

멀티호밍을 위한 데이터 플로우 분배 방안

송정화⁰ 이미정
이화여자대학교 컴퓨터학과
(jhsong⁰, lmj)@ewha.ac.kr

Data flow distribution method for multihoming

Jeong-Hwa Song⁰, Mee-Jeong Lee
Dept. of Computer Science and Engineering Ewha Womans

요 약

무선 네트워크 환경은 유선 네트워크 환경에 비해서 상대적으로 부족한 대역폭을 가진다. 또, 무선 네트워크 환경에서는 이동에 의해 빈번한 핸드오프가 일어나고 이는 패킷 손실, 지연 증가 등으로 인한 성능 저하를 야기한다. 하나의 엔드 시스템이 여러 개의 IP 주소를 가지고, 동시에 그 경로를 이용할 수 있다면, 이러한 문제를 완화시킬 수 있을 것이다. 이 논문에서는 하나의 엔드 시스템이 여러 개의 IP 주소를 가지는 경우에 그러한 여러 개의 경로상에 데이터 플로우를 분배하기 위한 알고리즘을 제안한다.

1. 서 론

무선 네트워크 환경은 유선 네트워크 환경에 비해서 상대적으로 부족한 대역폭을 가진다. 또, 무선 네트워크 환경에서는 이동에 의해 빈번한 핸드오프가 일어나고 이는 패킷 손실, 지연 증가 등으로 인한 성능 저하를 야기한다. 하나의 엔드 시스템이 여러 개의 IP 주소를 가지고, 동시에 그 경로를 이용할 수 있다면, 여러 경로의 대역폭을 사용할 수 있게 되어 부족한 대역폭으로 인한 문제를 완화시킬 수 있을 것이다[1]. 핸드오프의 경우에도 현재 모바일 노드가 위치하는 경로뿐만 아니라 모바일 노드가 이동할 것으로 예상되는 곳으로 데이터를 보냄으로써, 핸드오프 후에 사용 가능한 자원을 파악하기 위한 지연을 줄이는 데에 이용할 수도 있다.

이 논문에서는 하나의 엔드 시스템이 여러 개의 IP 주소를 가지는 경우에 그러한 여러 개의 경로상에 데이터 플로우를 분배하기 위한 알고리즘을 제안한다. 이 알고리즘은 트랜스포트 계층에서의 접근으로써, 멀티호밍을 지원하는 SCTP 프로토콜에 기반을 둔다[2][3].

2. 다중 경로상에 데이터 플로우 분배 알고리즘

이 장에서는 하나의 엔드 시스템이 여러 개의 IP 주소를 가질 때, 애플리케이션으로부터 받은 하나의 사용자 데이터 플로우를 트랜스포트 계층에서 여러 개의 경로상에 나누기 위한 알고리즘을 제시하고, 이에 따른 분석을 한다. 경로의 수는 두 개로 가정을 하고, 각 경로에 대한 지연을 각 경로의 RTT의 반으로 추정한다. 지연이 긴 경로로 하나의 패킷을 먼저 보내고, 그 후에 지연이 짧은 경로로 n 개의 패킷을 보내는 방식으로 데이터 플로우를 분배하려 한다. 이는 다중 경로 이용시에 야기되는 순서에 맞지 않게 패킷이 수신자에

도착하는 경우를 최소화하면서도 다중 경로의 대역폭을 이용하여 처리율을 높이고자 함이다.

2.1 데이터 플로우 분배 알고리즘

- 1) 긴 지연을 가지는 경로로 패킷 하나를 보내는 동안 짧은 지연을 가지는 경로로 보낼 수 있는 패킷의 수 n 을 계산한다. 긴 지연을 가지는 경로를 p_l 이라 하고, 짧은 지연을 가지는 경로를 p_s 라 하자. t 는 전송시간을 의미한다.

$$t(n+1) + p_s \leq t + p_l$$

$$n \leq \frac{p_l - p_s}{t}$$

$$n = \left\lfloor \frac{p_l - p_s}{t} \right\rfloor$$

- 2) 각 경로의 CW 값을 조건으로 하여 각각의 경로로 전송할 수 있는 패킷의 수를 결정한다. 여기서 CW 는 혼잡 윈도우와 수신자 윈도우 중 작은 값이다. 짧은 지연 경로로 전송되는 패킷 수 $m_s = \min(n, CW_s)$ 이고, 긴 지연 경로로 전송되는 패킷 수 $m_l = \min(1, CW_l)$ 이 된다. 즉, 한번(한 라운드)에 보낼 수 있는 전체 패킷의 수는 다음 식과 같다.

$$RW = \begin{cases} CW_s & , \text{if } CW_s = 0 \\ \min(n, CW_s) + \min(1, CW_l) & , \text{if } CW_l > 0 \end{cases} \quad (1)$$

- 3) 각 경우에 따라 패킷을 전송한다.

- ① $CW_s = 0$, 즉 $m_s = 0$ 인 경우는 긴 지연 경로만 사용하는 경우로 이 라운드에 보내는 총 패킷 수(RW) = 1 이다. 따라서, 하나의 패킷만을 긴 지연 경로로 보낸다.

② $CW_i = 0$, 즉 $m_i = 0$ 인 경우는 짧은 지연 경로만 사용하는 경우로 $RW=CW_s$ 이다. 따라서 짧은 지연 경로로 CW_s 개의 패킷 보낸다. 이 경우에는 긴 지연 경로의 CW_i 가 0보다 커지는 때를 체크해야 한다.

③ $m_s > 0$ 이고 $m_i > 0$ 인 경우 $RW=m_s+1$ 이고, 이 때 전송 방식은 다음과 같다. 먼저 긴 지연 경로로 m_s+1 개 중 가장 순차 번호가 높은 패킷 하나를 전송한 후, 나머지 m_s 개를 짧은 지연 경로로 순차 번호 순으로 전송한다.

4) RW 개의 패킷을 보낸 후, 다시 2) 과정부터 반복한다.

이 알고리즘에서 라운드는 RW 개의 패킷을 보내기까지를 말하는 것으로, RW 개 중 첫번째 패킷을 보낼 때의 시간부터 마지막 패킷을 보내기까지의 시간이 하나의 라운드가 된다. 이때, 라운드간 간격 결정이 성능에 있어서 하나의 변수가 된다.

2.2 라운드간 간격 결정

한 라운드동안 보낼 수 있는 총 패킷은 (1)식에서 알 수 있는 것처럼, 짧은 지연 경로만을 이용하여 패킷을 보내는 경우($CW_i=0$)에는 n 과 상관없이 CW_s 만큼의 패킷을 보낼 수 있고, 그렇지 않은 경우에는 각 경로의 CW 의 제약이 없을 때, 한 라운드에 최대 $n+1$ 개의 패킷을 전송하는 것이 가능하다. 한 라운드동안 보내는 패킷의 수에 따라서 라운드간의 간격 결정은 달라지게 된다. 한 라운드에 총 CW_s 개 패킷 전송 시에는, 전송 순서에 의해서 패킷이 순서에 어긋나게 수신자 측에 도착하는 경우는 발생하지 않으므로 라운드간 간격은 고려하지 않아도 된다. 또, 한 라운드에 $n+1$ 개 전송 시에는 정확히 순서대로 도착하게 되므로, 순서에 어긋나게 도착하는 패킷이 없도록 하기 위한 라운드간의 간격은 필요가 없다. 그러나, CW 의 제약으로 인해, 한 라운드에 $n+1$ 보다 작은 수의 패킷만을 전송하는 경우에는, 그로 인해 순서에 어긋나게 수신자 측에 도달하는 경우가 발생하며, 라운드간의 간격을 결정하는 데에 따라 그 성능은 달라지게 된다. 따라서 이 때에는 라운드간의 적절한 간격 결정이 필요하다.

2.2.1 라운드간의 간격 결정 기준

라운드간의 간격 결정은 패킷 전송의 목적에 따라 다르다. 여기서는 순서에 어긋나게 도착하는 패킷의 수를 최소화하는 것과, 대역폭의 활용률을 최대화하는 것을 목적으로 하는 각각 다른 두 경우에 대해 살펴본다. 각각의 목적에 따라서 라운드간의 간격은 다르게 주어져야 한다. 먼저 송신자 측에서 보낸 패킷이 어떠한 순서로 수신자 측에 도착하게 되는지 일반적인 예제를 보자.

이 그림에서 RW_i 는 i 라운드에 보낸 총 패킷 수이고, R_i 는 i 라운드가 시작되는 시간이며, 이는 다음과 같이 쓸 수 있다.

$$R_i = \sum_{j=1}^{i-1} (RW_j \cdot t + idle_j)$$

$idle_i$ 는 i 라운드의 유휴 시간이며, k_i 는 $i+1$ 이상의 라운드에서 보낸 패킷 중, i 라운드의 긴 지연 경로로 보낸 패킷보다 먼저 도착할 가능성이 있는 최대 패킷 수를 나타낸다.

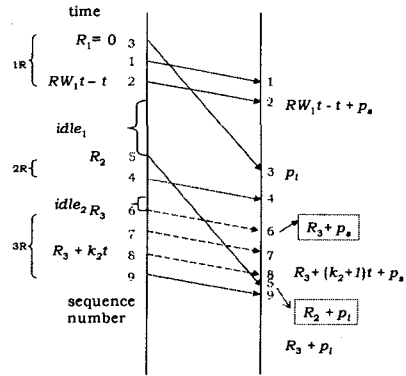


그림 1 일반적인 예제

그림1에서 점선으로 나타낸 부분이 순서에 맞지 않게 도착하는 패킷이다. 이 그림을 바탕으로 다음 두 절에서 적절한 유휴 시간을 정하고, 그에 따른 분석을 이루어지겠다.

2.2.2 순서에 어긋나는 패킷 도착 최소화

순서에 어긋나게 도착하는 패킷의 수를 최소화하기 위해서는 충분한 유휴 시간을 가져야 한다. 이 때, 충분한 유휴 시간은, 다음과 같이 계산된다.

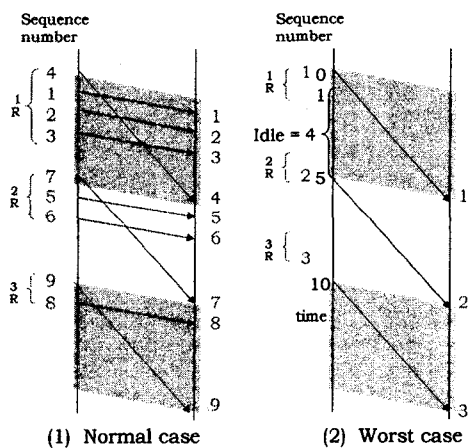
$$R_{i+1} + p_s \geq R_i + p_i$$

$$\sum_{j=1}^i (RW_j \cdot t + idle_j) + p_s \geq \sum_{j=1}^{i-1} (RW_j \cdot t + idle_j) + p_i$$

$$RW_i \cdot t + idle_i + p_s \geq p_i$$

$$idle_i \geq p_i - p_s - RW_i \cdot t$$

즉, $i+1$ 라운드를 시작하기 위해서는 최소 $p_i - p_s - RW_i \cdot t$ 시간만큼을 기다려야 한다. 이 때 i 라운드에 낭비되는 최소 대역폭은, $(p_i - p_s - RW_i \cdot t) \times$ 링크 용량이다.



(1) Normal case

(2) Worst case

그림 2 충분한 유휴 시간이 주어진 경우

그림 2는 순서에 어긋나게 도착하는 패킷의 수를 최소화하기

위해 충분한 유휴 시간인 주어졌을 때, 낭비되는 대역폭을 기준으로 하여 일반적인 경우와 가장 나쁜 경우를 나타낸 것이다. 가장 나쁜 경우는 각 라운드에 긴 지연 경로로만 패킷을 보내는 경우로써, $p_l - p_s - t$ 시간 후에 다음 라운드 전송이 가능하다. 즉, $(p_l - p_s - t) \times$ 링크 용량 만큼의 대역폭이 낭비된다.

2.2.3 대역폭 활용률 최대화

사용 가능한 대역폭의 활용률을 최대화하기 위해서는 대역폭이 낭비되지 않도록 라운드간의 유휴 시간을 0으로 두는 것이 좋다. 즉, 하나의 라운드가 끝나고 다음 라운드를 시작하기 위해 기다리지 않고 바로 다음 라운드를 시작하는 것이다. 그러나, 이때는 대역폭 활용을 최대화하는 대신 순서에 어긋나게 패킷이 수신자 측에 도착하는 상황을 야기하게 된다. 따라서, 이 경우에는 순서에 맞지 않게 도착하는 패킷을 저장하고 있어야 하며, 이를 위해 충분한 크기의 저장공간이 필요하게 된다. 이 때의 버퍼의 크기를 계산하면, 다음과 같다.

$$R_{i+1} + (k_i - 1) \cdot t + p_s < R_i + p_l$$

$$\sum_{j=1}^i (RW_j \cdot t + idle_j) + (k_i - 1) \cdot t + p_s < \sum_{j=1}^{i-1} (RW_j \cdot t + idle_j) + p_l$$

$$RW_i \cdot t + (k_i - 1) \cdot t + p_s < p_l$$

$$k_i < \frac{p_l - p_s}{t} - RW_i + 1$$

k_i 는 i 라운드에 필요한 최소 버퍼 크기이다. 따라서, i 라운드에 필요한 버퍼의 크기는 k_i 이상이어야 한다.

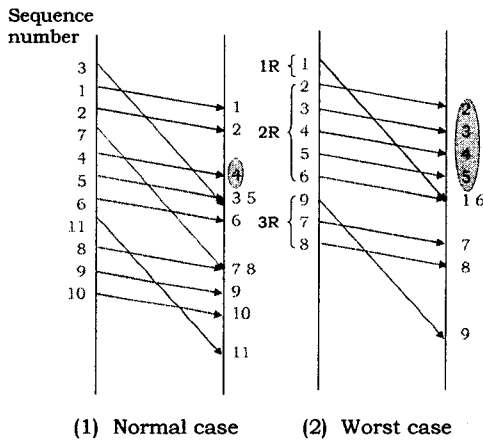


그림 3 유휴 시간이 전혀 주어지지 않은 경우

그림 3은 사용 가능한 대역폭의 활용률을 최대화하기 위하여 유휴 시간이 전혀 주어지지 않았을 때, 순서에 맞지 않게 도착할 수 있는 패킷 수를 일반적인 경우와 가장 나쁜 경우로 나타낸 것이다.

가장 나쁜 경우는 순서에 맞지 않게 도착할 수 있는 패킷의 수 k_i 가 가장 큰 경우이므로, RW_i 가 1인 경우이다. 즉, 이번 라운드에서 긴 지연 경로로만 하나의 패킷을 보낼 때 순서에 맞지 않게 패킷이 도착할 가능성이 가장 많게 된다. 실제로

다음 라운드에 몇 개의 패킷을 보낼 때 순서에 맞지 않게 도착하는 패킷이 가장 많은지를 계산하면 다음과 같다.

$$k_i < \frac{p_l - p_s}{t} - RW_i + 1$$

$$k_i < n + \alpha - RW_i + 1 \quad \left[\frac{p_l - p_s}{t} \right] = n$$

$$k_i < n + \alpha \quad \frac{p_l - p_s}{t} = n + \alpha, \quad 0 \leq \alpha < 1$$

$$k_i \leq n$$

따라서, 가장 나쁜 경우는 i 라운드에서 긴 지연 경로로 하나의 패킷을 보내고, $i+1$ 라운드에서 짧은 지연 경로로만 n 개 이상의 패킷을 보내는 경우이다. 이 때, 순서에 맞지 않게 수신자 측에 도착하는 패킷의 수는 n 개이며, 수신자는 n 이상의 버퍼 공간이 필요하게 된다.

2.2.4 라운드수와 순서에 어긋나게 도착하는 패킷 수의 관계

2.2.3절에서 순서에 어긋나게 도착하는 패킷의 수를 계산할 때, 연속된 두 개의 라운드만을 고려하여 주었다. 이것은 k_i 가 i 라운드에서 순서에 맞지 않게 도착할 가능성이 있는 패킷의 최대값을 계산한 값이므로, $i+1$ 라운드에서 보내진 패킷뿐 아니라, $i+2$ 이상의 라운드에서 보낸 패킷이 i 라운드의 긴 지연 경로로 보낸 패킷보다 먼저 도착하는 경우까지 포함한다. 즉, 두 개의 연속된 라운드만을 고려하면 된다.

3. 결 론

본 논문에서는 사용자 애플리케이션이 보낸 하나의 데이터 플로우를 두 개의 경로로, 각 경로의 지연을 반영하여 나누어 보내는 알고리즘을 제시하였다. 이 때, 패킷을 보내는 순서를 정함으로써, 수신자 측에 최대한 순서대로 도달 할 수 있도록 하였다. 즉, 가장 먼저 느린 경로로 순차 번호가 높은 패킷을 보내고, 그 후에 빠른 경로로 나머지 패킷을 순차 번호 순으로 보내는 방식을 제안했다. 실제 전송되어지는 패킷의 수는 각 경로의 $cwnd$ 와 $rwnd$ 중 작은 값에 의해서 정해지며, 이로 인한 순서에 어긋나는 패킷 도착 문제에 대해 유휴 시간을 두어 해결 또는 보안을 하였다.

향후 연구로는 본 논문에서 제안한 알고리즘을 구현하도록 한다. 순서에 어긋나는 패킷 도착을 최소화하기 위한 방안과 대역폭 활용률을 최대화하기 위한 방안을 각각 구현하여 시뮬레이션을 통해 이 두 방안이 트랜스포트 계층의 처리율에 미치는 영향을 분석할 것이다.

4. 참고문헌

[1] D. S. Phatak and Tom Goff, "A Novel Mechanism for Data Streaming Across Multiple IP Links for Improving Throughput and Reliability in Mobile Environments", *IEEE INFOCOM'02, New York*, June 2002

[2] M. Riegel, M. Tuexen, "Mobile SCTP", <draft-riegel-tuexen-mobile-sctp-00.txt>, February 20, 2002

[3] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.