receivers get the bits later than the closer ones).

Figure 1 describes the general scheme in the case when $n = r$. We need that the Sender's bits travel synchronously on the t long-distance channels. (Note, that this requirement can be assured by using the same wavelength optical signals on each channel, and by compensating for the distance-differences at the Senders side by installing fiber loops: this way the signals - if sent simultaneously by all the senders - will travel synchronously). However, we need not assume that the signals reach all the Receivers at the same time: the Receivers are allowed to be scattered along the long-distance channels (see Figure 2).

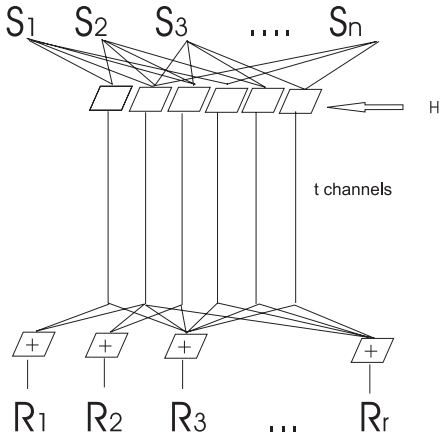

Figure 1: The common scheme

An arbitrary number of Senders may send messages to the same Receiver.

We also assume that the Receivers have access only to some subsets of the t long-distance channels. The security of the network depends on this property: Receivers who had access for all the channels would be able to decode all the messages. Note, that this assumption is realistic, since in high-speed optical networks intrusion and tampering with the fibers and optical components (that is, gaining access to the non-allowed channels) seems to be avoidable by using common security measures.

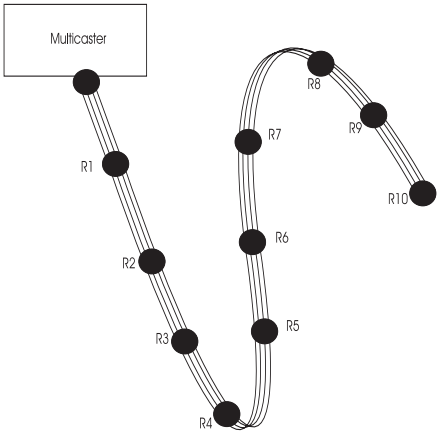

Figure 2: Application for multicasting

1.2 The common properties of our networks

Both in our networks the Senders use random numbers for security (in Network 1 for scheduling the transmissions, in Network 2 for the encoding). These random numbers are generated by the Senders cooperatively in Network 1 and privately in Network 2.

Generating random bits cooperatively in Network 1 may cause difficulties. This can be circumvented by generating the random bits locally at the head-station (denoted by H on Figure 1) of the long-distance channels, and sending to the Senders.

We should remark, that in Network 2, XOR gates (i.e., MOD 2 additions) are used only at the Receiver's side, and not at the Senders' side.

Both of our networks facilitate the re-configuration of the Sender-Receiver correspondence without any change in the hardware; that is, the same network can be used to send bits from S_i to $R_{\pi(i)}$, for $i = 1, 2, \ldots, n$, and for any function π. The Sender/Receiver rôles (and the direction of the communication) cannot be exchanged in our networks; however, if we place the same network with the R_i's as senders and S_i's as receivers next to our network, then everybody can send and also receive messages. Because of this quite usual trick, we will speak only about one-directional communication from the Senders to Receivers.

Both of our networks seem to be applicable in long-distance communication (Figure 1) and also in high-speed multicasting environment (Figure 2).

In Figure 1, we may suppose that the Senders are in the U.S. and the Receivers are in Europe, and the small number of high-speed (and high-cost) channels crosses the Atlantic.

Figure 2 illustrates the multicasting application. Here the multicaster (e.g., the server of some Video-on-Demand provider) plays the rôle of the Senders. In this case the synchronization and the generation of common random bits are easier than in the general case, when the Senders can be spatially distributed. The Receivers are the users, who are not allowed to tamper with the fixed connections to the transmitting channels (i.e., some channels cannot be received by some users). The Receivers are allowed to be at different spatial positions of the communication channels, so this application may be used in a VoD service of a big city. For example, the fibers may be placed into service tunnels in the ground below the streets, together with the expensive optical gates, and the homes can be connected to the network with short, inexpensive fibers and optical devices.

The 100 Gbit/s transmission rate seems to be possible commercially soon. This means, that theoretically, on a single 100 Gbit/s channel (without counting overheads) from 10 000 to 100 000 users can be served simultaneously with individually chosen DVD-quality video programming (which needs 1 Mbit/s to 10 Mbit/s transmission rate), if routers do not slow down the transmission. In our networks simply there are no routers.

The following table summarizes the properties of our networks.

1.3 Summary of the properties of our networks

Netw.	No. of channels	Rand. numb.	Security	time for xmit n bits
n chan.	n	no	yes	1
1 chan.	1	no	no	n
Netw. 1	$n^{o(1)}$	sync.	yes	$(n+1)\log 6$
Netw. 2	$O(\log n)$	priv. gen.	yes	see note

Note: The last column contains the total time needed to transmit 1 bit by Sender S_i to Receiver $R_{\pi(i)}$, for $i = 1, 2, \ldots, n$. This schedule of transmission, however, is advantageous mostly in Network 1 (its security also depends on this schedule). In Network 2, packets of length d can be sent to n Receivers with using $n(d + O(\log n))$ transmission time (in other networks we do not send packets, just bits). However, Network 2 is much more adequate for sending long messages only to a small set of Receivers. For example, if n bits is sent only to R_1 in one packet, then the time is only $n + \log n$.

1.4 Remarks on the Overhead of our Networks

The traditional packet-switching networks contains routers for directing the packets to their respective addresses. That means, that router W is connected to Receivers $R_1, R_2, \ldots, R_s$, and if a packet is sent to, say R_1, then R_2 will not get that packet.

It turned out, that in high-speed optical networks traditional (electronic) routers slow down the traffic considerably, and designing all-optical, high-speed routers is still a challenge for today's technology. Consequently, the first router that divides the traffic of the optical channel slows down the traffic significantly and so decreases the throughput of the optical channel.

Our networks do not contain routers at all. There is a price of this simplicity: in our networks an encoded form of every message is heard by every receiver, but those, who are not supposed to read it, could not decode it.

2. NETWORK 1

In packet-switched networks, the Receivers should know their own identity (say, an IP or MAC address) in order to pick up only those packets from the transmission channels, which are addressed to them. Note, that in Network 1, described in this section, the Receivers need not know even their own identity: the bits, intended to be sent to them, will find them securely and automatically.

2.1 Preliminaries

Let m be a composite number $m = p_1^{e_1} p_2^{e_2} \cdots p_\ell^{e_\ell}$, and let $\ell > 1$.

In [4] it is proven that the second degree, $2r$-variable polynomial

$$P(x,y) = \sum_{i \neq j, i,j=1}^{r} a_{ij} x_i y_j,$$

can be computed with $r^{o(1)}$ multiplications of homogeneous terms, where coefficients a_{ij} satisfy that

$$a_{ij} \equiv 1 \pmod{p_u^{e_u}},$$

for at least one $u : 1 \leq u \leq \ell$, and, moreover, if for some v: $a_{ij} \not\equiv 1 \pmod{p_v^{e_v}}$, then $a_{ij} \equiv 0 \pmod{p_v^{e_v}}$. (Note, that for prime moduli, one needs at least r multiplications to compute such a polynomial.) Consequently, polynomial

$$Q(x,y) = \left(\sum_{i=1}^{r} x_i\right)\left(\sum_{i=1}^{r} y_i\right) - P(x,y)$$

can be computed also with only $t = r^{o(1)}$ homogeneous multiplications:

$$Q(x,y) = \sum_{j=1}^{t} \left(\sum_{i=1}^{r} b_{ij} x_i\right)\left(\sum_{i=1}^{r} c_{ij} y_i\right), \qquad (1)$$

and it has the following properties:

(i) the coefficients of the $x_i y_i$ terms equals to 1, for $i = 1, 2, \ldots, r$,

(ii) any other coefficient is 0 modulo at least one prime-power divisor of m.

In particular, for $m = 6$, the coefficient of any $x_i y_j$ (where $i \neq j$), is either 0, or 4 or 3 modulo 6.

2.2 Linear Combinations

From now on, we fix m to be 6. For a general, non-prime-power composite m, our results are analogous, but they are more uncomfortable to state.

Let us consider a vector of r variables $x = (x_1, x_2, \ldots, x_r)$, and let us compute $z = (z_1, z_2, \ldots, z_t) \in \{0,1,2,3,4,5\}^t$, where

$$z_j = \sum_{i=1}^{r} b_{ij} x_i \bmod 6 \qquad (2)$$

that is, simply a linear combination, determined by (1), for $j = 1, 2, \ldots, t$. Or, in matrix notation, where $B = \{b_{ij}\}$ denotes an $r \times t$ matrix with entries $\{b_{ij}\}$ of equation (1): $z = xB \bmod 6$.

Certainly, different x's may lead to the same z, since $t = r^{o(1)} << r$, for large enough r.

Now, from (1), polynomial $Q(x,y)$ with any substitution of the y's can be computed from the $t = r^{o(1)}$ values of z. In particular, plugging in $y^{(i)} = (0,0,\ldots,\overbrace{1}^{i},0,\ldots,0)$, we will get modulo 6 the following polynomial:

$$x_i + 4(x_{i_1} + x_{i_2} + \cdots + x_{i_\ell}) + 3(x_{j_1} + x_{j_2} + \cdots + x_{j_k}) \quad (3)$$

where different indices denote different numbers.

Or, in other words, with the notation of $C = \{c_{ij}\}$ as an $r \times t$ matrix of c_{ij} entries from (1),

$$x' = x + 4xU + 3xV = xBC^T = zC^T, \qquad (4)$$

where $U = \{u_{ij}\}$ and $V = \{v_{ij}\}$ are some $r \times r$ matrices with 0's in the diagonal, and with the following property: for all i and j, either u_{ij} or $v_{ij} \equiv 0 \pmod 6$, for all i and j.

Note, that matrices B and C are independent of vector x, that is, they are constant matrices.

We describe our network 1 now. First, assume that the number of Senders and Receivers are the same: $n = r$, and, moreover, such that R_i gets messages from S_i, for $i = 1, 2, \ldots, r$; handling other functions π is given in Section 2.4.

Our network 1 is constructed as follows: The vector z with t mod 6 coordinates is computed as in (2). The coordinates of this vector is transmitted through $t = r^{o(1)}$ channels (these are the long-distance channels). Then, at the Receivers' side, vector z is transformed to length-r vector x', as in (4).

Note, that this transformation simply means computing x'_i as the sum of several bits, arrived on some of the t long-distance channels (we allow that some of them are added more than once, but less than 5 times: this multiplicity is equal to their coefficients, determined by (4)). Note also, that this computation is done locally at each Receiver.

Getting back the actual vector x needs some further steps, what we call *filtering*, and we detail it in the next section.

2.3 The Protocol

We describe the transmission-protocol in rounds. In every round, every sender S_i will transmit securely a bit x_i to the corresponding receiver, R_i, $i = 1, 2, \ldots, r$. In u consecutive rounds, every sender will send u bits, that is, sending u-bit messages needs u rounds of the following protocol.

A round is performed as follows:

Step 1 - Encoding - From the bits of $x = (x_1, x_2, \ldots, x_r)$ the mod 6 integers $z = (z_1, z_2, \ldots, z_t)$ is computed by linear combinations taken modulo 6: $z = xB \bmod 6$, as in (2).

Step 2 - Transmission - The mod 6 numbers $z_1, z_2, \ldots, z_t$ are sent on t channels to the receivers.

Step 3 - Decoding - The linear transformation $x' = (x'_1, x'_2, \ldots, x'_r) = zC^T$ is computed modulo 6 at the receivers' side, and number x'_i is given to receiver R_i, $i = 1, 2, \ldots, r$. (Note, that because of the obvious information-theoretical reasons, generally it is not possible to retrieve bit x_i from integer x'_i).

Step 4 - Pre-Filtering - A random $\mu : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ permutation is generated at the sender's side. Then for $j = 1, 2, \ldots, n$ Steps 1, 2 and 3 are repeated for $x^{\mu(j)} \in \{0,1\}^n$ instead of x, where $x^{\mu(j)}$ coincides with x, except on position $\mu(j)$, whereas $x^{\mu(j)}$ is 0 if it was 1 in x or 1 if it was 0 in x. Let x''_i denote the coordinate i of $x^{\mu(j)}BC^T$.

Step 5 - Post-Filtering - Now, receiver R_i stores value x'_i in its memory, and follows the next program after receiving any new x''_i, originating in Step 4:

if $x''_i = x'_i$ it does nothing;

if $x''_i = x'_i - 1$ then R_i concludes that $x_i = 1$;

if $x''_i \equiv x'_i \pm 3$ then it does nothing;

if $x''_i \equiv x'_i \pm 4$ then it does nothing;

if $x''_i = x'_i + 1$ then R_i concludes that $x_i = 0$.

THEOREM 1. *After performing one round, receiver R_i retrieves the bit x_i, for $i = 1, 2, \ldots, n$.*

PROOF. Clearly, x_i' is equal to quantity, given in formula (3); so decreasing any non-zero x_j in the sum of formula (3) by 1, leads to either a decrease of 1 of the sum (in the case when exactly x_i was decreased, or by 0 (when an x_j was decreased with a 0 coefficient in formula (3)), or by 4 or 3 modulo 6, (when the coefficient of the decreased variable was 4 or 3, respectively). Similarly, an increase by 1 will result an increase by 1 or 0 or 3 or 4 modulo 6, where 0, 3 and 4 means that not the variable with coefficient 1 was increased. Consequently, a change by value 1 means that $x_i = 0$, by value -1 means that $x_i = 1$. □

2.4 Building Network 1 for the General Addressing Function

First, consider the case when $n = r$ and π is a permutation, other than the identity. Network 1 can easily be reconfigured as follows. Suppose, that Sender S_i needs to send bits to Receiver $R_{\pi(i)}$ for some permutation π. Then – since all the Senders are connected to all the channels – Sender S_i will simply send the same messages as Sender $S_{\pi(i)}$ would have sent to $R_{\pi(i)}$. Note, that no wiring and no MOD 6 gates at the wires will be changed. This reasoning shows that the re-configuration is easy.

Let us consider now (not necessarily equal) n and r, and a function π from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, r\}$. If π is an injection (that is, no Receiver gets messages from two different Senders), then the original network protocol (and filtering) works. Suppose now, that S_1, S_2, S_3 want to send messages to - say - R_1. Then we play the original network protocol with the substitution $x_1 + x_2 + x_3$ for x_1 and 0 for x_2 and x_3. Then, $x_1 + x_2 + x_3$ will appear at R_1 with coefficient 1. Now, in the filtering process, only those random μ permutations may be used which fix the order of the image of the first three numbers, say $\mu(1) < \mu(2)\mu(3)$. This property facilitates that R_1 can recollect the bits of the long sequences which is sent to her by S_1, S_2 and S_3, respectively. Clearly, this method can be generalized to any π.

2.4.1 An alternative £ltering method

The following modification of the filtering steps of the protocol relies on the fact that if we one-by-one increase the value of g, then $3g$ will have period 2, and $4g$ will have period 3 modulo 6, but g itself will have period 6, modulo 6.

We cover only the basic case here ($n = r$, π is the identity).

So, we can modify Steps 4 and 5 as follows:

Step 4' - Pre-filtering-variant- A random $\mu : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ permutation is generated at the sender's side. Then for $j = 1, 2, \ldots, n$, Steps 1, 2 and 3 are repeated for six values of $x^{\mu(j)} \in \{0, 1, 2, 3, 4, 5\}^n$ instead of x, where $x^{\mu(j)}$ coincides with x, with a possible exception of position $\mu(j)$, whereas $x^{\mu(j)}$ takes on values u, for $u = 0, 1, 2, 3, 4$, and 5, in this order, in case of $x_{\mu(j)} = 1$, and $x^{\mu(j)}$ takes on values u, for $u = 0, 5, 4, 3, 2, 1$, in this order, in case of $x_{\mu(j)} = 0$. Let x_i'' denote the coordinate i of $x^{\mu(j)}BC^T$.

Step 5' - Post-Filtering-variant - Receiver R_i stores value x_i' in its memory, and follows the following program after receiving the 6 new x_i'''s in Step 4:

if if the period of x_i'' is 6 then R_i concludes that $x_i = 1$ if the values are increasing and concludes that x_i=0 if the values are decreasing;

if if the period of x_i'' is less than 6 then it does nothing.

Note, that this filtering method can be more applicable than the original one when the signals are physically represented as waves or pulses, because then the filtering step can be performed as filtering out waves or pulses of higher frequency (= lower period).

2.5 The Security of Network 1

The security of the network-protocol relies on the independently generated random permutations μ in each round.

Let us review, what R_i can learn from the bits, addressed to others. Clearly, R_i will know the number of the 1-bits with coefficient 4 and also the number of the 1-bits with coefficient 3 in formula (3), but R_i will not know the identity of the 1-bits. Note, that R_i will not know anything about the values of those bits which has coefficient 0 in formula (3). So we have proved the following

THEOREM 2. *After each round of the protocol, Receiver R_i learns its own bit, and also the number of the 1-bits with coefficient 3 in formula (3), and also the number of the 1-bits with coefficient 4 in formula (3), and nothing else, for $i = 1, 2, \ldots, n$.*

In other words, due to the random order of listing the bits in the filtering step, the Receivers learn only the sum of two fixed subsets of the bits, beside their own bit.

2.5.1 Note on the number of channels

In the practice we can decrease the number of channels if the modulus is slightly increased: this way more than 500 000 000 pairs can communicate securely in one direction using 32 channels.

Without going into details, choosing $m = 17$ for modulus,

$$\binom{32}{17} > 500000000$$

pairs can communicate on 32 channels. The price of this is that the security will be decreased a little bit: Every Receiver can learn the sum of 16 pairwise disjoint, fixed subsets of bits instead of just 2 fixed subsets, and for one round $(n+1)\lceil \log 17 \rceil = 5n + 5$ bits will be transmitted instead of $(n+1)\lceil \log 6 \rceil = 3n + 3$.

Note, that with modulus 17 (which is a prime) we cannot get the good asymptotic results as with 6.

3. NETWORK 2

We do not make here assumptions on the number of Senders. Suppose, that the number of Receivers (in Europe) is r, and let k denote the smallest integer such that $\binom{2k}{k} \geq r$. Clearly $k = O(\log r)$. Suppose, that we would like to transmit data on optical lines through the Atlantic, from the U.S. to Europe and suppose, that we have $2k$ dedicated optical fibers under the Atlantic. We will use packets of length $r + 2k$ for communication as an example, we will discuss the altering of the packet length later. Every Receiver has a $2k$ bit long unique address, to be defined in the next paragraph. Note that, as we have an all-optical network, the address cannot be used for directing the packets, when a Sender sends a packet, all Receivers will get a packet (but not the same one).

Suppose further that every Sender is capable to send one-one packet to the $2k$ fibers at the same time (not necessarily the same packet), and we suppose the $2k$ packets reach the

"optical switch" at the European end of the fibers at the same time. We also assume that there are no collisions (a device at the American end of the fibers may take care about this). The $2k$ element set of the fibers will be denoted by F.

We assign a k element subset of F for each Receiver uniquely, and the address of a Receiver will be the characteristic vector of this subset. The optical switch contains optical XOR gates only (see [7] or [5] on constructions of optical XOR gates). For each Receiver we XOR the data of her k fibers and forward the result to her.

The algorithm of a Sender S is the following. Suppose we are going to send a message to a Receiver R having address x. Taking x we can calculate that R is connected to fiber set $A \subseteq F$, where $|A| = k$. Now we compose a packet that will start with $2k$ bits of address x then followed by r "useful" bits (these will be the next r data bits we are going to send). Let y denote the $r + 2k$ bit long packet we made.

Now we choose a fiber $f \in A$ and for each $e \in F \setminus \{f\}$ we construct a random packet p_e of length $r + 2k$. Then we construct p_f as

$$p_f = y \oplus \bigoplus_{e \in A \setminus \{f\}} p_e.$$

After this procedure we may send the $2k$ p_e packets at the same time to the corresponding fibers.

The algorithm of the Receivers is quite simple. They know their own address of length $2k$. When a packet arrives they check whether the first $2k$ bits are equal to their address, and if this is the case then they read the following r bits and consider them as data. So the equipment of a Receiver do not have to be more clever than an Ethernet card.

As the consequence of the algorithm of the Senders one can see that the Receiver R intended by the current Sender will recognize that the packet is for her, and read the right data (as she gets exactly packet y). What happens with the other Receivers? We claim that

- they all will hear absolutely random noise, and
- they will recognize that the packet is not for them except with a negligible probability.

To see the first statement we prove slightly more.

LEMMA 3. *For a subset B of fiber set F*

$$\bigoplus_{e \in B} p_e$$

is exactly y if $B = A$ and consists of $r + 2k$ independent random bits if $\emptyset \neq B \neq A$.

PROOF. The first part of this statement is an obvious consequence of the procedure of the Senders. For the second part first observe, that if the selected fiber f is not an element of B then the sum above is a MOD 2 sum of $|B|$ independent random sequences. The next thing to understand is that p_f itself consists of independent random bits because it is a MOD 2 sum of a fixed sequence y and a random sequence $\bigoplus_{e \in A \setminus \{f\}} p_e$. Now suppose $f \in B$ but $B \neq A$. If $B \subset A$ then for $C = A \setminus B$ we have $C \neq \emptyset$ and $f \notin C$, so $\bigoplus_{e \in C} p_e$ is a random sequence, and so is $\bigoplus_{e \in B} p_e = y \oplus \bigoplus_{e \in C} p_e$. If there is an $e' \in B \setminus A$ then independently of what $q = \bigoplus_{e \in B \setminus \{e'\}} p_e$ is, $q \oplus p_{e'}$ consists of independent random bits. □

For the defective recognition, the probability that a particular Receiver other than R thinks that the packet is for her is at most

$$\frac{1}{2^{2k}} < \frac{1}{n}$$

Summarizing, the aimed Receiver R gets the right data, all other users do not get any information neither about the data nor about the addressee. But with a very small probability a Receiver may think that the packet is for her, she also gets noise as data and none useful information. This probability is small enough if r is large, as we assumed here.

The overhead of this protocol is $\frac{O(\log r)}{r}$ ($2k$ address bits contrasted with the r data bits), of course, if we increase the length of the packets then this value can be made arbitrary small.

If r cannot be assumed so large, then the probability, that a Receiver not intended by S falsely recognize the packet as her own, is not small enough. To fix this problem, we assign also an extended address for each Receiver that is l copies of her original address, in succession. Now the packet length will be $2lk + r$ and starts with the extended address of the intended Receiver and, of course, the Receivers look for their extended address at the beginning of each packet. Now, the above error probability can be decreased to $\frac{1}{2^{2lk}}$. On the other hand we increased the overhead.

It is worth to see some particular numbers that fit the practice. First of all, the packet length should not be too large, we may use packets of length 1024. The number of leased fibers is usually some nice number, let it be $32 = 2k$. With this number we can choose the number of Receivers $r = 500$ million. Further, we choose to repeat the address bits twice, so the length of an extended address is 64 and a packet contains 960 useful data bits. The probability, that a Receiver not intended by S falsely recognize the packet as her own, is

$$\frac{1}{2^{64}} < \frac{1}{1000000000000000000}$$

This probability is so small that the expected time for a user to get a "bad frame" is more than 100 years.

The drawback of this method that some triple of Receivers together can decipher the message for R, but not an arbitrary triple and not any two of them. If we increase the number of fibers from $2k = O(\log r)$ to $k^2 = O(\log^2 r)$ then we can assign a set of fibers to each user in such a way, that no coalition of $k - 1$ users can decipher of a message sent to a user not in the coalition. For this assignment we use the momentum curve in the k-dimensional space over field $GF(2^k)$. It has 2^k points, any k of them are linearly independent over $GF(2^k)$. Each point can be represented as a binary vector of length k^2 in such a way that any k of them are linearly independent over $GF(2)$. These vectors can be used as the characteristic vectors of the "per user" fiber sets.

4. ACKNOWLEDGMENTS

The authors acknowledge the partial support of the OTKA T 046234 grant.

5. REFERENCES

[1] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke. Optimal oblivious routing in polynomial time. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 383–388. ACM Press, 2003.

[2] S. Chatterjee and S. Pawlowski. All optical networks. *Communications of the ACM*, 42(6):74–83, 1999.

[3] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. In *SIGCOMM*, pages 109–120, 1999.

[4] V. Grolmusz. Computing elementary symmetric polynomials with a sub-polynomial number of multiplications. *SIAM Journal on Computing*, 32(6):1475–1487, 2003.

[5] K. Hall and K. A. Rauschenbach. All-optical bit pattern generation and matching. *Electron. Lett.*, 32:1214, 1996.

[6] P. Hawkes and G. Rose. Rewriting variables: the complexity of fast algebraic attacks on stream ciphers. Technical report, eprint.iacr.org/2004/081/, 2004.

[7] M. Jinno and T. Matsumoto. Nonlinear Sagnac interferometer switch and its applications. *IEEE J. Quantum Electron.*, 28:875, 1992.

[8] S. A. Plotkin. Competitive routing of virtual circuits in ATM networks. *IEEE Journal of Selected Areas in Communications*, 13(6):1128–1136, 1995.

[9] A. Poustie, R. J. Manning, A. E. Kelly, and K. J. Blow. All-optical binary counter. *Optics Express*, 6:69–74, 2000.