Alcom-FT Technical Report Series
ALCOMFT-TR-01-65

Efficient Communication Establishment in Adverse Communication Environments *

S. NIKOLETSEAS
Computer Technology Institute and Patras University, Greece

P. SPIRAKIS
Computer Technology Institute and Patras University, Greece

Abstract

We consider a large network of n nodes which supports only the following *unreliable basic communication primitive:* when a node requests communication then this request *may fail*, independently of other requests, with probability $f < 1$. Even if it succeeds, the request is answered by returning a stable link to a *random* node of the network. Furthermore, each node is allowed to perform *at most g such requests* where g is constant (independent of n).

We present a protocol that manages (with probability tending to 1) to establish a *very long path* (i.e. of length $\Theta(n)$) *of stable links* in such a network, provided $g > \frac{4\ln 2}{1-f}$. This protocol thus manages to *establish end-to-end communication* for a large part of the network, even in the (worst) case when the failure probaility f is constant and the number g of random requests is constant too. This protocol is *optimal* in the sense that no linear path can be established with a sublinear number of requests. We also show that our protocol is *fair* in the sense that any node can equiprobably be in the established path. We expect this protocol to be useful for establishing communication in uncontrolled *adverse* communication environments.

Keywords

Networks, Reliability, Communication Primitives, Markov Processes, End-to-End Communication

1 Introduction

Unreliable communication networks have been extensively studied because they pose fundamental problems in distributed computing. The end-to-end communication problem in such networks is to send information from a sender node to a

*This work was partially supported by the EU Projects ESPRIT LTR ALCOM-IT, IST FET-OPEN ALCOM-FT, IMPROVING RTN ARACNE and the Greek GSRT Project PENED99-ALKAD.

receiver node. Without such capabilities (at least for a big network part) it is not possible to perform distributed computations. Also, as the size of the network increases, the likelihood of a "fault" also increases. Good protocols for end-to-end communication allow distributed algorithms to treat an unreliable network as a reliable channel.

We consider here a network of n nodes which supports only one very poor communication primitive: each node may request for a *random* direct connection. Such a request either *fails* (independently of other such requests, with probability $f < 1$), or, if it succeeds, it returns *an established direct link* to some network node, randomly and equiprobably chosen. Furthermore, we assume that each node may request such links *at most g times* (g a constant). Once a direct link is established, the nodes may use it to exchange information for as many times as they wish.

Many different protocols for end-to-end communication have been developed when links (and/or intermediate nodes) may fail ([1], [2], [3], [4], [5], [6], [8], [9]). Also, previous work of the authors examined structural properties of unreliable networks ([10], [11], [12]).

In this paper, we present a simple protocol which establishes (almost surely) a *very long path* in the network (involving a constant fraction, at least, of all the nodes), provided $g(1-f) > 4\ln 2$. Note that our result shows that *even when the number of random requests g is constant and the failure probability f is constant too such a long path can still be established* (and thus solve the end-to-end communication problem for a big subset of the network node pairs).

Our communication establishment protocol can start only from a single node. Other nodes become *awake* (and start participating in the protocol) via reception of a special message after a direct successful link establishment. In addition, our protocol gives *equal chance* to any network node to participate in the long path. Thus it has a strong *fairness* property. To our knowledge, this is the first time that a protocol is presented to achieve such a long path in a distributed way under so adverse communication conditions.

2 Motivation

The problem considered here is a special case of the problem of establishing (in a distributed way) a *stable communication subnetwork* in an adverse communication environment. A long path is such a subnetwork and has many advantages in its use (sessions, pipelining, flow of high-volume of data such as video if the links allow such speeds etc).

In adverse communication environments, a failure in a link request may model interference, jams or noise. The limit g on the number of attempts models either power limitations or situations where such attempts are intercepted by adversaries and a high number of attempts may compromise the security of communication or may lead to the detection of the exact position of the sender.

The return of a *random* stable link models situations where the targeted nodes are "listening" (able to receive) each only at random, locally selected time intervals. In mobile communication stations it may model random proximity.

From another point of view, our research aims in discovering as weak communication primitive operations as possible, which can be used as building blocks for the construction of higher, more dependable, layers of communication protocols.

3 The Model

The network consists of a set V of n nodes. Initially, exactly one node u_0 is *active* and the others are *sleeping*. An active node is allowed to request for at most g direct random link establishments. Such requests are executed one-by-one and each may independently *fail* with probability $f < 1$. If it succeeds, then a direct link to a *randomly chosen network node* is permanently established. The model allows repetitions, i.e. the same node may be returned in two different successful requests. If a link is established, then the sending node is notified and the receiving node is now active.

For the purpose of *restarting a protocol* we assume the existence of a special message

$$M =< restart, U >$$

which can be sent by any active node and *always succeeds*. Here, U is a subset of V and must consist only of sleeping nodes. M then selects one node, x, of U randomly and the protocol restarts in the subnetwork $\{x, U - x\}$.

In fact, we will show in the paper how to *simulate* M with just unreliable random communication attempts.

A protocol P for such a network is a computation that each active node may execute together with any information exchange with other active nodes which must be carried out along *established links*. The update of the established link information is a duty of the protocol.

4 Summary of Results

We present here a protocol P_1 which manages to establish a path of length $\Theta(n)$ in the network provided $g > \frac{4\ln 2}{1-f}$. Our protocol has the property that any node has the same chance to be in the constructed path (fairness). Note that even in the worst case of *constant* failure probability f, only a small, *constant* number g of random attempts per node is enough to establish a long path. Thus, P_1 is *optimal* with respect to the number of direct random communication requests, in the sense that no linear path can be established with a sublinear number of requests.

We also extend P_1 to a protocol P_2 which *does not need* the special *restart* message M described in the preceeding section, and has essentially the same properties.

5 The Protocol P_1

In the sequel, let $g = 2\lambda$ ($\lambda > 1$ an integer). We group the g allowed requests per node into two equal sets: the green set and the red set, each consisting of λ random link requests. We use two sets of requests so that we have two chances of independence in the technical analysis: first, we try to establish the long path link by link considering only green edges. When extension of the constructed path using green edges is not possible anymore, and a red edge can be used to extend the path, we use it and then we continue using only green edges again. If at some step path extension is not possible using red edges either, we backtrack to the first vertex from which red edges have not bewen tried yet (and thus their use might be successful in extending the path). If this process leads to a path which shrinks to a single vertex that we can no longer leave, we start a new process of path construction from a new initial vertex. All this is done by the distributed protocol by using suitable communication messages and a special token for the distributed control in the way we explain below.

More formally, our protocol maintains a triple (P_k, U_k, R_k) where P_k is a (directed) *path* of established links (the *current* path after k logical protocol "steps"), U_k is a set of sleeping vertices and R_k is the set of *red* nodes (whose green set of requests have been used). Let V be the set of network nodes ($|V| = n$).

The protocol starts at the awake node x_0 (the sets of the corresponding triple are $P_0 = x_0, U_0 = V - \{x_0\}$ and $R_0 = \emptyset$).

Our protocol is motivated by the nice proof of F. de la Vega ([13]) for the existence of long paths in (sparse) random graphs $G_{n,p}$ with $p = \frac{c}{n}$, where $c > 0$ a constant. The differences are (i) the distributed version of our protocol and (ii) the significantly weaker random graph model that our network allows for, i.e. at most g requested random links, each with a failure probability f. It is not intuitively obvious that the result of [13] extends to such a weaker model of random graphs.

Suppose that the protocol has achieved a directed path P_k of established links with initial vertex x_k, end vertex y_k. The end vertex of the path holds a special token T. Suppose also $|U_k| = u_k$. If $U_k = \emptyset$ the protocol terminates. Else, there are three cases:

Case 1: $y_k \notin R_k$. Node y_k (which has the token) tries its green requests one by one until either a (green) link $y_k - U_k$ is established or all such requests fail. In the former case P_k is extended by the node y_{k+1} (formerly in U_k) with the established link i.e. $P_{k+1} = P_k\, y_{k+1}$, $U_{k+1} = U_k - \{y_{k+1}\}$ and $R_{k+1} = R_k$ and the token is passed to y_{k+1}. In the case of no green requests and no new link, y_k sets $P_{k+1} = P_k$, $U_{k+1} = U_k$ and $R_{k+1} = R_k \cup \{y_k\}$ and informs the nodes in the current path about the new triple (P_k, U_k, R_k).

Case 2: $y_k \in R_k$. Try the red requests out of y_k one by one. If y_k manages to link to a node in U_k (say y_{k+1}) then y_k extends P_k to y_{k+1}, sets $U_{k+1} = U_k - \{y_{k+1}\}$, $R_{k+1} = R_k$ and the token goes to y_{k+1}. Else, y_k sends a message backwards in P_k to find the first node whose red requests have not been tried. Call it y_{k+1}. The token passes then to y_{k+1}. The path now shrinks to just the part from x_k to y_{k+1} and y_{k+1} is added to R_k. Also $U_{k+1} = U_k$. The last updates are easily done by circulation of a message from y_k backwards till y_{k+1} is found. Then y_{k+1} resets the triple and informs its path.

Case 3: Now the path has shrunk to a single vertex ($x_k = y_k$). We try its red requests and if a link with U_k is established, the new node will get the token and continue (as in Case 2) with its green requests. Else, y_k cannot do anything but to send the restart message $M(U_k)$ and the protocol starts again from a new awaken vertex x_{k+1}, i.e. $P_{k+1} = x_{k+1}$, and executes only in the remaining sleeping nodes i.e. $U_{k+1} = U_k - \{x_{k+1}\}$ and $R_{k+1} = R_k$.

6 Properties of the Protocol P_1 and its Analysis

Let $l(k)$ be the length of P_k and let $r_k = |R_k|$. Also let $u_0 = |U_0| = n-1$. By definition $n - u_0 + r_0 = 1$. In each case our protocol increases at most one of $n - u_k$ and r_k by at most one. Thus,

$$(n - u_{k+1}) + r_{k+1} \leq (n - u_k) + r_k + 1$$

i.e.

$$(n - u_k) + r_k \leq k + 1$$

for every k. Also,

$$V - U_k - \{vertices\ of\ P_k\} \subseteq R_k$$

thus

$$l(k) = |P_k| - 1 \geq n - u_k - r_k - 1 \geq n - u_k - (k + 1 - (n - u_k)) - 1$$

i.e.

$$l(k) \geq 2(n - u_k) - k - 2 \tag{1}$$

Let $H_k = \{P_i, U_i, R_i, E_i\}_{i=0}^{k}$ where E_i is the set of the requests for random links already done. Briefly, H_k is the *history* of the protocol up to logical time k.

At each logical step k the protocol performs some requests for random links (green or red).

The probability of failing to get a link of the form $y_k - U_k$ (and thus failing to extend the path) is precisely

$$\begin{aligned}\Pr\{u_{k+1} = j | u_k = j, H_k = H\} &= \sum_{t=0}^{\lambda} \binom{\lambda}{t} \left(\frac{u_k}{n}\right)^t f^t \left(1 - \frac{u_k}{n}\right)^{\lambda - t} \\ &= \left(1 - \frac{u_k}{n}(1-f)\right)^{\lambda}\end{aligned}$$

for all k, j and H. Thus, this probability *depends only* on u_k (since n, f and λ are given parameters of the protocol) and since u_{k+1} is either u_k or $u_k - 1$ we get:

Lemma 1 *The sequence $\{u_k\}$ is a Markov Chain.*

Definition 1 *Let*

$$X_i = \max\{k - l : u_k = u_l = i\}$$

for $1 \le i \le n-1$, so that $X_i + 1$ is the length of the sojourn time of u_k in state i, and let

$$Y_j = \sum_{i=j+1}^{n-1} (X_i + 1)$$

i.e. the hitting time of state j of u_k.

By definition then

$$\Pr\{u_k \le j\} = \Pr\{Y_j \le k\} \tag{2}$$

The above, in combination with equation (1), is enough to show that protocol P_1 indeed constructs a very long path with probability tending to 1, provided that for some pair (j, k) we have $\Pr\{Y_j \le k\} \to 1$ and $2j + k$ is small.

From the Markov property of u_k, $\{X_i\}$ are *independent* geometric random variables of mean

$$E(X_i) = \frac{\left(1 - \frac{i}{n}(1-f)\right)^{\lambda}}{1 - \left(1 - \frac{i}{n}(1-f)\right)^{\lambda}}$$

and variance

$$\sigma^2(X_i) = \frac{\left(1 - \frac{i}{n}(1-f)\right)^{\lambda}}{\left[1 - \left(1 - \frac{i}{n}(1-f)\right)^{\lambda}\right]^2}$$

However

$$\left(1 - \frac{i}{n}(1-f)\right)^{\lambda} \le \exp\left(-\frac{\lambda i}{n}(1-f)\right)$$

Let $r = \frac{\lambda}{n}(1-f)$. Choose $j = \lceil \frac{\ln 2}{r} \rceil$. Then

$$\left(1 - \frac{j}{n}(1-f)\right)^{\lambda} \le e^{-jr} \le \frac{1}{2}$$

thus

$$\sigma^2(Y_j) = \sum_{i=j+1}^{n-1} \sigma^2(X_i) \leq 2(n-j+1) < 2n \tag{3}$$

We also need the following upper bound on $E(Y_j)$:

Lemma 2

$$E(Y_j) < n - \frac{1}{r} e^{-rn}$$

for $j = \lceil \frac{\ln 2}{r} \rceil$ *and* $r = \frac{\lambda}{n}(1-f)$

Proof. We have clearly

$$E(X_i + 1) \leq \frac{1}{1 - e^{-r_i}}$$

thus

$$\begin{aligned} E(Y_j) &= \sum_{i=j+1}^{n-1} E(X_i+1) \leq \sum_{i=j+1}^{n-1} \frac{1}{1-e^{-r_i}} \\ &\leq \int_j^{n-1} (1-e^{-rx})^{-1}\,dx \leq \frac{1}{r}\int_{\ln 2}^{rn} (1-e^{-y})^{-1}\,dy \\ &= \frac{1}{r}\left[y - \sum_{m=1}^{\infty} \frac{1}{m} e^{-my} \right]_{\ln 2}^{rn} = \frac{1}{r}\left[rn - \sum_{m=1}^{\infty} \frac{1}{m} e^{-mrn} \right] \\ &< n - \frac{1}{r} e^{-rn} \end{aligned}$$

because

$$\sum_{m=1}^{\infty} \frac{1}{m} 2^{-m} = \log 2$$

□

Thus, by applying the following version of Chebyshev's inequality:

$$\Pr\{Y_j < E(Y_j) + w(n)\sigma(Y_j)\} \geq 1 - \frac{1}{w(n)}$$

and by choosing any $w(n) \to \infty$ as $n \to \infty$ (e.g. $w(n) = \log\log\log n$) we get

$$\Pr\left\{ Y_j < n - \frac{e^{-rn}}{r} + w(n)\sqrt{n} \right\} \to 1$$

Hence, by equations (1) and (2), with probability tending to 1 our protocol constructs a path of length at least

$$2\left(n - \frac{\ln 2}{r}\right) - n + \frac{1}{r} e^{-rn} - w(n)\sqrt{n} - 2 \geq n\left(1 - \frac{2\ln 2}{\lambda(1-f)}\right)$$

Thus we have shown:

Theorem 1 *Protocol P_1 constructs with probability tending to 1 a path of length at least*

$$n\left(1-\frac{4\ln 2}{g(1-f)}\right)$$

Since the basic communication primitive always returns a random vertex (if it succeeds) we also have

Theorem 2 (Fairness) *Each node of V can be in the path constructed by P_1 equiprobably.*

7 Removing the need for the restart message

In our analysis in the previous section we demonstrated a pair (j,k) with

$$j=\frac{2n\ln 2}{g(1-f)}=\frac{n\ln 2}{\lambda(1-f)}$$

and

$$k=n-\frac{n}{\lambda(1-f)}e^{-\lambda(1-f)}+\omega(n)\sqrt{n}$$

such that

$$\Pr\{u_k\leq j\}\to 1 \text{ as } 1-\frac{1}{w(n)}$$

by choosing any $w(n)\to\infty$ as $n\to\infty$ (e.g. $w(n)=\log\log\log n$).

Thus, the protocol succeeds within a linear number of logical steps and when u_k is still $\Theta(n)$. In order to find the expected number of restart messages needed by protocol P_1, we model the stochastic process $l(k)$ (i.e. the path length) as a random walk in one dimension. By the description of the protocol, if p_k is the probability that $l(k)$ is extended by 1 (via green requests) then

$$p_k=1-\left(1-\frac{u_k}{n}(1-f)\right)^{\lambda}$$

and if q_k denotes the probability of shrinking to the previous node (of unused red requests) then clearly

$$q_k=1-p_k$$

Definition 2 *Let a round of the protocol P_1 be the period until the protocol sends a restart message.*

Definition 3 *Let $D(z)$ be the expected duration of a round, conditioned on the event that the length of the path is $z(k)$.*

Then, clearly

$$D_{z(k)} = p_k D_{z(k)+1} + q_k D_{z(k)-1} + 1$$

with boundary conditions

$$D_0 = 0, D_n = 0$$

This is a difference equation which is very difficult to solve because p_k, q_k vary with time. However, we note easily that

$$q_k \geq f^{\lambda} \tag{4}$$

Now, note that initially $u_0 = n-1$ and consider the sequence of periods at which $u_k \geq \frac{n}{2}$. Then for these periods

$$q = \left(1 - \frac{u_k}{n}(1-f)\right)^{\lambda} \leq \left(1 - \frac{(1-f)}{2}\right)^{\lambda} \leq \exp\left(-\frac{\lambda(1-f)}{2}\right)$$

and to get

$$p - q > \varepsilon$$

(where ε a cosntant) we need

$$p > \frac{1}{2} + \varepsilon$$

i.e

$$2 < (1-2\varepsilon)\exp\left(\frac{\lambda(1-f)}{2}\right)$$

which is true because of the condition $\lambda(1-f) > 4\ln 2$ (for ε small but constant).

Thus, for the sequence of these periods, the walk is stochastically dominated by a walk W with p, q constant and $p - q = \varepsilon$, in the sense that

$$\Pr\left\{D_z \geq x\right\} \geq \Pr\left\{D_z^W \geq x\right\}, \forall x > 0$$

with high probability. But, from known properties of the difference equation (see e.g. [7]) for the random walk presented above, we get

$$E\left(D_z^W\right) \simeq \frac{z}{q-p} - \frac{n}{q-p} = \Theta\left(\frac{n}{\varepsilon}\right)$$

But the number of returns to the origin of our stochastic process is bounded by the maximum value of the logical step k divided by $E\left(D_z\right)$ and thus it is a constant. So, we get:

Lemma 3 *The expected number of restart messages of P_1 is bounded above by a constant.*

Let us now allow (instead of a restart message) a new protocol P_2 which keeps $g = 3\lambda$ requests per node. We modify P_1 so that the node which keeps the token always stores the maximum path constructed up to now.

When the protocol shrinks to a single node (i.e. at end of a period) then that node notifies all the nodes of the maximum path seen in the past, to try to use their third batch of request messages in order to restart the protocol. From the previous analysis, the total number of requests that can be used for restart purposes is

$$\lambda \cdot (\text{length of maximum path})$$

This is with high probability at least $\alpha \log n$ for any constant α. The probability of failure to restart the protocol is then just the probability of these extra requests all failing to hit U_k. But this is at most

$$\left(1 - \frac{u_k}{n}(1-f)\right)^{\alpha \log n}$$

and since

$$u_k = \Theta(n) = \frac{4 \ln 2}{g(1-f)} n$$

we get a failure probability to restart the protocol less than

$$\left(1 - \frac{2 \ln 2}{g}\right)^{\alpha \log n} < n^{-2\alpha(\log 2)/g}$$

By choosing e.g. $\alpha > \frac{2g}{\log 2}$ we get a failure probability at most $\frac{1}{n^2}$ per restart attempt leading to $o\left(\frac{1}{n}\right)$ total failure probability to restart the protocol.

Thus, we get a modified protocol P_2 which uses only the unreliable basic communication primitive and satisfies:

Theorem 3 *There exists a protocol P_2 which does not need the special restart message. It uses only a linear total number of unreliable basic communication requests, and still achieves a $\Theta(n)$ path of stable links with probability tending to 1 as $n \to \infty$.*

References

[1] Adler and Fich. The Complexity of End-to-End Communication in Memoryless Networks. *ACM PODC*, 239–248, 1999.

[2] Afek, Awerbuch, Gafni, Mansour, Rosen and Shavit. Slide - The Key to Polynomial End-to-End Communication. *J. of Algorithms*, vol. 22, no 1, 158–186, 1997.

[3] Afek and Gafni. End-to-End Communication in Unreliable Networks. *7th ACM PODC*, 131–148, 1988.

[4] Awerbuch, Mansour and Shavit. Polynomial End-to-End Communication. *30th FOCS*, 358–363, 1989.

[5] Awerbuch, Patt-Shamir and Varghese. Self-Stabilizing End-to-End Communication. *J. of High Speed Networks*, vol. 5, no 4, 365–381, 1996.

[6] Dolev and Welch. Crash Relient Communication in Dynamic Networks. *IEEE Transactions on Computers*, vol. 46, 14–26, 1997.

[7] Feller. An Introduction to Probability Theory and its Applications. *John Wiley*, New York, 1957.

[8] Kushilevitz, Ostrofsky and Rosen. Log-Space Polynomial End-to-End Communication. *28th STOC*, 559–568, 1995.

[9] Ladner, LaMarca and Tempero. Counting Protocols for Reliable End-to-End Transmission. *JCSS*, vol. 56, no 1, 96–111, 1998.

[10] Nikoletseas, Palem, Spirakis and Yung. Short Vertex Disjoint Paths and Multiconnectivity in Random Graphs: Reliable Network Computing. *21st ICALP*, 508 – 515, 1994.

[11] Nikoletseas and Spirakis. Expander Properties in Random Regular Graphs with Edge Faults. *12th STACS*, 421 – 432, 1995.

[12] Nikoletseas, Pantziou, Psycharis and Spirakis. On the Reliability of Fat-Trees. *3rd International European Conference on Parallel Processing (Euro-Par)*, 208 – 217, 1997.

[13] de la Vega, Long Paths in Random Graphs. *Studia Sci. Math. Hungar.*, 14, 335-340, 1979.

Sotiris E. Nikoletseas is with the Computer Technology Institute (CTI), 61 Riga Fereou Street, 26221 Patras, Greece. Also, with the Computer Engineering and Informatics Department, University of Patras, Greece. Email: nikole@cti.gr

Paul G. Spirakis is with the Computer Technology Institute (CTI), 61 Riga Fereou Street, 26221 Patras, Greece. Also, with the Computer Engineering and Informatics Department, University of Patras, Greece. Email: spirakis@cti.gr