

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE


[Membership](#) [Publications/Services](#) [Standards](#) [Conferences](#) [Careers/Jobs](#)
IEEE Xplore®
 RELEASE 1.6

Welcome

United States Patent and Trademark Office

[Help](#) [FAQ](#) [Terms](#) [IEEE Peer Review](#)

Quick Links

» Search Absl

Welcome to IEEE Xplore®

- Home
- What Can I Access?
- Log-out

[Search Results](#) [PDF FULL-TEXT 776 KB] [PREV](#) [NEXT](#) [DOWNLOAD CITATION](#)

 Order Reuse Permissions
 RIGHT LINK

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

On four-connecting a triconnected graph

Hsu, T.

Dept. of Comput. Sci., Texas Univ., Austin, TX, USA;

This paper appears in: Foundations of Computer Science, 1992. Proceedings., Annual Symposium on

Search

- By Author
- Basic
- Advanced

Meeting Date: 10/24/1992 - 10/27/1992

Publication Date: 24-27 Oct. 1992

Location: Pittsburgh, PA USA

On page(s): 70 - 79

Reference Cited: 37

Inspec Accession Number: 4488295

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Abstract:

The author considers the problem of finding a smallest set of edges whose addition f connects a triconnected graph. This is a fundamental graph-theoretic problem that has applications in designing reliable **networks**. He presents an $O(n\alpha(m,n)+m)$ time sequential algorithm for four-connecting an undirected graph G that is triconnected by adding the smallest number of edges, where n and m are the number of vertices and edges in G , respectively, and $\alpha(m, n)$ is the inverse Ackermann function. He presents a new lower bound for the number of edges needed to four-connect a triconnected graph. The form of this lower bound is different from the form of the lower bound known for biconnectivity augmentation and triconnectivity augmentation. The new lower bound applies for arbitrary k , and gives a tighter lower bound than the one known earlier for the number of edges needed to **k-connect** a $(k-1)$ -connect graph. For $k=4$, he shows that this lower bound is tight by giving an efficient algorithm for finding a set of edges with required size whose addition four-connects a triconnected graph.

Index Terms:

[computational complexity](#) [computational geometry](#) [four-connecting](#) [graph theory](#) [graph-theoretic problem](#) [inverse Ackermann function](#) [reliable networks](#) [triconnected graph](#) [computational complexity](#) [computational geometry](#) [four-connecting](#) [graph theory](#) [graph-theoretic problem](#) [inverse Ackermann function](#) [reliable networks](#) [triconnected graph](#)

Documents that cite this document

There are no citing documents available in IEEE Xplore at this time.

[Search Results](#) [\[PDF FULL-TEXT 776 KB\]](#) [PREV](#) [NEXT](#) [DOWNLOAD CITATION](#)

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

A Flexible Architecture for Multi-Hop Optical Networks

A. Jaekel, S. Bandyopadhyay

and

A. Sengupta

School of Computer Science,

University of Windsor,

Windsor, Ontario N9B 3P4, CANADA

Department of Computer Science

University of South Carolina

Columbia, SC 29208

Abstract

It is desirable to have low diameter logical topologies for multihop lightwave networks. Researchers have investigated regular topologies for such networks. Only a few of these (e.g., GEMNET [8]) are scalable to allow the addition of new nodes to an existing network. Adding new nodes to such networks requires a major change in routing scheme. For example, in a multistar implementation, a large number of retuning of transmitters and receivers and/or renumbering nodes are needed for [8]. In this paper, we present a scalable logical topology which is not regular but it has a low diameter. This topology is interesting since it allows the network to be expanded indefinitely and new nodes can be added with a relatively small change to the network. In this paper we have presented the new topology, an algorithm to add nodes to the network and two routing schemes.

Keywords: *Optical networks, multihop networks, scalable logical topology, low diameter networks.*

1. Introduction

Optical networks [1] are interconnections of high-speed broadband fibers using *lightpaths*. Each lightpath provides traverses one or more fibers and uses one wavelength division multiplexed (WDM) channel per fiber. In a multihop network, each node has a small number of lightpaths to a few other nodes in the network. The physical topology of the network determines how the lightpaths get defined. For a multistar implementation of the physical topology, a lightpath $u \rightarrow v$ is established when node u broadcasts to a passive optical coupler at a particular wavelength and the node v picks up the optical signal by tuning its receiver to the same wavelength. For a wavelength routed network, a lightpath $u \rightarrow v$ might be established through one or several fibers interconnected by router nodes. The lightpath definition between the nodes in an optical network is usually represented by a directed graph (or digraph) $G = (V, E)$ (where V is the set of nodes and E is the set of the edges) with each node of G representing a

node of the network and each edge (denoted by $u \rightarrow v$) representing a lightpath from u to v . G is usually called the logical topology of the network. When the lightpath $u \rightarrow v$ does not exist, the communication from a node u to a node v occurs by using a (graph-theoretic) path (denoted by $u \rightarrow x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_{k-1} \rightarrow v$) in G using k hops through the intermediate nodes x_1, x_2, \dots, x_{k-1} . The information is buffered at intermediate nodes and, to reduce the communication delay, the number of hops should be small. If a shortest graph-theoretic path is used to establish a communication from u to v , the maximum hop distance is the *diameter* of G . Clearly, the lightpaths need to be defined such that G has a small diameter and low average hop distance. The indegree and outdegree of each node should be low to reduce the network cost. However, a reduction of the degree usually implies an increase in the diameter of the digraph, that is, larger communication delays. The design of the logical topology of a network turns out to be a difficult problem in view of these contradictory requirements. Several different logical topologies have been proposed in the literature. An excellent review of multihop networks is presented in [1].

Both regular and irregular structures have been studied for multihop structures [2], [3], [4], [5], [6], [7]. All the proposed regular topologies (e.g., shuffle nets, de Bruijn graphs, ~~torus~~, ~~hypercubes~~) enjoy the property of simple routing algorithms, thereby avoiding the need of complex routing tables. Since the diameter of a digraph with n nodes and maximum outdegree d is of $O(\log_d n)$, most of the topologies attempt to reduce the diameter to $O(\log_d n)$. One common property of these network topologies is the number of nodes in the network must be given by some well-defined formula involving network parameters. This makes the topology non-scalable. In short, addition of a node to an existing network is virtually impossible. In [8], the principle of shuffle interconnection between nodes in a shufflenet [4] is generalized (the generalized version can have any number of nodes in each column) to obtain a scalable network topology called GEMNET. A similar idea of generalizing

the Kautz graph has been studied in [9] showing a better diameter and network throughput than GEMNET. Both these scalable topologies are given by regular digraphs.

One topology that has been studied for optical networks is the bidirectional ring network. In such networks, each node has two incoming lightpaths and two outgoing lightpaths. In terms of the graph model, each node has one outgoing edge to and one incoming edge from the preceding and the following node in the network. Adding a new node to such a ring network involves redefining a fixed number of edges and can be repeated indefinitely.

Our motivation was to develop a topology which has the advantages of a ring network with respect to scalability and the advantages of a regular topology with respect to low diameter. In other words, our topology has to satisfy the following characteristics:

- The diameter should be small
- The routing strategy should be simple
- It should be possible to add new nodes to the network indefinitely with the least possible perturbation of the network.
- Each node in the network should have a predefined upper limit on the number of incoming and outgoing edges.

In this paper we introduce a new scalable topology for multihop networks where the graph is not, in general, regular. Given integers n and d , our proposed topology can be defined for n nodes with a fixed number of incoming and outgoing edges in the network. The major advantage of our scheme is that, as a new node is added to the network, most of the existing edges of the logical topology are not changed, implying that the routing schemes between the existing nodes need little modification. The edges to and from the new added node can be implemented by defining new lightpaths which is small in number, namely, $O(d)$. For multistar implementation, for example, this can be accomplished by retuning $O(d)$ transmitters and receivers.

The paper is organized as follows. In section 2, we describe the proposed topology and derive its pertinent properties. Section 3 presents two routing schemes for the proposed topology and establishes that the diameter is $O(\log_d n)$. Our experiments in section 4 show that, for a network with n nodes and having an indegree of at most $d+1$, an outdegree of d and the average hop distance is approximately $\log_d n$. We have concluded with a critical summary in section 4.

2. Scalable topology for multihop networks

2.1 Proposed interconnection topology

Given two integers n and d , $d \leq n$, we define the interconnection topology of the network as a digraph G in the following. As mentioned earlier, the digraph is not

regular - the indegree and outdegree of a node varies from l to $d+1$. We will assume that there is no k , such that

$n = d^k$; if $n = d^k$ for some k , our proposed topology is the same as given by [2]. Let k be the integer such that

$d^k < n < d^{k+1}$. Let Z_k be the set of all $(k+1)$ -digit strings

choosing digits from $Z = \{0, 1, 2, \dots, d-1\}$ and let any string of Z_k be denoted by $x_0 x_1 \dots x_k$. We divide Z_k

into $k+2$ sets S_0, S_1, \dots, S_{k+1} such that all strings in Z_k having x_j as the left most occurrence of 0 is included in S_j ,

$0 \leq j \leq k$ and all strings with no occurrence of 0 (i.e. $x_j \neq 0, 0 \leq j \leq k$) is included in S_{k+1} . We note that

$$|S_{k+1}| = (d-1)^{k+1} \quad \text{and} \quad |S_j| = (d-1)^j d^{k-j},$$

$0 \leq j \leq k$. We define an ordering relation between every pair of strings in Z_k . Each string in S_i is smaller than each

string in S_j if $i < j$. For two strings $\sigma_1, \sigma_2 \in S_j$,

$0 \leq j \leq k+1$, if $\sigma_1 = x_0 x_1 \dots x_k$ and $\sigma_2 = y_0 y_1 \dots y_k$

and t is the largest integer such that $x_t \neq y_t$, then $\sigma_1 < \sigma_2$ if $x_t < y_t$.

Definition: For any string $\sigma_1 = x_0 x_1 \dots x_i \dots x_j \dots x_k$, the string $\sigma_2 = x_0 x_1 \dots x_j \dots x_i \dots x_k$ obtained by interchanging the digits in the i^{th} and the j^{th} position in σ_1 , will be called the *i-j-image* of σ_1 .

Clearly, if σ_2 is the *i-j-image* of σ_1 then σ_1 is the *i-j-image* of σ_2 and if $x_i = x_j$, σ_1 and σ_2 represent the same node.

We will represent each node of the interconnection topology by a distinct string $x_0 x_1 \dots x_k$ of Z_k . As

$d^k < n < d^{k+1}$, all strings of Z_k will not be used to represent the nodes in G . We will use n smallest strings from Z_k to represent the nodes of G . Suppose the largest string representing a node is in S_M . We will use a node and its string representation interchangeably. We will use the term *used* string to denote a string of Z_k which has been already used to represent some node in G . All other strings of Z_k will be called *unused* strings.

Property 1: all strings of S_0 are used strings.

Property 2: if $\sigma \in S_j$ is an used string, then all strings

of S_0, S_1, \dots, S_{j-1} are also used strings.

Property 3: If $\sigma_1 = 0x_1\dots x_k$, σ_2 is the 0-1-image of σ_1 and $x_1 \neq 0$, then $\sigma_2 \in S_1$.

Property 4: If $\sigma_1 = 0x_1\dots x_k$, $x_1 \neq 0$ and σ_2 , the 0-1-image of σ_1 , is an unused string, then all strings of the form $x_1x_2\dots x_kj$, $0 \leq j \leq d-1$ are unused strings.

The proofs for Properties 1 - 4 are trivial and are omitted.

We now define the edge set of the digraph G . Let any node u in G be represented by $x_0x_1\dots x_k$. The outgoing edges from node u are defined as follows:

- There is an edge $x_0x_1x_2\dots x_k \rightarrow x_1x_2\dots x_kj$ whenever $x_1x_2\dots x_kj$ is an used string, for some $j \in Z$,
- There is an edge $0x_1x_2\dots x_k \rightarrow x_10x_2\dots x_k$ whenever the following conditions hold:
 - a) $x_1x_2\dots x_kj$ is an unused string for at least one $j \in Z$ and
 - b) $x_10\dots x_k$, the 0-1-image of u , is an used string
- There is an edge $0x_1x_2\dots x_k \rightarrow 0x_2\dots x_kj$ for all $j \in Z$ whenever the following conditions hold:
 - a) $x_1 \neq 0$ and
 - b) $x_10x_2\dots x_k$, the 0-1-image of u , is an unused string

We note that if $u \in S_j$, $j > 0$, node $v = x_1x_2\dots x_kj$ always exists (from property 2, since $v \in S_{j-1}$). As an example, we show a network with 5 nodes for $d=2, k=2$ in figure 1. We have used a solid line for an edge of the type $x_0x_1x_2\dots x_k \rightarrow x_1x_2\dots x_kj$, a line of dots for and a line of dashes and dots for an edge of the type $0x_1x_2\dots x_k \rightarrow 0x_2\dots x_kj$. We note that the edge from 010 to 100 satisfies the condition for both an edge of the type $x_0x_1x_2\dots x_k \rightarrow x_1x_2\dots x_kj$ and an edge of the type $0x_1x_2\dots x_k \rightarrow x_10x_2\dots x_k$.

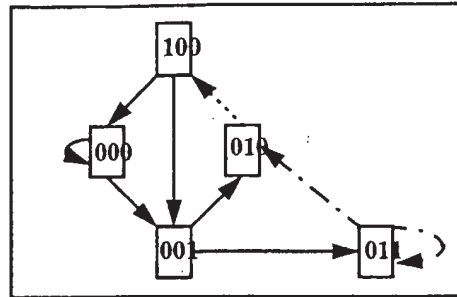


Figure 1: Interconnection topology with $d=2, k=2$ for $n=5$ nodes.

2.2 Limits on Nodal Degree

In this section, we derive the upper limits for the indegree and the outdegree of each node in the network. We will show that, by not enforcing the regularity, we can easily achieve scalability. As we add new nodes to the network, minor modifications of the edges in the logical topology suffice, in contrast to large number of changes in the edge-set as required by other proposed methods.

Theorem 1: In the proposed topology, each node has an outdegree of up to d .

Proof: Let u be a node in the network given by $x_0x_1\dots x_k \in S_j$. We consider the following three cases:

- i) $0 < j \leq k$: For every v given by $x_1x_2\dots x_kt$ for all t , $0 \leq t \leq d-1$ is an used string since $v \in S_{j-1}$. Therefore the edge $u \rightarrow v$ exists in the network. If $u \in S_j$, $j > 0$, these are the only edges from u . Hence, u has outdegree d .
- ii) $j = 0$: According to our topology defined above, u will have an edge to $x_1x_2\dots x_kj$ whenever $x_1x_2\dots x_kj$ is an used string for some $j \in Z$. We have three sub-cases to consider:
 - If $x_1x_2\dots x_kj$ is an used string for all j , $0 \leq j < d$ then u has outdegree d .
 - Otherwise, if p of the strings $x_1x_2\dots x_kj$ are used strings, for some j , $0 \leq j < d$ and the 0-1-image of u is also an used string, then u has edges to all the p nodes with used strings of the form $x_1x_2\dots x_kj$ and to the 0-1-image of u . Hence u has outdegree $p+1$. Here u has an outdegree of at least 1 and at most d .
 - Otherwise, if the 0-1-image of u is an unused string, then all strings of the form $x_1x_2\dots x_kj$ are unused

strings (Property 4) and u has d outgoing edges to nodes of the form $0x_2x_3\dots x_kj$, $0 \leq j < d$. Hence u has outdegree d .

iii) $j = k + 1$: If p of the strings $x_1x_2\dots x_kj$ are used strings, for some j , $0 \leq j < d$, then u has outdegree of p . We note that $x_1x_2\dots x_k0 \in S_k$ is an used string. Therefore $1 \leq p \leq d$, and u has an outdegree of at least 1 and at most d .

Theorem 2: In the proposed topology, each node has an indegree of up to $d+1$.

Proof: Let us consider the indegree of any node v given by $y_0y_1\dots y_k \in S_j$. As described in 2.1, there may be three type of edges to node v as follows:

- An edge $ty_0y_1\dots y_{k-1} \rightarrow y_0y_1\dots y_k$ whenever $ty_0y_1\dots y_{k-1}$ is an used string, for some $t \in Z$. There may be at most d edges of this type to v .
- If $y_1 = 0$, $y_0 \neq 0$ there may be an edge $0y_0y_2\dots y_k \rightarrow y_0y_1\dots y_k$
- If $y_0 = 0$ and $ty_0y_1\dots y_{k-1}$ is an unused string for some $t \in Z$, there is an edge $0ty_1\dots y_{k-1} \rightarrow y_0y_1\dots y_k$. There may be at most d edges of this type to v .

We have to consider 3 cases, $j = 0$, $j = 1$ and $j > 1$. If $j > 1$, the only edges are of the type $ty_0y_1\dots y_{k-1} \rightarrow y_0y_1\dots y_k$ and there can be up to d such edges. If $j = 1$, in addition to the edges are of the type $ty_0y_1\dots y_{k-1} \rightarrow y_0y_1\dots y_k$, there can be only one edge of the type $0y_0y_2\dots y_k \rightarrow y_0y_1\dots y_k$. Thus the total number of edges cannot exceed $d + 1$, in this case. If $j = 0$, an edge of the type $0ty_1\dots y_{k-1} \rightarrow y_0y_1\dots y_k$ exists if and only if the corresponding edge of type $ty_0y_1\dots y_{k-1} \rightarrow y_0y_1\dots y_k$ does not exist in the network. Therefore, there are always exactly d incoming edges to v in this case.

2.3 Node Addition to an Existing Network

In this section we consider the changes in the logical topology that should occur when a new node is added to the network. We show that at most $O(d)$ edge changes in G would suffice when a new node is added to the network. When a multistar implementation is considered, this means

$O(d)$ retuning of transmitters and receivers, whereas for a wavelength routed network, this means redefinition of $O(d)$ lightpaths. In contrast, for other proposed topologies [8], [9] the number of edge modifications needed was $O(nd)$. As discussed in the previous section, the nodes are assigned the smallest strings defined earlier. Addition of a new node u implies that we will assign the smallest unused string to the newly added node. Let the string be $x_0x_1\dots x_k \in S_j$. We consider the following three cases:

- i) $1 < j \leq k$: For every v given by $x_1x_2\dots x_kt$, $0 \leq t \leq d - 1$, $v \in S_{j-1}$. Therefore v is an used string and we have to add a new edge $u \rightarrow v$ to the network. The node given by $w_0 = 0x_0x_1\dots x_{k-1}$ is guaranteed to be an used string, since $w_0 \in S_0$ and we have to add a new edge $w_0 \rightarrow u$ to the network. If $x_k = d - 1$, we have to delete the edge from w_0 to its 0-1-image at this time. For every w given by $tx_0x_1\dots x_{k-1}$, $1 \leq t \leq d - 1$, $w \in S_{j+1}$ and is an unused string. Therefore w_0 is the only predecessor of u .
- ii) $j = k + 1$: If $v = x_1x_2\dots x_kt$, $0 \leq t \leq p - 1$ is an used string, we add a new edge $u \rightarrow v$ to the network. We note that $x_1x_2\dots x_k0 \in S_k$ is an used string. Therefore, there is at least one v such that $u \rightarrow v$ exists. Similarly, if $w = tx_0x_1\dots x_{k-1}$, $0 \leq t \leq p - 1$ is an used string, we add a new edge $w \rightarrow u$ to the network. We note that $w_0 = 0x_0x_1\dots x_{k-1} \in S_0$ is an used string. Therefore, there is at least one w such that $w \rightarrow u$ exists. If $x_k = d - 1$, we delete the edge from w_0 to its 0-1-image at this time.
- iii) $j = 1$: Let $w_c = 0x_0x_2\dots x_k$ be the 0-1-image of u . Before inserting u , the node $0x_0x_2\dots x_k$ was connected to all nodes $v = 0x_2\dots x_kt$, $0 \leq t \leq d - 1$ (case iii in our topology given in 2.1). We have to
 - delete the edge $w_c \rightarrow v$ for each node $v = 0x_2\dots x_kt$ in the network.
 - add an edge $u \rightarrow v$ for each node $v = 0x_2\dots x_kt$ in the network.
 - add a new edge $w_0 = 0x_0x_1\dots x_{k-1} \rightarrow u$ to the network

- If $w_c \neq w_0$, add an edge $w_c \rightarrow u$ to the network.
- If $x_k = d - 1$, and $w_0 \neq 0x_0000\dots0$ delete the edge from w_0 to its 0-1-image.

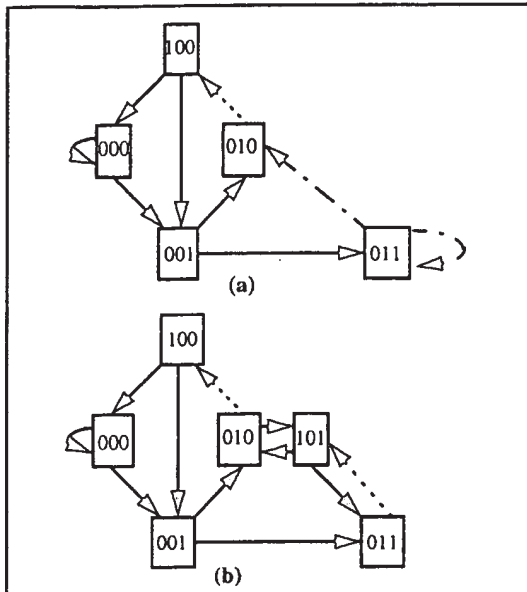


Figure 2: Expanding a topology with $d=2, k=2$ from (a) $n=5$ to (b) $n=6$ nodes.

Figure 2(a) shows again the network with 5 nodes given in Figure 1. We choose the smallest unused string $u = 101$ to represent the new node being inserted. The node u will have outgoing edges (shown by solid lines) to all nodes of the form $01j$, to nodes 010 and 011 . The 0-1 image of u is node 011 . Hence all edges from 011 to nodes 010 and 011 are deleted and a new edge from 101 to 011 is inserted (shown by a dashed line). Also a new edge is inserted from node 010 to 101 . The final network is shown in Figure 2(b)

3. Routing strategy

In this section, we present two routing schemes in the proposed topology from any source node S to any destination node D . Let S be given by the string $x_0x_1\dots x_k \in S_j$ and D be given by the string $y_0y_1\dots y_k \in S_l$.

3.1 Routing scheme

Let l be the length of the longest suffix of the string $x_0x_1\dots x_k$ that is also a prefix of $y_0y_1\dots y_k$ and let

$\sigma(S, D)$ denote the string $x_0x_1\dots x_ky_ly_{l+1}y_{l+2}\dots y_k$ of length $2(k+1)-l$. Since $\sigma(S, D)$ is of length $2(k+1)-l$, it has $(k+1)-l+1$ substrings, each of length $(k+1)$. Two of these substrings represent S and D . Since S and D are nodes in the network, these two substrings are used strings. If all the remaining $k-l$ substrings of $\sigma(S, D)$ having length $k+1$ are also used strings, then a routing path from S to D of length $k+1-l$ exists as given by the sequence of nodes given in (1) below.

$$S = x_0x_1\dots x_k \rightarrow x_1x_2\dots x_ky_ly_{l+1} \rightarrow x_2\dots x_{2k-1}x_ky_ly_{l+1} \rightarrow \dots \rightarrow x_ky_ly_{l+1}\dots y_{k-2}y_{k-1} \rightarrow y_0y_1\dots y_k = D \quad (1)$$

In other words, if all the $k-l+2$ substrings of $\sigma(S, D)$ are used strings, we can use $\sigma(S, D)$ to represent the path from S to D in (1).

Property 5: If all the $k-l+2$ substrings of $\sigma(S, D)$ are used strings, $\sigma(S, D)$ represents the shortest path from S to D .

However, if some of the substrings of $\sigma(S, D)$ are not used strings, then some of the corresponding nodes do not currently appear in the network and hence this path does not exist. We note that any two consecutive strings in $\sigma(S, D)$ is given by $\alpha\beta$, where $\alpha = x_ix_{i+1}\dots x_ky_ly_{l+1}\dots y_{l+i}$, $0 \leq i \leq k-l-1$, and

$$\beta = x_{i+1}x_{i+2}\dots x_ky_ly_{l+1}\dots y_{l+i}y_{l+i+1}. \text{ Let } \beta \text{ be the first unused string in (1). According to our topology, either } \alpha \in S_0 \text{ or } \alpha \in S_{k+1}.$$

Property 6: If $\alpha \in S_0$ and

$\gamma = x_{i+1}0x_{i+2}\dots x_ky_ly_{l+1}\dots y_{l+i}$, the 0-1-image of α is an unused string, then

- $\sigma(S, \alpha)$ represents a path from S to α of length i ,
- there exists a path $\alpha \rightarrow \gamma \rightarrow \delta = 0x_{i+2}\dots x_ky_ly_{l+1}\dots y_{l+i}y_{l+i+1}$
- $\sigma(\delta, D)$ is a string of length $k+2-l-i$

Property 7: If $\alpha \in S_0$ and

$\gamma = x_{i+1}0x_{i+2}\dots x_ky_ly_{l+1}\dots y_{l+i}$, the 0-1-image of α is an unused string, then

- $\sigma(S, \alpha)$ represents a path from S to α of length i ,
- there exists a path

$$\alpha \rightarrow \delta = 0x_{i+2} \dots x_k y_i y_{i+1} \dots y_{l+i} y_{l+i+1}$$

- $\sigma(\delta, D)$ is a string of length $k+2-l-i$

Properties 6 and 7 follow directly from our topology defined in 2.1.

Property 8: If a network contains all nodes in S_0, S_1, \dots, S_k then

- there exists an edge $S \rightarrow \gamma = x_1 x_2 \dots x_k 0$ and
- $\sigma(\gamma, D)$ represents a path from α to D of length that cannot exceed $k+1$.

Proof of Property 8: Since the network contains all nodes in S_0, S_1, \dots, S_k , $\gamma \in S_j$ for some j , $j \leq k$ and must exist. Our topology (section 2.1) ensures that the edge $S \rightarrow \gamma$ exists. The path given below consists only strings belonging to groups S_i , $0 \leq i \leq k$ and hence are used strings:

$\gamma \rightarrow x_2 \dots x_k 0 y_0 \rightarrow x_3 \dots x_k 0 y_0 \rightarrow \dots \rightarrow y_0 y_1 \dots y_k$. The number of edges in the path is $k+1$, hence the proof.

Theorem 3: The diameter of a network using the proposed topology cannot exceed $2(k+1)$.

Proof: We consider any source-destination pair (S, D) . If all the $k-l+2$ substrings of $\sigma(S, D)$ are used strings, $\sigma(S, D)$ represents the shortest path from S to D and cannot exceed $k+1$. If β is the first unused string in (1), and α is the preceding string then we have to consider two cases:

Case 1) $\alpha \in S_0$: In this situation we can apply property 6 if 0-1-image of α is an used string. Otherwise we can use property 7. If we can use property 6, it means we need two edges to insert the digit y_{l+i+1} . Alternatively, if we can use property 7, it means we need one edge to insert the digit y_{l+i+1} .

Case 2) $\alpha \in S_{k+1}$: In this situation we discard the partial path from S to α . The first edge in our new path will be $S = x_0 x_1 \dots x_k \rightarrow x_1 x_2 \dots x_k 0$. Property 8 guarantees that once we have this situation, we can always start all over again inserting digits y_0, y_1, \dots, y_k without ever encountering an unused string and requires a

maximum of $k+1$ edges. This represents the worst case since there may exist a shorter path by finding the longest suffix of $x_1 x_2 \dots x_k 0$ that matches the corresponding prefix of D . In this case the path cannot exceed $k+2$.

Case 1 can appear repeatedly. The worst situation is when we have to apply it to insert every digit of D . In other words, the path in this case can be as long as $2(k+1)$.

3.2 Example of routing

Let us consider the network of Figure 2(b). Suppose, $S = 011$ and $D = 001$. Since the only outgoing edge from 011 is to its 0-1-image 101, the first edge in the path is $011 \rightarrow 101$. From 101, we shift in the successive digits of the destination. So, the final path is given by $S = 011 \rightarrow 101 \rightarrow 010 \rightarrow 100 \rightarrow 001 = D$. In this particular example, there are no nodes belonging to group $k+1$. So, case 2 is not used.

4. Experiments to determine the average hop distance

We carried out some experiments to determine the average hop distance \bar{h} . In each of these experiments, we have started with a given value of d , the minimum indegree (or outdegree) and a specified value of an integer k . The network with d^k nodes is identical to that given in [8]. We have calculated the average hop distance \bar{h} of this network from the hop distances of every source/destinations pairs using the routing scheme described in the previous section. Then we have added a node to the network and calculated \bar{h} for the new network in the same way. We continued the process of adding nodes until the network contained d^{k+1} nodes. The results of the experiments are shown in Table 1 and reveal the following:

- The average hop distance is approximately $k+1$.
- The average hop distance starts at approximately k and increases to approximately $k+1$ as we start adding nodes to the network.

We interpret these results as follows. Even though the diameter is $2(k+1)$, the number of lightpaths through paths involving 0-1 images, which increase the number of hops, is relatively small. Our network is identical to that in [2] when the number of nodes in the network is d^k or d^{k+1} and, for these values, it is known that the network has a diameter of

k and k+1 respectively.

Table 1: Variation of average hop distance with number of nodes

Number of nodes	d	k	average hop \bar{h}
10	3	2	2.4333
13	3	2	2.6154
16	3	2	2.6618
19	3	2	2.4954
22	3	2	2.5974
25	3	2	2.5148
10	2	3	2.7000
12	2	3	2.9470
14	2	3	2.8022
16	2	3	2.8333
65	4	3	3.5954
75	4	3	3.8366
85	4	3	4.1077
95	4	3	4.2215
105	4	3	4.5172
115	4	3	4.5506
18	2	4	3.5915
20	2	4	3.67630
22	2	4	3.8636
24	2	4	4.30181
26	2	4	3.7908
28	2	4	3.7169

5. Conclusions

In this paper we have introduced a new graph as a logical network for multihop networks. We have shown that our network has an attractive average hop distance compared to existing networks. The main advantage of our

approach is the fact that we can very easily add new nodes to the network. This means that the perturbation of the network in terms of redefining edges in the network is very small in our architecture. The routing scheme in our network is very simple and avoids the use of routing tables.

Acknowledgments: The work of A. Jaekel and S. Bandyopadhyay has been supported by research grants from the Natural Science and Engineering Research Council of Canada. The work of A. Sengupta has been partially supported by Office of Naval Research grant # N00014-97-1-0806.

REFERENCES

- [1] B. Mukherjee, "WDM-based local lightwave networks part II: Multihop systems," *IEEE Network*, vol. 6, pp. 20-32, July 1992.
- [2] K. Sivarajan and R. Ramaswami, "Lightwave Networks Based on de Bruijn Graphs," *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, pp. 70-79, Feb 1994.
- [3] K. Sivarajan and R. Ramaswami, "Multihop Networks Based on de Bruijn Graphs." *IEEE INFOCOM '91*, pp. 1001-1011, Apr. 1991.
- [4] M. Hluchyj and M. Karol, "ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 9, pp.1386-1397, Oct. 1991.
- [5] B. Li and A. Ganz, "Virtual topologies for WDM star LANs: The regular structure approach," *IEEE INFOCOM '92*, pp.2134-2143, May 1992.
- [6] N. Maxemchuk, "Routing in the Manhattan street network," *IEEE Trans. on Communications*, vol. 35, pp. 503-512, May 1987.
- [7] P. Dowd, "Wavelength division multiple access channel hypercube processor interconnection," *IEEE Trans. on Computers*, 1992.
- [8] J. Innes, S. Banerjee and B. Mukherjee, "GEMNET : A generalized shuffle exchange based regular, scalable and modular multihop network based on WDM lightwave technology", *IEEE/ACM Trans. Networking*, Vol 3, No 4, Aug 1995.
- [9] A. Venkateswaran and A. Sengupta, "On a scalable topology for Lightwave networks", *Proc IEEE INFOCOM'96*, 1996.



Membership Publications/Services Standards Conferences Careers/Jobs



Welcome United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review

Quick Links

» Search Abst

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Search Results [PDF FULL-TEXT 580 KB] NEXT DOWNLOAD CITATION

Order Reuse Permissions RIGHT LINK

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

A flexible architecture for multihop optical networks

Jaekel, A. Bandyopadhyay, S. Sengupta, A. Sch. of Comput. Sci., Windsor Univ., Ont., Canada; This paper appears in: Computer Communications and Networks, 1998. Proceedings. 7th International Conference on

Search

- By Author
- Basic
- Advanced

Meeting Date: 10/12/1998 - 10/15/1998
Publication Date: 12-15 Oct. 1998
Location: Lafayette, LA USA
On page(s): 472 - 478
Reference Cited: 9
Number of Pages: xxii+929
Inspec Accession Number: 6226042

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Abstract:

It is desirable to have low diameter logical topologies for multihop lightwave network. Researchers have investigated regular topologies for such networks. Only a few of them (e.g., GEMNET) are scalable to allow the addition of new nodes to an existing network. Adding new nodes to such networks requires a major change in routing scheme. For example, in a multistar implementation a large number of retuning of transmitters at receivers anti/or renumbering nodes are needed for GEMNET. We present a scalable logical topology which is not regular but it has a low diameter. This topology is interesting since it allows the network to be expanded indefinitely and new nodes can be added with a relatively small change to the network. We present the new topology, an algorithm to add nodes to the network and two routing schemes.

Index Terms:

network topology optical fibre networks optical receivers optical transmitters telecommunications network routing wavelength division multiplexing GEMNET WDM algorithm flexible architecture low diameter logical topologies multihop lightwave networks multihop optical networks multistar implementation network nodes receivers regular topologies retuning routing scheme scalable logical topology transmitters

Documents that cite this document

There are no citing documents available in IEEE Xplore at this time.

Search Results [PDF FULL-TEXT 580 KB] NEXT DOWNLOAD CITATION

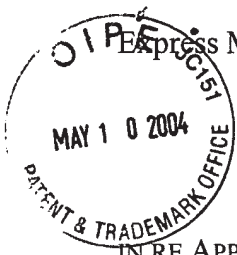
[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

05/12/04

2153/ \$

Attorney Docket No. 030048002US



Express Mail No. EV335515821US

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: FRED B. HOLT *ET AL.*
APPLICATION NO.: 09/629,570
FILED: JULY 31, 2000
FOR: **JOINING A BROADCAST CHANNEL**

EXAMINER: BRADLEY E. EDELMAN
ART UNIT: 2153
CONF. NO: 5411

Amendment Under 37 C.F.R. § 1.111

RECEIVED

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Sir:

MAY 17 2004

Technology Center 2100

The present communication responds to the Office Action dated January 12, 2004 in the above-identified application. Please extend the period of time for response to the Office Action by one month to expire on May 12, 2004. Enclosed is a Petition for Extension of Time and the corresponding fee. Please amend the application as follows:

Amendments to the Specification begin on page 2.

Amendments to the Claims are reflected in the listing of claims beginning on page 4.

Remarks/Arguments begin on page 8.

Amendments to the Specification:

In accordance with 37 CFR 1.72(b), an abstract of the disclosure has been included below. In addition, the status of the related cases listed on page 1 of the specification has been updated.

Therefore, please add the Abstract as shown below:

A technique for adding a participant to a network is provided. This technique allows for the simultaneous sharing of information among many participants in a network without the placement of a high overhead on the underlying communication network. To connect to the broadcast channel, a seeking computer first locates a computer that is fully connected to the broadcast channel. The seeking computer then establishes a connection with a number of the computers that are already connected to the broadcast channel. The technique for adding a participant to a network includes identifying a pair of participants that are connected to the network, disconnecting the participants of the identified pair from each other, and connecting each participant of the identified pair of participants to the added participant.

Please amend the "Cross-Reference to Related Applications" to read as follows:

This application is related to U.S. Patent Application No. 09/629,576, entitled "BROADCASTING NETWORK," filed on July 31, 2000 (Attorney Docket No. 030048001 US); U.S. Patent Application No. 09/629,570, entitled "JOINING A BROADCAST CHANNEL," filed on July 31, 2000 (Attorney Docket No. 030048002 US); U.S. Patent Application No. 09/629,577, "LEAVING A BROADCAST CHANNEL," filed on July 31, 2000 (Attorney Docket No. 030048003 US); U.S. Patent Application No. 09/629,575, entitled "BROADCASTING ON A BROADCAST CHANNEL," filed on July 31, 2000 (Attorney Docket No. 030048004 US); U.S. Patent Application No. 09/629,572, entitled "CONTACTING A BROADCAST CHANNEL," filed on July 31, 2000 (Attorney Docket No. 030048005 US);

U.S. Patent Application No. 09/629,023, entitled “DISTRIBUTED AUCTION SYSTEM,” filed on July 31, 2000 (Attorney Docket No. 030048006 US); U.S. Patent Application No. 09/629,043, entitled “AN INFORMATION DELIVERY SERVICE,” filed on July 31, 2000 (Attorney Docket No. 030048007 US); U.S. Patent Application No. 09/629,024, entitled “DISTRIBUTED CONFERENCING SYSTEM,” filed on July 31, 2000 (Attorney Docket No. 030048008 US); and U.S. Patent Application No. 09/629,042, entitled “DISTRIBUTED GAME ENVIRONMENT,” filed on July 31, 2000 (Attorney Docket No. 030048009 US), the disclosures of which are incorporated herein by reference.

Amendments to the Claims:

Following is a complete listing of the claims pending in the application, as amended:

1. (Currently amended) A computer-based, non-routing table based, non-switch based method for adding a participant to a network of participants, each participant being connected to three or more other participants, the method comprising:

identifying a pair of participants of the network that are connected wherein a seeking participant contacts a fully connected portal computer, which in turn sends an edge connection request to a number of randomly selected neighboring participants to which the seeking participant is to connect;

disconnecting the participants of the identified pair from each other; and

connecting each participant of the identified pair of participants to ~~the added~~ the seeking participant.

2. (Original) The method of claim 1 wherein each participant is connected to 4 participants.

3. (Original) The method of claim 1 wherein the identifying of a pair includes randomly selecting a pair of participants that are connected.

4. (Original) The method of claim 3 wherein the randomly selecting of a pair includes sending a message through the network on a randomly selected path.

5. (Original) The method of claim 4 wherein when a participant receives the message, the participant sends the message to a randomly selected participant to which it is connected.

6. (Currently amended) The method of claim 4 wherein the randomly selected path is ~~approximately~~ proportional to the diameter of the network.

7. (Original) The method of claim 1 wherein the participant to be added requests a portal computer to initiate the identifying of the pair of participants.

8. (Original) The method of claim 7 wherein the initiating of the identifying of the pair of participants includes the portal computer sending a message to a connected participant requesting an edge connection.

9. (Currently amended) The method of claim 8 wherein the portal computer indicates that the message is to travel a ~~certain~~ distance proportional to the diameter of the network and wherein the participant that receives the message after the message has traveled that ~~certain~~ distance is one of the participants of the identified pair of participants.

10. (Currently amended) The method of claim 9 wherein the certain distance is ~~approximately~~ twice the diameter of the network.

11. (Original) The method of claim 1 wherein the participants are connected via the Internet.

12. (Original) The method of claim 1 wherein the participants are connected via TCP/IP connections.

13. (Original) The method of claim 1 wherein the participants are computer processes.

14. (Currently amended) A computer-based, non-switch based method for adding nodes to a graph that is m-regular and m-connected to maintain the graph as m-regular, where m is four or greater, the method comprising:

identifying p pairs of nodes of the graph that are connected, where p is one half of m_2

wherein a seeking node contacts a fully connected portal node, which in turn

sends an edge connection request to a number of randomly selected neighboring

nodes to which the seeking node is to connect;

disconnecting the nodes of each identified pair from each other; and
 connecting each node of the identified pairs of nodes to ~~the added~~ the seeking node.

15. (Original) The method of claim 14 wherein identifying of the p pairs of nodes includes randomly selecting a pair of connected nodes.

16. (Original) The method of claim 14 wherein the nodes are computers and the connections are point-to-point communications connections.

17. (Original) The method of claim 14 wherein m is even.

18–31. (Previously cancelled)

32. (Currently amended) A computer-readable medium containing instructions for controlling a computer system to connect a participant to a network of participants, each participant being connected to three or more other participants, the network representing a broadcast channel wherein each participant forwards broadcast messages that it receives to all of its neighbor participants, wherein each participant connected to the broadcast channel receives all messages that are broadcast on the network, the network containing a method wherein messages are numbered sequentially so that messages received out of order are queued and rearranged to be in order, by a method comprising:

identifying a pair of participants of the network that are connected;
 disconnecting the participants of the identified pair from each other; and
 connecting each participant of the identified pair of participants to ~~the added~~ a seeking participant.

33. (Original) The computer-readable medium of claim 32 wherein each participant is connected to 4 participants.

34. (Original) The computer-readable medium of claim 32 wherein the identifying of a pair includes randomly selecting a pair of participants that are connected.

35. (Original) The computer-readable medium of claim 34 wherein the randomly selecting of a pair includes sending a message through the network on a randomly selected path.

36. (Original) The computer-readable medium of claim 35 wherein when a participant receives the message, the participant sends the message to a randomly selected participant to which it is connected.

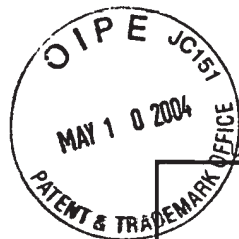
37. (Currently amended) The computer-readable medium of claim 35 wherein the randomly selected path is ~~approximately~~ twice a diameter of the network.

38. (Original) The computer-readable medium of claim 32 wherein the participant to be added requests a portal computer to initiate the identifying of the pair of participants.

39. (Original) The computer-readable medium of claim 38 wherein the initiating of the identifying of the pair of participants includes the portal computer sending a message to a connected participant requesting an edge connection.

40. (Currently amended) The computer-readable medium of claim 38 wherein the portal computer indicates that the message is to travel a ~~certain~~ distance that is twice the diameter of the network and wherein the participant that receives the message after the message has traveled that ~~certain~~ distance is one of the identified pair of participants.

41–49. (Previously cancelled)



Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM <i>(to be used for all correspondence after initial filing)</i>	Application Number	09/629,570	
	Filing Date	July 31, 2000	
	First Named Inventor	Fred B. Holt	
	Art Unit	2153	
	Examiner Name	Bradley E. Edelman	
Total Number of Pages in This Submission	26	Attorney Docket Number	030048002US

ENCLOSURES (Check all that apply)		
<input checked="" type="checkbox"/> Fee Transmittal Form <input checked="" type="checkbox"/> Fee Attached <input checked="" type="checkbox"/> Amendment/Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input checked="" type="checkbox"/> Petition for Extension of Time <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance communication to Group <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): Return Postcard
<div style="border: 1px solid black; padding: 2px; display: inline-block;">Remarks</div>		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Chun Ng
Signature	
Date	May 10, 2004

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as Express Mail No. EV335515821US in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name	Melody J. Almberg		
Signature		Date	5/10/2004

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

<h1 style="margin: 0;">FEE TRANSMITTAL</h1> <h2 style="margin: 0;">for FY 2004</h2> <p style="font-size: small; margin: 5px 0;">Effective 10/01/2003. Patent fees are subject to annual revision.</p>	Complete if Known	
	Express Mail No.	EV335515821US
	Application Number	09/629,570
	Filing Date	July 31, 2000
	First Named Inventor	Fred B. Holt
	Examiner Name	Bradley E. Edelman
<input type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27	Art Unit	2153
TOTAL AMOUNT OF PAYMENT	(\$) 110	
	Attorney Docket No.	030048002US

METHOD OF PAYMENT (check all that apply)

Check Credit card Money Other None Order

Deposit Account:
 Deposit Account Number 50-0665
 Deposit Account Name Perkins Coie LLP

The Commissioner is authorized to: (check all that apply)

Charge fee(s) indicated below Credit any overpayments

Charge any additional fee(s) during the pendency of this application

Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1205	160	2005	80	Provisional filing fee	
SUBTOTAL (1)					(\$) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

Total Claims	23	-49** =	0	X	=	=
Independent Claims	3	- 7** =	0	X	=	=
Multiple Dependent					=	=

Fee from below = Fee Paid

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1202	18	2202	9	Claims in excess of 20	
1201	86	2201	43	Independent claims in excess of 3	
1203	290	2203	145	Multiple dependent claim, if not paid	
1204	86	2204	43	** Reissue independent claims over original patent	
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent	
SUBTOTAL (2)					(\$) 0

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English Specification	
1812	2,520	1812	2,520	For filing a request for ex parte reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	110
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid **SUBTOTAL (3)** **(\$)** 110

SUBMITTED BY		(Complete if applicable)	
Name (Print/Type)	Chun Ng	Registration No. (Attorney/Agent)	36,878
Signature		Telephone	206-359-6488
		Date	05/10/2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

REMARKS

Reconsideration and withdrawal of the rejections set forth in the Office Action dated January 12, 2004 are respectfully requested.

I. Rejections under 35 U.S.C. § 112, first paragraph

Claims 1, 14, and 32 have been amended to include sufficient antecedent basis. In claim 1, the phrase "the added participant", which appears in the last line of the claim, has been changed to "the seeking participant". In addition, "a seeking participant" precedes "the seeking participant" in an earlier line of claim 1, providing sufficient antecedent basis. In claim 32, the phrase "the added participant", which appears in the last line of the claim, has been changed to "a seeking participant". In claim 14, the phrase "the added node", which appears in the last line of the claim, has been changed to "the seeking node". In addition, "a seeking node" precedes "the seeking node" in an earlier line of claim 14, providing sufficient antecedent basis.

II. Rejections under 35 U.S.C. § 112, second paragraph

Claim 6 has been amended to render the claim definite. The term "approximately proportional" has been changed to "proportional". Claim 10 has also been amended to render the claim definite. The term "approximately twice the diameter" has been changed to "twice the diameter". Claim 37 has been amended to render the claim definite. The term "approximately twice a diameter of the network" has been changed to "twice a diameter of the network".

III. Rejections under 35 U.S.C. § 102

A. The Applied Art

U.S. Patent No. 6,603,742 B1 to Steele, Jr. et al. (*Steele, Jr. et al.*) is directed to a technique for reconfiguring networks while it remains operational. *Steele, Jr. et al.* discloses a method for adding nodes to a network with minimal recabling. Column 3, lines 2-5. An interim routing table is used to route traffic around the part of the network affected by the adding of a

node. Column 11, lines 40-45. Each node in the network can connect to five other nodes. Column 4, lines 36-39, Column 4, lines 43-44. To add a node to a network, two links between two pairs of existing nodes are removed and five links are added to connect the new node to the network. Column 11, lines 25-31. For example, when upgrading from 7 to 8 nodes, the network administrator removes two links, 3-1 and 5-2, and adds five links, 7-1, 7-2, 7-3, 7-5, and 7-6. Column 12, lines 45-48.

B. Analysis

Distinctions between claim 1 and *Steele, Jr. et al.* will first be discussed, followed by distinctions between *Steele, Jr. et al.* and the remaining dependent claims.

As noted above, *Steele, Jr. et al.* discloses a technique for reconfiguring networks. Such a technique includes steps for disconnecting the participants of a pair from each other and connecting each participant to a seeking participant but does not include a step for identifying a pair of participants of the network that are fully connected. Column 12, lines 45-49. *Steele, Jr. et al.* fails to disclose a method for identifying a pair of participants of the network that are fully connected.

In contrast, claim 1 as amended includes the limitation of identifying a pair of participants of the network that are connected. For at least this reason, the applicant believes that claim 1 is patentable over *Steele, Jr. et al.*

The invention discloses an identification method in which a seeking participant contacts a fully connected portal computer. The portal computer directs the identification of a number of (for example four), randomly selected neighboring participants to which the seeking participant is to connect. *Steele, Jr. et al.* fails to disclose a portal computer that directs the identification of viable neighboring participants to which the seeking participant is to connect. Claim 1 has been amended to recite, among other limitations, the use of a portal computer for the identifying of "a

number of selected neighboring participants to which the seeking participant is to connect." *Steele, Jr. et al.* fails to disclose such a method for identifying neighboring participants for a seeking participant to connect to. For at least this reason, claim 1 is patentable over *Steele, Jr. et al.*

Further, the claimed does not make use of routing tables. *Steele, Jr. et al.* fails to disclose a non-table based routing method. Claim 1 has been amended to recite, among other limitations, "a computer-based, non-routing table based, non-switch based method for adding a participant to a network of participants". For at least this reason, claim 1 is patentable over *Steele, Jr. et al.*

Claim 2 discloses a connection scheme where "each participant is connected to 4 participants". *Steele, Jr. et al.* fails to disclose a connection scheme in which each participant is connected to 4 participants. Instead, *Steele, Jr. et al.* discloses a connection scheme in which each participant is connected to 5 other participants. Column 7, lines 14-33. For at least this reason, claim 2 is patentable over *Steele, Jr. et al.*

Anticipation a claim under 35 U.S.C. § 102 requires that the cited reference must teach every element of the claim.¹ *Steele, Jr. et al.* fails to disclose every limitation recited in claim 1. Since claim 1 is allowable, based on at least the above reasons, the claims that depend on claim 1 are likewise allowable.

¹ MPEP section 2131, p. 70 (Feb. 2003, Rev. 1). See also, *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1462 (Bd. Pat. App. & Interf. 1990) (to establish a *prima facie* case of anticipation, the Examiner must identify where "each and every facet of the claimed invention is disclosed in the applied reference."); *Glaverbel Société Anonyme v. Northlake Mktg. & Supply, Inc.*, 45 F.3d 1550, 1554 (Fed. Cir. 1995) (anticipation requires that each claim element must be identical to a corresponding element in the applied reference); *Atlas Powder Co. v. E.I. duPont De Nemours*, 750 F.2d 1569, 1574 (1984) (the failure to mention "a claimed element (in) a prior art reference is enough to negate anticipation by that reference").

IV. Rejections under 35 U.S.C. § 103, first paragraph

A. The Applied Art

A Flood Routing Method for Data Networks by Cho (*Cho*) is directed to a routing algorithm based on a flooding technique. *Cho* discloses a method in which flooding is used to find an optimal route to forward messages through. Flooding refers to a data broadcast technique that sends the duplicate of a packet to all neighboring nodes in a network. In *Cho*, flooding is not used to send the message, but is used to locate the optimal route for the message to be sent through. The method entails flooding a very short packet to explore an optimal route for the transmission of the message and to establish the data path via the selected route. Each node connected to the broadcast channel does not receive all messages that are broadcast on the broadcast channel. When a node receives a message, it does **not** forward that message to all of its neighboring nodes using flooding. In addition, *Cho* fails to disclose a method for rearranging a sequence of messages that are received out of order.

B. Analysis

As noted above, *Steele, Jr. et al.* discloses a method for adding nodes to a network with minimal recabling. *Steele, Jr. et al.* fails to disclose a method in which "each participant forwards broadcast messages that it receives to all of its neighbor participants". Claim 32 has been amended to clarify the language of previously pending claim 32. *Cho* discloses a method in which flooding is used to find an optimal route to forward messages through. *Cho* fails to disclose the use of flooding to forward messages. In *Cho*, flooding is used only to find an optimal route for data transmission and is not used to actually forward messages. *Cho* fails to disclose a system in which "each participant forwards broadcast messages that it receives to all of its neighbor participants". In *Cho*, each participant forwards messages only to a destination node once the optimal route has been selected. *Cho* fails to disclose a system in which "each

participant connected to the broadcast channel receives all messages that are broadcast on the network". In addition, Cho fails to disclose a method for addressing a sequence of messages that are received out of order in which "messages are numbered sequentially so that messages received out of order are queued and rearranged to be in order".

As explained below, there is no incentive or teaching to combine *Steele, Jr. et al.* and *Cho*. However, even if they were combined, neither *Steele, Jr. et al.* nor *Cho* teach or suggest the use of flooding to send messages to all nodes connected to a broadcast channel. In addition, neither *Steele, Jr. et al.* nor *Cho* teach or suggest the sequential numbering of messages to rearrange a sequence of messages that are received out of order. The invention of claim 32 includes forwarding messages to all neighboring nodes and numbering each message sequentially so that "messages received out of order are queued and rearranged to be in order", which are not disclosed in either *Steele, Jr. et al.* or *Cho*. For at least this reason, the applicant believes that claim 32 is patentable over the combination of *Steele, Jr. et al.* and *Cho*.

The independent claims are allowable not only because they recite limitations not found in the references (even if combined), but for at least the following additional reasons. For example, there is no motivation to combine the various references as suggested in the Office Action. According to the Manual of Patent Examining Procedure ("MPEP") and controlling case law, the motivation to combine references cannot be based on mere common knowledge and common sense as to benefits that would result from such a combination, but instead must be based on specific teachings in the prior art, such as a specific suggestion in a prior art reference. For example, last year the Federal Circuit rejected an argument by the PTO's Board of Patent Appeals and Interferences that the ability to combine the teachings of two prior art references to produce beneficial results was sufficient motivation to combine them, and thus overturned the

Board's finding of obviousness because of the failure to provide a specific motivation in the prior art to combine the two references.² The MPEP provides similar instructions.³

Conversely, and in a manner similar to that rejected by the Federal Circuit, the present Office Action lacks any description of a motivation to combine the references. Thus, if the current rejection is maintained, the applicant's representative requests that the Examiner explain with the required specificity where a suggestion or motivation in the references for so combining the references may be found.⁴

Steele et al. deals with a method for adding nodes to a network while *Cho* deals with finding an optimal route to forward messages in a network. The addition of nodes to a network represents a completely separate process from the forwarding of messages in a network. *Steele et al.* contains no specific teachings that would suggest combining *Steele et al.* with *Cho*. In other words, *Steele et al.* contains no specific teachings that would suggest finding an optimal route to forward messages in a network.

One may not use the application as a blueprint to pick and choose teachings from various prior art references to construct the claimed invention ("impermissible hindsight reconstruction").⁵ Assuming, for argument's sake, that it would be obvious to combine the teachings of *Steele et al.* with *Cho*, then *Steele et al.* would have done so because it would have

² In re Sang-Su Lee, 277 F.3d 1338, 1341-1343 (Fed. Cir. 2002).

³ Manual of Patent Examining Procedure, Section 2143 (noting that "the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure," citing in re Vaeck, 947 F.2d 488 (Fed. Cir. 1991)).

⁴ See, MPEP Section 2144.03.

⁵ See, e.g., In re Gorman, 933 F.2d 982,987 (Fed. Cir. 1991), ("One cannot use hindsight construction to pick and choose between isolated disclosures in the prior art to deprecate the claimed invention.").

provided at least some of the advantages of the presently claimed invention. *Steele et al.*'s failure to employ the teachings cited in *Cho* is persuasive proof that the combination recited in claim 32 is unobvious. For at least this reason, the applicant believes that claim 32 is patentable over the combination of *Steele et al.* and *Cho*.

Claim 33 discloses a connection scheme where "each participant is connected to 4 participants". *Steele, Jr. et al.* fails to disclose a connection scheme in which each participant is connected to 4 participants. Instead, *Steele, Jr. et al.* discloses a connection scheme in which each participant is connected to 5 other participants. Column 7, lines 14-33. For at least this reason, claim 33 is patentable over *Steele, Jr. et al.*

Since claim 32 is allowable, based on at least the above reasons, the claims that depend on claim 32 are likewise allowable. Thus, for at least this reason, claim 33 is patentable over the combination of *Steele, Jr. et al.* and *Cho*.

V. Rejections under 35 U.S.C. § 103, second paragraph

A. The Applied Art

U.S. Patent No. 6,490,247 B1 to Gilbert et al. (*Gilbert et al.*) is directed to a ring-ordered, dynamically reconfigurable computer network utilizing an existing communications system. *Gilbert et al.* discloses a method for adding a node to a network using a switching mechanism in which the nodes are ordered in a ring-like configuration as opposed to a hypercube configuration. Column 3, lines 28-35. The first step in adding a seeking node to the network consists of the seeking contacting a portal node that is fully connected to the network. Column 6, lines 31-33. The portal node that is contacted provides information regarding a neighboring node that is adjacent to the seeking node; the selection of the neighboring node is not random. Column 6, lines 40-42. The seeking node then contacts the neighboring node to request a connection. Column 6, lines 57-59. The portal node provides the relevant information regarding

the node that is adjacent to the neighboring node that is adjacent to the seeking node but does not request a connection.

U.S. Patent No. 6,553,020 B1 to Hughes et al. (*Hughes et al.*) is directed to a network for interconnecting nodes for communication across the network. *Hughes et al.* fails to disclose a system where a portal computer randomly selects four nodes to serve as neighboring nodes to the seeking node. *Hughes et al.* also fails to disclose a system in which the portal computer sends an edge connection request to the neighboring nodes.

B. Analysis

As noted above, *Gilbert et al.* discloses a method for adding a node to a network using a switching mechanism. *Gilbert et al.* fails to disclose a method in which a portal computer seeks "a number of randomly selected neighboring participants to which the seeking participant is to connect". In *Gilbert et al.*, the selection of the neighboring nodes is not random. Column 6, lines 40-49. Figure 6 of *Gilbert et al.* reveals that node 100 selects nodes 10 and 16; the selection of nodes 10 and 16 is not random since they are purposely adjacent to one another and since node 10 provides node 100 with information regarding the node adjacent to it, node 16. Column 6, lines 42-46. *Gilbert et al.* fails to disclose a method in which a portal computer "sends an edge connection request to a number of randomly selected neighboring participants to which the seeking participant is to connect". In *Gilbert et al.*, the seeking node, not the portal node, contacts the neighboring participants to which the seeking participant is to connect. Column 6, lines 57-61. *Gilbert et al.* fails to disclose a "non-switch based method for adding a participant to a network of participants". Column 3, lines 8-11. *Gilbert et al.* fails to disclose a method in which an additional node contacts "a number of randomly selected neighboring participants". Column 6, lines 30-32. *Hughes et al.* discloses a method in which an additional node contacts four neighboring participants. *Hughes et al.* fails to disclose a method in which a

portal computer seeks "four randomly selected neighboring participants to which the seeking participant is to connect". *Hughes et al.* also fails to disclose a method in which a portal computer "sends an edge connection request to four randomly selected neighboring participants to which the seeking participant is to connect".

As explained below, *Gilbert et al* and *Hughes et al.* would not be combined. However, even if they were combined, neither *Gilbert et al* nor *Hughes et al.* teach or suggest the random selection of neighboring participants. Claim 1 has been amended to recite, among other limitations, a method in which a portal computer seeks "four randomly selected neighboring participants to which the seeking participant is to connect". In other words, the invention of claim 1 includes randomly selecting neighboring participants to which the seeking participant is to connect, which is not disclosed in either *Gilbert et al* or *Hughes et al.* Even if they were combined, neither *Gilbert et al* nor *Hughes et al.* teach or suggest the sending of an edge connection request by the portal computer to the randomly selected neighboring participants to which the seeking participant is to connect. Claim 1 has been amended to recite, among other limitations, a method in which a portal computer "sends an edge connection request to four randomly selected neighboring participants to which the seeking participant is to connect". In other words, the invention of claim 1 includes the portal computer sending an edge connection request to the randomly selected neighboring participants to which the seeking participant is to connect, which is not disclosed in either *Gilbert et al* or *Hughes et al.* For at least these reasons, the applicant believes that claim 1 is patentable over the combination of *Gilbert et al* and *Hughes et al.*

In a similar fashion, claim 14 has been amended to recite, among other limitations, a method in which a portal computer seeks "four randomly selected neighboring nodes to which the seeking node is to connect". In other words, the invention of claim 14 includes randomly

selecting neighboring nodes to which the seeking node is to connect, which is not disclosed in either *Gilbert et al* or *Hughes et al*. Even if they were combined, neither *Gilbert et al* nor *Hughes et al* teach or suggest the random selection of neighboring nodes. In addition, even if they were combined, neither *Gilbert et al* nor *Hughes et al* teach or suggest the sending of an edge connection request by the portal computer to the randomly selected neighboring nodes to which the seeking node is to connect. Claim 14 has been amended to recite, among other limitations, a method in which a portal computer "sends an edge connection request to four randomly selected neighboring nodes to which the seeking node is to connect". In other words, the invention of claim 14 includes the portal computer sending an edge connection request to the randomly selected neighboring nodes to which the seeking node is to connect, which is not disclosed in either *Gilbert et al* or *Hughes et al*. For at least these reasons, the applicant believes that claim 14 is patentable over the combination of *Gilbert et al* and *Hughes et al*.

Since claim 1 is allowable, based on at least the above reasons, the claims that depend on claim 1 are likewise allowable. Thus, for at least this reason, claims 2-5, 7, 8, and 11-13 are patentable over the combination of *Gilbert et al* and *Hughes et al*. Since claim 14 is allowable, based on at least the above reasons, the claims that depend on claim 14 are likewise allowable. Thus, for at least this reason, claims 15-17 are patentable over the combination of *Gilbert et al* and *Hughes et al*.

If the current rejection is maintained, the applicant's representative requests that the Examiner explain with the required specificity where a suggestion or motivation in the references for so combining the references may be found.⁶

⁶ See, MPEP Section 2144.03.

Gilbert et al. deals with a method for adding nodes to a network while *Hughes et al.* deals with a network for interconnecting nodes for communication across the network. The addition of nodes to a network represents a completely separate process from the interconnection of nodes in a network. *Hughes et al.* contains no specific teachings that would suggest combining *Hughes et al.* with *Gilbert et al.* In other words, *Hughes et al.* contains no specific teachings that would suggest adding a node to a network.

As is known, one may not use the application as a blueprint to pick and choose teachings from various prior art references to construct the claimed invention ("impermissible hindsight reconstruction").⁷ Assuming, for argument's sake, that it would be obvious to combine the teachings of *Hughes et al.* with *Gilbert et al.*, then *Hughes et al.* would have done so because it would have provided at least some of the advantages of the presently claimed invention. *Hughes et al.*'s failure to employ the teachings cited in *Gilbert et al.* is persuasive proof that the combination is unobvious. For at least this reason, the applicant believes that claims 1 and 14 are patentable over the combination of *Hughes et al.* and *Gilbert et al.*

Since claim 1 is allowable, based on at least the above reasons, the claims that depend on claim 1 are likewise allowable. Thus, for at least this reason, claims 2-5, 7, 8, and 11-13 are patentable over the combination of *Gilbert et al.* and *Hughes et al.* Since claim 14 is allowable, based on at least the above reasons, the claims that depend on claim 14 are likewise allowable. Thus, for at least this reason, claims 15-17 are patentable over the combination of *Gilbert et al.* and *Hughes et al.*

⁷ See, e.g., *In re Gorman*, 933 F.2d 982,987 (Fed. Cir. 1991), ("One cannot use hindsight construction to pick and choose between isolated disclosures in the prior art to deprecate the claimed invention.").

VI. Rejections under 35 U.S.C. § 103, third paragraph

A. The Applied Art

A Flood Routing Method for Data Networks by Cho (*Cho*), U.S. Patent No. 6,490,247 B1 to Gilbert et al. (*Gilbert et al.*), and U.S. Patent No. 6,553,020 B1 to Hughes et al. (*Hughes et al.*) have already been disclosed in the above descriptions of the applied art.

B. Analysis

As noted previously, *Gilbert et al.* discloses a method for adding nodes to a network while *Hughes et al.* discloses a network for interconnecting nodes for communication across the network. The combination of *Gilbert et al.* and *Hughes et al.* fails to disclose a method in which "each participant forwards broadcast messages that it receives to all of its neighbor participants". *Cho* discloses a method in which flooding is used to find an optimal route to forward messages through. *Cho* fails to disclose the use of flooding to forward messages. In *Cho*, flooding is used only to find an optimal route for data transmission and is not used to actually forward messages. *Cho* fails to disclose a system in which "each participant forwards broadcast messages that it receives to all of its neighbor participants". In *Cho*, each participant forwards messages only to a destination node once the optimal route has been selected. *Cho* fails to disclose a system in which "each participant connected to the broadcast channel receives all messages that are broadcast on the network". In addition, *Cho* fails to disclose a method for addressing a sequence of messages that are received out of order in which "messages are numbered sequentially so that messages received out of order are queued and rearranged to be in order". Claim 32 has been amended to clarify the inherent language of previously pending claim 32. As explained below, *Gilbert et al.*, *Hughes et al.*, and *Cho* would not be combined. However, even if they were combined, *Gilbert et al.*, *Hughes et al.*, and *Cho* fail to teach or suggest the use of flooding to send messages to all nodes connected to a broadcast channel. In addition, *Gilbert et al.*, *Hughes*

et al., and *Cho* fail to teach or suggest the sequential numbering of messages to rearrange a sequence of messages that are received out of order. The invention of claim 32 includes forwarding messages to all neighboring nodes and numbering each message sequentially so that "messages received out of order are queued and rearranged to be in order", which are not disclosed in *Gilbert et al.*, *Hughes et al.*, or *Cho*. For at least these reasons, the applicant believes that claim 32 is patentable over the combination of *Gilbert et al.*, *Hughes et al.*, and *Cho*.

Since claim 32 is allowable, based on at least the above reasons, the claims that depend on claim 32 are likewise allowable. Thus, for at least this reason, claims 33-36, 38, and 39 are patentable over the combination of *Gilbert et al.*, *Hughes et al.*, and *Cho*.

Gilbert et al. deals with a method for adding nodes to a network, *Hughes et al.* deals with a network for interconnecting nodes for communication, and *Cho* deals with finding an optimal route to forward messages in a network. These three prior art references represent separate, distinct processes. The combination of *Gilbert et al.* and *Hughes et al.* contains no specific teachings that would suggest combining *Gilbert et al.* and *Hughes et al.* with *Cho*. In other words, the combination of *Gilbert et al.* and *Hughes et al.* contains no specific teachings that would suggest finding an optimal route to forward messages in a network.

Assuming, for argument's sake, that it would be obvious to combine the teachings of *Gilbert et al.* and *Hughes et al.* with *Cho*, then *Gilbert et al.* and *Hughes et al.* would have done so because it would have provided at least some of the advantages of the presently claimed invention. The failure of *Gilbert et al.* and *Hughes et al.* to employ the teachings cited in *Cho* is persuasive proof that the combination recited in claim 32 is unobvious. For at least this reason, the applicant believes that claim 32 is patentable over the combination of *Gilbert et al.* and *Hughes et al.* in view of *Cho*.

Since claim 32 is allowable, based on at least the above reasons, the claims that depend on claim 32 are likewise allowable. Thus, for at least this reason, claims 33-36, 38, and 39 are patentable over the combination of *Gilbert et al*, *Hughes et al.*, and *Cho*.


VII. Conclusion

In view of the foregoing, the claims pending in the application comply with the requirements of 35 U.S.C. § 112 and patentably define over the applied art. A Notice of Allowance is, therefore, respectfully requested. If the Examiner has any questions or believes a telephone conference would expedite prosecution of this application, the Examiner is encouraged to call the undersigned at (206) 359-6488.

Date: 5/10/04

Respectfully submitted,


Perkins Coie LLP


Chun M. Ng
Registration No. 36,878

Correspondence Address:

Customer No. 25096
Perkins Coie LLP
P.O. Box 1247
Seattle, Washington 98111-1247
(206) 359-6488

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. 1.136(a)		Docket Number (Optional) 030048002US	
	In re Application of Fred B. Holt		Filed 07/31/2000
	Application Number 09/629,570		
	For JOINING A BROADCAST CHANNEL		
	Group Art Unit 2153	Examiner Bradley E. Edelman	

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and appropriate non-small-entity fee are as follows (check time period desired):

- One month (37 CFR 1.17(a)(1))
- Two months (37 CFR 1.17(a)(2))
- Three months (37 CFR 1.17(a)(3))
- Four months (37 CFR 1.17(a)(4))
- Five months (37 CFR 1.17(a)(5))

RECEIVED
 MAY 17 2004
 Technology Center 2100

\$ 110
\$ 420
\$ 950
\$ 1,480
\$ 2,010

Applicant claims small entity status. See 37 CFR 1.27. Therefore, the fee amount shown above is reduced by one-half, and the resulting fee is: \$ _____.

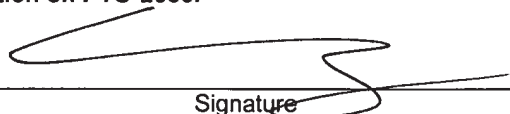
- A check in the amount of the fee is enclosed.
- Payment by credit card. Form PTO-2038 is attached.
- The Director has already been authorized to charge fees in this application to a Deposit Account.
- The Director is hereby authorized to charge any additional fees which may be required, or credit any overpayment, to Deposit Account No. 50-0665.
I have enclosed a duplicate copy of this sheet.

- I am the
- applicant/inventor
 - assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).
 - attorney or agent of record. Registration number _____.
 - attorney or agent under 37 CFR 1.34(a).
Registration number if acting under 37 CFR 1.34(a): 36,878.

05/13/2004 RECEIPT 00000141 09529570 110.00 CP 01 FC:1251

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

05/10/2004
Date


Signature

206-359-6488
Telephone Number

Chun Ng
Typed or printed name

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

Total of 1 forms is submitted.

PATENT APPLICATION FEE DETERMINATION RECORD

Effective December 29, 1999

Application or Docket Number

09/629570

CLAIMS AS FILED - PART I

SMALL ENTITY TYPE OR OTHER THAN SMALL ENTITY

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA
BASIC FEE		
TOTAL CLAIMS	48 minus 20 =	28
INDEPENDENT CLAIMS	7 minus 3 =	4
MULTIPLE DEPENDENT CLAIM PRESENT		

RATE	FEE	OR	RATE	FEE
	345.00			690.00
X\$ 9=			X\$18=	504.00
X39=			X78=	312.00
+130=			+260=	
TOTAL			TOTAL	1506.00

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT A	(Column 1)	(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	26	Minus ** 48	=
Independent	3	Minus *** 7	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X\$ 9=			X\$18=	
X39=			X78=	
+130=			+260=	
TOTAL ADDIT. FEE			TOTAL ADDIT. FEE	

AMENDMENT B	(Column 1)	(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	27	Minus ** 48	=
Independent	3	Minus *** 2	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X\$ 9=			X\$18=	
X39=			X78=	
+130=			+260=	
TOTAL ADDIT. FEE			TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1)	(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total		Minus **	=
Independent		Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X\$ 9=			X\$18=	
X39=			X78=	
+130=			+260=	
TOTAL ADDIT. FEE			TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Performance Analysis of Network Connective Probability of Multihop Network under Correlated Breakage

Shigeki Shiokawa and Iwao Sasase

Department of Electrical Engineering, Keio University
3-14-1 Hiyoshi, Kohoku, Yokohama, 223 JAPAN

Abstract—One of important properties of multihop network is the network connective probability which evaluate the connectivity of the network. The network connective probability is defined as the probability that when some nodes are broken, rest nodes connect each other. Multihop networks are classified to the regular network whose link assignment is regular and the random network whose link assignment is random. It has been shown that the network connective probability of regular network is larger than that of random network. However, all of these results is shown under independent node breakage. In this paper, we analyze the network connective probability of multihop networks under the correlated node breakage. It is shown that regular network has better performance of the network connective probability than random network under the independent breakage, on the other hand, random network has better performance than regular network under the correlated breakage.

1 Introduction

In recent years, multi-hop networks have been widely studied [1]-[8]. These networks must pass messages between source and destination nodes via intermediate links and nodes. Examples of them include ring, shuffle network (SN) [1],[2] and chordal network (CN)[3]. One of the very important performance measure of multi-hop network is the connectivity of the network. If some nodes are broken, it is needed for a network to guarantee the connection among non-broken nodes. Thus, the network connective probability defined as the probability that when some nodes are broken, rest links and nodes construct the connective network, should be a very important property to evaluate the connectivity of the network.

Multi-hop networks are classified to regular network and random network according to the way of link assignment. In the regular network, links are assigned regularly and examples of them include shufflenet and manhattan street network. On the other hand, in random network, link assignment is not regular but somewhat random and examples of them include connective semi-random network (CSRN) [6]. The network connective probabilities of some multi-hop networks have been analyzed and it has been shown that the network connective probability of regular network is larger than that of random network. However, all of them is analyzed under the condition that locations of broken nodes are independent each other. In the real network, there are some case that the locations of broken nodes have correlation, for example, links and nodes are broken in the same area under the case of disaster. Thus, it is significant and great of interest to analyze the network connective probability under the condition when the locations of broken nodes have correlations each other.

In this paper, we analyze the network connective probability of multi-hop network under the condition that locations of broken nodes have correlations each other, where we treat SN, CN and CSRN as the model for analysis. We realize the correlation as follows. At first, we note one node and break it and call this node the center broken node. And next, we note nodes whose links connect to the center broken nodes and break them at some probability. We define this probability as the correlated broken probability. Very interesting result is shown that under independent breakage of node, regular network has better performance of the network connective probability than random network, on the other hand, under the correlated breakage of node, random network has better performance than regular network.

In the section 2, we explain network model of SN, CN and CSRN which we analyze in the section 3. In the section 3, we analyze the network connective probability under the condition when the location of broken nodes have correlation each other. And we compare each of network connective probability in the section 4. In the last, we conclude our study.

2 Multihop network model

In this section, we explain the multihop network models used for analysis of the network connective probability. We treat three networks such as SN, CN and CSRN which consists of N nodes and p unidirected outgoing links per node.

Fig. 1 shows SN with 18 nodes and 2 outgoing links per node. To construct the SN, we arrange $N = kp^k$ ($k = 1, 2, \dots; p = 1, 2, \dots$) nodes in k columns of p^k nodes each. Moving from left to right, successive columns are connected by p^{k+1} outgoing links, arranged in a fixed shuffle pattern, with the last column connected to the first as if the entire graph were wrapped around a cylinder. Each of the p^k nodes in a column has p outgoing links directed to p different nodes in the next column. Numbering the nodes in a column from 0 to $p^k - 1$, nodes i has outgoing links directed to nodes $j, j + 1, \dots, j + p - 1$ in the next column, where $j = (i \bmod p^{k-1})p$. In Fig. 1, p is equal to 2 and k is equal to 2. Since the link assignment of SN is regular, SN is regular network.

Fig. 2 shows CN with 16 nodes and 2 outgoing links per node. To construct CN, at first, we construct unidirected ring network with N nodes and N unidirected links. And $p-1$ unidirected links are added from each node. Numbering nodes along ring network from 0 to $N - 1$, node i has outgoing links directed to nodes $(i + 1) \bmod N, (i + \tau_1) \bmod N, \dots, \text{and } (i + \tau_{p-1}) \bmod N$, where τ_j ($j = 1, 2, \dots, p - 1$) is defined as the chordal length. In Fig. 2, τ_1 is equal to 3. Since τ_i for every i are independent each other, CN is not regular network. However, CN has much regular elements such a symmetrical pattern of network.

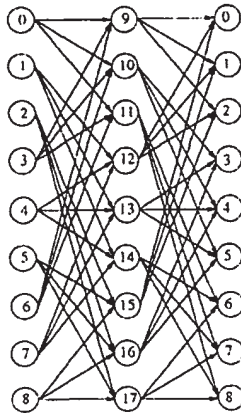


Figure 1. Shuffle network with $N = 18$ and $p = 2$.

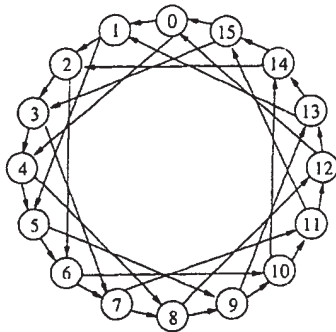


Figure 2. Chordal network with $N = 16$, $p = 2$ and $\tau_1 = 3$.

Fig. 3 shows CSRN with 16 nodes and 2 outgoing links from a node. Similarly with CN, CSRN includes unidirected ring network with N nodes and N unidirected links. And we add $p - 1$ links from each node whose directed nodes are randomly selected. In CSRN, the number of incoming links per node is not constant, for example, in Fig. 3, the number of incoming links into node 1 is 1 and the one into node 3 is 3. The link assignment of CSRN is random except for the part of ring network, thus CSRN is random network. It has been shown that since the number of incoming links per node is not constant, the network connective probability of CSRN is smaller than those of SN and CN when locations of broken nodes are independent each other. And that of SN is the same as that of CN, because the network connective probability depends on the number of incoming links come into every nodes.

3 Performance Analysis

Here, we analyze the network connective probability of SN, CN and CSRN under the condition that locations of broken nodes have correlation each other. Now, we explain the network connective probability in detail using Fig. 3. This figure shows the connective network which is defined as the network in which all nodes connect to every other nodes directly or indirectly. At first, we consider the case that the node 1 is broken. The node 1 has two outgoing links directed to nodes 2 and 3, and if the node 1 is broken, we can not use them. However, node 2 has two incoming links from nodes 1 and 14, and node 3 has three incoming links from nodes 1, 2 and 11. Therefore, even if node 1 is broken, rest nodes can construct

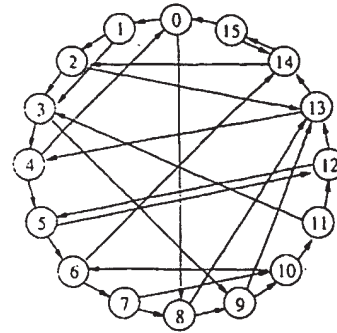


Figure 3. Connective semi-random network with $N = 16$ and $p = 2$.

the connective network. Next, we consider the case that node 0 is broken. The node 0 has two outgoing links directed to nodes 1 and 8, and if the node 0 is broken, we can not use them. Since node 1 has only one incoming link from node 0, even if only node 0 is broken, rest nodes can not connect to node 1, that is, they can not construct the connective network. Here, we define the network connective probability as the probability that when some nodes and links are broken, the rest nodes and links can construct the connective network.

Now, we explain the correlated node breakage using Fig. 3. At first, we note one node and break it, where this node is called as the center broken node. And then, we note nodes whose outgoing links come into the center broken node or whose incoming links go out of the center broken node, and break them at a probability defined as the correlated broken probability. In Fig 3, when we assume that the center broken node is the node 3, there are five nodes 1, 2, 4, 9 and 11 which have possibility to become correlated broken node. And they become the broken nodes at the correlated broken probability. It is obvious that none of them is broken when the correlated broken probability is 0 and all of them is broken when the correlated broken probability is 1.

In our study, we analyze the network connective probability that only nodes are broken. And we assume that the number of center broken node is one in the analysis. We denote the correlated broken probability by a and the network connective probability of SN, CN and CSRN by P_{SN} , P_{CN} and P_{CSRN} , respectively.

3.1 Shuffle Network

Because the number of incoming links per node in SN is the constant p , when broken node is only center broken node, the rest nodes can construct the connective network. There are $2p$ nodes have the possibility to become the correlated broken node. All of p nodes which have outgoing link come into the center broken node have the outgoing links directed to the same nodes. For example, in Fig. 1, if we assume that the node 9 is the center broken node, the nodes 0, 3 and 6 has outgoing links to node 9. And each of three nodes have two outgoing links directed to nodes 10 and 11. Therefore, only when all of them are broken, the rest nodes can not construct the connective network. On the other hand, all of outgoing links go out from p nodes which have incoming link from center broken node direct to different nodes. In Fig. 1, nodes 0, 1 and 2 have the incoming link from center broken node 9. And all of the outgoing links from their nodes direct to different nodes, thus even if all of them are broken, the rest nodes can construct the connective network. Thus, the network connective probability of SN is the probability that all of nodes whose outgoing links come

into the center broken node are broken, and it is derived as

$$P_{SN} = 1 - a^p. \quad (1)$$

3.2 Chordal Network

The network connective probability of CN with $p = 2$ is different from that with $p \geq 3$. At first, we consider the case with $p = 2$. When p is equal to 2, all of the outgoing links, from the nodes whose incoming links go out from the center broken node, direct to the same node. For example, in Fig. 2, when we assume that the center broken node is node 0, the outgoing links from it direct to nodes 1 and 4. And each of outgoing links from them directs to node 5. Therefore, only when all nodes whose incoming links go out from the center broken node are broken, the rest nodes can not construct the connective network. And we can obtain the network connective probability as

$$P_{CN} = 1 - a^2 \quad \text{for } p = 2. \quad (2)$$

And next, we consider the case that $p \geq 3$. In CN, when p is equal to or larger than three and each chordal length is selected properly, all of outgoing links from the nodes whose incoming links go out from the center broken node do not direct to the same nodes. And therefore, even if all of nodes which connect to the center broken nodes with incoming or outgoing links is broken, the rest nodes can construct the connective network, that is,

$$P_{CN} = 1 \quad \text{for } p \geq 3. \quad (3)$$

3.3 Connective Semi-Random Network

In CSRN, the number of the incoming links per node is not constant. Since the maximum number of incoming links is $N - 1$ and one link come into a node at least, the probability that the number of the incoming links come into a node is i , denoted as A_i , is

$$A_i = \begin{cases} 0, & \text{for } i = 0 \\ \binom{N-2}{i-1} \left(\frac{p}{N-2}\right)^{i-1} \left(1 - \frac{p}{N-2}\right)^{N-1-i} & \text{for } i \geq 1. \end{cases} \quad (4)$$

The nodes which have possibility to become the correlated broken nodes are those which connect to the center broken node by outgoing link or incoming link. When the number of the incoming link come into the center broken node is i , the sum of outgoing links and incoming links it have is $p + i$. However, the number of the nodes which have possibility to become the correlated broken nodes is not always $p + i$, because the p outgoing links have the possibility to overlap with one of i incoming links. For example, in Fig. 3, when the center broken nodes is node 5, the outgoing link to node 12 overlap with the incoming link from node 12. Therefore, in spite of the node 5 has four outgoing and incoming links, the number of the nodes which have possibility to become the correlated broken nodes when the node 5 is the center broken node is three.

And now, we derive the probability that the number of nodes which have possibility to become the correlated broken nodes is j , denoted as B_j . Before derive B_j , we derive the probability that q of p outgoing links which go out of a node overlap with r incoming links come into it, denoted as $C_{p,q,r}$. Here, we define regular link as the link which construct the ring network and random link as other link. We consider the two case. The one is the case that one of the incoming links overlap with the regular outgoing link, and the other case is that none of incoming links overlap with it. Since

the regular incoming link never overlap with the regular outgoing link, the probability to become the first case is $(r - 1)/(N - 2)$ and one to become the second case is $1 - (r - 1)/(N - 2)$. In the first case, $C_{p,q,r}$ is the same as the probability that each of $q - 1$ outgoing links among the $p - 1$ outgoing links except for the regular outgoing link overlap one of $r - 1$ incoming links, denoted as $C'_{p-1,q-1,r-1}$. And in the second case, $C_{p,q,r}$ is the same as the probability that each of q outgoing links among the $p - 1$ outgoing links except for the regular outgoing link overlap one of r incoming links, denoted as $C'_{p-1,q,r}$. Using $C'_{p',q',r'}$ given as follows,

$$C'_{p',q',r'} = \begin{cases} 0, & \text{for } q' < 0, r' \leq 0, q' > p', \\ & (p' + r' > N \text{ and } q' < p' + r' - N) \\ \frac{\binom{p'}{q'} r' P_{q'}^{N-2-r'} P_{p'-q'}}{N-2 P_{p'}}, & \text{otherwise,} \end{cases} \quad (5)$$

we can derive $C_{p,q,r}$ as

$$C_{p,q,r} = \left(\frac{r-1}{N-2}\right) C'_{p-1,q-1,r-1} + \left(1 - \frac{r-1}{N-2}\right) C'_{p-1,q,r}. \quad (6)$$

B_j can be derived as the sum of the probability that when the number of incoming links is $j - p + q$, q of p outgoing links overlap with one of incoming links. Therefore, we can obtain B_j as

$$B_j = \sum_{q=\max(0,p+1-j)}^p A_{j-p+q} C_{p,q,j-p+q}. \quad (7)$$

Here, we consider two nodes whose regular links connect to the center broken node. We call them regular node (R-node). And we define non-connective node (NC-node) as the node which have no incoming link. Even if a node has many incoming links, when all of source node of them are broken, it becomes NC-node. However, when the number of incoming link is equal to or greater than 2, the probability that all of source nodes of them are broken is very small compared with that when the number of incoming link is 1. Therefore, we assume the NC-node as the node which have only one incoming link and its source node is broken. That is, when the destination node of regular outgoing link of the broken node has only this regular incoming link and this node is not broken, it becomes the NC-node. Fig. 4 shows the center broken node and R-node. (a) shows the case that none of R-node is broken, (b) shows the case that one of them is broken, and (c) shows the case that both of them are broken. It is found that there is only one node which have possibility to become the NC-node in all case. The probability that this node becomes the NC-node is A_1 . When the number of broken nodes is k , we can consider the three case with $k = 1$, $k = 2$ and $k > 2$. In $k = 1$, this node is the center broken node and it certainly becomes the case (a) and never becomes the case (b) and (c). In $k = 2$, the one node is the center broken node and the other is the correlated broken node and it becomes the cases (a) or (b). And the probability to become the case (a) is $2/l$ and to become the case (b) is $1 - 2/l$ where l is the number of the nodes have possibility to become the correlated broken nodes. If $k > 2$, it becomes all the case. The number of broken nodes except for R-node in (a), (b) and (c) is k , $k - 1$ and $k - 2$, respectively. Furthermore, when the number of links connect to the center broken node is l , the probability that the number of correlated broken nodes is k , denoted as $t_{l,k}$ is

$$t_{l,k} = B_l \binom{l}{k} a^k (1 - a)^{l-k}. \quad (8)$$

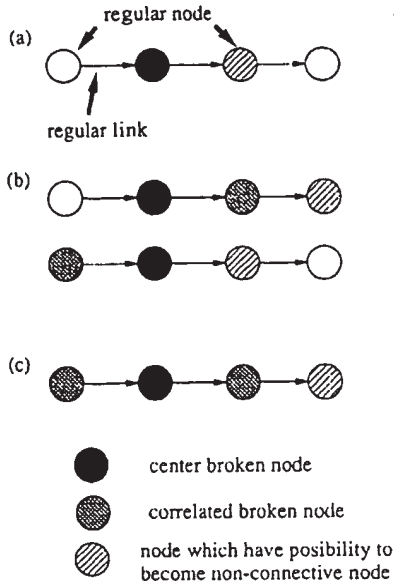


Figure 4. The center broken node and regular nodes.

And in this case, the probability to become the case of (a) is $\binom{k}{0} \frac{1-2P_k}{1P_k}$, to become the case of (b) is $\binom{k}{1} \frac{1-2P_{k-1}}{1P_k}$ and to become the case of (c) is $\binom{k}{2} \frac{1-2P_{k-2}}{1P_k}$. The network connective probability when the number of broken nodes is l , denoted as E_l , is derived in [8] as follows

$$E_l = \prod_{s=0}^{l-1} \frac{N - NA_1 - s}{N - s} \quad (9)$$

Therefore, using (8) and (9), we can obtain the network connective probability as

$$\begin{aligned} R_{CSR N} &= \sum_{l=p}^{N-1} t_{l,0}(1 - A_1) \\ &+ \sum_{l=p}^{N-1} t_{l,1} \left\{ \frac{2}{l}(1 - A_1) + (1 - \frac{2}{l})(1 - A_1)E_1 \right\} \\ &+ \sum_{k=2}^{N-1} \sum_{l=\max(p,k)}^{N-1} t_{l,k} \left\{ \frac{\binom{k}{0} 1-2P_k}{1P_k} (1 - A_1)E_k \right. \\ &\quad \left. + \frac{\binom{k}{1} 1-2P_{k-1}}{1P_k} (1 - A_1)E_{k-1} \right. \\ &\quad \left. + \frac{\binom{k}{2} 1-2P_{k-2}}{1P_k} (1 - A_1)E_{k-2} \right\}. \end{aligned} \quad (10)$$

4 Results

We show computer simulation and theoretical calculation results of the network connective probability under the correlated breakage.

Fig. 5 shows the network connective probability of SN, CN and CSR N with $p = 2$ versus the correlated broken probability. In this

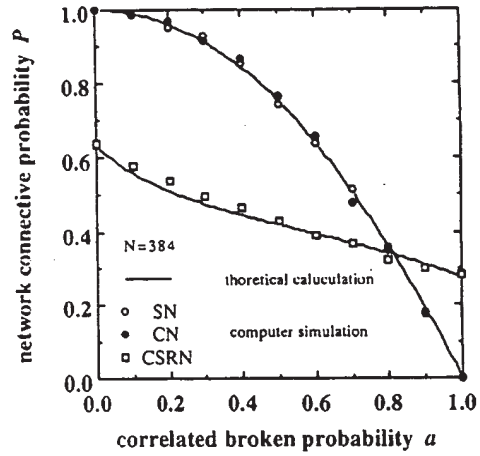


Figure 5. The network connective probability with $p = 2$ versus correlated broken probability.

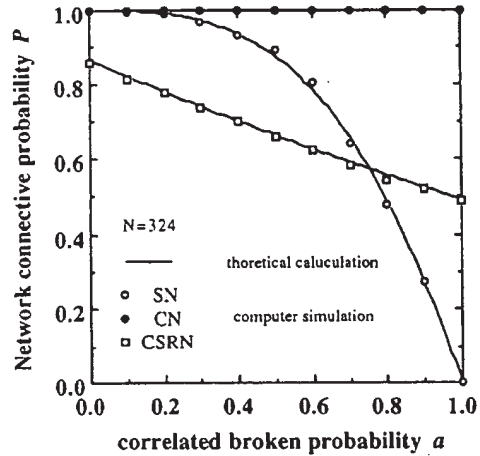


Figure 6. The network connective probability with $p = 3$ versus correlated broken probability.

figure, the chordal length of CN, τ_1 is 50. It is shown that the both the network connective probability of CN or SN is larger than that of CSR N in small a , however, in large a , the network connective probability of CN or SN is smaller than that of CSR N.

Fig. 6 shows the network connective probability of SN, CN and CSR N with $p = 3$ versus the correlated broken probability. In this figure, τ_1 is 50 and τ_2 is 120. The tendency of the network connective probability of SN and CSR N is the same as the case with $p = 2$. However, the tendency of the network connective probability of CN is not different from that with $p = 2$.

In CSR N, because the number of incoming links come into a node is not constant, even if p is large, there are some nodes whose number of incoming links is one. Therefore, the network connective probability itself is small. However, the link assignment of CSR N is random, the condition of correlated breakage is not so different from that of independent breakage. On the other hand, in SN, because the number of incoming links come into a node is constant, the network connective probability under the indepen-

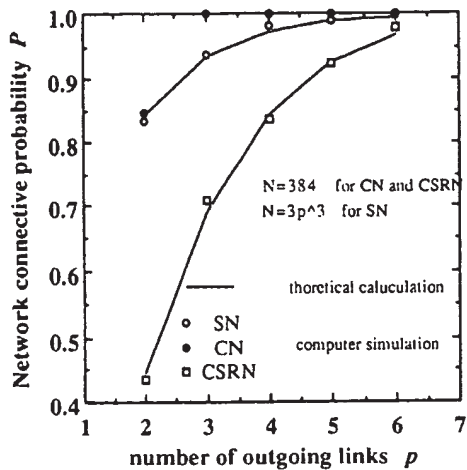


Figure 7. The network connective probability with $\alpha = 0.4$ versus the number of outgoing links per node.

ent breakage is large. However, because of regularity of the link assignment, that under the correlated breakage is small. In CN, when p is two, the link assignment is regular, however, when p is larger than two, every chordal length is random and independent each other, and the link assignment is random. Moreover, the number of incoming links per node of CN is the constant. Therefore, the network connective probability of CN is large under both independent and correlated breakage.

Figs. 7 and 8 show the network connective probability with $\alpha = 0.4$ and 0.8 versus p , respectively. It is shown that the larger α is, the smaller difference of network connective probability between SN and CSRN is, when α is small. On the other hand, when α is large, the larger p is, the larger difference of network connective probability between SN and CSRN is. The reason is as follows. When α is small, the network connective probability of CSRN is small. However, the larger p is, the smaller the number of nodes, whose number of incoming links is 1, is, and the closer to 1 the network connectivity is. In SN and CN, even if p is small, the network connective probability is somewhat large when α is small. When p is large, the network connective probability of CSRN is almost the same with small p . On the other hand, in SN, the tendency network connectivity versus p is almost the same, however, the larger α is, the smaller the value is.

As these results, CN has best performance of network connectivity. However, it has been shown that CN has much poorer performance of intermodal distance than other network. Thus, it is expected for the network to have good performance of both network connective probability and intermodal distance.

Conclusion

We theoretically analyze the network connective probability of multihop network under the correlated damage of node. We treat shuffleNet, chordal network and connective semi-random network. It is found that in the independent node breakage, the network whose number of incoming links is the constant has good performance of network connective probability, and found that in the correlated node breakage, the network whose link assignment

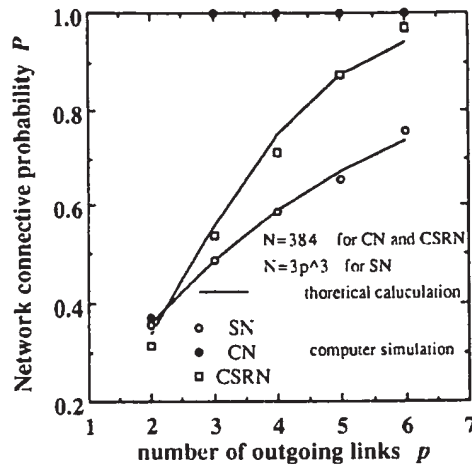


Figure 8. The network connective probability with $\alpha = 0.8$ versus the number of outgoing links per node.

is random has good performance of one.

Acknowledgement

This work is partly supported by Ministry of Education, Kanagawa Academy of Science and Technology, KDD Engineering and Consulting Inc., NTT Data Communication System Co., Hitachi Ltd. and Mitsubishi Electric Co..

References

- [1] M.G. Hluchyj, and M.J. Karol, "ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks", *INFOCOM '88*, New Orleans, LA., Mar. 1988.
- [2] M.J. Karol and S. Shaikh, "A simple adaptive routing scheme for shuffleNet multihop lightwave networks", *GLOBECOM '88*, Nov. 28, 1988-Dec. 1, 1988.
- [3] Bruce W. Arden and Hikyu Lee, "Analysis of Chordal Ring Network", *IEEE Trans. Comp.*, vol. C-30, No. 4, pp. 291-296, Apr. 1981.
- [4] K. W. Doty, "New designs for dense processor interconnection networks", *IEEE Trans. Comp.*, vol. C-33, No. 5, pp. 447-450, May. 1984.
- [5] H. J. Siegel, "Interconnection networks for SIMD machines", *Comput.* pp. 57-65, June 1979.
- [6] Christopher Rose, "Mean Internodal Distance in Regular and Random Multihop Networks", *IEEE Trans. Commun.*, vol. 40, No.8, pp. 1310-1318, Oct. 1992.
- [7] J. M. Peha and F. A. Tobagi, "Analyzing the fault tolerance of double-loop networks", *IEEE Trans. Networking*, vol. 2, No.4, pp. 363-373, Aug. 1994.
- [8] S. Shiokawa and I. Sasase, "Restricted Connective Semi-random Network," 1994 International Symposium on Information Theory and its Applications (ISITA '94), pp. 547-551, Sydney, Australia, November 20-24, 1994.


IEEE Xplore[®]
 RELEASE 1.6

 Welcome
 United States Patent and Trademark Office

[Help](#) | [FAQ](#) | [Terms](#) | [IEEE Peer Review](#)
[Quick Links](#)
[» Search Abst](#)

 Welcome to IEEE Xplore[®]

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

[Search Results](#) [PDF FULL-TEXT 484 KB] [PREV](#) [NEXT](#) [DOWNLOAD CITATION](#)
[Order Reuse Permissions](#)
[RIGHTS LINK](#)

Performance analysis of network connective probability multihop network under correlated breakage

Shiokawa, S. Sasase, I.

Dept. of Electr. Eng., Keio Univ., Yokohama, Japan;

This paper appears in: Communications, 1996. ICC 96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on

Meeting Date: 06/23/1996 - 06/27/1996

Publication Date: 23-27 June 1996

Location: Dallas, TX USA

On page(s): 1581 - 1585 vol.3

Volume: 3

Reference Cited: 8

Number of Pages: 3 vol. xxxix+1848

Inspec Accession Number: 5443424

Abstract:

One of important properties of a multihop network is the network connective probability which evaluate the connectivity of the network. The network connective probability is defined as the probability that when some nodes are broken, the rest of the **nodes connect** each other. Multihop **networks** are classified as a regular network whose link assignment is regular and a random network whose link assignment is random. It has been shown that the network connective probability of a regular network is larger than that of a random network. However, all of these results is shown under independent breakage. We analyze the network connective probability of multihop networks under correlated node breakage. It is shown that a regular network has a better performance the network connective probability than a random network under independent breakage. on the other hand, a random network has a better performance than a regular network under correlated breakage

Index Terms:

[correlation methods](#) [network topology](#) [probability](#) [random processes](#) [telecommunication](#) [network reliability](#) [correlated node breakage](#) [independent breakage](#) [link assignment](#) [multihop network](#) [network connective probability](#) [node breakage](#) [performance](#) [performance analysis](#) [random network](#) [regular network](#)

Documents that cite this document

There are no citing documents available in IEEE Xplore at this time.

[Search Results](#) [[PDF FULL-TEXT 484 KB](#)] [PREV](#) [NEXT](#) [DOWNLOAD CITATION](#)

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

A Flood Routing Method for Data Networks

Jaihyung Cho

Monash University
Clayton 3168, Victoria
Australia
jaihyung@dgs.monash.edu.au

James Breen

Monash University
Clayton 3168, Victoria
Australia
jwb@dgs.monash.edu.au

Abstract

In this paper, a new routing algorithm based on a flooding method is introduced. Flooding techniques have been used previously, e.g. for broadcasting the routing table in the ARPAnet [1] and other special purpose networks [3][4][5]. However, sending data using flooding can often saturate the network [2] and it is usually regarded as an inefficient broadcast mechanism. Our approach is to flood a very short packet to explore an optimal route without relying on a pre-established routing table, and an efficient flood control algorithm to reduce the signalling traffic overhead. This is an inherently robust mechanism in the face of a network configuration change, achieves automatic load sharing across alternative routes, and has potential to solve many contemporary routing problems. An earlier version of this mechanism was originally developed for virtual circuit establishment in the experimental Caroline ATM LAN [6][7] at Monash University.

1. Introduction

Flooding is a data broadcast technique which sends the duplicates of a packet to all neighboring nodes in a network. It is a very reliable method of data transmission because many copies of the original data are generated during the flooding phase, and the destination user can double check the correct reception of the original data. It is also a robust method because no matter how severely the network is damaged, flooding can guarantee at least one copy of the data will be transmitted to the destination, provided a path is available.

While the duplication of packets makes flooding a

generally inappropriate method for data transmission, our approach is to take advantage of the simplicity and robustness of flooding for routing purposes. Very short packets are sent over all possible routes to search for the optimal route of the requested QoS and the data path is established via the selected route. Since the Flood Routing algorithm strictly controls the unnecessary packet duplication, the traffic overhead caused from the flooding traffic is minimal.

Use of flooding for routing purposes has been suggested before [3][4][5], and it has been noted that it can be guaranteed to form a shortest path route[10]. And an earlier protocol was proposed and implemented for the experimental local area ATM network (Caroline [6][7]). However the earlier protocol had problems with scaling timer values, and also required complex mechanism to solve potential race and deadlock problem. Our proposal greatly simplifies the previous mechanism and reduces the earlier problems.

Chapter 2 explains the procedure for route establishment and the simulation results are presented in chapter 3. The advantages of the Flood Routing are reviewed specifically in chapter 4. Chapter 5 concludes this paper with suggesting some possible application area and the future study issues.

2. Flood Routing Mechanism

Figure 1, 3, 4 show the stepwise procedure of the route establishment.

In the Figure 1, the host A is requesting a connection set up to the target host B. In the initial

stage, a short connection request packet (CREQ) is delivered to the first hop router 1 and router 1 starts the flood of the CREQ packets.

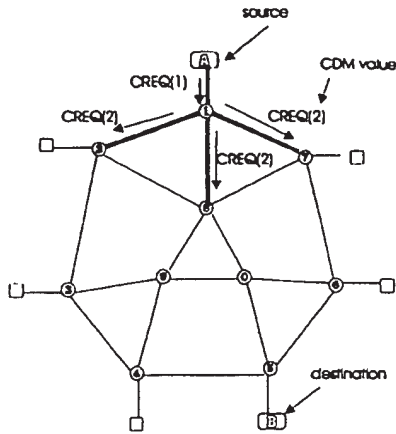


Figure 1

VC number (1byte=0)
Packet Type (1byte="CREQ")
CDM (1byte)
Source Address
Connection No (1byte)
Destination Address
QoS

Figure 2 CREQ Packet Format

Figure 2 shows the format of the CREQ packet. The CREQ packet contains a connection difficulty metric (CDM) field, QoS parameters and the source & destination addresses and connection number. The metric can be any accumulative measure representing the route difficulty, such as hop count, delay, buffer length, etc. The connection number is chosen by the source host to distinguish the different packet floods of the same source and destination.

When a router receives the CREQ packet, the router matches the packet information with the internal Flood Queue to see if the same packet has been received before. If the CREQ packet is new, it records the information in the Flood Queue, increases the CDM value, and forwards the packet to all output links with adequate capacity to meet the QoS except the received one. Thus the flood of CREQ packets propagate through the entire network.

The Flood Queue is a FIFO list which contains the

information relating to the best CREQ packet the router has received for each recent flood. As the flood packet of a new connection arrives and the information is pushed into the Flood Queue, the old information gradually moves to the rear and eventually is removed. The queuing delay from the insertion to the deletion depends on the queue size and the call frequency, and provided this delay is enough to cover the time for network wide flood propagation and reply, there is no need for a timer to wait to the completion of the flood.

Since the CDM value is increased as the CREQ packet passes the routers, the metric value represents the route difficulty that the CREQ packet has experienced. Because of the repeated duplication of the packet, a router may receive another copy of the CREQ packet. In this case, the router compares the metric values of the two packets and if the most recently arrived packet has the better metric value, it updates the information in the Flood Queue and repeats the flood action. Otherwise the packet is discarded. As a consequence, all the routers keep the record of the best partial route and the output link to use for setting up the virtual circuit.

Figure 3 shows the intermediate routers 2, 7, 8 have chosen the links toward the router 1 as the best candidate link. If one of them is requested for the path to the source node A, the router will use this link for the virtual circuit set up.

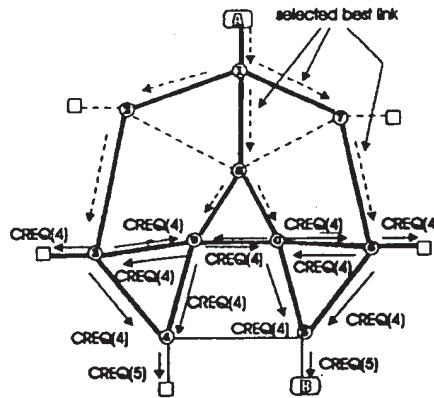


Figure 3

When the destination host receives a CREQ packet, it opens a short time-window to absorb possible further arriving CREQ packets. The expiration of the timer triggers the sending of the

connection acceptance (CACC) packet along the best links indicated by the CREQ packet with the lowest CDM. The CACC packet is relayed back to the source host by the routers which at the same time install the virtual circuit via the optimal route. Finally, when the source host receives the CACC packet, the host may initiate data transmission.

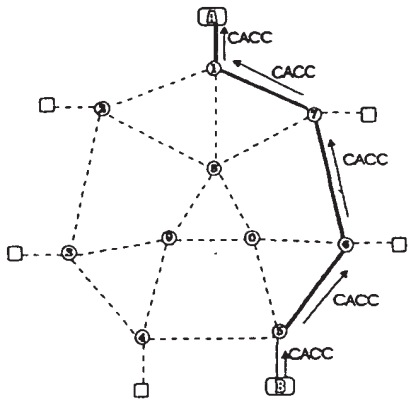


Figure 4

Note that bandwidth reservation occurs during the relay of the CACC packet. It is possible that the available QoS will have dropped below the requested level in one or more links. In this case, the source may either accept the lower QoS, or close the connection and try again.

More implementation details of the flooding protocol can be found in [9].

3. Simulation Result

One concern of Flood Routing is whether it will lead to congestion of the network by the signalling

traffic. A simulation was carried out using various network conditions. Figure 5 shows the number of flooding packets produced in a connection trial in a normal traffic condition on a network consisting of 5 switching nodes, 9 hosts and 16 links. The simulation tested the event of 2000 seconds.

The graph shows that the total number of flooding packets per connection converges on the lower bound 18 with some exceptions. This is slightly higher than the number of the network links (16). This shows how the flood control mechanism is efficient in that the routers usually generate only one flooding packet per output link and this duplication process is rarely repeated again. As a result, the total number of flooding packets per connection is nearly same as the number of network links.

Considering the small size of the flooding packet, the bandwidth consumed by the signalling traffic is small. Suppose an ATM network using the Flood Routing generates 1000 calls per seconds, the bandwidth consumption by the signalling traffic will only be about 424 Kbps (= 1 K * 53 byte) per link and this does not include any additional route management traffic such as the routing table update.

From the simulation, it is observed that the average number and the maximum number of the flooding packets depends on the network topology and the traffic condition. If the network is simple topology such as a tree or a star shape, the average number of the flooding packets is nearly identical to the number of the network links. If the network is a complex topology such as a complete mesh topology, and there is a high traffic load, the routers tend to generate more packets because of the racing of the flooding packets.

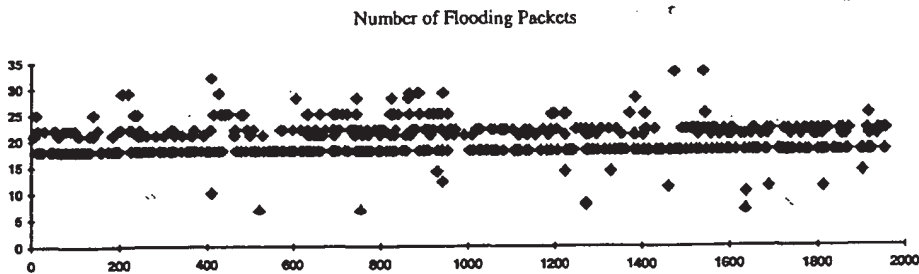


Figure 5

The connections established by Flood Routing successfully avoid busy links and disperse the communication paths to all possible routes. This reduced the chance of congestion and utilizes all network resources efficiently.

4. Advantages of the Flood Routing

The distinctive features of the Flood Routing method are :

(a) It facilitates the load sharing of available network resources. If many possible routes exist between two end points in a network, the Flood Routing can disperse different connections over different routes to share the network load. Figure 6 shows this example.

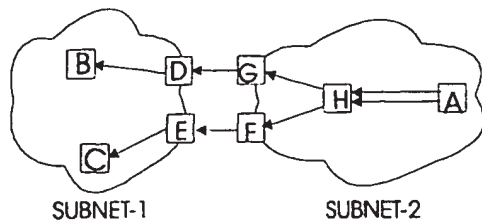


Figure 6 Example of Multipath Connection

In the sample network, there are more than two links exist between node A and H, and the node A used all links for different connections with balancing the load. More than two exterior routers are connecting the subnet 1 and the subnet 2, and the node H distributed the connections to all exterior routers. Therefore, all the network resources are utilized fully in Flood Routing network. This load sharing capability has been considered to be a difficult problem in table based routing algorithms.

(b) It automatically adapts to changes in the network configuration. For example, if the overall traffic between two end points has been increased, the network bandwidth can simply be expanded by adding more links between routers. The Flood Routing algorithm can recognize the additional links and use them for sharing the load in new connections.

(c) The method is robust. The Flood routing can achieve a successful connection even when the network is severely damaged, provided flooding packets can reach the destination. Once a flooding

packet reaches the destination, the connection can be established via the un-damaged part of the network which was searched by the packet. This is very useful property in networks which are vulnerable but which require high reliability, such as military networks.

(d) The method is simple to manage, as it makes no use of routing tables. This table-less routing method does not have the problem like "Convergence time" of the Distance Vector routing [8].

(e) It is possible to find the optimal route of the requested bandwidth or the quality of service. While the packet flood is progressing, bandwidth requirement and QoS constraints specified in the flooding packets are examined by the routers and the links that does not meet the requirements are excluded from the routing decision. As a result, the route constructed with the qualified links can meet the bandwidth and the QoS requirements, usually in the first attempt.

(f) It is a loop-free routing algorithm. The only possible case that the route may consist a loop can be caused from the corrupted metric information. However this can be detected by a check sum.

(g) Since the flooding method is basically a broadcast mechanism, it can be used for locating resources in network. Many network applications are best served by a broadcast facility, such as distributed data bases, address resolution, or mobile communications. Implementing broadcast in point-to-point networks is not straight forward. The flooding technique provides a means to solve this problem. In particular, locating a mobile user by Flood Routing, and establishing a dynamic route is an interesting issue. Application to a movable network in which entire network units including both the mobile users as well as the switching nodes and the wireless links is another potential research area.

5. Future Study and Conclusion

In this paper, we introduced a revised Flood Routing technique. Flood Routing is a novel approach to network routing which has the potential to solve many of the routing problems in contemporary networks. The basic Flood Routing presented in this paper has been developed to be used in an ATM style network, however we

believe a similar technique can also be applied to IP routing. Another promising area of application of this method would be military or mobile networks which require high mobility and reliability. Research to extend the point-to-point Flood Routing to optimal multi-point routing is now progressing. Further analysis of performance, and application to large scale networks are the future issues.

Routing Technique", Technical Report 96-5, Faculty of Computing and Information Technology, Department of Digital Systems, Monash University, January 1996

[10] A. S. Tanenbaum, "Computer Networks", Prentice Hall, 1989

References

- [1] R. Perlman, "Fault-tolerant Broadcast of Routing Information", Proc. IEEE Infocom '83, 1983
- [2] E. C. Rosen, "Vulnerabilities of Network Control Protocol: An Example", Computer Communication Review, July 1981, 11-16
- [3] V. O. K. Li and R. Chang, "Proposed Routing Algorithms for the U.S Army Mobile Subscriber Equipment (MSE) Network", Proceedings - IEEE Military Communications Conference, Monterey, CA, 1986, paper 39.4
- [4] M. Kavehrad and I.M.I Habbaqb, "A simple High Speed Optical Local Area Network Based on Flooding", IEEE Journal on Selected Areas in Communications, Vol. 6, No.6, July 1988
- [5] P. J. Lyons and A. J. McGregor, "MasseyNet: A University Oriented Local Area Network", IFIP Working Conference on the Implications of Interconnecting Microcomputers in Education, August 1986
- [6] C. Blackwood, R. Harris, A. T. McGregor and J. W. Breen, "The Caroline Project: An Experimental Local Area Cell-Switching Network", ATNAC-94, 1994
- [7] Rik Harris, "Routings in Large ATM Networks", Master of Computing Thesis, Department of Digital Systems, Monash University, 1995
- [8] W. D. Tajibnapis, "A Correctness Proof of a Topology Information maintenance Protocol for Distributed Computer Networks", Communications of the ACM, Vol.20, July 1977, 477-485
- [9] Jaihyung Cho, James Breen, "Caroline Flood

A Reliable Dissemination Protocol for Interactive Collaborative Applications

Rajendra Yavatkar, James Griffioen, and Madhu Sudan
Department of Computer Science
University of Kentucky
Lexington, KY 40506
{raj,griff,madhu}@dcs.uky.edu
(606) 257-3961

ABSTRACT

The widespread availability of networked multimedia workstations and PCs has caused a significant interest in the use of collaborative multimedia applications. Examples of such applications include distributed shared whiteboards, group editors, and distributed games or simulations. Such applications often involve many participants and typically require a specific form of multicast communication called *dissemination* in which a single sender must reliably transmit data to multiple receivers in a timely fashion. This paper describes the design and implementation of a reliable multicast transport protocol called *TMTP* (Tree-based Multicast Transport Protocol). *TMTP* exploits the efficient best-effort delivery mechanism of IP multicast for packet routing and delivery. However, for the purpose of scalable flow and error control, it dynamically organizes the participants into a hierarchical control tree. The control tree hierarchy employs *restricted nacks with suppression* and an *expanding ring search* to distribute the functions of state management and error recovery among many members, thereby allowing scalability to large numbers of receivers. An Mbone-based implementation of *TMTP* spanning the United States and Europe has been tested and experimental results are presented.

KEYWORDS

Reliable Multicast, Transport Protocols, Mbone, Interactive Multipoint Services, Collaboration

INTRODUCTION

Widespread availability of IP multicast [6, 2] has substantially increased the geographic span and portability of collaborative multimedia applications. Example ap-

plications include distributed shared whiteboards [15], group editors [7, 14], and distributed games or simulations. Such applications often involve a large number of participants and are interactive in nature with participants dynamically joining and leaving the applications. For example, a large-scale conferencing application (e.g., an IETF presentation) may involve hundreds of people who listen for a short time and then leave the conference. These applications typically require a specific form of multicast delivery called *dissemination*. Dissemination involves 1xN communication in which a single sender must reliably multicast a significant amount of data to multiple receivers. IP multicast provides scalable and efficient routing and delivery of IP packets to multiple receivers. However, it does not provide the reliability needed by these types of collaborative applications.

Our goal is to exploit the highly efficient best-effort delivery mechanisms of IP multicast to construct a scalable and efficient protocol for reliable dissemination. Reliable dissemination on the scale of tens or hundreds of participants scattered across the Internet requires carefully designed flow and error control algorithms that avoid the many potential bottlenecks. Potential bottlenecks include host processing capacity [18] and network resources. Host processing capacity becomes a bottleneck when the sender must maintain state information and process incoming acknowledgements and retransmission requests from a large number of receivers. Network resources become a bottleneck unless the frequency and scope of retransmissions is limited. For instance, loss of packets due to congestion in a small portion of the IP multicast tree should not lead to retransmission of packets to all the receivers. Frequent multicast retransmissions of packets also wastes valuable network bandwidth.

This paper describes the design and implementation of a reliable dissemination protocol called *TMTP* (Tree-based Multicast Transport Protocol) that includes the following features:

1. *TMTP* takes advantage of IP multicast for efficient

packet routing and delivery.

2. TMTP uses an *expanding ring search* to dynamically organize the dissemination group members into a *hierarchical control tree* as members join and leave a group.
3. TMTP achieves scalable reliable dissemination via the hierarchical control tree used for flow and error control. The control tree takes the flow and error control duties normally placed at the sender and distributes them across several nodes. This distribution of control also allows error recovery to proceed independently and concurrently in different portions of the network.
4. Error recovery is primarily driven by receivers who use a combination of *restricted negative acknowledgements with nack suppression* and periodic positive acknowledgements. In addition, the tree structure is exploited to restrict the scope of retransmissions to the region where packet loss occurs; thereby insulating the rest of the network from additional traffic.

We have completed a user-level implementation of TMTP based on IP/UDP multicast and have used it for a systematic performance evaluation of reliable dissemination across the current Internet Mbone. Our experiments involved as many as thirty group members located at several sites in the US and Europe. The results are impressive; TMTP meets our objective of scalability by significantly reducing the sender's processing load, the total number of retransmissions that occur, and the end-to-end latency as the number of receivers is increased.

Background

A considerable amount of research has been reported in the area of group communication. Several systems such as the ISIS system [1], the V kernel [4], Amoeba, the Psynch protocol [17], and various others have proposed group communication primitives for constructing distributed applications. However, all of these systems support a general group communication model (NxN communication) designed to provide reliable delivery with support for atomicity and/or causality or to simply support an unreliable, unordered multicast delivery. Similarly, transport protocols specifically designed to support group communication have also been designed before [13, 5, 3, 19, 9]. These protocols mainly concentrated on providing reliable broadcast over local area networks or broadcast links. Flow and error control mechanisms employed in networks with physical layer multicast capability are simple and do not necessarily scale well to a wide area network with unreliable packet delivery.

Earlier multicast protocols used conventional flow and error control mechanisms based on a *sender-*

initiated approach in which the sender disseminates packets and uses either a *Go-Back-N* or a *selective repeat* mechanism for error recovery. If used for reliable dissemination of information to a large number of receivers, this approach has several limitations. First, the sender must maintain and process a large amount of state information associated with each receiver. Second, the approach can lead to a *packet implosion* problem where a large number of ACKs or NACKs must be received and processed by the sender over a short interval. Overall, this can lead to severe bottlenecks at a sender resulting in an overall decrease in throughput [18].

An alternate approach based on *receiver-initiated* methods [19, 15] shifts the burden of reliable delivery to the receivers. Each receiver maintains state information and explicitly requests retransmission of lost packets by sending negative acknowledgements (NACKs). Under this approach, the receiver uses two kinds of timers. The first timer is used to detect lost packets when no new data is received for some time. The second timer is used to delay transmission of NACKs in the hope that some other receiver might generate a NACK (called *nack suppression*).

It has been shown that the receiver-initiated approach reduces the bottleneck at the sender and provides substantially better performance [18]. However, the receiver-initiated approach has some major drawbacks. First, the sender does not receive positive confirmation of reception of data from all the receivers and, therefore, must continue to buffer data for long periods of time. The second and most important drawback is that the end-to-end delay in delivery can be arbitrarily large as error recovery solely depends on the timeouts at the receiver unless the sender periodically polls the receivers to detect errors [19]. If the sender sends a train of packets and if the last few packets in the train are lost, receivers take a long time to recover causing unnecessary increases in end-to-end delay. Periodic polling of all receivers is not an efficient and practical solution in a wide area network. Third, the approach requires that a NACK must be multicast to all the receivers to allow suppression of NACKs at other receivers and, similarly, all the retransmissions must be multicast to all the receivers. However, this can result in unnecessary propagation of multicast traffic over a large geographic area even if the packet losses and recovery problems are restricted to a distant but small geographic area¹. Thus, the approach may unnecessarily waste valuable bandwidth.

In this paper we present an alternative approach that achieves scalable reliable dissemination by reducing the processing bottlenecks of sender-initiated approaches

¹ Assume that only a distant portion of the Internet is congested resulting in packet loss in the area. One or more receivers in this region may multicast repeated NACKS that must be processed by all the receivers and the resulting retransmissions must also be forwarded to and processed by all the receivers.

and avoiding the long recovery times of receiver-initiated approaches.

OVERVIEW OF OUR APPROACH

Under the TMTP dissemination model, a single sender multicasts a stream of information to a *dissemination group*. A *dissemination group* consists of processes scattered throughout the Internet, all interested in receiving the same data feed. A session directory service (similar to the session directory *sd* from LBL [12]) advertizes all active dissemination groups.

Before a transmitting process can begin to send its stream of information, the process must create a dissemination group. Once the dissemination group has been formed, interested processes can dynamically join the group to receive the data feed. The dissemination protocol does not provide any mechanism to insure that all receivers are present and listening before transmission begins. Although such a mechanism may be applicable in certain situations, we envision a highly dynamic dissemination system in which receiver processes usually join a data feed already in progress and/or leave a data feed prior to its termination. Consequently, the protocol makes no effort to coordinate the sender and receivers, and an application must rely on an external synchronization method when such coordination is necessary.

For the purposes of flow and error control, TMTP organizes the group participants into a hierarchy of subnets or *domains*. Typically, all the group members in the same subnet belong to a domain and a single *domain manager* acts as a representative on behalf of the domain for that particular group. The domain manager is responsible for recovering from errors and handling local retransmissions if one or more of the group members within its domain do not receive some packets.

In addition to handling error recovery for the local domain, each domain manager may also provide error recovery for other domain managers in its vicinity. For this purpose, the domain managers are organized into a *control tree* as shown in Figure 1. The sender in a dissemination group serves as the root of the tree and has at most K domain managers as children. Similarly, each domain manager will accept at most K other domain managers as children, resulting in a tree with maximum degree K . The value of K is chosen at the time of group creation and registration and does *not* include local group members in a domain (or subnet). The degree of the tree (K) limits the processing load on the sender and the internal nodes of the control tree. Consequently, the protocol overhead grows slowly, proportional to the $\log_K(\text{Number_Of_Receivers})$.

Packet transmission in TMTP proceeds as follows. When a sender wishes to send data, TMTP uses IP multicast to transmit packets to the entire group. The transmission rate is controlled using a sliding window based protocol described later. The control tree ensures reliable delivery to each member. Each node of

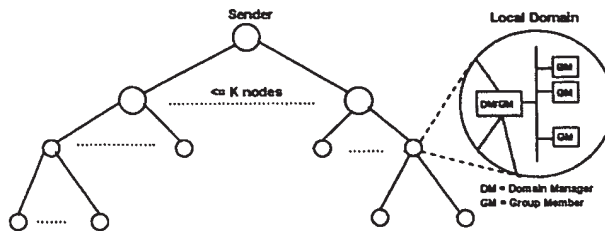


Figure 1: An example control tree with the maximum degree of each node restricted to K . Local group members within a domain are indicated by GM. There is no restriction on the number of local group members within a domain.

the control tree (including the root) is only responsible for handling the errors that arise in its immediate K children. Likewise, children only send periodic, positive acknowledgments to their immediate parent. When a child detects a missing packet, the child multicasts a NACK in combination with nack suppression. On the receipt of the NACK, its parent in the control tree multicasts the missing packet. To limit the scope of the multicast NACK and the ensuing multicast retransmission, TMTP uses the *Time-To-Live (TTL)* field to restrict the transmission radius of the message. As a result, error recovery is completely localized. Thus, a dissemination application such as a world-wide IETF conference would organize each geographic domain (e.g., the receivers in California vs. all the receivers in Australia) into separate subtrees so that error recovery in a region can proceed independently without causing additional traffic in other regions. TMTP's hierarchical structure also reduces the end-to-end delay because the retransmission requests need not propagate all the way back to the original sender. In addition, locally retransmitted packets will be received quickly by the affected receivers.

The control tree is self-organizing and does not rely on any centralized coordinator, being built dynamically as members join and leave the group. A new domain manager attaches to the control tree after discovering the closest node in the tree using an *expanded ring* search. Note that *the control tree is built solely at the transport layer and thus does not require any explicit support from, or modification to, the IP multicast infrastructure inside the routers.*

The following sections describe the details of the TMTP protocol.

GROUP MANAGEMENT

The session directory provides the following group management primitives:

CreateGroup(GName,CommType): A sender creates a new group (with identifier *GName*) using the *CreateGroup* routine. *CommType* specifies the type of communication pattern desired and may be ei-

ther *dissemination* or *concast*². If successful, CreateGroup returns an IP multicast address and a port number to use when transmitting the data.

JoinGroup(Gname): Processes that want to receive the data feed represented by *GName* call JoinGroup to become a member of the group. Join returns the transport level address (IP multicast address and port number) for the group which the new process uses to listen to the data feed.

LeaveGroup(Gname): Removes the caller from the dissemination group *GName*.

DeleteGroup(GName): When the transmission is complete, the sending process issues a DeleteGroup request to remove the group *GName* from the system. DeleteGroup also informs all participants, and domain managers that the group is no longer active.

CONTROL TREE MANAGEMENT

Each dissemination group has an associated control tree consisting of domain managers. Over the lifetime of the dissemination group, the control tree grows and shrinks dynamically in response to additions and deletions to and from the dissemination group membership. Specifically, the tree grows whenever the first process in a domain joins the group (i.e., a domain manager is created) and shrinks whenever the last process left in a domain leaves the group (i.e., a domain manager terminates).

There are only two operations associated with control tree management: *JoinTree* and *LeaveTree*. When a new domain manager is created, it executes the JoinTree protocol to become a member of the control tree. Likewise, domain managers that no longer have any local processes to support may choose to execute the LeaveTree protocol.

Figures 2 and 3 outline the protocols for joining and leaving the control tree. The join algorithm employs an *expanding ring search* to locate potential connection points into the control tree. A new domain manager begins an expanding ring search by multicasting a SEARCH_FOR_PARENT request message with a small time-to-live value (TTL). The small TTL value restricts the scope of the search to nearby control nodes by limiting the propagation of the multicast message. If the manager does not receive a response within some fixed timeout period, the manager resends the SEARCH_FOR_PARENT message using a larger TTL value. This process repeats until the manager receives a WILLING_TO_BE_PARENT message from one or more domain managers in the control tree. All existing domain managers that receive the SEARCH_FOR_PARENT message will respond with a

²Although this paper focuses on dissemination, TMTP also supports efficient concast style communication[10].

```

While (NotDone) {
  Multicast a SEARCH_FOR_PARENT msg
  Collect responses
  If (no responses)
    Increment TTL /* try again */
  Else
    Select closest respondent as parent
    Send JOIN_REQUEST to parent
    Wait for JOIN_CONFIRM reply
    If (JOIN_CONFIRM received)
      NotDone = False
    Else /* try again */
}

```

(A) New Domain Manger Algorithm

```

Receive request message
If (request is SEARCH_FOR_PARENT)
  If (MAX_CHILDREN not exceeded)
    Send WILLING_TO_BE_PARENT msg
  Else
    /* Do not respond */
Else If (request is JOIN_REQUEST)
  Add child to the tree
  Send JOIN_CONFIRM msg

```

(B) Existing Domain Manger Algorithm

Figure 2: The protocol used by domain managers to join the control tree. A new domain manager performs algorithm (A) while all other existing managers execute algorithm (B).

```

If (I_am_a_leaf_manager)
  Send LEAVE_TREE request
  to parent
  Receive LEAVE_CONFIRH
  Terminate
Else /* I am an internal manager */
  Fulfill all pending obligations
  Send FIND_NEW_PARENT message to children
  Receive FIND_NEW_PARENT reply from all children
  Send LEAVE_TREE request to parent
  Receive LEAVE_CONFIRH
  Terminate

```

Figure 3: The algorithm used to leave the control tree after the last local group member terminates.

WILLING_TO_BE_PARENT message unless they already support the maximum number of children. The new domain manager then selects the closest domain manager (based on the TTL values) and directly contacts the selected manager to become its child. For each domain, its manager maintains a *multicast radius* for the domain, which is the TTL distance to the farthest child within the domain. The domain manager keeps the children informed of the current multicast radius. As described later in the description of the error control part of TMTP, both parent and its children in a domain use the current multicast radius to restrict the scope of their multicast transmissions.

Before describing the LeaveTree protocol, note that a domain manager typically has two types of children. First, a domain manager supports the group members that reside within its local domain. Second, a domain manager may also act as a parent to one or more children domain managers. We say a manager is an *internal manager* of the tree if it has other domain managers as children. We say a manager is a *leaf manager* if it only supports group members from its local domain.

A domain manager may only leave the tree after its last local member leaves the group. At this point, the domain manager begins executing the LeaveTree protocol shown in Figure 3. The algorithm for leaf managers is straightforward. However, the algorithm for internal managers is complicated by the fact that internal managers are a crucial link in the control tree, continuously servicing flow and error control messages from other managers, even when there are no local domain members left. In short, a departing internal node must discontinue service at some point and possibly coordinate children with the rest of the tree to allow seamless reintegration of children into the tree. Several alternative algorithms can be devised to determine when and how service will be cutoff and children reintegrated. The level of service provided by these algorithms could range from “unrecoverable interrupted service” to “temporarily interrupted service” to “uninterrupted service”. Our current implementation provides “probably unin-

errupted service” which means children of the departing manager continue to receive the feed while they reintegrate themselves into the tree. However, errors that arise during the brief reintegration time might not be correctable. We are still investigating alternatives to this approach.

After a departing manager has fulfilled all obligations to its children and parent, the departing manager instructs its children to find a new parent. The children then begin the process of joining the tree all over again. Although we investigated several other possible algorithms, we chose the above algorithm for its simplicity. Other, more static algorithms, such as requiring orphaned children to attach themselves to their grandparents, often result in poorly constructed control trees. Forcing the children to restart the join procedure ensures that children will select the closest possible connection point. Other more complex dynamic methods can be used to speed up the selection of the closest connection point but, in our experience, the performance of our simple algorithm has been acceptable.

DELIVERY MANAGEMENT

TMTP couples its packet transmission strategy with a unique tree-based error and flow control protocol to provide efficient and reliable dissemination. Conventional flow and error control algorithms employ a sender or receiver-initiated approach. However, using the control tree, TMTP is able to combine the advantages of each approach while avoiding their disadvantages. Logically, TMTP’s delivery management protocol can be partitioned into three components: data transmission, error handling, and flow control. The following sections address each of these aspects.

The Transmission Protocol

The basic transmission protocol is quite simple and is best described via a simple example. Assume a sender process S has established a dissemination group X and wants to multicast data to group X. S begins by multicasting data to the $\langle IP_multicast_addr, port_no \rangle$ representing group X. The multicast packets travel directly to all group members via standard IP multicast. In addition, all the domain managers in the control tree listen and receive the packets directly.

As in the sender-initiated approach, the root S expects to receive positive acknowledgments in order to reclaim buffer space and implement flow control. However, to avoid the *ack implosion* problem of the sender initiated approach, the sender does not receive acknowledgments directly from all the group members and, instead, receives ACKs only from its K immediate children. Once a domain manager receives a multicast packet from the sender, it can send an acknowledgment for the packet to its parent because the branch of the tree the manager represents has successfully received the packet (even though the individual members may not have received the packet). That is, a domain manager

does not need to wait for ACKs from its children in order to send an ACK to the parent. In addition, each domain manager only periodically sends such ACKs to its parent. This feature substantially reduces ACK processing at the sender (and each domain manager).

Error Control

Before describing the details of TMTP's error control mechanism we must define an important concept called *limited scope multicast* messages. A limited scope multicast restricts the scope of a multicast message by setting the TTL value in the IP header to some small value which we call the multicast radius. The appropriate multicast radius to use is obtained from the expanding ring search that domain managers use to join the tree. Limited scope multicast messages prevent messages targeted to a particular region of the tree from propagating throughout the entire Internet.

TMTP employs error control techniques from both sender and receiver initiated approaches. Like the sender initiated approach, a TMTP traffic source (sender) requires periodic (unicast) positive acknowledgements and uses timeouts and (limited scope multicast) retransmissions to ensure reliable delivery to all its immediate children (domain managers). However, in addition to the sender, the domain managers in the control tree are also responsible for error control after they receive packets from the sender. Although the sender initially multicasts packets to the entire group, it is the domain manager's responsibility to ensure reliable delivery. Each domain manager also relies on periodic positive ACKs (from its immediate children), timeouts, and retransmissions to ensure reliable delivery to its children. When a retransmission timeout occurs, the sender (or domain manager) assumes the packet was lost and retransmits it using IP multicast (with a small TTL equal to the multicast radius for the local domain so that it only goes to its children).

In addition to the sender initiated approach, TMTP uses *restricted NACKs with NACK suppression* to respond quickly to packet losses. When a receiver notices a missing packet, the receiver generates a negative acknowledgment that is multicast to the parent and siblings using a restricted (small) TTL value. To avoid multiple receivers generating a NACK for the same packet, each receiver delays a random amount of time before transmitting its NACK. If the receiver hears a NACK from another sibling during the delay period, it suppresses its own NACK. This technique substantially reduces the load imposed by NACKs. When a domain manager receives a NACK, it immediately responds by multicasting the missing packet to the local domain using a limited scope multicast message.

Flow Control

TMTP achieves flow control by using a combination of rate-based and window-based techniques. The rate-based component of the protocol prohibits senders from

transmitting data faster than some predefined maximum transmission rate. The maximum rate is set when the group is created and never changes. Despite its static nature, a fixed rate helps avoid congestion arising from bursty traffic and packet loss at rate-dependent receivers while still providing the necessary quality-of-service without excessive overhead.

TMTP's primary means of flow control consists of a window-based approach used for both dissemination from the sender and retransmission from domain managers. Within a window, senders transmit at a fixed rate.

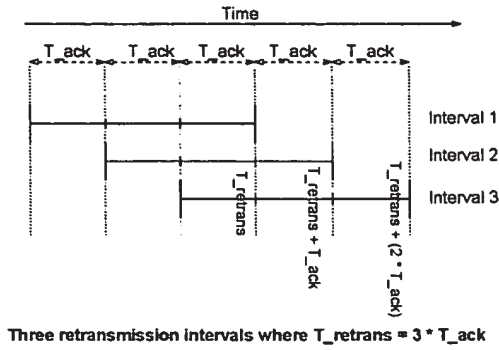
TMTP's window-based flow control differs slightly from conventional point-to-point window-based flow control. Note that retransmissions are very expensive because they are multicast. In addition, transient traffic conditions or congestion in one part of the network can put backpressure on the sender causing it to slow the data flow. To oversimplify, TMTP avoids both of these problems by partitioning the window and delaying retransmissions as long as possible. This increases the chance of a positive acknowledgement being received and it also allows domain managers to rectify transient behavior before it begins to cause backpressure.

TMTP uses two different timers to control the window size and the rate at which the window advances. $T_{retrans}$ defines a timeout period that begins when the first packet in a window is sent. Since the transfer rate is fixed, $T_{retrans}$ also defines the window size. A second timer, T_{ack} , defines the periodic interval at which each receiver is expected to unicast a positive ACK to its parent.

The sender specifies the value of T_{ack} based on the RTT to its farthest child. $T_{retrans}$ is chosen such that $T_{retrans} = n \times T_{ack}$, where n is an integer, $n \geq 2$. Both $T_{retrans}$ and T_{ack} are fixed at the beginning of transmission and do not change. A sender must allocate enough buffer space to hold packets that are transmitted over the $T_{retrans}$ period.

Figure 4 illustrates the windowing algorithm graphically. The sender starts a timer and begins transmitting data (at a fixed rate). Consider the packets transmitted during the first T_{ack} interval. Although the sender should see a positive ACK at time T_{ack} , the sender does not require one until time $T_{retrans}$. Instead, the sender continues to send packets during the second and third interval. After $T_{retrans}$ amount of time, the timer expires. At this point, the sender retransmits all unACK'd packets that were sent during the first T_{ack} interval. Retransmissions continue until all packets in the T_{ack} interval are acknowledged at which point the window is advanced by T_{ack} . On the receiving end, packets continue to arrive without being acknowledged until T_{ack} amount of time has expired³.

³However, a receiver may generate a *restricted NACK* as soon as it detects a missing packet.



At the end of the first interval, packets sent during the first T_{ack} period are retransmitted. At the end of the second interval, packets sent during the second T_{ack} period are retransmitted. At the end of the third interval, packets sent during the third T_{ack} period are retransmitted.

Figure 4: Different Stages in Sending Data

A domain manager must continue to hold packets in its buffer until all of its children have acknowledged them. If the children fail to acknowledge packets, the domain manager's window will not advance and its buffers will eventually fill up. As a result, the domain manager will drop and not acknowledge any new data from the sender, thereby causing backpressure to propagate up the tree which ultimately slows the flow of data.

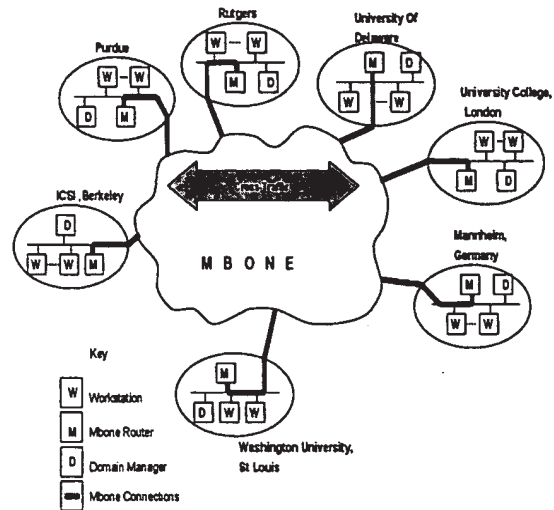
There are three reasons for using multiple T_{ack} intervals during a retransmission timeout interval ($T_{retrans}$). First, by requiring more than one positive ACK during the retransmission interval, TMTP protects itself from spurious retransmissions arising from lost ACKs. First, by requiring more than one positive ACK during the retransmission interval, TMTP protects itself from spurious retransmissions arising from lost ACKs. Second, a larger retransmission interval gives receivers sufficient time to recover missing packets using receiver-initiated recovery when only one (or a few) packets in a window are lost. This avoids unnecessary multicast retransmissions of a window full of data. Third, multiple T_{ack} intervals during the retransmission interval provide sufficient opportunity for a domain manager to recover from transient network load in its part of the subtree without unnecessarily applying backpressure to the sender.

We have chosen the value of the multiplying factor n to be 3 based on empirical evidence; the appropriate value depends on several factors including expected error rates, variance in RTT, and expected length of the intervals with transient, localized congestion. Further study is necessary to determine whether value of n should be chosen dynamically using an adaptive algorithm.

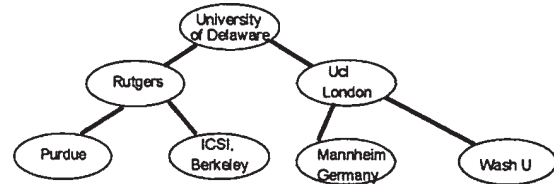
RESULTS

The Test Environment

Figure 5a illustrates the environment in which the experiments were run. Our tests involved seven geograph-



(5a) The Internet Mbone Used



(5b) The Control Tree

Figure 5: Figure 5a shows the test environment consisting of seven geographically distant sites connected by the Mbone. Figure 5b shows the corresponding control tree configuration used in the experiments.

ically distinct Internet Mbone sites across the United States and Europe: Washington University in St. Louis, Purdue University, the International Computer Science Institute at Berkeley, Rutgers University, the University of Delaware, University College at London, and the University of Mannheim in Germany. All of our experiments were conducted using standard IP multicast across the Internet Mbone and thus experienced real Internet delays, congestion, and packet loss.

As a point of comparison, we implemented a standard sender-initiated reliable multicast transport protocol both with and without window-base flow control (called *WIN_BASEP* and *BURST_BASEP* respectively). Under both protocols, the sender maintains state information for all receivers, expects positive ACKs from each receiver, and uses timeouts and global multicast retransmissions to recover from missing acknowledgments. The two BASEP protocols illustrate the performance bottlenecks related to processor load and end-to-end latency. All three protocols used the same packet size (1 Kbytes). TMTP and WIN_BASEP used a window size of 5. TMTP uses a transmission rate of 10 packets per

second, while both BASEP protocols transmit packets as fast as possible (up to the window size in the case of WIN_BASEP). Both BASEP protocols set the retransmission timeout period to be twice the RTT to the farthest site (approx. 2 seconds in our tests). TMTP uses a retransmission period of $T_{retrans} = N \times T_{ack}$. T_{ack} is dynamically set based on the RTT to the farthest group member (approximately 1.1 seconds for our tests). After some preliminary evaluation of different setting for N , our empirical results indicated that $N = 3$ provides sufficient time for local domains to recover without delaying acks unnecessarily or consuming too much buffer space. Consequently, $T_{retrans}$ was approximately 3.3 seconds in our tests. The following sections describe the performance measures used and detail the actual experiments performed.

Performance Measures

To evaluate the performance of our protocol, we identified two important measures of performance: *end-to-end delay* and *processing load*. In addition, we monitored the total number of retransmissions to estimate the amount of network traffic generated by TMTP.

From the application's perspective, the primary concern is the delay in reliably delivering the entire data feed (e.g., video, audio, or file data) to the multiple recipients of the group. To measure the end-to-end delay, we required that each receiving application send back a single positive acknowledgment (a GOT_IT message) to the sending application when the entire data transmission was complete. The sending application then calculated the end-to-end delay as the time between the beginning of the transmission and the time at which the last group member's final GOT_IT message is received.

From the network's perspective, the primary concern is network load and scalability of the algorithm. If the protocol provides low end-to-end delay but consumes large amounts of network resources, the protocol will not scale well, congesting the Internet by consuming shared resources required by other Internet users. There are two aspects to network load: processing load and bandwidth consumption. To measure the processing load at the sender, receivers, and domain managers, we monitored the following processing activities:

- receiving and processing a selective positive acknowledgment
- receiving and processing a negative acknowledgment
- handling a timer event (such as a retransmission timeout)
- performing a retransmission

Because it is hard to measure the amount of processing time needed for each of the events listed above (and highly dependent on the operating system and architecture), we have chosen to simply count the total number

of such events at the sender to estimate the processing load generated by a protocol.

The second important measure of network load is bandwidth consumption. The precise amount of bandwidth consumed by each protocol is much harder to quantify since we were unable to collect traces of traffic across the Mbone to determine the number of links traversed and the amount of bandwidth consumed over each link. However, our results indicate that TMTP generated far fewer retransmissions than the BASEP protocols, and most TMTP retransmissions are local to a particular domain. For example, under the BASEP protocols most timeouts/retransmissions occurred as a result of dropped ACKs. TMTP's hierarchy substantially reduced the number of lost ACKs, experiencing only 6 local retransmissions totaled across all domain managers (four occurring concurrently) as opposed to 9 global retransmission for BURST_BASEP (out of thirty 1K messages).

Experiments Performed

Each of our experiments measured the performance of a single dissemination group consisting of many processes evenly distributed across the seven sites pictured in Figure 5a. The total number of processes acting as receivers was varied between five and thirty processes. The five process case used only five domains while all other cases used seven domains. In each experiment, a sending process created a dissemination group, waited for the receiving processes to join the group and organize their domains into a control tree. Multiple tree configurations are possible depending on when, and in what order, domain managers join the tree. However to ensure consistency across tests, we held the tree configuration constant across all tests (see Figure 5b). After all receivers joined the group, the sender disseminated a data file to the group, and then waited for the final GOT_IT message from all receivers. The values reported for each test are averaged over at least five runs taken during weekdays at roughly the same time so that the observed Internet traffic conditions remain similar across tests.

To gauge the scalability of the protocol, we monitored the changes in processing load at the senders, receivers, and managers. To measure the effective throughput, we measured the changes in end-to-end delay as perceived by the sender. Both processing load and end-to-end delay were recorded under a variety of workloads. In the first set of tests, the sender transmitted a 30 Kbyte file to a varying number of receivers. The dissemination was considered complete when all the receivers correctly receive the entire file. In the second set of tests, the number of processes was fixed at 30 and we incrementally increased the file size from 3K to 30 Kbytes. The end-to-end delay is measured as the time between the beginning of the file transfer and the time at which the last group member's final GOT_IT message is received.

To measure the processing load, we counted the total

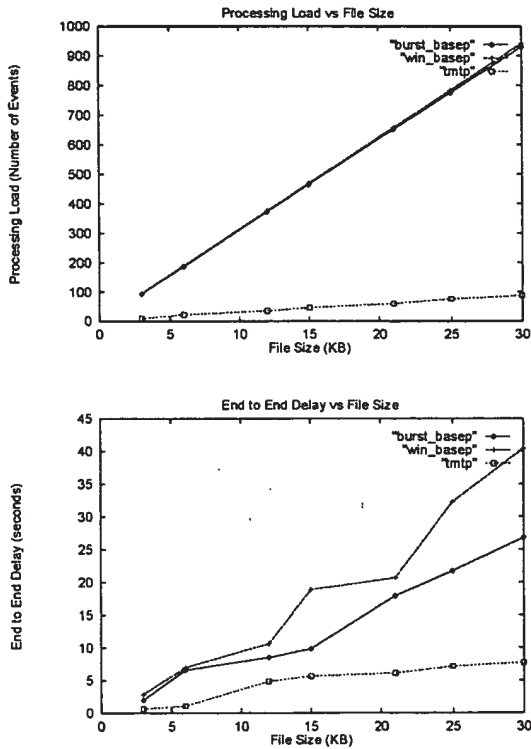


Figure 6: (a) Effect of the amount of data transmitted on the processing load. (b) Effect of the amount of data transmitted on the end-to-end delay. Figure b shows the time for the file transfer to complete at all the receivers. All measurements were taken with a dissemination group of size 30.

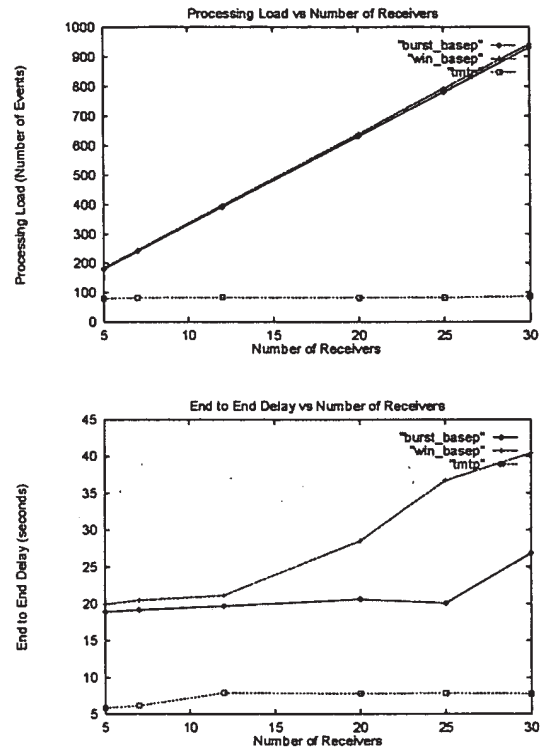


Figure 7: (a) Impact of group size (no. of receivers) on the processing load. (b) Impact of group size (no. of receivers) on the end-to-end delay. Figure b shows the time for the file transfer to complete at all the receivers. All measurements were taken for a dissemination of a 30 KB file.

number of events at the sender that contribute to the processing load. Similarly, we recorded the number of events at each domain manager. Figure 6 only shows the number of events processed at the sender. However, the balanced nature of our control tree meant the event processing load was spread equally among the sender and all domain managers. Consequently, the number of events processed at each domain manager is approximately the same as the number of events processed at the sender. Variations occurred based on the number of NACKs received.

Figures 6 and 7 show the results for each of the experiments performed. From these results we draw the following observations:

Impact of the Data Size

Figures 6a and 6b show how the file size affects the processing load and end-to-end delay. As the file size increases, the number of packets transmitted increases, thereby increasing the number of events (such as ACK/NACK processing or timer events) that affect the processing load at the sender (or a domain manager). Similarly, end-to-end delay is likely to increase due to time needed to deliver all the packets and due to increased probability of packet loss.

As the plots show, both the versions of the BASEP benchmark protocol show a significant increase in the processing load at the sender and the end-to-end delay. Note that the delay for WIN_BASEP (with flow control) is actually higher than BURST_BASEP (no flow control). This occurs because the WIN_BASEP sender expects acknowledgments from all its receivers before advancing the flow control window.

In the case of TMTP, the processing load shows only a small increase because the work is distributed among many nodes in the control tree. Consequently, the sender does not have to process acknowledgments or retransmission requests from all the receivers. TMTP's end-to-end delay is substantially lower than that of the BASEP protocols for all file sizes. Although all three protocols experience an increase in end-to-end delay resulting from larger data transmissions, packet losses, and retransmissions, TMTP's end-to-end delay rises at a significantly lower rate than that of the BASEP protocols. This occurs because error recovery in TMTP proceeds concurrently in different parts of the control tree rather than sequentially as in the BASEP cases.

Impact of the Group Size

Figures 7a and 7b show how the number of receivers (group size) affects the processing load and end-to-end delay.

Again, as the plots show, two versions of BASEP protocol show sharp increases in processing load with increase in number of receivers because the sender solely shoulders the responsibility for processing acknowledgments and retransmission requests (or timeouts) from each receiver. In the case of TMTP, the processing load

at the sender (and each domain manager) is limited by the maximum number of immediate children in the control tree and, therefore, shows almost no increase as the number of receivers is increased. This results from the fact that the number of domains remains at seven for more than seven receivers. An increase in the number of domains participating in the dissemination group would cause a slight load increase on domain managers who adopt the new children.

Figure 7a shows that the end-to-end delay of both BASEP protocols is significantly higher than that of TMTP. The primary reason for this difference stems from TMTP's receiver-initiated capabilities that respond to and correct errors quickly. In contrast, the BASEP protocols will not correct an error until a retransmission timeout occurs.

In the case of TMTP end-to-end delays increase gradually because error recovery proceeds concurrently and independently in different parts of the control tree as explained earlier. Figure 7b shows that the end-to-end delay stabilizes to almost a constant value beyond a point. That is, to a small extent, an artifact of our tests in which we did not add any new domains to the control tree, but rather only added new processes to the existing tree. However, in other experiments involving varying number of domains, we have observed a similar trend of gradual increase in end-to-end delays with increasing number of receivers at additional domains.

RELATED WORK

A considerable amount of work has been reported in the literature regarding reliable multicast [13, 5, 3, 18, 12, 1, 4, 19, 15, 8, 11, 16]. Most of the earlier approaches achieve reliable delivery using a *sender-initiated* approach which is not suitable for large-scale, delay-sensitive, reliable dissemination.

Pingali and others[18] recently analyzed and compared both sender- and receiver-initiated approaches to demonstrate the limitations of the sender-initiated approach for large-scale dissemination. Our work is also motivated by similar observations, but combines the elements of both the approaches to achieve fast, local error recovery.

The reliable multicast protocol used in LBL's whiteboard tool (*wb*) [15, 8] and the log-based reliable multicast protocol [11] are two recent examples of the receiver-initiated approach for reliable delivery. Unlike TMTP, these protocols do not combine sender-initiated with receiver-initiated approaches and differ significantly in flow control mechanisms and buffering mechanisms. Our work is related to the *wb* work in that the *wb* protocol also uses a *NACKs with NACK suppression* mechanism. The *wb* protocol reduces state management overhead and achieves high degree of fault tolerance by relying solely on the receiver to recover from a packet loss. However, the protocol incurs the overhead of global (sometimes redundant) multicasts; a receiver

multicasts a *repair request* to the entire group and one or more receivers in the group who have missing data (irrespective of their proximity to the complaining receiver) will multicast the missing packet(s) to the entire group even though the loss (or congestion) is restricted to a small region of the group topology. TMTP restricts the scope of multicast NACKs and retransmissions to the local domain to avoid generating redundant multicast transmissions over a wider region. Similar to TMTP, receivers using the wb protocol delay their NACKs to suppress duplicate NACKs in case another receiver multicasts a NACK. However, in the wb protocol, each receiver delays its NACK (and the response) by a random amount that depends on the RTT to the original sender. This can result in higher latency in recovering from packet losses. TMTP, on the other hand, uses localized recovery and, thus, the amount of random delay is bounded by the largest RTT between the local domain manager and one of the receivers in the domain. In addition, TMTP allows recovery from different errors to proceed concurrently in different domains to allow faster and efficient recovery.

Cheriton et. al.[8] have recently proposed a collection of strategies (called log-based receiver-reliable multicast or LRBM) for achieving large-scale, reliable multicast delivery. Some elements of LRBM are similar to TMTP's mechanisms to some extent. LRBM uses a hierarchy of logging servers with a primary log server responsible for sending positive acknowledgments to the multicast source. The primary log server stores the packets as long as an application desires and the receivers must recover from errors by contacting a logging server. A secondary server at each site may log received packets and satisfy local retransmission requests to reduce load on the primary server. Deployment of LRBM in the Internet is necessary to evaluate its performance in achieving reliable delivery in a wide area network environment.

Recently Paul et. al. [16] have proposed and are examining three multicast alternatives with features similar to those of TMTP. In contrast to these protocols, TMTP uses a multi-level hierarchical control tree and a dynamic group management protocol, as opposed to a static two-level hierarchy, to evenly distribute the protocol processing load and allow finer grained independent and concurrent error recovery. TMTP targets a best-effort multicast system such as IP multicast rather than an ATM-like network with allocated resources. TMTP imposes no additional load on network-level routers and requires no modification to the network-level routers, but yet incorporates both local retransmissions and combined acknowledgments. Furthermore, TMTP employs receiver-initiated recovery techniques (*restricted negative acknowledgments with nack suppression* combined with periodic positive acknowledgments) and a unique flow control mechanism that can provide quick recovery from transient congestion and lost acknowledg-

ments.

CONCLUSION

Based on our experimental results, we believe that TMTP can scale well to provide reliable delivery on a large scale without sacrificing end-to-end latency. Under TMTP, the network processing load increases very gradually, indicating that the protocol will scale well as the number of receivers increases. Moreover, TMTP provides significantly better application-level throughput because of the concurrency resulting from local retransmissions as shown by the end-to-end measurements.

References

- [1] Ken Birman and Thomas Joseph. Reliable communication in the presence of failures. *ACM Transactions on Computer Systems*, 5(1):47-76, Feb 1987.
- [2] S. Casner and S. Deering. First IETF Internet Audiocast. *ACM Computer Communication Review*, 22(3):92-97, July 1992.
- [3] J. Chang and N. Maxemchuck. Reliable Broadcast Protocols. *ACM Transactions on Computer Systems*, 2(3):251-273, August 1984.
- [4] David. R. Cheriton and W. Zwaenepoel. Distributed process groups in the V kernel. *ACM Transactions on Computer Systems*, 3(2):77-107, May 1985.
- [5] J. Crowcroft and K. Paliwoda. A Multicast Transport Protocol. In *Proceedings of ACM SIGCOMM '88*, pages 247-256, August 1988.
- [6] Stephen E. Deering and David R. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, 8(2):85-110, May 1990.
- [7] Prasun Dewan. A Guide to Suite: Version 1.0. Technical Report SERC-TR-60-P, Software Engineering Research Center, Purdue University, West Lafayette, IN, February 1990.
- [8] S. Floyd, V. Jacobsen, S. McCanne, C-G Liu, and L. Zhang. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. In *sigcomm95*, 1995. to appear.
- [9] I. Gopal and J. Jaffe. Point-to-multipoint Communication over Broadcast Links. *IEEE Transactions on Communications*, 32, September 1984.
- [10] James Griffioen and Rajendra Yavatkar. Clique: A Toolkit for Group Communication using IP Multicast. In *Proceedings of the Workshop on Services in Distributed and Networked Environments*, June 1994.

- [11] H.W. Holbrook, S.K. Singhal, and D.R. Cheriton. Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation. In *sigcomm95*, 1995. to appear.
- [12] Van Jacobson. *SD: Session Directory*. Lawrence Berkeley Laboratory, March 1993.
- [13] M Frans Kaashoek, A.S. Tanenbaum, S.F. Hummel, and H.E. Bal. An Efficient Reliable Broadcast Protocol. *ACM Operating Systems Review*, 23(4), October 1989.
- [14] Amit Mathur and Atul Prakash. Protocols for integrated audio and shared windows in collaborative systems. In *Proceedings of ACM Multimedia '94*, October 1994.
- [15] Steven McCanne. A Distributed Whiteboard for Network Conferencing. Technical report, Real Time Systems Group, Lawrence Berkeley Laboratory, Berkeley, CA, September 1992. unpublished report.
- [16] S. Paul, K. Sabnani, and D. Kristol. Multicast Transport Protocols for High Speed Networks. In *IEEE Int. Conf. on Network Protocols*, 1994 Oct.
- [17] L. Peterson, N. Buchholz, and R.D. Schlichting. Preserving and using context information in interprocess communication. *ACM Transactions on Computer Systems*, 7(3):217-246, August 1989.
- [18] Sridhar Pingali, Don Towsley, and James F. Kurose. A comparison of sender-initiated and receiver-initiated reliable multicast protocols. In *Proceedings of ACM SIGMETRICS '94*, volume 14, pages 221-230, 1994.
- [19] S. Ramakrishnan and B.N. Jain. A Negative Acknowledgement Protocol with Periodic Polling Protocol for Multicast over Lans. In *Proceedings of IEEE INFOCOMM '87*, pages 502-511, March-April 1987.

Routing Strategies for Fast Networks

Yossi Azar
DEC - Systems Research Center
130 Lytton Ave.
Palo-Alto, CA 94301

Joseph Naor
Department of Computer Science
Technion
Haifa 32000, Israel

Raphael Rom
Sun Microsystems
Mountain View, CA
and
Technion, Haifa Israel

Abstract

Modern fast packet switching networks forced to rethink the routing schemes that are used in more traditional networks. The reexamination is necessitated because in these fast networks switches on the message's route can afford to make only minimal and simple operation. For example, examining a table of a size proportional to the network size is out of the question.

In this paper we examine routing strategies for such networks based on flooding and predefined routes. Our concern is to get both efficient routing and an even (balanced) use of network resources. We present efficient algorithms for assigning weights to edges in a controlled flooding scheme but show that the flooding scheme is not likely to yield a balanced use of the resources. We then present efficient algorithms for choosing routes along: (i) breadth-first search trees; and (ii) shortest paths. We show that in both cases a balanced use of network resources can be guaranteed.

1 Introduction

Traditional computer networks were designed on the premise of fast processing capability and relatively slow communications channels. This manifested itself by burdening network nodes with frequent network management decisions such as flow control and routing [1, 2, 3]. In a typical packet-switching network the routing decision at every node is based on the packet's destination and on routing information stored locally. This routing information may become quite voluminous, increasing the per-packet processing time.

Changes in technology, applications, and network sizes have forced to rethink these strategies. Modern fast packet switching networks [4, 5] relegate most of the routing com-

This work was done while the author was in the department of Computer Science, Stanford University, CA 94305-2140, and was supported by a Weizmann Fellowship and contract ONR N00014-88-K-0166

Most of this work was done while the author was a postdoctoral fellow at the Computer Science Department, Stanford University and supported by contract ONR N00014-88-K-0166.

putation to the end-nodes leaving all but the minimal computation to the intermediate nodes once the packet is on its way. This paper considers and compares several routing strategies for such fast networks. We assume that links are of high capacity so that message length is of no great concern. Computation capability in intermediate nodes is assumed limited so that all decisions made enroute should be simple and could not rely, for example, on generating random numbers or on tables that grow with the size of the network.

The first to encounter similar problems were the designers of parallel computers. Their solution, in the form of an interconnection network, typically derives the route directly from the destination address [6]. This approach, however, is limited to specific types of network topology and a structured layout which cannot be assumed for a general network. Furthermore, deriving the route from the address in general conflicts with alternate routing approach.

Flow-based techniques, used in many existing networks [7, 8], are also inadequate for our environment. These routing strategies are destination based (typically require a table entry per destination) but more importantly, result in bifurcated routing necessitating intermediate nodes to generate random numbers.

Two strategies are considered in this paper - controlled flooding and fixed routing. Flooding is a routing strategy that guarantees fast arrivals with minimal enroute computation at the expense of excessive bandwidth use. The scheme we use here, first proposed in [9], limits the extent to which a message is flooded through the network. Essentially, each link is assigned a cost for traversing it, thereby limiting the extent of the flood. The problem is to assign the link costs so as to achieve best performance. We show two methods of computing optimal weights that are drawn from a polynomial range (as opposed to the exponential range proposed in [9]). However, we do show that the assignment does not result in a routing scheme that uses network resources in a balanced way.

In the fixed routing scheme the route of the message is determined at the source node and is included in the message. No further routing decision are done enroute. The problem is therefore to find a set of routes, one for each pair of nodes, such that all the network's links will be used in a

2A.4.1

balanced manner. We propose two methods to achieve this. In the first one, we force the messages to be routed along a (topological) breadth first search tree. The problem can be formulated as finding a set of rooted BFS trees such that the maximum load on a link is minimized. Notice that no link in the network remains unused. We provide polynomial algorithms to generate such a set of balanced routes.

In the second method, routing is done along paths that do not necessarily form trees. One of the shortest paths between every pair of nodes is designated as the path along which these two nodes exchange messages. We prove that a set of paths can be chosen that yields a balanced load. We define the notion of a balanced load with respect to randomized choices of paths, i.e., every pair chooses uniformly in random one of the shortest paths connecting them. We first show that with high probability the load on every edge will be close to its expected value. We then show how to construct deterministically in polynomial time such a set of balanced paths via the method of conditional probabilities.

2 Routing Along Trees

In this section we consider the option of routing along fixed BFS trees. Routing along trees can be viewed in two ways: (1) the tree rooted at a node specifies the routes used by the root when acting as a source of messages, or (2) the tree rooted at the node specifies the routes used by the other nodes with the root serving as the destination. From a design standpoint these are identical and in both we strive to balance the load on the links as much as possible.

As before we consider the network as a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$. In addition we single out a vertex r called the root. The graph is divided into layers relative to root r by conducting a breadth-first search on G from r (i.e., we construct a tree of the shortest paths from r to all the other nodes in the graph). In this division, layer i , $0 \leq i \leq n-1$, contains all the vertices whose distance from r is i . The corresponding resultant tree is denoted T_r . Note that for a given G and r , the layers are defined uniquely but the BFS tree is not. Also note that given a BFS tree, the edges of the original graph connect vertices only from adjacent layers or in the same layer.

Let $v \in V$ be some vertex in layer i (for some $1 \leq i \leq n-1$). Define d_v^i as the number of neighbors of v at layer $i-1$ in graph G rooted at r ; by convention $d_v^0 = 0$. The following proposition establishes relations which we shall use later on.

Proposition 2.1 For any graph G

1. The number of different BFS trees from root r is $\prod_{v \in G-r} d_v^1$
2. For any r , $\sum_{v \in V} d_v^1 \leq m$

Proof:

1. All the BFS trees can be constructed by having each vertex $v \in G-r$ choose independently a parent out of its neighbors in the previous layer, and each such construction corresponds to a legal and different BFS tree rooted at r . Hence the claim follows.
2. Each edge contributes unity to the sum if its two endpoint vertices are not in the same layer, and zero otherwise. Thus, this sum is exactly equal to the number of edges connecting vertices of different (and therefore adjacent) layers.

□

2.1 Homogeneous Sources

In this section we assume that each node sends (or receives) the same amount of data to every other node, and our aim, as we indicated, is to use the resources evenly. To that end we define the load on an edge as follows. Assume that for every vertex r in the graph we are given a single BFS tree rooted at that vertex (thus determining node's r routing). The load on an edge is defined (relative to this set of trees) as the number of trees which contain this edge. Formally, we are given a set $\{T_r\}_{r \in V}$ containing a single T_r for every $r \in V$ and we define the load of an edge as

$$l(e) = |\{r \in V | e \in T_r\}|.$$

Note that $l(e) \leq n$ and $\sum_{e \in E} l(e) = n(n-1)$, since there are n BFS trees with $n-1$ edges in each and each edge in a BFS tree contributes a unity to the sum. The capacity of an edge e , denoted $c(e)$, is defined as the maximum number of BFS trees that may contain it.

Our goal is to choose a set $\{T_r\}_{r \in V}$ such that the maximum load of the edges is minimized. We do this by solving a more general problem in which edges have limited capacities that are not necessarily equal. Assume that we are given the edge capacity $c(e)$ for each edge $e \in E$. We are seeking a feasible solution that is, a set $\{T_r\}_{r \in V}$ such that $l(e) \leq c(e)$ for all e . A solution for the capacitated problem can be easily used to solve the problem of minimizing the maximum load (in the uncapacitated problem). We just let $c(e) = c$ for all e and perform a binary search on $1 \leq c \leq n$, thereby increasing the complexity by a factor of $\log n$.

In order to solve the capacitated problem we define the following bipartite graph $H = (A \cup B, F)$. Side A consists of $n(n-1)$ vertices denoted by pairs (r, v) for all $v, r \in V, v \neq r$ (this pair will subsequently be interpreted as a root r and some vertex v in G). Side B consists of m vertices, each corresponding to (and denoted by) an edge e for all $e \in E$. Each vertex $(r, v) \in A$ is connected to a vertex $e \in B$ iff $\exists T_r$ (i.e., a tree rooted at r) in which $e \in E$ connects v to a vertex from the previous level.

Note that the degree of vertex (r, v) is d_v^r as per the definition of d_v^r . Also, from proposition 2.1 $|F| = \sum_{v,r} d_v^r \leq \sum_r m = nm$.

The key observation is that in order to solve our problem we need to find $n(n-1)$ edges in the graph H such that the degree of each vertex in A is exactly 1 (matching), and the degree of vertex $e \in B$ is at most $c(e)$. These edges define the n BFS trees in G . Specifically, the edges of T_r are the vertices in B which are adjacent to the vertices (r, v) for all $v \in G - r$. We present two algorithms for finding these trees.

Algorithm 1. Each vertex $e \in B$ with all its incident edges is duplicated $c(e)$ times, generating an "exploded" graph. Now, it is clear that solving the problem is equivalent to finding a perfect matching for side A into side B . The number of vertices in the exploded graph is $n(n-1) + \sum_e c(e) < n^2 + mn$ and the number of edges is at most $n|F| \leq n^2n$. The complexity of computing a maximum matching in a bipartite graph is $O(|E|\sqrt{|V|}) = O(m^{3/2}n^{5/2})$ [11].

The latter complexity can be improved by the next algorithm.

Algorithm 2. Add to the graph $H = (A \cup B, F)$ a source node s and sink t . Add directed edges from s to all the vertices in A , each with capacity 1, and directed edges from each vertex $e \in B$ to t , each with capacity $c(e)$. Finally, direct all the edges from A to B and assign each the capacity 1 (any capacity greater than 1 will also do).

Consider an integer flow problem with source s and destination t obeying the specified capacities. It is clear that any such legal flow starts with some edges from s to A with flow 1. Then, each vertex in A that has an incoming edge with one unit of flow also has one outgoing edge with one unit flow to a vertex in B . Finally, all the flow reaching B continues to t . Thus we conclude that there is a feasible solution to our problem iff the maximum flow between s and t is exactly $n(n-1)$.

We will use Dinic's algorithm for finding the max-flow [12]. A careful analysis of the algorithm for our case yields a better complexity than more recent max-flow algorithms that perform better on general graphs. We first give a short review of Dinic's algorithm. The algorithm has $O(|V|)$ phases; at each phase only augmenting paths of length $i, 1 \leq i \leq |V|$, are considered. The invariant maintained at

phase i is that there are no augmenting paths of length less than i . The complexity of each phase is $O(|E||V|)$ in general graphs and $O(|E|)$ in 0-1 networks.

We first convert our graph into a 0-1 network. Each edge of capacity $c(e)$ is duplicated into $c(e)$ unity capacity edges which yields a 0-1 network. Since $c(e) \leq n$ for every edge e , the total number of new edges is at most nm and thus the number of edges remains $O(nm)$. As mentioned before, the complexity of Dinic's algorithm for 0-1 network is $O(|E||V|)$ which in our case becomes

$$O((n^2 + m)[n^2 + mn + mn]) = O(n^2 \cdot mn) = O(mn^3)$$

In fact, the running time can be reduced to $O(mn^2)$. In our graph, there are no edges between vertices in A and also none between vertices in B , and there will not be such in any of the residual graphs. In fact, the residual graph will always start with s , end with t , have only vertices of A in the other even numbered layers and only vertices of B in the other odd-numbered layers. Moreover, the vertices of A will always have, in any residual graph, at most one incoming edge. Let us run the first $n-1$ phases of Dinic's algorithm (where each phase takes time $O(|F|) = O(nm)$). In phase n there will be at least n layers of A (unless we have already finished), one of them having at most $n(n-1)/n = n-1$ vertices. The incoming edges into this layer of A define a cut separating s from t whose capacity is at most $n-1$. Thus, Dinic's algorithm will terminate after at most additional $n-1$ phases, which gives the desired time bound.

2.2 Heterogeneous Sources

The situation at hand in this section is similar to that of the previous subsection except that we no longer assume homogeneous traffic but rather that each node generates a different amount of traffic. Translated into our model, this results in a problem with weighted trees. Formally, let the relative traffic intensity associated with node r be $w(r)$ (assumed to be an integer). This means that the tree associated with r (where r is the root) has a weight of $w(r)$ and we seek a set of BFS trees $\{T_r\}_{r \in V}$ with load $l(e) \leq c(e)$ for all e , where the load $l(e)$ is defined in the natural way, i.e.,

$$l(e) = \left\{ \sum_r w(r) | e \in T_r \right\}$$

The Capacitated Problem of the previous subsection is the special case of our problem with $w(r) = 1$ for all $r \in V$. While the Capacitated Problem in the homogeneous case has an efficient solution, we prove that in the heterogeneous case this problem is NP-complete (it is clear that the problem belongs to class NP). We base our proof on a reduction from the "knapsack" problem which is known to be NP-complete [13], defined as follows.

2A.4.3

0172

The Knapsack Problem: Given are integers $x_1 \dots x_n$ and s . Are there $u_i \in \{0, 1\}$, $1 \leq i \leq n$, such that $\sum u_i x_i = s$?

The Reduction: Consider a graph whose vertices are $v_1, \dots, v_n, u_1, u_2, t$. Connect v_i to v_j for $1 \leq i \leq n, j = 1, 2$ and connect u_1 and u_2 to t . Let the weight of the sources be $w(v_i) = x_i$ for all i , $w(u_1) = w(u_2) = w(t) = 0$. Finally, let the capacities of the edges be $c(u_1 t) = s, c(u_2 t) = \sum_i x_i - s$, and infinite (or big enough) for all the rest. It is clear that each BFS tree from $v_i, 1 \leq i \leq n$, contains exactly one of the edges $u_1 t$ or $u_2 t$. Since $c(u_1 t) + c(u_2 t) = \sum_i x_i$, there is a solution iff there is a subset of the integers x_i that sums up to s .

Note that it is possible to eliminate the zero weights (and have the proof still hold) by assigning $w(u_1) = w(u_2) = w(t) = 1$ and also adding 2 to the capacities of the edges $u_1 t$ and $u_2 t$.

2.3 Randomized Capacity Bounds

In this section we develop upper bounds on the capacities that are needed for the edges in the Capacitated Problem of the homogeneous case (section 2.1) in order to achieve "good" load balancing. Our reference is a random tree routing scheme in which every node, whenever it needs to send a message, randomly and uniformly chooses a BFS tree in which it is a root, and routes according to this tree. Intuitively, such a routing scheme is likely to achieve a good balancing.

We start by calculating P_r^e - the probability that an edge e participates in a randomly and uniformly chosen BFS tree rooted at r . Let x_r^e be an indicator random variable indicating whether edge e belongs to the BFS tree rooted at r . By our definition

$$l(e) = \sum_{r \in V} x_r^e.$$

Consider an edge $e = (x, y)$. If both x and y are in the same layer (i.e., equidistant from r), then $P_r^e = 0$. Otherwise, they belong to adjacent layers (without loss of generality let x be the vertex that is further away from r), and $P_r^e = \frac{1}{d_x^r}$.

Let $\bar{l}(e)$ be the expected load of e . Clearly $E[x_r^e] = P_r^e$ and also

$$\bar{l}(e) = E \left[\sum_{r \in V} x_r^e \right] = \sum_{r \in V} E[x_r^e] = \sum_{r \in V} P_r^e$$

$$\begin{aligned} \sum_{e \in E} \bar{l}(e) &= \sum_{r \in V} \sum_{e \in E} P_r^e \\ &= \sum_r \sum_{x \neq r} \frac{1}{d_x^r} \cdot d_x^r = \sum_r (n-1) = n(n-1). \end{aligned}$$

Since $\sum_{e \in E} \bar{l}(e) = n(n-1)$ and also $\sum_{e \in E} l(e) = n(n-1)$, we cannot expect to find a set of BFS trees in which

$l(e) \leq \bar{l}(e)$ for every edge e ($\bar{l}(e)$ is not necessarily an integer for instance). However, we can find a set which is almost as good. We show that there always exists a set of BFS trees $\{T_r\}_{r \in V}$ such that the load on any edge satisfies the following:

$$l(e) \leq \bar{l}(e) + 2\sqrt{\bar{l}(e) \log n}.$$

We will prove the claim via the probabilistic method; one can easily find such a set by applying the algorithm from section 2.1 as we are guaranteed that a solution exists.

To prove the bound on the load, we show that for each edge e , the probability that $l(e)$ exceeds the claimed bound is less than $\frac{1}{2m}$. Hence, there is a positive probability that the claim holds for all edges in the network. From Chernoff's bounds it can be shown that for all $\lambda \geq 0$,

$$\text{Prob}[l(e) > (1 + \gamma)\bar{l}(e)] \leq \frac{E[e^{\lambda l(e)}]}{e^{(1+\gamma)\lambda \bar{l}(e)}}$$

and it can be shown [14] that there exists a choice of λ such that

$$\frac{E[e^{\lambda l(e)}]}{e^{(1+\gamma)\lambda \bar{l}(e)}} \leq e^{-\gamma^2 \bar{l}(e)/2}.$$

Assigning $\gamma = 2\sqrt{\frac{\log n}{\bar{l}(e)}}$, results in

$$\text{Prob}[l(e) > \bar{l}(e) + 2\sqrt{\bar{l}(e) \log n}] \leq \frac{1}{n^2} < \frac{1}{2m}$$

which finally yields

$$\text{Prob}[\forall e, l(e) \leq \bar{l}(e) + 2\sqrt{\bar{l}(e) \log n}] > \frac{1}{2}$$

meaning that a solution exists with a high probability.

3 Routing Along Shortest Paths

In this section we consider a different option of routing namely, routing along paths that do not necessarily form trees. One of the shortest paths between every pair of nodes is designated as the path along which these two nodes exchange messages. We prove that a set of paths can be chosen that yields a balanced load.

The proof we present follows the exact same lines of the proof in section 2.3 and we adopt the same notation. Again, our reference for a good load balancing is the random path routing scheme

We first evaluate P_e^{uv} - the probability that an edge e participates in a randomly and uniformly chosen shortest path connecting vertices u and v . (We will denote this event by the indicator variable x_e^{uv}). To compute this probability, we must count the shortest paths connecting u and v that contain edge e . Let $M_p(u, v)$ denote the number of paths of

length p between the vertices u and v . The number of shortest paths between u and v can be computed in polynomial time by the following recursive formula. Let the vertices adjacent to u be a_1, \dots, a_d and let p be the length of the shortest path from u to v , then

$$M_p(u, v) = \sum_{i=1}^d M_{p-1}(a_i, v).$$

We consider a pair of nodes u and v and an edge $e = (x, y)$ (assume without loss of generality that vertex x is closer to u than vertex y). Denote by p_{uv} the distance between the vertices u and v , by p_{ux} the distance between u to x , and by p_{yv} the distance between v and y . Define $p' = p_{uv} - p_{ux} - 1$. If $p_{yv} > p'$, then $P_e^{uv} = 0$; otherwise,

$$P_e^{uv} = \frac{M_{p_{ux}}(u, x) \cdot M_{p_{yv}}(y, v)}{M_{p_{uv}}(u, v)}.$$

Similar to the derivation in section 2.3 the expected load on an edge e is $\bar{l}(e) = \sum_{u,v \in V} P_e^{uv}$ and thus we cannot expect to find a set of shortest paths in which $l(e) \leq \bar{l}(e)$ for every edge e . However, again, we can find a set which is almost as good, namely, a set of shortest paths such that the load on any edge satisfies

$$l(e) \leq \bar{l}(e) + 2\sqrt{\bar{l}(e) \log n}.$$

An edge whose load does not satisfy the above condition is called an *overloaded* edge. If there are no overloaded edges, then the set of paths is called a *good set*. We will prove that a good set of paths exists via the probabilistic method and then show how to find such a set of paths deterministically.

Let every pair of vertices choose its path uniformly in random (among the shortest paths between them). We show that with high probability, the set of paths chosen is good. The random variable $l(e)$ is a sum of $\binom{n}{2}$ indicator variables x_e^{uv} . These variables are independent because each pair of vertices chooses its path independently of the other pairs. If we show that the probability that edge e is overloaded is less than $\frac{1}{2m}$, then with high probability the claim holds for all edges in the network. As stated in Section 2.3, it can be shown that for all $\lambda \geq 0$,

$$\text{Prob}[l(e) > (1 + \gamma)\bar{l}(e)] \leq \frac{E[e^{\lambda l(e)}]}{e^{(1+\gamma)\lambda \bar{l}(e)}}$$

furthermore, there exists a choice of λ [14] such that

$$\frac{E[e^{\lambda l(e)}]}{e^{(1+\gamma)\lambda \bar{l}(e)}} \leq e^{-\gamma^2 \bar{l}(e)/2}$$

Similar to Section 2.3, assigning $\gamma = 2\sqrt{\frac{\log n}{\bar{l}(e)}}$, results in

$$\text{Prob}[l(e) > \bar{l}(e) + 2\sqrt{\bar{l}(e) \log n}] \leq \frac{1}{n^2} < \frac{1}{2m}$$

which finally yields

$$\text{Prob}[\forall e, l(e) \leq \bar{l}(e) + 2\sqrt{\bar{l}(e) \log n}] > \frac{1}{2}$$

as was claimed.

Having established that there exists a good set of paths we now show how to find this good set deterministically in polynomial time by the *method of conditional probabilities* [15],[16]. This method was introduced by Spencer [15] with the intention of converting probabilistic proofs of existence of combinatorial structures into efficient deterministic algorithms for actually constructing these structures. The idea is to perform a binary search of the sample space associated with the random variables so as to find a good set. At each step of the binary search, the current sample space is split into two halves and the conditional probability of obtaining a good set is computed for each half. The search is then restricted to the half having a higher conditional probability. The search terminates when only one sample point remains in the subspace, which must belong to a good set.

To apply this method to our case for finding a good set of paths, we will consider the indicator variables one-by-one. In a typical step of the algorithm, the value of some of the indicator variables has already been set, one variable is currently being considered, and the rest are chosen in random. (By choosing in random we mean that, for the pair of vertices which is now being considered, the remainder of the path is chosen uniformly in random.) At each step we will compute the (conditional) probability of finding a good set if the variable considered is set to 0 and if it is set to 1.

We denote by P_j the probability of finding a *bad* set of paths after the variable considered at step j has already been assigned a value and by P_j^i the probability of obtaining a bad set of paths by assigning the value i , for $i = 0, 1$, to the variable considered at step j . Initially, it follows from the existence proof that the probability of choosing a good set of paths is positive; we inductively maintain that $P_j < 1$ for $j \geq 1$, and hence, either $P_j^0 < 1$ or $P_j^1 < 1$.

For the sake of simplicity, assume the following on the order in which the variables are considered:

- For a pair of vertices u and v , for all edges e , the variables x_e^{uv} are considered consecutively.
- For a pair of vertices u and v , the edges are considered according to their distance from u . (Ties are broken arbitrarily).

For example, suppose that we are considering the variable x_e^{uv} where $e = (a, b)$ and assume that vertex a is closer to u than b . Notice that by assigning a value to x_e^{uv} ,

- The probability P_j^{uv} may change for edges f for which x_f^{uv} has not been determined yet. (These changes

2A.4.5