A Gossip Algorithm for Bus Networks with Buses of Limited Length

Satoshi Fujita[1], Christian Laforest[2], and Stephane Perennes[3]

[1] Faculty of Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima, 739, Japan
[2] LRI, Université de Paris-Sud, Bât. 490, 91405 Orsay Cedex, France
[3] Project SLOOP I3S-CNRS/INRIA/Université de Nice–Sophia Antipolis, 930 route des Colles, F–06903 Sophia-Antipolis Cedex, France

Abstract. In this paper we consider the **gossiping problem** in bus networks in which any subset of vertices (i.e., nodes) of size $l' \leq l$, where l is a given constant greater than or equal to 2, is connected by a shared bus of length l'. We call such a network **complete bus network**, since it is a generalization of complete point-to-point network K_n. The proposed algorithm finishes a gossip in complete bus networks in a short time under a feasible and practical communication model. For example, it finishes a gossip only in $1.137 \log_2 n + O(1)$ steps for $l = 3$.

1 Introduction

A **bus network** with vertex set V is a communication network in which vertices in V (representing nodes of a physical network) are connected by a set of shared buses. The number of vertices connected to a bus is called the **length** of the bus. In recent years, bus networks have received considerable attention from many researchers as a versatile communication topology for parallel processors, since it is inherently more powerful than usual point-to-point networks. In spite of a lot of important works in this area, there has not been found a general bus network topology which is recognized as the "best" one. A criterion of the goodness of communication topology is the performance for executing several communication patterns which are commonly used in parallel processing. In this paper, we evaluate the performance of bus networks by focusing on a communication pattern known as a **gossiping** in the literature. Note that it requires a lot of messages to be transmitted for finishing a gossiping.

In this paper, the communication in bus networks is assumed to proceed as follows [8]:

- Communications proceed step by step, and the transmission of a message along a bus takes one time unit (called a **step**) regardless of the length of the transmitted message and the length of the bus. (That is, we are interested here in the total number of steps taken by a gossiping.)
- A vertex can either send or receive a message to/from at most one bus at a given step, even if the vertex is connected to several buses.

- Given a bus, only one vertex connected with the bus can send a message through it at a given step. In addition, all vertices connected with the bus can simultaneously receive the transmitted message (i.e., each bus is accessed in CREW manner).

During a step, several message transmissions can occur simultaneously as long as all of the above three constraints are satisfied. Note that if all buses have length two, it coincides with the communication in point-to-point networks (i.e., graphs) under the single-port, half-duplex model, so-called "telegraph model" (surveys on communications in graphs can be found in [9, 12]).

In this paper, we focus on a special class of bus networks defined as follows.

Definition 1. Complete bus network $cbn(n,l)$, where $2 \le l \le n$, is a bus network consisting of n vertices and each bus is of length at most l. For any subset of vertices of size $l' \le l$, there is a bus of length l' which connects all vertices in the subset.

For each $u \in V$, let $m(u)$ be the piece of information initially held by vertex u. The **gossiping problem** is the problem of disseminating $m(u)$ to all the other vertices in V, for all $u \in V$. This problem in bus networks has been studied in the literature [8, 10, 11, 13, 14]; e.g., Fujita and Yamashita [10, 11] considered the problem in *mesh-bus networks* in which all vertices are arranged on a two-dimensional array and vertices in each row and vertices in each column, respectively, are connected with a bus; Hily and Sotteau [13, 14] extended the result on two-dimensional arrays to the case of d-dimensional arrays for $d \ge 3$; and Fraigniaud and Laforest [8] studied bus networks of minimal size in which a gossip finishes in a minimum time. Some other results have been obtained in [1, 2, 6, 7] concerning the design of particular bus networks and the broadcasting operation in them.

Let $g_l(n)$ denote the minimum number of steps necessary to finish a gossip in $cbn(n,l)$ under our model[4]. In [8], it is shown that $g_n(n) \ge \lceil \log_2 n \rceil + 1$ holds for any $n \ge 2$. Since $g_l(n) \ge g_n(n)$ obviously holds (indeed, one may use only buses of length at most l), we have the following theorem.

Theorem 2. (*Lower Bound*) *For all* $2 \le l \le n$, $g_l(n) \ge \lceil \log_2 n \rceil + 1$.

The upper bound we will prove on $g_l(n)$ is related to some numbers τ_l that are defined as follows:

Definition 3. τ_l is the root of greatest modulus of the polynomial $X^l - X^{l-1} - \cdots - X - 1$. It has been proved that τ_l is a real and belongs to $[1, 2]$ (see [3] and [15], section 5.4.2, ex. 7).

When $l = 2$, buses of length 2 are indeed "edges" and our model is equivalent to the model so-called *half-duplex model* for point-to-point networks. In that case, a tight bound has been derived:

[4] $g_l(n)$ represents a fundamental lower bound on the number of steps for finishing a gossip in bus networks with n vertices and buses of length at most l under our model.

Theorem 4. $g_2(n) = \frac{1}{\log_2(\tau_2)} \log_2 n + O(1)$.

A proof of the upper bound was given by Entringer and Slater [4], and the proof of the lower bound was independently given by Krumme *et al.* [16], Sunderam and Winkler [18], Even and Monien [5], and Labahn and Warnke [17].

The objective of our paper is to find an exact value of $g_l(n)$ in terms of n for any $l \geq 2$. In Section 2, we propose a gossip algorithm for $cbn(n, l)$ to give an upper bound on $g_l(n)$. It is shown that this upper bound is very close to the lower bound given in Theorem 2, and we conjecture that the bound is optimal within a constant number of steps. Section 3 concludes the paper with future directions of this research.

2 An Upper Bound on $g_l(n)$

The time of our algorithm will be denoted $T_l(n)$.

2.1 When $n = 0 \pmod{l}$

For simplicity we describe first an algorithm valid when $n = ql$ for some integer q. The set of vertices in those networks can be viewed as a rectangle of q columns and l lines. Vertex of line i and column j will be consequently labeled (i, j) with $i \in Z_l$ and $j \in Z_q$. The set of vertices in line i will be denoted L_i $(= \{(i,j) \mid j \in Z_q\})$; in the same way, the set of vertices in column j will be denoted C_j $(= \{(i,j) \mid i \in Z_l\})$.

The algorithm runs in two phases.

Phase 1. A description of Phase 1 is given as follows.

Description of Phase 1 *In parallel, in each column C_j for $j \in Z_q$, perform a gossip.*

Phase 1 takes exactly $\lceil \log_2 l \rceil + 1$ steps according to the results obtained in [8].

Remark. Note that at the end of Phase 1, any vertex (i, j) in column C_j knows the whole information originally contained in column j; i.e., $m_j = \bigcup_{u \in C_j} m(u)$. For simplicity of notations we will say that *a vertex v knows the information of columns $[j, j']$, if v knows the information originally contained in the consecutives columns $j, j+1, j+2, \cdots, j'$* (remind that addition is performed modulo q).

Phase 2 The second phase is executed only when $q \geq 2$. We can refer it abusively as a "gossip between columns". In what follows, we shall denote by "time t" the time after the t^{th} step of Phase 2, that is phase 2 starts at time 0. At the beginning of Phase 2, any vertex knows the information of at least one column. In particular, vertex (i, j) knows the information of columns $[j, j]$.

We will describe an algorithm such that at time t each vertex (i,j) of line L_i, knows the information of the $F_i(t)$ consecutive columns $[j, j+F_i(t)-1]$, where $F_i(t)$ depends only on i and t. This situation can be represented by the vector:

$$F(t) = \begin{pmatrix} F_0(t) \\ F_1(t) \\ \cdots \\ \cdots \\ F_{l-1}(t) \end{pmatrix}$$

where $F_i(t)$ will be called the **amount of information** known by line L_i.

Description of Phase 2 At time 0, to perform step 1 we choose a line L_{i_0} and any vertex (i_0, j) of this line sends its information to the vertices of the set $\{(i, j-1), i \neq i_0\}$. So at time 1, each vertex (i,j) of a line $L_i (i \neq i_0)$ will know the information of columns $[j, j+1]$. A vertex (i_0, j) will still know only the information of column j. At time 1, we choose a line $L_{i_1} (i_1 \neq i_0)$; then during step 2, a vertex (i_1, j) sends its information to $\{(i, j - F_i(1)), i \neq i_1\}$. Explicitly (i_1, j) sends its information to $(i_0, j-1)$ and to $(i, j-2)$ for $i \neq i_0, i_1$.

Let $i_t \in Z_l$ be such that $F_{i_t}(t)$ is maximum, that is $F_{i_t}(t) = max_{i \in Z_l}(F_i(t))$. Then, at step $t+1$ each vertex (i_t, j) sends its information to the vertices $\{(i, j - F_i(t)), i \neq i_t\}$.

For example (see table 1), in the case $l = 4$, at time 3 if we have chosen $i_0 = 0, i_1 = 1, i_2 = 2$, then the maximum of $F_i(3)$ is 8 which is attained for line 3; so the senders are in line 3. During step 4 a vertex $(3,j)$ sends its information (namely that of columns $[j, j+7]$) to $\{(0, j-7), (1, j-6), (2, j-4)\}$.

At time $t+1$ a vertex $(i,j), i \neq i_t$ will know all the information of columns $[j, j+F_i(t)-1]$ (known at time t) plus that of columns $[j+F_i(t), j+F_i(t)+F_{i_t}(t)-1]$ that is of columns $[j, j+F_i(t)+F_{i_t}(t)-1]$. Vertices of line i_t will receive no new information. Consequently we obtain the following recurrence relations:

- $\forall i$, $F_i(0) = 1$;
- $F_{i_t}(t+1) = F_{i_t}(t)$; and
- for $i \neq i_t$, $F_i(t+1) = F_{i_t}(t) + F_i(t)$.

As during phase 2, at time t, any vertex (i,j) knows the information of $F_i(t)$ consecutive columns; phase 2 completes the gossip at the first time T for which $\forall i, F_i(T) \geq q$. Note that according to the previous description the algorithm can be implemented under our model in $cbn(n,l)$.

Lemma 5. *The time of Phase 2 is at most* $\lceil \log_{\tau_l}(q) \rceil + 1$.

Proof. Let $G(t)$ be the vector of values $F_i(t)$ which are sorted in a decreasing order. We will denote $G_i(t)$ the i^{th} coordinate of $G(t)$, as example $max_{i \in Z_l}(F_i(t)) =$

Table 1. The first values of $F_i(t)$ for $l = 4$.

t	0	1	2	3	4	5	6	7	8
$F_0(t)$	1	1	3	7	15	15	44	100	208
$F_1(t)$	1	2	2	6	14	29	29	85	193
$F_2(t)$	1	2	4	4	12	27	56	56	164
$F_3(t)$	1	2	4	8	8	23	52	108	108

$F_{i_t}(t) = G_0(t)$. Then, $G(t)$ clearly satisfies:

$$G(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \cdots \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad G(t+1) = \begin{pmatrix} G_0(t) + G_1(t) \\ G_0(t) + G_2(t) \\ G_0(t) + G_3(t) \\ \cdots \\ G_0(t) + G_{l-1}(t) \\ G_0(t) \end{pmatrix}$$

The relation between $G(t+1)$ and $G(t)$ can be represented by a usual linear recursion $G(t+1) = QG(t)$, where

$$Q = \begin{pmatrix} 1\,1\,0 \cdots 0\,0 \\ 1\,0\,1 \cdots 0\,0 \\ \cdots \\ 1\,0\,0 \cdots 0\,1 \\ 1\,0\,0 \cdots 0\,0 \end{pmatrix}$$

One can check that the characteristic polynomial of Q is up to a sign of $X^l - X^{l-1} - X^{l-2} - \cdots X - 1$. Consequently, $Q^l - Q^{l-1} - Q^{l-2} - \cdots - Q - I = 0$, and as for $t \geq 0$, $G(t+l) = Q^l G(t)$ we have:

$$t \geq 0, \quad G(l+t) \;=\; G(l+t-1) + G(l+t-2) + \cdots + G(t). \tag{1}$$

Note that the smallest coordinate of $G(t)$ is $G_{l-1}(t) = G_0(t-1)$, though coordinates of $G(t)$ lie in the interval $[G_0(t-1), G_0(t)]$. Consequently, we will only analyze the behavior of $G_0(t)$ value which will be denoted $M(t)$. Phase 2 will be completed at time T if and only if $M(T-1) \geq q$.

We will not derive an exact evaluation [5] of $M(t)$. Instead we just point out that $M(t) = 2^t$ for $t \leq l-1$, moreover for $l \leq t$, equation (1) implies that:

$$M(t) = M(t-1) + M(t-2) + \cdots + M(t-l) \tag{2}$$

Now note that the two sequences $M(t)$ and $U(t) = (\tau_l)^t$ are the solutions of the same linear recursion (2). As all the coefficients of the recursion are positive and as $\forall t \in \{0, 1, \cdots, l-1\}$, $M(t) = 2^t \geq (\tau_l)^t$ (recall that $\tau_l < 2$), we can claim that for any time t, $M(t)$ is greater than $(\tau_l)^t$. Consequently we have $T \leq \lceil \log_{\tau_l}(q) \rceil + 1$. □

[5] Actually one can shows that the generating function of $M(t)$ is $\frac{1-z^l}{1-2z+z^{l+1}}$, and prove then that $M(t) = c_l(\tau_l)^t + o(t)$ with $c_l = \frac{1-(\frac{1}{\tau_l})^l}{2-(l+1)(\frac{1}{\tau_l})^l}\tau_l > 1$.

Corollary 6. $g_l(ql) \leq T_l(ql) \leq \lceil \log_2 l \rceil + \lceil \log_{\tau_l}(q) \rceil + 2$

Proof. By summing the times of Phases 1 and 2 □

2.2 When $n \neq 0 \pmod{l}$

Let $n = ql + r$ for some $1 \leq r < l$. If $q = 0$, since we have $n < l$, the gossiping can be achieved in an optimal time $\lceil \log_2(n) \rceil + 1$ (see [8]). If $q \geq 1$, we have to slightly modify the algorithm in such a way to take into account those r remaining vertices. In our algorithm, before starting Phase 1, each of those r vertices send their information to a vertex in column C_0. Note that it takes only one step since $r < l$. At that point, the whole information of V has been concentrated on the first q columns and we can apply the previous algorithm to the ql vertices. One additional step is enough for completing the whole gossip operation because at the end of Phase 2, ql ($\geq r$) vertices have known the whole information. So when $n = ql + r$, the total time to complete the gossiping is at most $T_l(ql) + 2$.

2.3 Total Time

By the previous subsections, the total execution time $T_l(n)$ of our gossip algorithm is bounded above by

$$T_l(n) \leq \log_2 l + \left(\frac{1}{\log_2 \tau_l} \right) \log_2 \left(\frac{n}{l} \right) + O(1) \tag{3}$$

It is worth noting that for any fixed l, an upper bound on the gossip time by the algorithm is asymptotically given by $T_l(n) \leq (1/\log_2 \tau_l) \log_2 n + O(1)$. For comparison, let us consider the performance of the following naive gossip algorithm: it first accumulates all the pieces of information into a vertex (i.e., an expert) in $\lceil \log_2 n \rceil$steps, then the expert vertex broadcasts the accumulated information to all the other vertices in $\lceil \log_l n \rceil$ steps. The asymptotic behavior of the algorithm is given by $\{1 + (1/\log_2 l)\} \log_2 n$.

Table 2. Coefficients of the $\log_2 n$ term.

l	x	$1/\log_2 x$	$1 + (1/\log_2 l)$
3	1.839286755	1.137466951	1.630929753
4	1.927561975	1.056214652	1.500000000
5	1.965948237	1.025404040	1.430676558
6	1.983582843	1.012034454	1.386852807
7	1.991964197	1.005842216	1.356207187
8	1.996031180	1.002873979	1.333333333

Table 2 shows a numerical result on the asymptotic behavior of the above two algorithms; i.e., it compares the numerical approximation of the coefficient of the $\log_2 n$ term of two algorithms for small l's. Note that the performance of the naive algorithm is worse than that of an optimal algorithm for complete graphs. The readers can verify that $1/\log_2 \tau_l$, the coefficient of our algorithm, is smaller than $1 + (1/\log_2 l)$, the coefficient of a naive algorithm. According to Table 2, $\frac{1}{\log_2 \tau_l}$ is quickly converging to 1. Indeed, this fact is proved in [3] by the evaluation $\frac{1}{\log_2(\tau_l)} \sim 1 + \frac{\log_2(e)}{2^l}$, where e is the base of the exponential.

3 Concluding Remarks

In this paper, we have proposed a new gossip algorithm for complete bus networks. This implies that

$$g_l(n) \leq T_l(n) = \frac{\log_2(n)}{\log_2(\tau_l)} + O(1).$$

We conjecture that it is optimal within an additive constant number of steps, that is $g_l(n) = \frac{\log_2(n)}{\log_2(\tau_l)} + O(1)$. Note that this conjecture is proved in the case $l = 2$ (see theorem 4) , but we have not been able to extend the proof in the general case. It would be interesting to prove that the bound is tight at least for its order: i.e., $g_l(n) \sim \frac{\log_2(n)}{\log_2(\tau_l)}$. This problem should be essential for analyzing the time complexity of the gossiping in bus networks. Indeed, it would gives a generic lower bound valid for any bus network with buses of limited length.

Another problem we want to solve is to construct bus networks with the smallest number of buses of length at most l, for given l, in which a gossip takes $g_l(n)$ steps. It is the problem of finding *minimum gossip bus networks* with a given parameter l. For that question one can already remark that the algorithm presented is valid for a bus network of degree $\Theta(\log n)$ obtained by considering the network containing only the buses used during the algorithm.

Acknowledgment: This work is supported in part by Scientific Research Grants-in-Aid from the Ministry of Education, Science and Culture of Japan 08680372. The two last authors are partially supported by the opèration "Rumeur" of the French GDR/PRC PRS and also by the French/Canadian PICS of the CNRS.

We are grateful to J-C. Bermond and J. Yu for their advice.

References

1. J.-C. Bermond, J. Bond, and S. Djelloul. Dense bus networks of Diameter 2. In *Interconnection Networks and Mapping and Scheduling Parallel Computation*, DIMACS Workshop, Volume 21, D. F. Hsu, A. L. Rosenberg, D. Sotteau Editors, American Mathematical Society (1994) 9–18.

2. J-C. Bermond and F.O. Ergincan. Bus Interconnection Networks. To appear in *Discrete Applied Mathematics* (1996).
3. J.-C. Bermond, P. Hell, A. L. Liestman, and J. G. Peters. Broadcasting in Bounded Degree Graphs. SIAM J. Disc. Math. **5(1)** (1992) 10–24.
4. R. C. Entringer and P. J. Slater. Gossips and telegraphs. *J. Franklin Inst.* **307** (1979) 353–360.
5. S. Even and B. Monien. On the number of rounds necessary to disseminate information. In *Proc. 1st ACM Symp. on Parallel Algorithms and Architectures,* (1989) 318–327.
6. A. Ferreira, A. Goldman vel Lejbman, and S. Song. Bus based parallel computers: a viable way for massive parallelism. *PARLE'94* (Lecture Notes in Computer Science 817), Springer Verlag (1994) 553–564.
7. A. Ferreira, A. Goldman Vel Lejbman, and S.W. Song. Broadcasting in bus interconnection networks. In *Proc. Parallel Processing: CONPAR 94 - VAPP VI* (Lecture Notes in Computer Science 854). Springer Verlag (1994) 797–807.
8. P. Fraigniaud and C. Laforest. Minimum gossip bus networks. *Networks,* **27** (1996) 239–251.
9. P. Fraigniaud and E. Lazard. Methods and Problems of Communication in Usual Networks. *Discrete Applied Mathematics* **53** (1994) 79–133.
10. S. Fujita and M. Yamashita. Optimal gossiping in mesh-bus computers. *Parallel Processing Letters* **3(4)** (1993) 357–361.
11. S. Fujita, M. Yamashita, and T. Ae. Gossiping on mesh-bus computers by packets. In *Proc. of Int'l Symp. on Parallel Architectures, Algorithms and Networks (ISPAN)* (1994) 222–229.
12. S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks* **18** (1986) 319–349.
13. A. Hily and D. Sotteau. Communication in bus networks. *Parallel and Distributed Computing,* (Lecture Notes in Computer Science 805). Springer Verlag (1994) 197–206.
14. A. Hily and D. Sotteau. Gossiping in d-dimensional mesh-bus networks. *Parallel Processing Letters* **6(1)** (1996) 101–113.
15. D. E. Knuth. The Art of Computer Programming, Vol.3 / Sorting and Searching, Addison-Wesley Publishing Company (1973).
16. D. W. Krumme, G. Cybenko, and K. N. Venkataraman. Gossiping in minimal time. *SIAM J. Comput.,* **21(1)** (1992) 111–139.
17. R. Labahn and I. Warnke. Quick gossiping by multi-telegraphs. In Bodendiek and R. Henn, editors, *Topics in Combinatorics and Graph Theory* Phisica-Verlag Heidelberg (1990) 451–458.
18. V. S. Sunderam and P. Winkler. Fast information sharing in a complete network. *Discrete Applied Math.* **42** (1993) 75–86.