International Symposium on Information Theory and its Applications, ISITA2004
Parma, Italy, October 10–13, 2004

Processing along the way: forwarding vs. coding

Daniela Tuninetti and Christina Fragouli

EPFL, Lausanne, Switzerland
Email: daniela.tuninetti@epfl.ch, christina.fragouli@epfl.ch

Abstract

We consider a source that transmits to a receiver by routing the information packets over a communication network and examine rate benefits that finite complexity processing at the intermediate nodes may offer. We show that the processing capabilities of the intermediate nodes affect not only the end-to-end achievable rate, but also the optimal routing strategy. For example, there exist network configurations where the maximal throughput is achieved only by coding across independent information streams.

1. Introduction

The success of the Internet has made large scale communication networks part of everyday life. In the wireless world, ad-hoc networks promise to offer equally exciting applications. In a network environment the information transverses a number of channels before reaching the destination, as opposed to a single channel. When the Internet first emerged, coding was employed only at physical layer, and thus was oblivious to the nature of the network, and transparent to higher layers. For several years it has been a controversial question whether coding at higher levels could offer benefits.

In practical networks, multicast is an instance where coding does offer a benefit. In [6, 7] elegant end-to-end coding schemes have been proposed that take into account the behavior of the network as it is experienced at higher layers.

From a theoretical point of view, it is well known that if intermediate nodes are allowed to decode and re-encode the information send by the source, without any complexity and/or delay constraints, then the capacity between a sender and a receiver is upper bounded by the *min-cut capacity* of the network, as described in [3].

Recently it was demonstrated that even for lossless links, allowing intermediate nodes to process the information can increase the achievable rate in a multicasting scenario [1] with respect to simple rounting. The proposed approach termed *network coding* basically requires intermediate nodes to perform linear combinations of the incoming packets. The complexity of the computations is bounded as a function of the number of receivers [1, 4, 5]. Thus an interesting observation is that allowing intermediate nodes to perform finite complexity processing may not only increase the achievable rates, but actually achieve the min-cut capacity of the network.

Motivated by this observation, in this work we investigate what benefits finite complexity processing at intermediate nodes may offer in networks with noisy channels. We examine the problem using information-theoretic tools. A similar problem was mentioned in [8], where however the main focus of the work was on finding the ordering of a cascade of binary channels leading to the largest possible capacity.

We consider the following model. A source transmits information to a destination over a network represented as a graph. Each edge of the graph models an independent channel which accounts for factors such as traffic congestion, protocol used, and interference. Each node represents the same type of device, with the exception of the source and the destination.

We assume that source and destination can process long sequences of data bits, and perform optimal (in a sense to be defined later) coding and decoding. Intermediate nodes however can only process blocks of N channel symbols. We use N as measure of complexity as it allows to bound the physical resources necessary for processing such as time and memory requirements. Moreover, it is well suited to environments where information is transmitted in packets.

The goal of this paper is to get insights on whether, and under what conditions, finite complexity processing at intermediate nodes increases the network throughput.

We start by examining a linear network that consists of a single path of several edges/channels. Our contributions include the formulation of an optimization problem to calculate the optimal coding rate at in-

C. Fragouli was supported by the Fonds National Suisse award No. 200021-103836/1.

termediate nodes. We show that moderate block length suffices to achieves more that 90% of the possible network throughput over fairly long networks. Moreover, we show that the intermediate processing capability should scale logarithmically with the length of the path to achieve a desired (constant) end-to-end rate.

We then consider a general network. We compare the cases where intermediate nodes are allowed simple forwarding and finite complexity processing. We show that the type of intermediate processing not only affects the end-to-end rate, but also implies a different "routing" strategy for each case. We give an example where the best performance is achieved only by coding across independent information streams.

The paper is organized as follows. Section 2 considers the linear network model, Section 3 examines the general model and Section 4 concludes the paper.

2. Linear Networks

A linear network model comprises one source node, one destination node and a series of $L-1$ intermediate nodes as depicted in Fig. 1.

Figure 1: Source and receiver connected by 3 channels.

The edges between intermediate nodes correspond to identical Discrete Memoryless Channels without feedback, indicated in the following as "$(\mathcal{X}, W)$ physical channels", where $\mathcal{X}$ denotes the input alphabet and W denotes the channel transition probability matrix. We assume identical input and output alphabets.

The source and the destination are not subject to any processing constraints. That is, the source can use a channel capacity achieving code and the destination can perform Maximum Likelihood (ML) decoding. The intermediate nodes are only allowed to perform memoryless processing over "chunks" of length N of the codeword send by the source.

We distinguish three cases.

- $N = \infty$: Intermediate nodes are allowed to decode and re-encode the whole codeword send by the source. We say that this case corresponds to $N = \infty$ as, in general, the capacity achieving code used by the source has extremely large block length. We also refer to this case as *perfect processing* because the achievable rate equals the capacity of each $(\mathcal{X}, W)$ physical channel [3], which is the maximum possible rate on linear networks.

  Example 1 *Consider a linear network consisting of L Binary Symmetric Channels with cross-over probability $\epsilon_0 < 1/2$, indicated in the following as BCS(ϵ_0). In this case the maximum end-to-end reliable rate is*

  $$R^{(opt-proc)} = \log(2) - \mathcal{H}(\epsilon_0)$$

  where $\mathcal{H}(t) = -t\log(t) - (1-t)\log(1-t)$. Notice that $R^{(opt-proc)}$ does not depend on L, and is equal to the channel capacity.

Figure 2: Source and receiver connected with a path of length $L = 3$.

- $N = 0$: Intermediate nodes just forward/route the information without any further processing. Since a message traverses L channels before arriving at destination, the overall channel between the source and the destination has transition probability matrix W^L.

  Example 2 *A cascade of L BSC(ϵ_0) is equivalent to a BCS with cross-over probability*

  $$\epsilon_L = \frac{1-(1-2\epsilon_0)^L}{2} > \epsilon_0$$

  In this case the maximum reliable rate is

  $$R^{(no-proc)} = \log(2) - \mathcal{H}(\epsilon_L)$$

  notice that as $L \to \infty$, $\epsilon_L \to \frac{1}{2}$ and hence $R^{(no-proc)} \to 0$.

- $0 < N < \infty$: This is the case of interest in this work and will be examined in the next Section.

2.1. Finite Processing

Here finite complexity processing consists of an "inner" channel code of rate r_c (measured in nats) and block length N over each physical channel $(\mathcal{X}, W)$. We refer to the entity that includes the finite complexity coding, the physical channel and the finite complexity decoder as a "relay".

At the transmitter side the relays produce one out of $M = \mathrm{e}^{Nr_c}$ codewords of length N whose symbols are then send one at a time over the physical channel. At the receiver side blocks of N channel symbols are processed to recover the transmitted codeword. Once the receiver has estimated the transmit codeword, it forwards the estimate to the following relay. Fig. 2 depicts this procedure.

This process at the L relays turns the whole network between the source and the destination into the point-to-point "expanded" channel $(\mathcal{C}_c, (W')^L)$. Here $\mathcal{C}_c$ indicates the set of codewords used by the relays and W' the transition probability matrix from each transmitter to the next receiver, i.e., the (i, j) entry of W' is the probability of deciding for the j-th codeword given that the i-th one was transmitted, $(i, j) \in \{1, \cdots, M\}^2$. Since each message hops through L relays before reaching the destination, and since all the relays perform the same processing, the overall channel has transition probability matrix $(W')^L$.

The source communicates to the destination by using a capacity achieving "outer" code of rate r_{source} whose coded symbols are the codewords of the "inner" code $\mathcal{C}_c$. Since one use of the expanded channel requires N uses of the physical channel, the overall transmission rate is

$$R = \frac{r_{\text{source}}}{N} \quad (1)$$

For a fixed N, the maximum rate of communication R is given by the Shannon capacity of the expanded channel $(\mathcal{C}_c, (W')^L)$. Our next task is hence the characterization of $(W')^L$.

We take $\mathcal{C}_c$ to be a "good" channel code and the decoding rule to be ML. Then, for any N and any r_c, the probability of decoding error given that the m-th codeword was transmitted, $m = 1, \cdots, M$, is bounded by [2, Pag. 140]

$$P_m \leq 4\mathrm{e}^{-N\,E(r_c)} \triangleq \delta \quad (2)$$

where $E(r_c)$ is the random error exponent of the channel W.

In Appendix A we show that the condition in (2) allows us to upper-bound the rate R by using a worst-case argument. Intuitively, since the probability of correct decision for each codeword is as least as large as $1-\delta$, we get a "worst case channel" by "setting"

$$\begin{aligned} [W']_{m,m} &= 1-\delta && \text{for} \quad m = 1, \cdots, M \\ [W']_{m,j} &= \frac{\delta}{M-1} && \text{for} \quad j \neq m \quad j = 1, \cdots, M \end{aligned} \quad (3)$$

Given W' defined by (3) it follows that

$$(W')^L = \left(1 - \delta_L \frac{M}{M-1}\right) \mathbf{I} + \delta_L \frac{1}{M-1} \boldsymbol{v}\boldsymbol{v}^T \quad (4)$$

where $\mathbf{I}$ is the identity matrix of dimension M, $\boldsymbol{v}$ is a column vector of length M with all one entries and

$$\delta_L = \frac{1 - \left(1 - \delta \frac{M}{M-1}\right)^L}{\frac{M}{M-1}} \quad (5)$$

By computing the capacity of channel (4) we can bound the source transmission rate as

$$\max\{r_{\text{source}}\} \geq \log(M) - \delta_L \log(M-1) - \mathcal{H}(\delta_L)$$

Figure 3: Normalized achievable R/C for a linear network consisting of L BCS(0.1).

and thus we can get a rate greater or equal to

$$R = r_c(1-\delta_L) - \mathcal{H}(\delta_L)/N \simeq r_c(1-\delta)^L \quad (6)$$

The approximation in (6) ($\log(M-1) \simeq \log(M)$, $M/(M+1) \simeq 1$, $1/N \simeq 0$) holds for sufficiently large N and admits an "outage capacity" interpretation: a rate r_c is received without error if the information does not undergo any error along the L channels, which happens with probability $(1-\delta)^L$.

Notice that equation (6) captures in a single simple compact expression: a) the physical channel characteristics through the error exponent $E(\cdot)$, b) the complexity of the process at the relays through the pair (N, r_c) and c) the network topology through the number of traversed links L.

With the throughput expression in (6) we can answer questions like: given the number of links to be traversed L, and the maximum blocklength N, what is the optimal rate at which the network should be operated? Since (6) is a positive, continuous and differentiable function in $r_c \in [0, C]$, where C is the capacity of channel W, and since $R(0) = R(C) = 0$, the optimal rate at physical layer is the unique solution of

$$1 = 4\mathrm{e}^{-NE(r_c)} \left(1 + r_c L N \left|\frac{\mathrm{d}E(r_c)}{\mathrm{d}r_c}\right|\right) \quad (7)$$

Example 3 *The error exponent, in parametric form, of a BSC(ϵ_0) is given by [2, page 147]*

$$\begin{aligned} t &\in \left[\epsilon_0, \frac{\sqrt{\epsilon_0}}{\sqrt{\epsilon_0} + \sqrt{1-\epsilon_0}}\right] \\ r_c &= \log(2) - \mathcal{H}(t), \quad E(r_c) = D(t\|\epsilon_0) \quad \textit{and} \\ r_c &\in \left[0, \log(2) - \mathcal{H}\left(\frac{\sqrt{\epsilon_0}}{\sqrt{\epsilon_0} + \sqrt{1-\epsilon_0}}\right)\right] \\ E(r_c) &= \log(2) - \log(\sqrt{\epsilon_0} + \sqrt{1-\epsilon_0}) - r_c \end{aligned}$$

where $D(t\|\epsilon_0) = t\log\left(\frac{t}{\epsilon_0}\right) + (1-t)\log\left(\frac{1-t}{1-\epsilon_0}\right)$ *for* $(t, \epsilon_0) \in [0,1] \times [0,1]$. *Fig. 3 shows the normalized achievable R/C vs. N, with r_c given by (7), for $\epsilon_0 = 0.1$ and several values of L. It is clear that with moderate block length N a large fraction of capacity can be achieved, and actually the benefits are more pronounced for smaller N.*

2.2. Scaling

In this section we examine how the processing length N should scale with the network length L given that a rate

$$R = (1 - \xi) r_c$$

for $\xi \in [0, 1]$ must be guaranteed at the destination. By solving (6) with respect to N, with the approximation $(1-\delta)^L \simeq 1 - L\delta$, we obtain the following scaling law

$$N \simeq \frac{1}{E(r_c)} \log\left(\frac{4L}{\xi}\right) \tag{8}$$

Equation (8) quantifies the intuitive notion that in order to achieve hight rates ($\xi \to 0$ and/or $r_c \to C$) the intermediate nodes must increase their computation capability. That is, the longer the network, the larger the block length. From (8) it appears that the difficulty to achieve high rates comes from the channel noise ($E(r_c) \to 0$ as $r_c \to C$) rather than from the number of links to traverse since processing complexity only increases logarithmically with the network length.

3. General Networks

In our general network model all the assumptions made in Section 2 still apply, however the intermediate nodes are not longer restricted to form a line. For illustration purposes we shall use the network depicted in Fig. 4. We distinguish again between the cases of perfect processing, forwarding and finite processing, presented respectively in Sections 3.1, 3.2 and 3.3.

Figure 4: An example of a network connecting a source and a receiver.

3.1. Perfect Processing

Intermediate nodes are allowed to decode and reencode the whole codeword sent by the source. In this case the capacity is equal to the min-cut capacity between the source and the destination [3, Th.14.10.1].

Optimal routing reduces to identifying edge-disjoint max-flow paths, for example using the Ford-Fulkerson algorithm, and send through them independent information streams. The length of the paths does not affect the total capacity. However, identifying the minimum length max-flow paths allows to minimize the employed resources.

Example 4 *In Fig. 4, let each edge corresponds to a BSC(ϵ_0). The source can transmit to the destination at rate* $R = 2(1 - \mathcal{H}(\epsilon_0))$ *as the min-cut equals 2. A set of max-flow paths is* $\{(AB,\ BF),\ (AC,\ CF)\}$.

3.2. Forwarding

Each intermediate node is only allowed to forward the received information. For simplicity we shall assume that all edges of the network correspond to the same channel, and thus the length of a path indicates how good the end-to-end channel is (the shorter the path the less noisy the channel).

What is significantly different in this case is that the destination can receive multiple noisy observations of the same information stream from different incoming edges, and hence, by optimally combining them, can increase the end-to-end rate. That is, at intermediate nodes forwarding each information stream along branching paths that independently arrive at the receiver, generate *path diversity.* We propose the following routing algorithm:

- Identify the shortest max-flow paths from the source to the closest min-cut to the destination.
- After the min-cut, branch each max-flow path to a number of paths that meet at the destination.

Indeed, only one path can get routed through each edge of the min-cut. After the min-cut, if resources are available, having multiple observations of each information stream allows to increase the rate. Notice the similarity of the described path-diversity scheme with a multi antenna wireless system.

Example 5 *In the example of Fig. 4, edges AB and AC form the min-cut closest to the destination. As before, the paths* $(AB,\ BF)$ *and* $(AC,\ CF)$ *can be used to transmit the information streams* X_1 *and* X_2. *Additionally, as illustrated in Fig. 4, path* $(BD,\ DF)$ *or path* $(CD,\ DF)$ *can be used to provide the destination with an additional observation of* X_1 *or* X_2.

Example 6 *To gain an intuition of the rate improvements path-diversity can offer, since each information stream can be treated independently, we consider the configuration in Fig. 5 that consists of a BSC(ϵ_0) corresponding to the edge AB followed by L parallel BSC(ϵ). Fig. 6 plots the end-to-end rate for several values of the parameters.*

Figure 5: The BSC(ϵ_0) between nodes A and B is followed by L BSC(ϵ) channels between nodes B and C.

Figure 6: End-to-end rate for the configuration in Fig. 5 as a function of ϵ, for different values of ϵ_0 and L.

3.3. Finite Processing

In both the previous cases, the information streams corresponding to independent data were kept separated, in the first case (perfect processing) because there was no benefit, and in the second case (forwarding) because there were no processing capabilities in intermediate nodes to enable anything else. An interesting question is, given a very small processing capability, for example allowing intermediate nodes only symbol by symbol processing, whether "mixing" independent information along paths allows to increase the end-to-end rate.

The answer we give in this section is positive. That is, there exist networks, where the optimal rate under finite processing constraints can only be achieved when coding is applied jointly across information streams that carry independent information. Thus, unlike the previous two cases, the optimal routing is no longer edge-disjoint for independent information streams, which has a similar flavor to network coding.

In network coding the context is slightly different in that we assume error-free edges of a given capacity. In that context it is shown that rate benefits are achieved by mixing independent information streams when multicasting (where a source transmits information to multiple receivers) [1], while there are no rate benefits in the unicast case (where a source transmits information to a single receiver). Example 7 demonstrates that, when the edges of the graph model noisy channels and we have finite processing capabilities at intermediate nodes, rate benefits can also be achieved in the unicast case by allowing intermediate nodes to code across independent information streams.

Example 7 *Consider the configuration in Fig. 4, where node D can perform bit by bit processing. Assume that all the edges of the graph represent noiseless channels, except for edges BF, DF, and CF that model $BSC(\epsilon)$. Fig. 7 plots the achievable rate as a function of ϵ in the cases where edge DF is not used, forwards bit X_1, and carries the binary sum (over F_2) of X_1 and X_2. The binary sum achieves the optimal rate*[1]. *Numerical results over a number of different configurations strengthed our conclusion that the optimal rate cannot always be achieved when keeping indepedent information streams seperate.*

Figure 7: Achievable rate when transmitting the binary sum $X_1 + X_2$, just X_1, or nothing over the edge DF.

[1]We verified through exhaustive search over all the functions of X_1 and X_2 that the rate achieved with the binary sum is indeed optimal.

4. Conclusions

In this paper we examined from an information theory point of view what are the rate benefits that finite processing at intermediate nodes of a network can offer, and how the overall routing and coding problem has to be adapted to achieve the optimal rate.

A. The worst channel case

Let $I(Q, W)$ denote the mutual information between input X and output Y of a channel with transition matrix W when the input X is distributed according to Q,

$$I(Q, W) = \sum_{i=1}^{M}\sum_{j=1}^{M} Q_i [W]_{i,j} \log \frac{[W]_{i,j}}{\sum_{m=1}^{M} Q_m [W]_{m,j}}$$

i.e., $[W]_{m,j} = \Pr[Y = j | X = m]$ and $Q_m = \Pr[X = m]$. Let W' denote the "expanded channel" matrix induced by the length-N process at the relays. Assume $L = 1$. Our goal is to determine $\max_Q I(Q, W')$. However, from (2) we only know that for every $m \in \{1 \cdots M\}$

$$1 - [W]_{m,m} = \sum_{j \neq m} [W]_{m,j} \leq \delta \tag{9}$$

Clearly

$$\max_Q I(Q, W') \geq \min_W \left\{ \max_Q I(Q, W) \right\} \tag{10}$$

where the minimization is performed within the of channels that satisfy (9). Because both the sets where Q and W vary are convex, and because $I(Q, W)$ is convex-∩ in Q and convex-∪ in W, minimization and maximization in (10) can be swapped. Hence,

$$\begin{aligned}
&\max_Q \min_W I(Q, W) = \max_Q \min_W \{H(Y) - H(Y|X)\} \\
&\leq \max_Q \min_W \{ \log(M) - \sum_m Q_m H(Y|X = m)\} \\
&\leq \max_Q \left\{ \log(M) - \sum_m Q_m \max_W \{H(Y|X = m)\} \right\} \\
&= \max_Q \left\{ \log(M) - \sum_m Q_m H(\boldsymbol{v}) \right\} = \log(M) - H(\boldsymbol{v}) \\
&= \log(M) - \delta \log(M - 1) - \mathcal{H}(\delta)
\end{aligned} \tag{11}$$

where $\boldsymbol{v} = [1-\delta, \delta/(M-1), \cdots, \delta/(M-1)]$ and $H(\boldsymbol{v}) = \sum_m v_m \log(1/v_m)$. The first inequality in deriving (11) follows from [2, Th. 2.3.1 pag. 23] and the second last equality follows from [2, Ex. 2.14 pag. 508]. Since (11) is achieved by a uniform input Q and a channel W whose rows equal all possible permutations of $\boldsymbol{v}$, we conclude that (11) is actually $\max_Q \min_W I(Q, W)$ and hence

$$\max_Q I(Q, W') \geq \log(M) - \delta \log(M-1) - \mathcal{H}(\delta)$$

The same holds for a cascade of L channels.

References

[1] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE IT* Vol. 46, pp. 1204–1216, July 2000.

[2] R. G. Gallager, "Information Theory and Reliable Communication," *J. Willey and Sons,* 1986.

[3] T. Cover and J. Thomas, "Elements of Information Theory," *Wiley Series.*

[4] C. Fragouli, E. Soljanin and A. Shokrollahi "Network coding as a Coloring Problem," *CISS 2004,* Princeton University, available at *http://algo.epfl.ch*.

[5] M. Feder, D. Ron, and A. Tavory, "Bounds on linear codes for network multicast", *Electronic Colloquium on Computational Complexity*, Report No. 33, 2003.

[6] A. Shokrollahi "Raptor Codes", *submitted to IEEE IT.*

[7] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, V. Stemann. "Practical Loss-Resilient Codes", *ACM Symposium on Theory of Computing*, 1997.

[8] A. B. Kiely, J. T. Coffey, "On the capacity of a cascade of channels" *IEEE IT,* Vol. 39, No. 4, pp. 1310–1321, July 1993.