Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computer Networks 50 (2006) 1630–1638

Computer Networks

www.elsevier.com/locate/comnet

Header-size lower bounds for end-to-end communication in memoryless networks [☆]

Pierre Fraigniaud [a,*], Cyril Gavoille [b]

[a] CNRS, Lab. de Recherche en Informatique, University Paris-Sud, Batiment 490, 91405 Orsay Cedex, France

[b] University of Bordeaux 1, LaBRI, 33405 Talence Cedex, France

Available online 25 October 2005

Abstract

The end-to-end communication problem is a protocol design problem, for sending a packet from a specified source-node s to a specified target-node t, through an unreliable asynchronous memoryless communication network. The protocol must insure reception and termination. In this paper, we measure the complexity of the protocol in term of header size, i.e., the quantity of information that must be attached to the packets to insure their delivery. We show that headers of $\Omega(\log\log\tau)$ bits are required in every network, where τ denotes the treewidth of the network. In planar networks, $\Omega(\log\tau)$ bits are required.

Keywords: End-to-end; Sequence transmission; Treewidth

1. Introduction

The *end-to-end communication* problem is the problem of sending a (sequence of) packet(s) from a specified source-node s to a specified target-node t, through an *unreliable* communication network G (see, e.g., [13,29]). The *sequence transmission* [38] problem and the *reliable communication* [24] problem are other names for the end-to-end communication problem (cf. the survey [25]). By an unreliable network, it is generally meant that links can lose, reorder, and duplicate packets. Moreover, networks are assumed to be *asynchronous*, i.e., the time for a packet to traverse a link is finite but otherwise unbounded. In particular, a processor cannot distinguish between an inoperational link and an operational link which is just very slow. Hence, an instance of the end-to-end communication problem is described by an (unreliable and asynchronous) network, modeled by an undirected graph G, and two nodes s and t of G. Solving the problem consists in designing a distributed protocol which (1) allows s to send a packet, or a sequence of packets, to t

[☆] The two authors are supported by the Actions Spécifiques CNRS "Dynamo" and "Algorithmique des grands graphes", and by the project "PairAPair" of the ACI Masses de Données. P. Fraigniaud receives additional support from the INRIA project "Grand Large". A preliminary version of this paper appeared in the proceedings of the 17th International Symposium on Distributed Computing (DISC), Sorrento, Italy, October 1–3, 2003.

[*] Corresponding author.

E-mail addresses: pierre@lri.fr (P. Fraigniaud), gavoille@labri.fr (C. Gavoille).

1389-1286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2005.09.025

through the network G, and (2) generates a finite amount of traffic for each packet. In other words, the end-to-end protocol must satisfy the two following requirements: (1) *reception*, i.e., the target must eventually receive at least one copy of each packet sent by the source; and (2) *termination*, i.e., after a finite time, no copy of the packet(s) remains in the network.

1.1. The oblivious single-packet end-to-end communication problem

This paper considers a static model, that is, we assume that each link is either operational or not. If there exists at least one operational path from s to t in G, it is the role of the protocol to find such a non-faulty path, which is of course a priori unknown. Note that the case of dynamic faults, that is when links can alternate between being operational and inoperational, can be treated similarly by assuming, as in [1,14,28], infinitely frequent path stability, i.e., infinitely often there is a path P from s to t such that a packet sent from s along P will arrive at t (see also [3]).

Performance of end-to-end communication protocols is commonly measured in terms of (1) the amount of communication performed over the links of the network, and (2) the amount of storage space used by intermediate nodes in the networks. *Oblivious* protocols, a.k.a. *memoryless* protocols, take their routing decision at every node x (i.e., on which link(s) x has to forward a packet) based solely on the content of the header of the packet. In particular, a node does not store any knowledge about the traffic that previously passed through it. It can however forward several copies of the received packet, and it can modify the header of this packet. As mentioned in [1], the practical advantage of oblivious protocols is their high tolerance to processor crashes. Indeed, as soon as a node recovers from a crash, the protocol is ready to proceed with no risk of corruption due to an altered writable memory (RAM). Moreover, the extremal behavior of oblivious protocols (in the sense that they consume no local memory at all) allows concentrating the analysis of end-to-end communication protocols on the amount of information transmitted over the links of the network. More specifically, this paper focuses on minimizing the *packet-header size*, i.e., the quantity of additional information that must be attached to every packet to insure its correct delivery.

The header content has two distinct functions: (1) to control the order in which packets arrive at the destination; (2) to find a route from the source s to the destination t (reception), and to insure that residual packets are eventually removed from the network (termination). In this paper, we are interested in the routing part of the problem, that is, in the problem of finding the non-faulty route from the source to the destination, while insuring termination. We will therefore concentrate our analysis on the process of sending a single packet from s to t. In other words, we consider the *single-packet* end-to-end communication problem, as opposed to the *stream-of-packets* end-to-end communication problem, the latter problem requiring the transmission of a sequence of packets from the source to the destination [22]. Hence, let us summarize our problem.

1.1.1. Our problem

We are given an unreliable and asynchronous network G, and two nodes s and t of G. We consider the design of an oblivious distributed protocol which allows the transmission of a packet from s to t (if there is a fault-free path between s and t in G), and which eventually leaves the network empty of packets. Such a protocol is required to use packet-headers of small size. The quality of the protocol is indeed measured by the maximum size of the headers involved during its execution.

1.1.2. Previous work

In the context of static link failures (i.e., every link is operational or not but its status does not change during the execution of the routing protocol), the *hop-count* protocol [32] uses headers of size $O(\log n)$ in n-node networks. It proceeds by flooding the network as follows. The source sends a copy of the original packet to all its neighbors, with header 1. A node receiving a packet whose header contains the hop count $i < n - 1$ updates the header by replacing i by $i + 1$, and forwards a copy of the packet to each of its neighbors. A node receiving a packet whose header contains the hop count $n - 1$ removes the packet from the network. If s and t are connected despite the faulty links, then a path of length at most $n - 1$ exists between s and t, and therefore at least one copy of the packet sent by s eventually arrives at t. Moreover, the remaining copies of the packet are removed from the network after a finite time since no packet can traverse n or more links.