

# 네트워크 지원 비디오 전송에서의 효율적인 혼잡제어

박학선<sup>o</sup> 서주하

강원대학교 컴퓨터정보통신공학과, 강원대학교 컴퓨터정보통신공학과 교수  
sunsson@empal.com<sup>o</sup>, jhseo@cc.kangwon.ac.kr

## Efficient Congestion Control in Network-supported Layered Video Multicast

Hagsun Park<sup>o</sup> Juha Seo

Department of Computer and Communication in Kangwon National University

### 요 약

이전의 계층화된 비디오 멀티캐스트에서는 각 계층의 크기를 그 하위 계층의 총합과 같도록 하여 TCP와 유사한 혼잡제어를 했었다. 하지만, 혼잡 시 비디오 데이터의 대역폭을 1/2로 줄이는 것은 사용자에게 인지되는 비디오 품질에 있어 큰 변화를 보이게 되므로 바람직하지 않고, 멀티캐스트를 사용함으로써 네트워크 전체에 트래픽을 줄이는 세션에게 아무런 이득도 주지 못하였다. 따라서 제안하는 방법에서는 기존 계층을 동일한 크기의 계층으로 나누고, 혼잡의 정도에 따라 계층의 감소 간격을 적절히 함으로써, 혼잡 해소 시에는 남아 있는 계층으로 데이터를 전달할 수 있게 하였다. 이러한 결과로 혼잡 시 비디오 품질이 급격히 떨어지는 것을 방지할 수 있고, 멀티캐스트를 사용하는 세션에게 이득을 줄 수 있다.

## 1. 서 론

계층화된 비디오 멀티캐스트 기법은 비디오 스트림을 계층화된 인코딩 기법을 통해 여러 개의 계층화된 비디오 스트림을 생성하여 멀티캐스트 한다. 계층화된 비디오 멀티캐스트에서는 혼잡에 대처하기 위해 전송을 제어할 필요로 하는데, 이를 주도하는 주체에 따라 수신자 주도의 계층화된 비디오 멀티캐스트 기법(RLM, Receiver-driven Layered Multicast)[1][2][3]과 네트워크 지원의 계층화된 비디오 멀티캐스트 기법(NLM, Network-supported Layered Multicast)[4]으로 나뉜다.

RLM에서는 송신자가 비디오 데이터를 여러 개의 계층으로 나누어 각 계층마다 별도의 멀티캐스트 그룹을 형성하여 전송한다. 따라서 하나의 멀티캐스트 그룹은 계층으로 표현되고 하위 계층은 기본적인 영상 정보를, 상위 계층은 고품질의 영상 정보를 갖는다. 수신자는 네트워크 상태를 파악하여 혼잡이 발생했을 때는 상위 계층을 내리기 위해 해당 그룹에서 탈퇴하고, 부하가 적을 때는 한 계층을 올리기 위해 상위 그룹에 가입한다.

하지만 RLM에서는 혼잡 발생시 RED(Random Early Detection)나 Drop-tail 방법을 사용함으로써 인해, 기본 계층 패킷이 랜덤하게 폐기되고 이는 수신 측의 품질 저하를 야기한다. 또한 네트워크의 상태를 수신자가 판별하기 때문에 혼잡에 느리게 반응하고, 혼잡 발생 시 IGMP leave delay[5]로 인해 멀티캐스트 그룹을 떠날 때 지연이 발생한다.

이에 비해 NLM[4]은 RLM이 지닌 비디오 품질의 계속적인 변동과 불공평성을 해결하기 위해 라우터에게 우선 순위가 따른 전송을 제어 기능을 추가하였다. 상대적으로 중요한 기본 계층(base Layer)에 앞서 우선순위가 낮은 향상 계층(enhancement layer)을 먼저 탈락시킴으로써 품질저하를 막을 수 있고, 라우터가 네트워크의 상태를 판단하기 때문에 RLM보다 빨리 혼잡에 대응해 우선 순위가 낮은 계층을 감소시킬 수 있다.

하지만 NLM에서 계층화된 비디오 멀티캐스트가 TCP와 유사한 혼잡제어를 위해 각 계층의 크기를 그 하위 계층 크기의 총합과 같도록 했기 때문에, 혼잡 발생시 향상 계층을 하나 감소시키는 것은 비디오 품질에 있어 큰 차이를 보이게 되고, 비디오 데이터를 멀티캐스트 함으로써 네트워크 전체에 트래픽을 줄이는 세

션에게 아무런 이득도 주지 못한다.

이를 해결하기 위해 이 논문에서는 비디오 데이터를 더 많은 수의 같은 크기를 갖는 계층으로 나누고, 혼잡 발생 시 계층의 감소 간격을 적절히 해서 혼잡 정도에 따라 계층을 탈락시켜 혼잡이 해소되면 남아 있는 계층으로 데이터를 전달하게 함으로써 비디오 품질이 급격히 떨어지는 것을 방지할 수 있고, 멀티캐스트를 사용하는 세션에게 이득을 줄 수도 있다.

앞으로 2절에서는 기본적인 서비스 모델을 보이고, 3절에서는 네트워크의 상태에 따른 계층 전송 제어를 설명하며, 4절에서는 계층의 증가/감소 요구를 설명하고 5절에서는 결론으로 마무리 하겠다.

## 2. 서비스 모델

### 2.1 계층의 구성

기존의 계층 구성은 TCP와 유사한 혼잡 제어를 위해, 한 계층의 대역폭이 그 하위 모든 계층의 대역폭의 합과 같도록 하여 혼잡 발생 시 TCP처럼 대역폭을 1/2로 낮춤으로써 사용자에게 비디오 품질에 큰 변화를 보이게 되므로 바람직하지 않고, 멀티캐스트를 사용해서 비디오 데이터를 전달함으로써 네트워크 전체에 트래픽을 줄이는 세션에게 아무런 이득도 없다. 따라서 제안하는 알고리즘에서는 기존의 계층을 모두 동일한 크기의 작은 계층으로 나누고, 네트워크에 혼잡이 발생했을 때 같은 크기로 나누어진 계층을 하나씩 줄이면서 혼잡이 해소되었는지 파악한다. 일정간격 후에 혼잡이 해소되지 않으면 계층을 하나 더 줄이는 방법을 혼잡이 해소될 때까지 적용한다. 만약 혼잡이 해소되면, 남아 있는 계층으로 데이터를 전달해 혼잡 시 전송율이 절반으로 떨어져 비디오 품질이 급격히 저하되는 것을 방지할 수 있고 멀티캐스트를 사용하는 세션에게 이득을 줄 수 있다.

### 2.2 계층 상태표(Layer State Table)

멀티캐스트 트리의 중간에 위치한 능동노드는 계층 상태표(Layer State Table)를 유지하고 있고, 각 출력큐와 세션별로  $L_{max}$ 와  $L_{cur}$ 을 유지하고 있다.  $L_{cur}$ 은 출력큐를 통해 그 세션이 전송할 수 있는 계층의 크기이고,  $L_{max}$ 은 현 노드에서 상향노드로부터 받고 있는 한 세션의 최대 계층이다.  $L_{max}(S_i)+1$  서비스 노드는 계층 증가 요구를 할 때 그 세션의 계층을 전송하고 있

는 노드를 말하며 다음 절에서 설명할 계층 상태 검사 패킷(Layer State Probe Packet)을 통해 알 수 있다. 표1은 계층 상태표의 예를 보여주고 있다.

표 1 한 Active Node에서 계층 상태표의 예

Prev_Node	Output_queue Q <sub>j</sub>	Source address	Session S <sub>i</sub>	L <sub>max</sub> (S <sub>i</sub> )	L <sub>cur</sub> (Q <sub>j</sub> , S <sub>i</sub> )	L <sub>max</sub> (S <sub>i</sub> )+1 서비스노드
AN <sub>a</sub>	Q <sub>1</sub>	Src1	1	3	3	AN <sub>1</sub>
		Src2	2	2	2	AN <sub>2</sub>
		Src3	3	4	3	AN <sub>3</sub>
		Src4	4	1	1	AN <sub>4</sub>
	Q <sub>2</sub>	Src1	1	3	2	AN <sub>1</sub>
		Src2	2	2	2	AN <sub>2</sub>
		Src3	3	4	4	AN <sub>3</sub>
		Src4	4	1	1	AN <sub>4</sub>

2.3 계층 상태 검사 패킷(Layer State Probe Packet)

능동노드는 상황에 따라 자신의 상향 노드들에게 필요한 계층의 증가를 요구하거나 쓸모없게 된 계층의 감소를 요구할 수 있다. 계층 상태 검사 패킷은 송신자를 포함한 능동노드가 세션별로 자신이 서비스할 수 있는 최고 계층을 기록해 전송함으로써 하향의 노드가 한 세션의 계층 증가를 요구할 때, 그 세션의 계층을 전송하고 있는 노드와 최초 송신자로부터의 홉 거리를 알게 한다. 홉 거리는 계층 증가 요구가 상향의 노드에 혼잡을 일으키는가를 판단하기 위해 요구한 계층이 도착할 때까지 기다리기 위한 detect\_period를 설정하기 위해 필요하다. 그림 1은 계층 상태 검사 패킷의 포맷을 나타낸 것이다.

Type	Size	Prev_node	Source Address	Multicast Group ID	Layer 수	
서비스노드(L <sub>0</sub> )	Hop(L <sub>0</sub> )	서비스노드(L <sub>1</sub> )	서비스노드(L <sub>1</sub> )	Hop(L <sub>1</sub> )	서비스노드(L <sub>2</sub> )	Hop(L <sub>2</sub> )
서비스노드(L <sub>2</sub> )	Hop(L <sub>2</sub> )	서비스노드(L <sub>3</sub> )	서비스노드(L <sub>3</sub> )	Hop(L <sub>3</sub> )	서비스노드(L <sub>4</sub> )	Hop(L <sub>4</sub> )

그림 1 계층 상태 검사 패킷의 포맷  
송신자를 포함한 능동노드는 세션별로 자신이 상향으로부터 전달받고 있는 계층을 계층 상태표를 참조해서 하향노드들로 전송한다. 그림 2와 같은 멀티캐스트 트리를 갖는다면, 최초 송신자는 모든 계층을 전송할 수 있으므로 계층 상태 검사 패킷의 각 계층의 서비스 노드를 S, 송신자로부터의 홉 거리를 0으로 표시한다. 이를 수신한 능동노드 A는 계층 상태표에서 계층 L<sub>4</sub>까지 수신하고 있으므로 계층 L<sub>0</sub>~L<sub>4</sub>의 서비스노드를 A로, 송신자로부터 홉 거리는 1로 표시해 하향의 노드에 전송한다. 그리고 나머지 계층은 위로부터 전달된 정보를 그대로 유지한다.

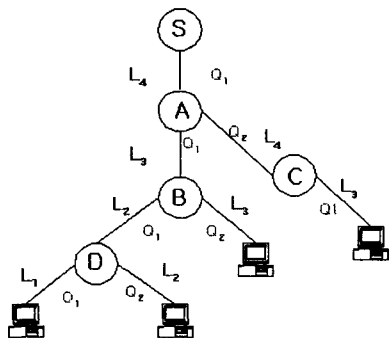


그림 2 멀티캐스트 트리의 일반적인 구성  
표 2는 이렇게 각 능동노드가 표시해 전송하는 계층 상태 검사 패킷의 예를 나타낸 것이다.

표 2 각 능동노드가 표시한 서비스계층과 홉 수

Layer \ 능동노드	L <sub>0</sub>		L <sub>1</sub>		L <sub>2</sub>		L <sub>3</sub>		L <sub>4</sub>		...	
	서비스노드	홉	서비스노드	홉	서비스노드	홉	서비스노드	홉	서비스노드	홉	서비스노드	홉
S	S	0	S	0	S	0	S	0	S	0	S	0
A	A	1	A	1	A	1	A	1	A	1	S	0
B	B	2	B	2	B	2	B	2	A	1	S	0
C	C	2	C	2	C	2	C	2	C	2	S	0
D	D	3	D	3	D	3	B	2	A	1	S	0
...	...	...	...	...	...	...	...	...	...	...	...	...

3. 계층 전송 제어

3.1 네트워크의 상태감지

비디오 데이터가 능동노드에 도착하면 계층과 세션에 대한 정보를 계층상태표에서 찾고 데이터 패킷이 전달될 각 출력큐에 대해 혼잡상태가 아니라면 데이터 캡슐을 출력큐로 적재하고, 혼잡상태라면 필터에 의해 필터링의 대상인가에 따라 필터링되거나 출력큐에 적재된다.

혼잡상태가 아니어서 패킷을 적재한 것이 출력큐에 혼잡을 일으킬 수도 있기 때문에 적재시마다 큐길이를 검사해서 혼잡이 발생했는지를 판단한다. 이 논문에서는 혼잡 상태를 판단하기 위해 평균 큐 길이를 사용하는데, 이는 버스티한 데이터에 의한 순간적인 큐 변화에 민감하게 반응하지 않기 위해서이다. 평균 큐 길이 qlen<sup>i</sup>는[6]

$$qlen^i = qweight \times q_{cur}^i + (1 - qweight) \times qlen^i$$

를 통해 구한다. 평균 큐 길이 qlen<sup>i</sup>가 qmin보다 작으면 "unloaded", qmax보다 크면 "혼잡(congestion)", 그리고 qmin과 qmax 사이에 있으면 "loaded"상태라고 한다.

3.2 계층 증감에 따른 전송을 제어(필터링)

능동노드는 네트워크의 상태와 하향노드로부터 계층의 증가를 요구하는 패킷이 수신되었을 때, 계층을 증가시키는 ADD process 혹은 DROP process를 한다. 계층 증감 요구의 패킷을 수신한 경우의 설명은 다음으로 미룬다.

우선 DROP process에 대해 먼저 살펴보면, 출력큐의 큐 길이가 qlen<sup>i</sup>가 qmax를 넘어 혼잡상태라고 판단하면 그 출력큐를 통해 가장 많은 누적 계층으로 전송하는 세션을 선택해서 계층 감소의 타겟으로 설정한다.[4] 이는 모든 세션이 보낼 수 있는 계층의 수와 각 계층의 데이터양이 같다는 가정 하에 이루어진다. 그리고 기본 계층은 어떤 경우여도 전송되어야 하므로 타겟으로 설정하지 않는다.

1. i<sub>0</sub>=i, if L<sub>cur</sub>(Q<sub>j</sub>, S<sub>i</sub>)=max<sub>k</sub>{L<sub>cur</sub>(Q<sub>j</sub>, S<sub>k</sub>)}
2. L<sub>cur</sub>(Q<sub>j</sub>, S<sub>0</sub>)≥2

이 과정으로 선택된 세션의 최상위 계층의 패킷이 해당 출력큐로 전송되지 않도록 필터링 한다. 계층 감소 후에는 drop\_intv<sup>i</sup>가 지나기 전에는 다시 계층 감소를 하지 않는데, 이는 효과가 출력큐에 나타나기를 기다리기 위한 것이다.

DROP process는 필터를 통해 출력큐로 유입되는 데이터를 바로 필터링할 수 있으므로 그 효과를 빨리 볼 수 있고, 혼잡이 계속된다면 여러 번의 계층 감소를 통해 혼잡을 일찍 해소할 수 있다.

ADD process는 출력큐가 "unloaded"상태일 때, 가장 낮은 누적 계층으로 전송하는 세션을 선택해서 그 계층을 하나 증가시킨다. 이 때 선택된 세션의 계층은 능동노드가 수신하고 있는 계층보다는 작아야 한다.[4]

1. i<sub>1</sub>=i, if L<sub>cur</sub>(Q<sub>j</sub>, S<sub>i</sub>)= min<sub>k</sub>{L<sub>cur</sub>(Q<sub>j</sub>, S<sub>k</sub>)}
2. L<sub>cur</sub>(Q<sub>j</sub>, S<sub>i</sub>)+1≤L<sub>max</sub>(S<sub>i</sub>)

계층 증가 후에는 add\_intvl<sup>i</sup>이 지나기 전에는 다시 계층 증가를 하지 않는데, 이는 계층 증가가 혼잡을 야기하는지를 검사하기 위한 것이다.

### 3.3 drop\_intvl<sup>i</sup>, add\_intvl<sup>i</sup>

계층을 증가시키거나 감소시키는 주기를 적절히 하지 않으면 세션들의 전송율이 빈번히 바뀌게 된다.

DROP process를 통해 선택된 세션의 최상위 계층을 필터링하기 시작하더라도 이미 출력큐에 적재된 동일 세션의 최상위 계층이 다 전송되기까지 기다려야 한다. 하지만, 그 계층이 출력큐의 어느 위치에 있는가에 따라 달라질 수 있으므로 최대 qmax개의 패킷을 하향 링크의 전송 속도R<sub>i</sub>로 다 전송할 때까지를 drop\_intvl<sup>i</sup>로 한다. drop\_intvl<sup>i</sup>후에도 혼잡이 계속되면 또 다른 세션을 선택해서 필터링하고 drop\_intvl<sup>i</sup>을 기다린다.

$$\text{drop\_intvl}^i = \frac{q_{\max}(\text{packets})}{R_i(\text{packets/sec})}$$

ADD process를 통해 선택된 세션의 계층을 하나 증가시켜 전송하게 되면, 그 출력큐는 물론 수신자까지의 하향의 노드에 혼잡을 야기할 수 있다. 따라서 한 세션의 계층을 증가시킨 후에 또 계층을 증가시키려면 그 세션을 수신하는 수신자까지 전달되는 동안에 혼잡이 발생하는지를 알아야 한다. 혼잡발생을 최소한 감지하기 위해서 능동노드부터 제일 가까운 수신자까지의 홉 거리를 가지고 구할 수 있는데, 이는 수신자가 초기에 그 세션에 조인했을 때 보내온 홉 거리 정보들 중 가장 짧은 홉 거리 정보를 계층상태표에 기록했다가 사용한다. 모든 링크의 RTT는 네트워크의 상태에 따라 다를 수 있고 이를 측정하는 것은 어려우므로 RTT<sub>0</sub>로 같다고 가정하면, add\_intvl<sup>i</sup>는

$$\text{add\_intvl}^i = \text{Hop}_{\text{closest\_receiver}} \times RTT_0$$

이다. 증가된 계층을 수신한 하향의 노드는 그 세션이 전달되는 출력큐를 봐서 "혼잡"상태라면 상향의 노드에게 알리고, "loaded"상태였지만 "혼잡" 가능성이 있으면 필터링하고 상향의 노드에게 알린다. add\_intvl<sup>i</sup>이후에도 여전히 출력큐가 "unloaded"상태이고 하향의 노드로부터 혼잡이라는 응답이 없으면 또 다른 세션을 계층 증가의 타겟으로 선택하고 그 세션에 대해서도 위와 같이 add\_intvl<sup>i</sup>을 기다린다.

## 4. 계층 증가/감소 요구

### 4.1 계층 감소 요구 : DROP\_REQ(S<sub>i</sub>, L<sub>ai</sub>)

상향노드에서 ADD process를 통해 임의로 전송한 패킷을 하향 노드에서 수용할 수 없을 수도 있다. 이런 경우에는 필터링을 하고 있게 되고 상향의 노드에 혼잡을 발생시키지 않는 한 계속 수신한다. 이는 후에 계층 증가의 타겟 세션으로 선택될지 모를 계층을 필터링함으로써 ADD latency를 줄일 수 있기 때문이다. 하지만, 다른 출력큐가 "unloaded"이어서 선택된 계층 또는 하향노드로부터 도착한 계층 증가 요구의 계층이 필터링하고 있는 세션의 계층과 다를 수 있다. 이 경우에는 필터링하고 있는 계층의 감소 요구와 함께 새롭게 발생한 계층 증가 요구를 상향노드에게 전송한다. 계층 감소 요구를 수신한 상향의 노드는 요구를 보내온 링크의 출력큐가 두 계층을 전송해도 여전히 "unloaded"라면 밑에서 사용하지 않는 세션의 계층 우선순위를 낮추어 전송한다. 만약 혼잡이 발생하면, 우선순위가 낮은 세션의 계층을 먼저 감소시킨다.

### 4.2 계층 증가 요구 : ADD\_REQ(S<sub>i</sub>, L<sub>ai</sub>)

ADD process에 의해 선택된 세션의 계층을 상향의 노드로부터 전송받아 필터링하고 있다면 필터링을 멈추고 전송을 시작해 ADD latency를 줄일 수 있다. 하지만 선택된 세션의 계

층을 전송받고 있지 않다면 즉시 계층 증가를 할 수 없다. 이 경우에는 그 계층을 전송하는 송신자에게 계층 증가를 요구해야 한다. 계층 상태 검사 패킷을 통해 그 계층을 전송하는 노드를 알고 있으므로 그 노드까지의 홉 거리 d와 RTT<sub>0</sub>의 곱으로써 detect\_period를 설정한다. detect\_period는 계층 증가가 상향의 노드에 혼잡을 발생시키는지를 감지할 때까지 기다리는 시간이다.

$$d = \text{hop of } L_{[a_i-1]} - \text{hop of } L_{[a_i]} \text{ in LSP Packet}$$

$$\text{detect\_period} = \sum_{i=0}^d RTT_i = d \times RTT_0$$

detect\_period 이후에 증가를 요구한 계층이 전송되지 않는다면 계층 증가가 실패했다고 판단하고, 또 다른 세션을 선택해서 ADD process를 수행한다.

## 5 결론

계층의 크기를 더 많은 수의 동일한 크기로 나누고 혼잡 시 계층 감소 주기를 적절히 함으로써 혼잡에 빠르게 대비할 수 있고 혼잡이 일찍 해소되면, 전송율을 절반으로 떨어트리지 않고 남아 있는 계층으로 데이터를 전달할 수 있으므로 수신자가 인지하는 비디오 품질에 큰 변화를 보이지 않을 수 있다. 계층 증가 역시 하향의 노드에 혼잡을 주지 않으므로 빈번한 전송율 변동을 겪지 않게 된다.

또 상향노드에서 출력큐가 "unload"일 때 전송한 계층이 하향노드에 혼잡을 발생시키지 않는다면 이를 수신해서 필터링함으로써 후에 계층 증가의 타겟 세션으로 선택되었을 때 ADD latency를 줄일 수 있을 것이다.

## 참고 문헌

- [1] S.McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered multicast. *In Proc. of ACM SIGCOMM'96*. Aug. 1996
- [2] X. Li, S. Paul, and M. H. Ammar. Multi-session Rate control for Layered Video Multicast. *In Proc. of Multimedia Computing and Networking 1999*, Jan. 1999
- [3] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like congestion control for layered multicast data transfer. *In Proc. of IEEE INFOCOM'98*, Mar. 1998
- [4] Kiyohide NAKAUCHI, Hiroyuki MORIKAWA, Tomonori AOYAMA, "A Network-Supported Approach to Layered Multicast", *Proceedings of the 2001 IEEE International Conference on Communications Record - Volume 4*, 1227-1231, 20010611
- [5] S. Bhattacharjee, K. L. Calvert, and E. W. Zegura. Network Support for Multicast Video Distribution. Technical Report(GIT-CC-98-16), 1998
- [6] S. Floyd and V. Jacobson. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, Vol.1, No.4, pp.397-413, Aug 1993