

최적 라우팅에 관한 연구

김정옥^o, 이석필, 박상희
연세대학교 전기공학과

A Study on Optimal Routing of Computer Networks
using Neural Networks

Jung-Ook Kim, Seok-Pil Lee, Sang-Hui Park
Dept. of Electrical Eng. Yonsei University

Abstract

An optimal routing method using hysteresis binary neurons with link failure probability is proposed in this paper. The link failures in computer networks can degrade the performance of the entire networks. We assume the time between successive link failures is exponentially distributed with parameter λ and the failures are independent. The link failure probability is used for neural networks to find the shortest paths of given source-destination pairs. By using the probability of link failures and hysteresis binary neurons we implement an optimal routing method that can takes routes by coping with link failures.

1. 서론

컴퓨터 네트워크를 효율적으로 이용하고 사용자가 다른 컴퓨터와 정보를 주고 받는 데 걸리는 시간이 최소화 되려면 네트워크안의 데이터가 임의의 시작 노드에서 목적 노드까지 최소 비용으로 전달되어야 한다. 따라서 시작 노드와 목적 노드 사이에 데이터를 보내기 위해 사용 가능한 링크의 집합을 최선으로 선택하려는 것을 라우팅의 목적이라 할 때 이것은 비교적 적당한 전송속도로 가능한 한 시간 지연을 줄이고 네트워크의 throughput을 크게 할 수 있는 경제적인 방법을 찾는 것이라고 할 수 있다[1,2,3].

최적 라우팅은 비용함수의 결정요소로서 링크간 시간지연을 선택하며 정의된 비용함수의 최소 미분값을 갖는 방향으로 최단 경로를 선택하여 전체적인 비용함수를 최소화하는 것이 그 목적이다[1,3]. 최적 라우팅을 구현할 때 최단 경로를 선택하는 방법으로 많이 사용되는 알고리즘으로는 Dijkstra와 Bellman등의 순차적 알고리즘이 있다[1,2,3]. 이러한 순차적 알고리즘은 그 자체로서는 링크의 결손이나 변화를 파악할 수 없으며 만일 임의의 링크에서 결손이 생기거나 데이터 흐름이 집중되어 혼잡 현상이 발생할 경우 이러한 외부 환경 변화를 라우팅에 반영하기 위해서는 별도의 bit를 할당하여 이러한 정보를 이웃하는 모든 노드끼리 서로 교환한뒤 새로운 경로를 찾아 선택된 경로로 이전의 데이터를 다시 보내는 과정이 필요하게 된다. 그 결과 패킷 제어에 사용되는 데이터양이 네트워크에 증가하게 되고 이것이 또다른 링크 결손이나 혼잡을 야기시킬 수 있다[1,4]. 따라서 신뢰성 있는 네트워크를 구축하기 위해 네트워크의 환경 상태를 주기적으로 파악하는 것은 필수적이며 링크의 결손시에는 그에 따른 적절한 조치가 취해져야 한다.

본 논문은 신경회로망[5,6,7]을 패킷 교환 컴퓨터 네트워크의 라우팅에 적용하여 노드간 시간 지연이 최소가 되고 데이터 흐름이 최적화될 수 있으며 링크 결손과 같은 네트워크의 환경변화에 적용할 수 있는 라우팅 방법을 구현하고자 한다[8,9,10,11].

2. 컴퓨터 네트워크의 최적 라우팅

컴퓨터 네트워크의 라우팅은 throughput이 크고 시간 지연이 작을수록 그 성능이 우수하다고 할 수 있다[1,3]. Quasi-static

모델을 사용하여 네트워크의 최적 라우팅을 설명하면 다음과 같다[1,3,12,13]. 네트워크의 노드 i 와 j 간의 링크를 나타내는 노드쌍 (i,j) 의 전송용량을 C_{ij} , 링크의 데이터 흐름을 F_{ij} 라고 할 때 M/M/1 모델을 사용하여 구한 네트워크안의 전체적인 평균 시간 지연과 제한 조건은 다음과 같다.

$$D(F) = \sum_{(i,j) \in L} \frac{F_{ij}}{C_{ij} - F_{ij}} \quad (2.1)$$

$$\forall i, j \in N, \forall (i, j) \in L$$

$$F_{ij} = \sum_{w \in W} x_p \quad (2.2)$$

$$r_w = \sum_{p \in P_w} x_p, \forall w \in W \quad (2.3)$$

$$x_p \geq 0, \forall p \in P_w, \forall w \in W \quad (2.4)$$

$$f_{ij} \leq C_{ij}, \forall i, j \in N, \forall (i, j) \in L \quad (2.5)$$

여기서 N 은 전체 노드 집합, 노드쌍 (i,j) 는 노드 i 와 j 를 연결하는 링크, L 은 전체 링크 집합, w 는 시작-목적 노드쌍 (source-destination pair), W 는 가능한 모든 SD쌍 w 의 집합, r_w 는 w 에 대한 데이터 도착률(arrival rate), P_w 는 w 에 대한 시작 노드와 도착 노드를 연결하는 모든 가능한 경로의 집합, p 는 P_w 의 한 원소로 임의의 경로를 나타내며, x_p 는 임의의 경로 p 에 대한 데이터 흐름을 나타낸다.

다음과 같은 경로 벡터 \vec{x} 를 이용하여 최소화 시키려는 비용함수를 나타내는 식 (2.1)을 다시 정리하면 다음과 같다.

$$\vec{x} = [x_1, \dots, x_p, \dots] \quad p \in P_w, w \in W$$

$$D(\vec{x}) = \sum_{(i,j) \in L} \frac{K_{ij} \vec{x}}{C_{ij} - K_{ij} \vec{x}} \quad (2.6)$$

$$i, j \in N, (i, j) \in L$$

여기서 벡터 K_{ij} 는 행벡터로 링크 (i,j) 가 경로 p 에 있으면 벡터 K_{ij} 의 p 번째 성분이 1 이 되고 그렇지 않으면 p 번째 성분이 0 이 되는 벡터이다. 따라서 최적의 라우팅 문제는 위의 식 (2.3), (2.4), (2.5)를 제한조건으로 식 (2.6)을 최소화하는 것으로 나타낼 수 있다[1,3,12,13]. 위에서 정의된 최적 라우팅 문제를 해결하기 위해 제안된 기존의 방법을 살펴보면 Flow Deviation (FD) 방법[1,3,14], Extremal Flow(EF) 방법[12], Gradient Projection(GP) 방법을 사용한 분산화된 비동기 최적 라우팅 방법[1, 3,13]등이 있다.

3. 신경회로망을 이용한 최적 라우팅

신경회로망을 이용하여 최단 경로를 선택할때 최소화하려는 비용함수를 식 (3.1)과 같이 선택한다[8,9,10]. 여기에서 Dx_i 는 링크 (i,j) 의 평균 시간 지연의 최소 미분 변화율로 식 (3.2)와 같다. Vx_i 는 x 행 i 열 뉴런 출력으로 x -사이의 링크가 선택된 경로에서 사용되었으면 1을, 그렇지 않으면 0을 나타내는 변수이다. $Q_{x,i}$ 는 노드 x 와 i 간 링크 (x,i) 에 결손이 발생할 확률을 나타내는 것으로 연이어 일어나는 링크 결손 사이의 시간은 식 (3.3)과 같은 지수적 확률분포를 따르며 링크 결손은 서로 독립적이라고 가정한다[4].

$$\begin{aligned}
E = & \frac{A1}{2} \sum_{x=1, x \neq d} \sum_{i=1, i \neq x, i \neq s} D_{xi} V_{xi} \\
& + \frac{A2}{2} \sum_{x=1, x \neq d} \sum_{i=1, i \neq x, i \neq s} Q_{xi} V_{xi} \\
& + \frac{A3}{2} \sum_{x=1} \sum_{i=1, i \neq x} (V_{xi} - \sum_{i=1, i \neq x} V_{ix})^2 \quad (3.1) \\
& + \frac{A4}{2} \sum_{x=1, i=1, i \neq x} V_{xi} (1 - V_{xi}) \\
& + \frac{A5}{2} (1 - V_d)
\end{aligned}$$

$$D_{xi} = \frac{C_{xi}}{(C_{xi} - F_{xi})^2} \quad (3.2)$$

$$\begin{aligned}
Q_{xi} = & \int_0^t \lambda e^{-\lambda t} T_w dt \\
0 \leq & Q_{xi} \leq 1, \quad 0 \leq t \leq T_w
\end{aligned} \quad (3.3)$$

식 (3.3)에서 λ 는 결손율을, t 는 램덤 변수를, T_w 는 SD쌍의 경로를 결정하는데 걸리는 시간을 나타낸다. 만일 Q_{xi} 가 1이면 이것은 링크 (x,i) 가 SD쌍의 경로를 결정하는 동안 계속 결손 상태에 있기 때문에 그 링크로 데이터가 전혀 전달될 수 없음을 나타내므로 경로를 선택할 때 그 링크에 해당하는 노드들이 서로 연결되어 있지 않은 것으로 간주하게 된다. 마찬가지로 만일 Q_{xi} 가 0의 값을 가지면 링크 (x,i) 로 흐르는 데이터는 모두 완전히 전달 가능함을 나타낸다. 따라서 Q_{xi} 를 에너지 함수에 고려함으로써 링크의 결손 가능성에 대한 정보가 라우팅 과정 속에 포함된다. 뉴런의 입력 update 방법에서는 식 (3.4)와 같은 운동방정식을 사용한다[15,16].

$$\begin{aligned}
\frac{dU_{ij}}{dt} = & -AU_{ij} + BE_{ij} \\
U_{ij}(t+1) = & U_{ij}(t) + \Delta U_{ij}(t) \quad (3.4)
\end{aligned}$$

A와 B는 상수이며 E_{ij} 는 excitation 함수를 나타내고 I_{ij} 는 inhibition 함수를 나타낸다. 따라서 뉴런 입력의 미소변화량은 E_{ij} 와 I_{ij} 의 선형합으로 나타낼 수 있다. 식 (3.4)를 이용하여 뉴런 입력의 미소 변화를 구하면 다음의 식 (3.5)와 같다.

$$\begin{aligned}
\Delta U_{xi} = & -\frac{A1}{2} D_{xi} (1 - \delta_{xd} \delta_{is}) \\
& - \frac{A2}{2} Q_{xi} (1 - \delta_{xd} \delta_{is}) \\
& - A3 \sum_{j=1, j \neq x} [V_{xj} - V_{jx}] \\
& + A3 \sum_{j=1, j \neq i} [V_{ij} - V_{ji}] \quad (3.5) \\
& - \frac{A4}{2} (1 - 2V_{xi}) + \frac{A5}{2} \delta_{xd} \delta_{is}
\end{aligned}$$

$$U_{xi}(t+1) = U_{xi}(t) + \Delta U_{xi}(t)$$

이것은 (3.1)에 나타난 것과 같은 비용(에너지) 함수의 뉴런 출력에 대한 변화율을 사용하여 hysteresis binary 뉴런의 입력 미소 변화를 구한 것이다. A1항에 의해 링크의 순시적 최소 미분 길이값의 영향이, A2항에 의해 링크의 결손 확률이 각 뉴런의 입력에 영향을 미치게 된다. A3항에 의해 입출력 arc의 수가 같도록 함으로써 각 행과 열에 하나의 뉴런이 존재하도록 만들며 A4항은 안정적 수렴을 위해, A5항은 페루프를 형성하여 뉴런의 확률이 시작되게 한다. 따라서 이러한 뉴런 입력 미소 변화를 사용하여 뉴런의 출력이 안정한 값을 가지며 수렴할 때까지 iteration을 계속해 나감으로써 결손이 발생한 링크가 있을 경우 이를 피해가며 최단 경로를 찾아내게 된다. 그림 3.1은 5노드 5 commodity(SD쌍의 흐름) 네트워크를 나타낸 것이다. 그림 3.1의 네트워크에서 시작 노드가 0, 도착 노드가 5인 commodity의 최단 경로가 0-1-4로 결정되었다면 뉴런의 출력은 그림 3.2(a)와 같이 나타나며, 만약 SD쌍이 (3,0)일 때 경로가 3-4-1-0으로 결정되었다면 뉴런의 출력은 그림 3.2(b)와 같이 나타나게 된다[7,8,9].

뉴런의 입출력 관계를 결정하는 함수를 그림 3.3에 나타내었다 [15,16]. 그림 3.3(a),(b),(c)는 binary 함수, 시그모이드 함수, hysteresis 함수를 각각 나타낸다. 그림 3.3(c)에 나타난 것과 같이 뉴런의 입력값이 UTP(upper trip point)보다 크게 되면 뉴런의 출력값은 1을 갖게 되고, LTP(lower trip point)보다 작게 되면 0을 갖게 된다. UTP와 LTP사이에서는 뉴런의 출력값은 이전의 값과 동일하게 되어 뉴런의 출력값은 현재값의 함수일 뿐 아니라 과거값의 함수이기도 한 성질을 보유하게 된다. 라우팅에 사용되는 뉴런은 그 출력이 해당 링크가 최단 경로에 있는

면 1, 없으면 0을 나타내기만 하면 되므로 hysteresis binary 뉴런을 사용함으로써 신경회로망을 사용할 때 혼허 겪게되는 현상인 느린 수렴 현상을 극복할 수 있게 된다[15,16].

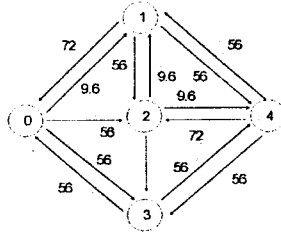


그림 3.1 5 노드 5 commodity 네트워크

0	1	2	3	4
0	1	0	0	0
1	0	0	0	1
2	0	0	0	0
3	0	0	0	0
4	1	0	0	0

(a) 경로 (0-1-4)

0	1	2	3	4
0	0	0	1	0
1	1	0	0	0
2	0	0	0	0
3	0	0	0	1
4	0	1	0	0

(b) 경로 (3-4-1-0)

그림 3.2 해당 경로를 나타내는 뉴런 출력

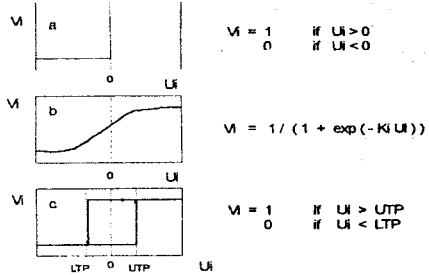


그림 3.3 뉴런의 입출력 관계를 나타내는 함수

4. 시뮬레이션 결과 및 비교 검토

최적 라우팅 방법으로는 FD 방법을 선택하였다[1,3,14]. 그림 3.1의 네트워크에 대하여 각 commodity에 따라 가능한 모든 경로를 나타낸것이 표 4.1에 나타나 있다. 이 경로는 일정한 카운터 수안에서 가능한 모든 경로를 찾을 수 있는 범람 라우팅(flooding) 방법을 사용하면 구할 수 있다[1,2]. 경로 벡터 \vec{x} 의 초기치는 식 (4.1)과 같이 설정한다[9,14]. 이 때 초기 비용함수 값은 식 (2.6)을 이용하여 구하면 78.434가 된다.

$$\begin{aligned}
\vec{x}^0 = & [x_0^0, x_1^0, \dots, x_{29}^0] \\
x_6^0 = & 50, x_7^0 = 45, x_{12}^0 = 35, \\
x_{19}^0 = & 10, x_{20}^0 = 5, x_{25}^0 = 30, \\
x_p^0 = & 0, \quad \forall p \in P_w, \quad \forall w \in W
\end{aligned} \quad (4.1)$$

표 4.1 5 commodity 네트워크의 가능한 모든 경로

SD쌍	경로번호	노드 순서	SD쌍	경로번호	노드 순서
(0,4)	0	0-1-4	(1,3)	15	1-4-2-3
	1	0-1-2-4		16	1-2-4-3
	2	0-1-2-3-4		17	1-0-2-3
	3	0-2-1-4		18	1-0-2-4-3
	4	0-2-4		19	3-0-2
	5	0-2-3-4		20	3-0-1-2
(0,2)	6	0-3-4	(3,2)	21	3-0-1-4-2
	7	0-2		22	3-4-2
	8	0-1-2		23	3-4-1-2
	9	0-1-4-2		24	3-4-1-0-2
	10	0-3-4-2		25	4-1-0
	11	0-3-4-1-2		26	4-1-2-3-0
(1,3)	12	1-2-3	27	4-2-1-0	
	13	1-0-3	28	4-2-3-0	
	14	1-4-3	29	4-3-0	

4.1 최적 라우팅 방법의 비교

최단 경로를 결정하는 세가지 방법을 사용하여 시뮬레이션한 결과 수렴한 비용을 iteration 회수에 따라 비교한 것이 그림 4.1에 나타나 있다. 비용 함수의 변화량이 10^{-2} 이내가 되면 수렴한 것으로 간주하였다.

- 계열 1 : Dijkstra의 방법을 FD에 적용한 것
- 계열 2 : Hopfield 네트워크를 FD에 적용한 것
- 계열 3 : 제안된 hysteresis binary 뉴런을 FD에 적용한 것

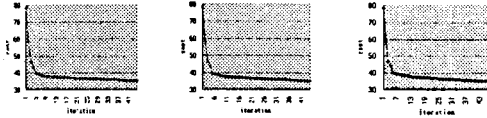


그림 4.1 5 commodity 네트워크의 최적 라우팅 비교

그림에서 보는 것과 같이 FD방법에서 비용 함수가 수렴하게 되는 iteration 수가 세 방법 모두 거의 비슷하였다. 그러나 Hopfield의 신경회로망을 사용할 경우 각 경로를 결정하는데 약 6000번의 반복학습이 필요한데 비해 제안된 방법은 약 2000번의 반복 학습으로 경로를 결정하기 때문에 전체적으로 감소하게 되는 시간은 아래 식 (4.2)와 같다.

$$T_{hs} = T_{hop} - N_c \times T(N_i) \quad (4.2)$$

여기서 T_{hs} 는 hysteresis 방법을 사용할 때의 시간이고 T_{hop} 는 Hopfield의 방법을 사용할 때의 시간을 나타내며 N_c 는 commodity의 갯수이고 N_i 는 Hopfield와 hysteresis 방법을 사용했을 때의 iteration 수 차이로 약 4000이고 $T(N_i)$ 는 N_i 번 동안 지속된 시간을 나타낸다.

4.2 링크 결손에 대한 결과

본 논문에서 제안한 hysteresis binary 뉴런을 이용한 최적 라우팅 방법의 특징은 링크 결손이 발생할 수 있는 확률이 라우팅 결정에 반영된다는 것이다. 그림 3.1과 같은 네트워크에서 만약 링크 (0,1)에 결손이 발생할 확률이 크다면 그 링크는 식 (3.2)에서 λ 값이 큰 확률 분포를 가질 것이다. 즉 결손이 발생할 링크를 포함하는 경로로는 데이터의 흐름이 점점 감소할 것이며 이에 따라 다른 경로의 데이터 흐름의 양은 증가할 것이다.

표 4.2 정상적인 동작을 할 때 각 링크의 데이터 흐름 양

	0	1	2	3	4
0		7.753476	51.484937	46.581397	0.000000
1	20.999283		34.038471	0.000000	9.687174
2	0.000000	1.202169		38.807321	2.472711
3	19.820156	0.000000	0.000000		50.761957
4	0.000000	20.999283	16.929163	5.193396	

표 4.3 링크 (0,1)의 결손율이 10일 때 각 링크의 데이터 흐름 양

	0	1	2	3	4
0		5.345946	52.069099	50.206100	0.000000
1	26.625490		35.322614	0.000000	5.023332
2	0.000000	0.000000		33.098785	0.964555
3	16.596855	0.000000	0.000000		51.984056
4	0.000000	26.625490	6.070727	5.276725	

표 4.2, 4.3은 정상적으로 동작 중일 때, 링크 (0,1)에 결손이 발생하여 결손 확률분포의 λ 가 10일 때의 수렴값을 각각 나타낸 것이다. 링크 (0,1)은 각 표에서 진하게 표시한 부분이며 결손 확률이 높아질수록 해당 링크에 흐르는 데이터의 양이 줄어들게 됨을 알 수 있다. 링크 (0,1)에 결손이 발생함에 따라 노드 0에서 노드 4로 가는 commodity 1의 흐름은 링크 (0,2)와 링크 (0,3)으로 더 많이 보내지게 됨을 표의 값을 통해 알 수 있다.

그림 4.2(a),(b)는 정상 상태일 때, 링크 (0,1)의 결손이 발생할 때의 비용함수의 수렴값을 나타낸 것이다. 여기서 최종 수렴값은 정상 상태에 도달할 수 있는 값보다 링크에 결손이 발생했을 때 더 증가함을 알 수 있다.

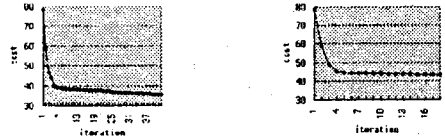


그림 4.2 링크 (0,1)에 결손이 발생하였을 때의 비용함수 비교

5. 결론

본 논문에서는 신경회로망을 패킷 교환 컴퓨터 네트워크의 라우팅에 적용하여 링크 결손과 같은 네트워크의 환경변화에 적용하며 최단 경로를 찾을 수 있는 라우팅 방법을 제시하고 이를 FD방법에 적용하여 시뮬레이션을 통해 기존의 최적 라우팅 방법과 비교, 검토하였다.

- 본 논문에서 얻은 결과를 정리하면 다음과 같다.
- Hysteresis binary 뉴런을 사용하여 최단 경로를 구한 뒤 최적 라우팅 방법(FD)에 적용한 결과 Hopfield의 신경회로망을 사용하였을 때 보다 경로 선택시 수렴하는데 걸리는 시간이 감소함을 알 수 있었다.
- 제안된 신경회로망은 링크 결손 확률을 뉴런의 update 과정에 포함시킴으로써 링크 결손이 발생하면 이를 최단 경로 결정에 반영하여 주어진 SD상에 대한 경로를 선택할 때 결손확률이 높은 링크가 포함되지 않도록 경로를 선택하여 데이터를 전달하는 라우팅을 구현할 수 있었다.

6. 참고문헌

- [1] D. P. Bertsekas, R. G. Gallager, Data Networks, Englewood Cliffs, NJ: Prentice-Hall, 2nd Ed., 1992
- [2] W. Stallings, Data and Computer Communications, Macmillan, 4th Ed., 1994
- [3] A. Ephremides, S. Verdu, "Control and optimization methods in communication network problems", IEEE Trans. on Automat. Cont., vol. 34, NO. 9, Sept., pp. 930-942, 1989
- [4] B. Gavish, I. Neuman, "Routing in a network with unreliable components", IEEE Trans. on Comm., vol. 40, NO. 7, July, pp. 1248-1257, 1992
- [5] L. Fausett, Fundamentals of Neural Networks : Architectures, Algorithms, and Applications, Prentice-Hall, 1994
- [6] S. Haykin, Neural Networks : A Comprehensive Foundation, IEEE Press, 1994
- [7] J. J. Hopfield, D. W. Tank, "Neural computations of decisions in optimization problems", Biol. Cyber., vol. 52, pp. 141-152, 1985
- [8] H. E. Rauch, T. Winarske, "Neural networks for routing communication traffic", IEEE Cont. Sys. Mag., pp. 26-30, Apr., 1988
- [9] K. Ali, F. Kamoun, "Neural network for shortest path computation and routing in computer networks", IEEE, Trans. Neural Network, Vol. 4, NO. 6, Nov., pp. 941-954, 1993
- [10] J. E. Wieselthier, C. M. Barnhart, A. Ephremides, "A neural network approach to routing without interference in multihop radio networks", IEEE Trans. on Comm., vol. 42, NO. 1, Jan., pp. 166-177, 1994
- [11] S. Lee, S. Chang, "Neural networks for routing of communication networks with unreliable components", IEEE Trans. on Neu. Net., vol. 4, NO. 5, Sept., pp. 854-863, 1993
- [12] D. G. Cantor, M. Gerla, "Optimum routing in a packet switch computer network", IEEE Trans. Comp., vol. 23, NO. 10, pp. 1062-1069, 1974
- [13] J. N. Tsitsiklis, D. P. Bertsekas, "Distributed asynchronous optimal routing in data network", IEEE Trans. Automat. Contr., vol. AC-31, pp. 325-332, 1986
- [14] L. Kleinrock, Queuing Systems Volume II : Computer Applications, Addison Wesley, 1976
- [15] Y. W. Leung, "Neural scheduling algorithms for time-multiplex switches", IEEE JSAC, vol. 12, NO. 9, Dec., pp. 1481-1487, 1994
- [16] Y. Takefuji, K. C. Lee, "An artificial hysteresis binary neuron : a model suppressing the oscillatory behaviors of neural dynamics", Biol. Cyber., vol. 64, pp. 353-356, 1991