ASYMMETRIC COMMUNICATION PROTOCOLS VIA HOTLINK ASSIGNMENTS*

Prosenjit Bose† Danny Krizanc‡ Stefan Langerman§ Pat Morin†

Abstract. We exhibit a relationship between the asymmetric communication problem of Adler and Maggs (1998) and the hotlink assignment problem of Bose *et al* (2000). By generalizing previous results on the hotlink problem and then exploiting this relationship we present a new asymmetric communication protocol with different performance bounds than previous protocols.

Keywords: Communication Protocols, Asynchronous Communication, Hotlink Assignment.

1 Introduction

In the last decade or so there have been a number of services introduced that provide asymmetric broadband communication to homes and small offices. With these services, clients can typically download at rates ranging from 250KBps to 2MBps but can only upload at much lower rates, often an order of magnitude less than the download rates. Sometimes this discrepancy is created artificially by Internet service providers in order to discourage clients from running servers on these connections. In other cases, it is intrinsic to the technology being used. An example of the latter case is home satellite Internet connections in which downloads are received through a satellite dish, but uploads take place over the telephone line.

Motivated by this, Adler and Maggs [1] describe a model of an asymmetric communication channel that captures (roughly) the asymmetry in these kinds of networks. In this model, a client is attempting to send an n-bit request string to a server, where the request string is drawn according to some probability distribution $D : \{0,1\}^n \rightarrow [0,1]$. The client does not know the distribution D, but the server has access to D through *black box* queries in which the server can specify a k-bit string s, ($k \leq n$) and the black box returns the cumulative probability of all requests that have s as a prefix.

A protocol is a $[\sigma, \phi, \lambda, \rho]$-protocol if, using this protocol, σ, ϕ, λ, and ρ are upper bounds on the expected numbers of: bits sent by the server, bits sent by the client, black box queries, and rounds, respectively. Table 1 summarizes previous results on protocols for asymmetric communication channels. In this table, $H = -\sum_{r\in\{0,1\}^n} D(r) \log D(r)$ is *entropy* of the distribution D and $k \geq 1$ is a free integer parameter of the algorithm.[1]

In this paper, we exhibit a relationship between the asymmetric communication problem of Adler

*This work was partly supported by NSERC, CRM, FCAR and MITACS.

†School of Computer Science, Carleton University, {jit,morin}@cs.carleton.ca

‡Wesleyan University, krizanc@caucus.cs.wesleyan.edu

§School of Computer Science, McGill University, sl@cgm.cs.mcgill.ca

[1] Throughout this paper, when the base of a logarithm is not explicitly specified, it is assumed to be base 2.

Ref.	σ	ϕ	λ	ρ
[1]	$3n$	$1.71H+1$	$3n$	$1.71H+1$
[1]	$O(n)$	$O(H+1)$	2^n	$O(1)$
[1]	$O(n)$	$O(H+1)$	$n2^k/k$	$O(\min\{n/k, H+1\})$
[6]	$3n$	$1.08H+1$	$3n$	$1.08H+1$
[7]	$n(H+2)$	$H+2$	$n(H+3)$	$H+2$
[7]	$O(2^k nH)$	$H+2$	$O(2^k nH)$	$(H+1)/k+2$
[4]	$nHk+1$	$H\log_{k-1}k+1$	$O(nk)$	$H/\log k+1$
Here	$n(k+2)$	$\frac{H\log(k+2)}{\log(k+2)-1}+\log(k+2)$	$n(k+2)$	$\frac{H}{\log(k+2)-1}+1$

Table 1: Previous and new results on asymmetric communication protocols.

and Maggs [1] and the hotlink assignment problem of Bose *et al* [2]. We show that hotlink assignments can be viewed as asymmetric communication protocols. When we apply this knowledge along with improved results on hotlink assignment, we obtain new results on asymmetric communication protocols. In particular, we obtain the last row of Table 1, which is the first protocol that, for any constant $\epsilon > 0$, can achieve $\phi = (1+\epsilon)H + c_\epsilon$, while maintaining $\sigma = c_\epsilon n$, where c_ϵ is a function that depends only the value of ϵ and is therefore constant for any fixed ϵ.

The remainder of this paper is organized as follows: In Section 2 we describe the hotlink assignment problem and generalize an approximation algorithm for this problem due to Kranakis *et al* [5]. In Section 3 we show how a hotlink assignment leads to an asymmetric communication protocol and how the hotlink assignment described in Section 2 leads to a particularly efficient protocol. Finally, Section 4 summarizes and makes some concluding remarks.

2 Hotlink Assignment

Motivated by the goal of improving web site performance by making the most popular pages more accessible, Bose *et al* [2] introduced the hotlink assignment problem. Let T be a rooted tree with m leaves and a probability distribution D on its leaves, and let the entropy of this distribution be H. The cost, $c(v)$ of a vertex v of T is the sum of the probabilities of all leaves in the subtree rooted at v. The *cost* $c(u,v)$ of an edge (u,v) of T, where u is the parent of v is equal to $c(v)$. The cost, $c(T)$ of T is the sum of all its edge costs. Equivalently, if $L(T)$ is the set of leaves of T, and $d(v,T)$ denotes the length of the path from the root of T to v then

$$c(T) = \sum_{v \in L(T)} c(v) d(v,T) \enspace .$$

An *adoption* operation at a node u of T creates a new tree T' by taking a descendant w of u, detaching w from its parent and making u the parent of w (see Fig. 1). Obviously, adoptions can change the costs of the edges and internal nodes of T'. A tree T' is a *k-hotlink assignment* of T if it can be obtained in the following way: Perform up to k adoptions at the root r of T to obtain a new tree T_1 and then create k-hotlink assignments for each of the children of r in T_1. The *k-hotlink assignment problem* is to find the hotlink assignment of T with minimum cost.

In [5], an approximation algorithm that obtains a near-optimal solution to the 1-hotlink assignment for binary trees is given. A natural generalization of that method to the k-hotlink assignment

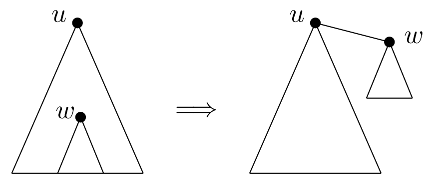

Figure 1: The adoption of w by u.

problem on binary trees may be described as follows: When processing a vertex u, we perform adoptions at node u until each child w of u is either a leaf or satisfies $c(w) \le \frac{2}{k+2}c(u)$. To determine which descendant w of u to adopt next, we start at the non-leaf child of u with maximum cost and repeatedly move to the child of the current node with largest cost until reaching a node w such that, either $c(w) \le \frac{2}{k+2}c(u)$ or w is a leaf.

The following lemma shows that the above algorithm actually does produce a k-hotlink assignment.

Lemma 1. *During the processing of u, at most k adoptions are performed.*

Proof. By construction, every descendant w of u that is adopted while processing u is either a leaf, or has weight $c(w) \le \frac{2}{k+2}c(u)$. Thus, we need only show that, after at most k adoptions, $c(x) \le \frac{2}{k+2}c(u)$ and $c(y) \le \frac{2}{k+2}c(u)$ where x and y are the two children of u in the original binary tree T. We will show something stronger; after at most k adoptions, $c(x) + c(y) \le \frac{2}{k+2}c(u)$.

We claim that every node that u adopts has weight at least $\frac{1}{k+2}c(u)$. To prove this, we argue that the procedure for finding the node w never moves to a node v with cost less than $\frac{1}{k+2}c(u)$. Observe that every descendant of u has at most two children and the algorithm only moves to a child of the current node v if $c(v) > \frac{2}{k+2}c(u)$. But then, the child v' that the procedure moves to has cost at least $c(v') \ge \frac{1}{2}c(v) > \frac{1}{k+2}c(u)$. Therefore, the procedure never moves to a node of cost less than $\frac{1}{k+2}c(u)$ and the algorithm never has u adopt a node of weight less than $\frac{1}{k+2}c(u)$, as claimed.

Therefore, each adoption performed at u reduces $c(x) + c(y)$ by at least $\frac{1}{k+2}c(u)$ and after at most k adoptions

$$c(x) + c(y) \le \left(1 - \frac{k}{k+2}\right) c(u) = \left(\frac{2}{k+2}\right) c(u) \ ,$$

as required. □

Theorem 1. *A hotlink assignment obtained by the above algorithm has a cost of at most*

$$c(T') \le \frac{H}{\log(k+2) - 1} + 1 \ .$$

Proof. The hotlink assignment produced by the algorithm has the property that for every edge (u, v) where u is a parent of v and v is not a leaf, $c(v) \le \frac{2}{k+2}c(u)$. An immediate consequence of this is that the length of the path from the root of T' to any leaf v is at most

$$d(v, T') \le \frac{\log(1/c(v))}{\log((k+2)/2)} + 1 = \frac{\log(1/c(v))}{\log(k+2) - 1} + 1 \ .$$

Therefore,

$$\begin{aligned} c(T') &= \sum_{v \in L(T')} c(v) d(v, T') \\ &\le \sum_{v \in L(T')} \left(c(v) \left(\frac{\log(1/c(v))}{\log(k+2) - 1} + 1 \right) \right) \\ &= \frac{H}{\log(k+2) - 1} + 1 \ , \end{aligned}$$

as required. □

Remark: For the case $k = 1$, a different, more detailed, analysis of the hotlink assignment algorithm shows that the cost of the resulting tree is at most $H/(\log 3 - 2/3) + 3/2 \approx 1.08H + 3/2$ [5, 6]. Repeating this analysis for larger values of k does not yield as much improvement. In fact for even k, the detailed analysis yields the same result and yields only a minor improvement for odd values of k.

3 Hotlink Assignments are Asymmetric Communication Protocols

Next we show how hotlink assignments and asymmetric communication protocols are related by showing how hotlink assignments give asymmetric communication protocols. In this protocol, the server sends bit strings of variable length and requires that the client know the lengths of the strings. Like Adler and Maggs [1], we assume that the server can do this, and the client can determine the length of these strings. When this is not the case, the server and client can use a variable-length encoding to prepend each string with its length. This allows the server to send an m-bit string using $m + \log m + O(\log \log m)$ bits [3].

Let T be a complete binary tree with 2^n leaves and label each edge of T with a 0 or 1 depending on whether the edge leads to a left or right child, respectively. Label each vertex v of T with the sequence of edge labels encountered on the path from the root of T to v. In this way, we obtain a bijection between the leaves of T and $\{0, 1\}^n$.

Suppose we have a k-hotlink assignment T' for T. Then we use the following communication protocol to send the n-bit request string r from the client to the server. The client and server split r into two parts, r_1 and r_2 where r_1 is the prefix of r that has already been transmitted and r_2 is the suffix of r that remains to be transmitted. Initially, r_1 is empty and the server does not know r_2.

During a round of the protocol, the server considers the node u of T' whose label is r_1. For each child w of u, the server sends the label of w less the prefix r_1 to the client. The client then responds with the index of the longest label l that is a prefix of r_2. The server and client both append l to r_1, and the client removes l from r_2. The client and server both continue in this manner until the entire request string has been transmitted (this happens when the length of r_1 is n).

It is clear that this protocol correctly sends the request string r from the client to the server. It is also clear that, if r is the label of the leaf v in T, then the number of bits sent by the client is $\lceil \log(k+2) \rceil d(v, T')$. If D is simultaneously the distribution on request strings and the distribution on leaves of T (so that $D(r) = D(\text{leaf labelled } r)$) then the expected number of bits sent by the client in this protocol is

$$\phi = c(T')\lceil\log(k+2)\rceil \ .$$

If T' is a k-hotlink assignment produced by the algorithm of the previous section then, by Theorem 1,

$$\phi \leq \frac{H\lceil\log(k+2)\rceil}{\log(k+2)-1} + \lceil\log(k+2)\rceil \ ,$$

and

$$\rho \leq \frac{H}{\log(k+2)-1} + 1$$

is an upper bound on the expected number of rounds used by the protocol.

Next we analyze the number of bits sent by the server if T' is a hotlink assignment obtained by the algorithm of the previous section. Let E_i be the expected number of strings sent by the server to the client that include the ith bit position of r. If it receives any such string, the client accepts it with probability at least $1/(k+2)$. Therefore, the expected number of times the client accepts a string that includes the ith bit position of r is

$$A_i \geq E_i/(k+2) \ .$$

However, the client only accepts a string that contains the ith bit position of r once, after which no other strings contain the ith bit position of r, so $A_i = 1$. Therefore, $E_i \leq k+2$. Summing this over all i, the expected number of bits sent by the server is

$$\sigma \leq n(k+2) \ .$$

Finally, we consider the expected number of black box queries required by this algorithm. Observe that we can compute the tree T online, so that we need only compute the adoptions at a vertex u if the client has accepted the prefix that is the label of u. In computing these adoptions, the only time we consider the costs of vertices is when moving from one node u to the child of u with larger cost. If we already know $c(v)$, then this can be implemented as one black box query by querying, say, the left child of v. Furthermore, each such query corresponds to exactly one more bit sent by the server to the client. Therefore, the expected number of black box queries performed by the server is

$$\lambda = \sigma \leq n(k+2) \ .$$

We have just proven

Theorem 2. *There exists an*

$$\left[n(k+2), \left(\frac{H}{\log(k+2)-1}+1\right)\lceil\log(k+2)\rceil, n(k+2), \frac{H}{\log(k+2)-1}\right]$$

asymmetric communication protocol.

4 Conclusions and Remarks

We have shown a relationship between the hotlink assignment problem and protocols for asymmetric communication channels. By generalizing previous results for the hotlink assignment problem and then exploiting this relationship we give the first protocol for asymmetric communication channels that achieves $\phi = (1+\epsilon)H + c_\epsilon$ for any constant $\epsilon > 0$ while maintaining $\sigma = c_\epsilon n$, where c_ϵ is a function that depends only on ϵ.

Previously proposed asymmetric communication protocols all assume that the universe of requests is the set of all n-bit strings. One nice property of the relationship between asymmetric communication protocols and hotlink assignments is that hotlink assignments are not restricted to complete binary trees. A consequence of this is that, if the client and server are attempting to exchange requests from any universe U of binary strings, a hotlink assignment of the binary trie containing all the strings in U gives an efficient asymmetric communication protocol.

References

[1] Micah Adler and Bruce Maggs. Protocols for asymmetric communication channels. In *Proceedings of 39th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 522–533, 1998.

[2] P. Bose, J. Czyzowicz, L. Gasienicz, E. Kranakis, D. Krizanc, A. Pelc, and M. Vargas Martin. Strategies for hotlink assignments. In *Proceedings of International Symposium on Algorithms and Computation (ISAAC 2000)*, pages 23–34, 2000.

[3] P. Elias. Universal codeword sets and the representation of the integers. *IEEE Transactions on Information Theory*, 21:194–203, 1975.

[4] S. GhaziZadeh, M. Ghodsi, and A. Saberi. A new protocol for asymmetric communication: Reaching the lower bounds. *Scientia Iranica*, 8, 2001.

[5] E. Kranakis, D. Krizanc, and S. Shende. Approximate hotlink assignment. In *Proceedings of International Symposium on Algorithms and Computation (ISAAC 2001)*, pages 756–767, 2001.

[6] E. S. Laber and L. G. Holanda. Improved bounds for asymmetric communication protocols. *Information Processing Letters*, 83:205–209, 2002.

[7] John Watkinson, Micah Adler, and Faith Fich. New protocols for asymmetric communication channels. In *Proceedings of 8th International Colloquium on Structural Information and Communication Complexity (Sirocco)*, pages 337–350, 2001.