1991年度 韓國情報科學會 가을 學術發表論文集 Vol. 18, No. 2

이중 루프망에서의 최단 경로 알고리즘

○ 성 경아, 좌 경룡, 이 광형

한국과학기술원 전산학과

Shorthest Path Algorithm in Double Loop Networks

Kyoung-A Seong, Kyoung-Yong Chwa and H. Lee-Kwang

Dept. of Computer Science, KAIST

ABSTRACT

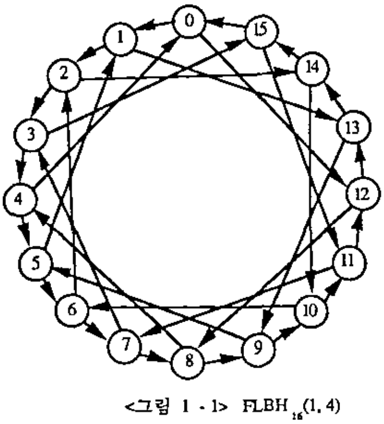
The routing problem in the computer communication network is to determine a routing path from a source to a destination node. We are especially concerned for algorithm with the smallest communication cost. If communication cost is a length of the path, this problem will be the shorthest path problem in the connected graph.

$FLBH_N(1, h)$ network is a double loop network with N nodes, such that each node has a forward link connecting to its neighbor and a backward link connecting to a node at some distance h where h is called the long hop length. In this paper, we first characterize properties of the shorthest path and the structure of $FLBH_N(1, h)$ networks. Then we divide the $FLBH_N(1, h)$ networks into several levels using the properties. Second, we present an optimal shorthest path algorithm for $FLBH_N(1, h)$ networks with $O(\lg h)$ time and $O(\lg h)$ space complexity.

1. 서론

망(network)에서의 라우팅 문제(routing problem)는 출발 노드(source node)에서 종착 노드(destination node)로의 라우팅 경로(routing path)를 결정하는 문제를 말한다[SS80]. 이때 최소 비용을 가지는 경로를 찾는 문제가 관심이 될 수 있다. 그리고 경로의 길이를 비용으로 정하면 이 문제는 망상의 임의의 노드사이의 최단 경로(shortest path)를 찾는 문제가 된다.

본 논문에서는 이중 루프 구조중의 하나인 Raghavendra [RMA85, RS85]에 의해 제안된 Forward Loop Backward Hop (FLBH) 망에서의 최단 경로를 구하는 알고리즘을 소개한다. N개의 노드를 가지는 망에서, 순방향 기본 루프의 링크를 짧은홉(short hop)이라 하고 역방향의 링크를 긴홉(long hop)이라 할 때, 짧은홉의 크기는 1이고, 긴홉의 크기 h를 긴홉의 길이(long hop length)라고 하면, FLBH 망내의 임의의 노드 x는 (x + 1) mod N 노드와 (x - h) mod N노드하고 연결되는 구조를 가진다. 이때, 노드의 이름은 루프의 시계 반대 방향의 순서로 0 ~ N-1의 정수 번호라고 가정한다. N개의 노드를 가진 망의 경우 긴홉의 길이가 h인 FLBH를 $FLBH_N(1, h)$라 표기한다.



<그림 1 - 1> $FLBH_{16}(1, 4)$

<그림 1 - 1>는 노드의 개수가 16이고 긴홉의 길이가 4인 $FLBH_{16}(1, 4)$를 보여준다. $FLBH_N(1, h)$ 망의 경로를 구하는 알고리즘으로 Raghavendra와 Silvester[RS85]가 제안한 O(1)의 시간 복잡도를 가지는 알고리즘이 있으나 이 알고리즘으로는 모든 형태의 $FLBH_N(1, h)$의 임의의 두 노드 사이의 최단 경로를 구하지는 못한다. 그리고 Dijkstra[HS78]의 최단 경로 알고리즘은 $FLBH_N(1, h)$ 망의 경우에는 한 노드에 인접하는 노드들의 수가 2로 일정하므로

O(N)의 시간 복잡도를 가진다 본 논문에서는 $FLBH_N(1, h)$ 망의 구조를 분석하여 그 성질을 밝힌 후, 이 성질들을 이용하여 O([log h])의 시간 복잡도를 갖는 새로운 최단 경로 알고리즘을 소개한다

본 논문의 구성은 다음과 같다 먼저 2장에서는 출발 노드에서 결정된 경로의 정보를 이용하여 중간 노드를 거쳐 종착 노드로 메시지를 보내는 라우팅 알고리즘과 위에서 언급한 Raghavendra와 Silvester의 알고리즘을 소개한다. 3장에서는 $FLBH_N(1, h)$ 망에서 임의의 두 노드 사이의 최단 경로가 가지는 특성들을 밝혀냄으로써, $FLBH_N(1, h)$ 망의 구조를 유형별로 분석한다 그리고 4장에서는 유형 1 구조를 가지는 $FLBH_N(1, h)$ 망의 최단 경로를 구하는 알고리즘 $PATH_1$을 알아본다

2. 라우팅 알고리즘과 $PATH_0$ 알고리즘

$FLBH_N(1, h)$의 임의의 노드 x에서 y로의 최단 경로에 대한 정보를 Path (x, y)라 할때 이것은 (n_1, n_2)의 순서쌍으로 표시할 수 있다 이때 n_1은 긴홉의 개수를 말하며, n_2는 짧은홉의 개수를 말한다 이때 경로의 길이는 $n_1 + n_2$ 이다 일단 n_1, n_2의 값이 결정되고 나면, 긴홉과 짧은홉의 사용 순서에는 관계가 없이 항상 종착 노드에 도달할 수 있으므로[RS85], 주어진 Path (x, y)에 대해 여러개의 최단 경로가 가능하다.

$FLBH_N(1, h)$의 라우팅 알고리즘은 출발 노드와 종착 노드간의 경로를 결정하는 PATH 알고리즘과 결정된 경로에 따라 정보를 전달하는 ROUT 알고리즘으로 나뉜다

2.1. 라우팅 알고리즘

PATH를 통해 얻어진 라우팅 정보 *Path (x, y)*는 종착 노드로 보낼 메시지의 한 부분에 첨가되어 중간 노드로 보내진다 중간 노드에서는 이 정보를 가지고 알고리즘 ROUT를 수행하여 그 노드가 종착 노드인가를 알아보고, 아닌 경우에는 다음 노드를 결정하여 Path (x, y)를 수정한 후 결정된 다음 노드로 메시지를 보내는 작업을 계속 한다 n_1, n_2가 모두 0이면 그 노드가 종착 노드가 되는데 이때는 보내진 메시지를 호스트로 전달한다

알고리즘 ROUT는 다음과 같다 이 알고리즘에서는 다음 노드를 선택할 때 순방향 링크를 우선적으로 선택하게 했다

알고리즘 ROUT (x, n_1, n_2);

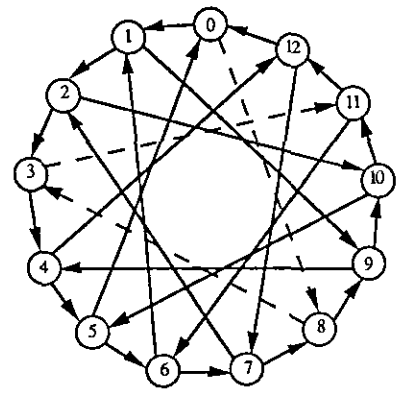
```
begin  if  n2 is not 0
         then   begin
           n2 := n2 - 1,    다음 노드 z 는 ( x + 1 ) mod N 이다;
           end
       else      if   n1 is not 0
                 then  begin
                 n1 := n1 - 1,  다음 노드 z는 ( x - h ) mod N 이다 ,
                 end
                 else  메시지를 호스트 노드로 보낸다 ,
end
```

2.2. $PATH_0$ 알고리즘

Raghavendra와 Silvester[RS85]가 제안한 $FLBH_N(1, h)$의 최단 경로를 구하는 알고리즘을 $PATH_0$라 하면 다음과 같다

알고리즘 $PATH_0$ (N, h, x, y)

```
begin
  m = ⌊N/(h+1)⌋,
  if  ( x - y ) mod N > m * h
    then  begin  n1 := 0;   n2 = ( y - x ) mod N ,  end
    else  begin
            n1 = ⌊(x - y) mod N / h⌋,
            n2 = n2 * h - ( x - y ) ;
          end
end.
```



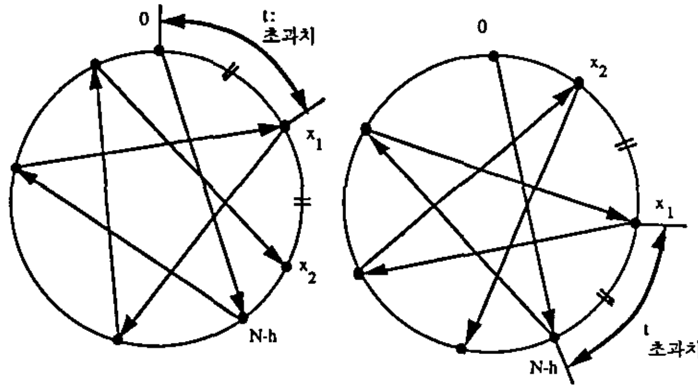
<그림 2 - 1> $FLBH_{13}(1, 5)$

$PATH_0$ 알고리즘은 < 그림 1 - 2 >의 $FLBH_{16}(1, 4)$ 의 경우에는 모든 출발 노드와 종착 노드에 대해서 최단 경로를 구한다 그러나 < 그림 2 - 1 > 에서 보여주는 $FLBH_{13}(1, 5)$ 의 경우에는 모든 노드 사이의 최단 경로를 구해 주지는 못한다 예를 들어 출발 노드가 0이고 종착 노드가 11 일때 $PATH_0$에 의해 얻어진 Path (0, 11)은 (1, 3)이지만 0 노드에서 11 노드로 가는 최단 경로는 (3, 0)이므로 $PATH_0$가 최적화 알고리즘이 될 수 없다

3. $FLBH_N(1, h)$ 망의 구조

3.1 최단 경로의 성질

<그림 2 - 1> $FLBH_{13}(1, 5)$ 에서의 Path(0, 11)의 경우의 최단 경로의 성질을 알아보기 위하여, 먼저 2.2 장에서 소개한 $PATH_0$ 알고리즘이 최단 경로를 찾는 방법을 살펴보자 예를 들어 <그림 1 - 2>의 $FLBH_{16}(1, 4)$ 에서 Path(0, 6)을 구하기 위하여 먼저 출발 노드로부터 종착 노드까지의 시계 방향 거리를 구해보면 10이다 그러면 최단 경로의 긴홉의 갯수, n_1은 이 값을 긴홉의 길이인 4로 나눈 올림값인 3이고, 짧은홉의 갯수, n_2는 계산된 긴홉으로 이동했을때의 도착 노드인 4에서 6으로 다시 되돌아오기 위한 거리로 Path(0, 6)의 경우에는 2가 된다 즉 시계 방향 거리를 계산된 긴홉의 개수와 긴홉의 크기를 곱한 값에서 빼준 값이다



(a) 경우 1 (b) 경우 2

<그림 3-1> 초과치가 발생하는 $FLBH_N(1, h)$ 망의 최단 경로

이번에는 $FLBH_{13}(1, 5)$ 에서 Path(0, 11)를 구해보자. 이때의 출발 노드에서 종착 노드로의 시계 방향 거리는 2이다. 그러나 올바른 최단 경로를 구하기 위해서는 이 거리에 망을 한바퀴 회전하는 거리, 즉 전체 노드의 개수, N의 값인 13을 더해 주어야만 한다 그러면 이렇게 얻어진 거리, 15를 5로 나눈 올림값인 3이 최단 경로의 긴홉의 개수가 되고 그때의 짧은홉의 개수를 위와 같은 방법으로 계산하면 0이 된다 그러므로 일단 원래의거리에 더해지는 망을 회전하는 회전 거리를 구할 수 있다면, 모든 $FLBH_N(1, h)$ 망의 최단 경로를 $PATH_0$ 에서와 같은 방법으로 구할 수 있다 이제 몇가지 용어를 정의한다

출발 노드로부터 종착 노드까지의 시계 방향 거리를 라우팅 거리라 하고 d라 고 표기한다 설명의 편의를 위해 앞으로 출발 노드는 항상 0 노드로 고정시키고, 최단 경로의 구성은 우선 더이상 경로의 길이를 줄일 수 없을 때까지 긴홉을 이용하여 이동시킨 다음, 최종적으로 짧은홉을 이용하여 종착 노드로 이동시킨다고 가정 한다

<그림 3-1>는 $FLBH_N(1, h)$ 망에서 긴홉을 이용한 최단 경로의 일부만을 표현한 것이다 이 중에서 우선은 (a)의 경우만을 살펴보자 그림에서 보면, 긴홉을 이용한 경로는 출발 노드에서 시계 방향으로 망을 회전하는 모양으로 나타난다 또한 0노드에서 x_1까지의 시계 방향 거리는 x_1에서 x_2까지의 시계 방향 거리와 항상 일치한다 이 거리는 경로가 망을 회전하여 처음으로 출발 노드를 넘어선 거리로 초과치라고 부르고 기호로는 t로 표기 하며 이런 경우를 초과치가 발생했다고 한다. 그리고 처음으로 초과치를 발생시키는 회전량, 즉 x_1에 도달할 때까지 긴홉으로 망을 회전하는 회전수를 초과 회전수, C라 하고 그때의 긴홉의 개수를 초과 회전 길이, B라 하자.

이제 더 일반적인 경우를 생각해 보자. 위에서 정의한 초과치를 발생시키는 노드는 출발 노드에서 초과치의 양만큼씩 시계 방향으로 이동한 노드들이다. 그러나 시계 방향으로 이동한 수와 초과치의 곱이 처음으로 긴홉의 크기, h를 넘어서게 되면 그때의 긴홉의 개수보다 하나 작은 개수의 긴홉으로 이동한 노드에서 먼저 출발 노드를 넘어서게 된다 이런 경우에는 출발 노드로부터 그 노드까지의 시계 방향 거리로 두번째 초과치를 정의한다 이때 두번째 초과치는 처음의 초과치보다는 작은 값이다

물론 또다시 새로운 초과치가 발생할 수도 있다 그러므로 i 번째까지 초과치를 발생시키는 $FLBH_N(1, h)$ 망의 구조를 유형 i 구조라 하고 그때의 각각의 초과치, 초과 회전수, 초과 회전 길이는 아래 첨자를 붙여 $t_j, C_j, B_j (j = 0, 1, .., i)$로 나타내기로 한다

<그림 3-1>의 (b) 는 새로운 초과치의 값이 바로 전단계의 초과치의 반보다 큰 경우이다 이런 경우를 경우 2라 하고 앞에서와 같은 경우에는 경우 1이라고 하자. 그리고 i번째 단계가 경우 2일때는 초과치 t_i를 t_{i-1}에서 이때의 초과치를 뺀 값으로 조정하여 정의한다. 그러면 i단계에서의 초과치를 발생시키는 노드들은 그림에서 보듯이 경우 1일때와는 달리 그 단계를 한번씩 진행할 때마다 시계 반대 방향으로 이동하게 된다.

정리 3-1은 각 단계의 정보를 구하는 식을 보여주고 있다

정리 3-1 : 유형 i 구조를 갖는 $FLBH_N(1, h)$ 망의 각 단계의 정보들은 다음과 같이 구할 수 있다

(i) $b_j = \left\lfloor \frac{t_{j-2}}{t_{j-1}} \right\rfloor$

(ii) $B_0 = 1,\ B_1 = b_1,$

$B_j = B_{j-1} * b_j - B_{j-2}$, j-1 단계가 경우 1이었을때

$B_{j-2} + (B_{j-1} - B_{j-2}) * b_j$, j-1 단계가 경우 2이었을때

(iii) $C_0 = 0,\ C_1 = 1,$

$C_j = C_{j-1} * b_j - C_{j-2}$, j-1 단계가 경우 1이었을때

$C_{j-2} + (C_{j-1} - C_{j-2}) * b_j$, j-1 단계가 경우 2이었을때

(iv) $t_{-1} = N,\ t_0 = h,$

$t_j = B_j * h - C_j * N$, j-1 단계가 경우 1이었을때

$C_j * N - B_j * h$, j-1 단계가 경우 2이었을때

이때 $\left\lfloor \frac{t_{j-1}}{t_j} \right\rfloor < 3$ 이면 $t_j = t_{j-1} - t_j$이다

3.2 구조의 분석

$FLBH_N(1, h)$ 망이 어떤 유형을 가지는지는 다음의 정의를 통해 알 수 있다.

정리 3-2 : $i+1$ 단계에서 다음 조건을 처음으로 만족하면 $FLBH_N(1, h)$ 망은 유형 i 구조를 가진다

(i) $i-1$ 단계에서 경우 1이고 i 단계가 경우 1일때, $B_i - B_{i-1} > t_{i-1} - t_i$

(ii) i-1 단계에서 경우 1이고 i 단계가 경우 2일때, $B_i - B_{i-1} > t_i$
(iii) i-1 단계에서 경우 2이고 i 단계가 경우 1일때, $B_i > t_i$
(vi) i-1 단계에서 경우 2이고 i 단계가 경우 2일때, $B_i > t_{i-1} \cdot t_i$

단계의 증가는 새로운 초과치가 발생했을때 일어난다 그런데 정리 3-1의 (iv)에서 초과치를 항상 그전 단계의 초과치의 반보다 작은 값이 되도록 정의했으므로, 새로운 초과치가 발생하는 회수에 대해 다음과 같은 정리가 만족된다.

정리 3-3 : 임의의 $FLBH_N(1, h)$ 망이 유형 i 구조를 가질때 i 값은 항상 다음 조건을 만족한다.

$$i < \lfloor \lg h \rfloor$$

4. 최단 경로 알고리즘

이장에서는 유형 1 구조일 경우에 최단 경로를 구하는 알고리즘 $PATH_1$만을 소개하기로 하겠다. 3장에서 설명한 라우팅 거리 d는 다음과 같으며, 이것은 최단 경로가 (q, r)인 모든 두 노드 에서 같은 값을 가진다.

(1) 경우 1일때,

$$d = q*h - \left\lfloor \frac{q}{b_1} \right\rfloor * N - r$$

(2) 경우 2일때,

$$d = q*h - \left(\left\lfloor \frac{q-b_1}{b_1-1} \right\rfloor + \left\lfloor \frac{q - \left\lfloor \frac{q-b_1}{b_1-1} \right\rfloor * (b_1-1)}{b_1} \right\rfloor \right) * N - r$$

우선 유형 1 구조를 가지는 $FLBH_N(1, h)$의 몇가지 성질을 살펴보자

소정리 4-1 . 유형 1 구조를 가지는 $FLBH_N(1, h)$ 망에서 모든 x, y노드에 대한 최단 경로, Path (x, y)의 n_2값, 즉 짧은홉의 개수 중에서 가장 큰 것을 r_{max}라고 하면 $r_{max} < t_1$이다

출발 노드와 종착 노드 사이의 최단 경로가 정해졌을 때 긴홉으로 이루어진 경로가 기본 루프를 회전한 회전수를 초과량이라 한다 초과량을 구하면 최단 경로는 앞 장에서 정의한 라우팅 거리 d에 초과량만큼의 N을 더한 값을 근거로 언어질 수 있다

먼저 경우 1일때 초과량과 최단 경로를 구하는 방법을 알아본다. 라우팅 거리 d 에서

$$c_1 = \left\lfloor \frac{q}{b_1} \right\rfloor$$

이 초과량이다 이 값이 c_1 이라면 $t_1 = b_1 * h - N$이므로,

$$d = q*h - c_1*N - r = (q - c_1*b_1)*h + c_1*t_1 - r$$

이다 소정리 4-1에서 $t_1 > r_{max}$이고 유형 1 구조를 가지므로

$$\left\lfloor \frac{d}{h} \right\rfloor = q - c_1*b_1 \qquad d - \left\lfloor \frac{d}{h} \right\rfloor * h = c_1*t_1 - r$$

이다 또다시 소정리 4-1에 의해

$$\left\lceil \frac{c_1*t_1 - r}{t_1} \right\rceil = c_1$$

이 된다 일단 초과량, c_1을 구해지면 최단 경로는 다음과 같다

$$\left\lceil \frac{c_1 + d}{h} \right\rceil = n_1, \qquad n_1*h - (c_1+d) = n_2$$

경우 2일때도 비슷한 방법으로 최단 경로를 구한다 이때는 초과량,

$$c_1 = \left\lfloor \frac{(q+b_1-1)-b_1}{b_1-1} \right\rfloor + \left\lfloor \frac{q - \left\lfloor \frac{(q+b_1-1)-b_1}{b_1-1} \right\rfloor * (b_1-1)}{b_1} \right\rfloor$$

이다 알고리즘 $PATH_1$을 정리하면 다음과 같다

알고리즘 $PATH_1$(N, h, x, y)

begin
f := (y - x) mod N; d = (x - y) mod N;
if (경우 1) then

$$c_1 := \left\lceil \frac{d - \left\lfloor \frac{d}{h} \right\rfloor * h}{t_1} \right\rceil,$$

else (경우 2)

$$c_1 := \left\lceil \frac{\left\lfloor \frac{d}{h} \right\rfloor * h - d}{t_1} \right\rceil, \quad n_1 := \left\lceil \frac{c_1*N + d}{h} \right\rceil,$$

$$n_2 := n_1*h - (c_1*N + d),$$

if $(n_1 + n_2 > f)$ then begin $n_1 = 0$; $n_2 = f$; end,
end;

5. 결론

본 논문에서는 $FLBH_N(1, h)$ 망의 최단 경로의 성질을 살펴보아 망을 여러 유형의 구조로 분류한 후, 이에 근거한 최적화 PATH 알고리즘을 구현하였는데 $FLBH_N(1, h)$는 $O(\lfloor \lg h \rfloor)$유형의 구조를 가질 수 있으며, 유형 i 구조일때 $PATH_i$ 알고리즘의 시간 복잡도와 공간 복잡도는 $O(i)$이다 즉 $O(\lfloor \lg h \rfloor)$의 시간 복잡도와 공간 복잡도를 가지는 알고리즘을 얻을 수 있다

본 논문에서는 FLBH 망의 순방향 링크, 즉 짧은홉의 크기를 1로 한정하고 있으나 이 크기를 임의의 크기, s로 일반화시킨 $FLBH_N(s, h)$에서도 비슷한 연구가 가능할 것으로 보인다

참 고 문 헌

[HS78] E. Horowitz & S Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, 1978

[RMA85] C S Raghavendra, Mario Gerla & Algirdas Avizienis, *Reliable Loop Topologies for Large Local Computer Networks*, IEEE Trans Comput, Vol 34, No 1, Jan 1985, 46 ~ 55

[RS85] C S Raghavendra & J A Silvester, *Double Loop Network Architectures - A Performance Study*, IEEE Trans Comm, Vol 33, No 2, Feb 1985, 185 ~ 187

[SS80] Mischa Schwartz & Thomas E. Stern, *Routing Techniques Used in Computer Communication Networks*, IEEE Trans. Comm, Vol 28, No 4, Apr 1980, 539 ~ 552