

# 동적 임계점을 이용한 스트리밍 서버의 Fail-Over 구조

전성규<sup>o</sup> 차호정  
 연세대학교 컴퓨터과학과  
 (skjeon,hjcha)@cs.yonsei.ac.kr

## A Fail-Over Structure for Streaming Server based on Dynamic Threshold

Seongkwu Jeon<sup>o</sup> Hojung Cha  
 Dept. of Computer Science, Yonsei University

### 요약

본 논문은 스트리밍 환경에서 동적 임계점에 기반한 스트리밍 서버의 고장탐지와 Fail-Over 구조를 제안한다. 지속적이며 투명한 서비스를 사용자에게 지원하기 위해서는 서버 고장에 대한 서비스 중단 시점을 정확하고 빠르게 탐지해야 한다. 기존의 서버 고장탐지 방식은 고정된 시간, 즉 최대 서버 응답시간을 사용하여 서버의 고장을 탐지함으로써 네트워크 상태에 따라 고장탐지 시간이 증가하는 단점을 가진다. 임계점이 서버 응답지연시간을 초과하게 되면 서버에게 고장여부를 확인하기 위한 질의를 보내는데 제안된 기법은 이러한 임계점을 네트워크 상태에 따라 서버응답 지연시간의 변동에 동적으로 적응되도록 하는 방법이다. 동적 임계점을 사용하여 네트워크 상태에 가장 적합한 임계점을 찾을 수 있다는 것을 보이고 성능을 검증한다.

### 1. 서론

스트리밍 환경에서 사용자에게 지원되는 서비스가 지속적이며 투명하게 수행되기 위해서는 서버의 고장여부를 감시해야 하고, 서버의 고장이 발생되었을 때는 신속하고 정확하게 탐지해야 한다. 서버고장을 탐지하게 되면 현재의 세션을 새로운 서버와 연결하기 위해 고장난 서버의 시간과 세션정보를 바탕으로 새로운 세션을 맺어 사용자에게 투명한 서비스를 지속적으로 제공해야 한다. 하지만 서버의 고장탐지를 위한 질의를 계속 보낼 수 없기 때문에 일정한 간격으로 서버와 감시시간의 'heartbeat' 신호를 상호 교환한다[1]. [2], [3]에서는 최대 서버 응답지연시간을 임계점으로 사용하여 이 값을 초과하는 서버 응답지연이 발생될 때만 서버에게 질의를 보냄으로써 서버의 고장여부를 확인하게 되는데, 이러한 방법은 상호간 고장탐지용 프로토콜을 사용해야 하며 네트워크의 상태를 고려하지 않는다. 따라서, 네트워크 상태가 악화될때 서버 고장탐지 시간이 증가하는 단점을 가지고 있다.

본 논문에서는 서버의 응답지연시간을 기준으로 결정되는 임계점을 단순히 최대 값으로 고정하지 않고 네트워크 상태 변화에 따라 적응적으로 이동하는 동적 임계점을 사용하여 서버의 고장을 탐지한다. 동적 임계점을 사용하게 되면 네트워크 상태 변화에 따라 서버의 응답지연시간을 예측할 수 있기 때문에 효율적인 서버 고장탐지를 수행 할 수 있다. 또한, 서버의 고장을 탐지하기 위해 별도의 모니터를 사용하지 않고 세션을 제어하는 프록시를 이용하여 서버의 고장탐지와 Fail-Over를 수행한다. 논문의 구성은 다음과 같다. 2장에서 스트리밍 환경에서 서버의 고장탐지와 Fail-Over를 수행하기 위한 시스템 구성을 제시하고, 3장에서는 동적 임계점에 기반한 Fail-Over 구조에 대해 설명한다. 4장에서는 구현 및 실험결과에 대한 분석을 기술하며, 5장에서 결론을 맺는다.

### 2. 시스템 구성

본 논문에서 제안하고 있는 시스템은 스트리밍 표준 프로토콜인 RTSP[4], RTP[5]를 사용하는 서버와 클라이언트 및 중간에서 모든 세션에 대한 제어와 관리를 담당하는 프록시 서버로 구성되어 있다[그림1].

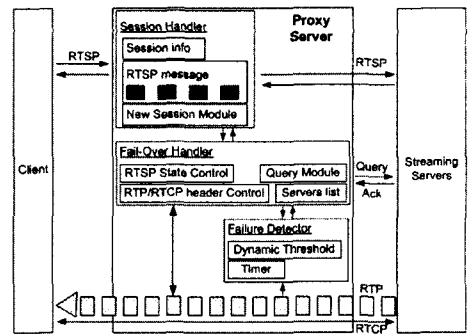


그림 1. 스트리밍 환경의 Fail-Over 시스템 구성

스트리밍 환경에서 서버의 고장탐지를 수행하기 위한 프록시는 세션 관리자와 Fail-Over 관리자 및 고장탐지 모듈을 포함해야 한다. 세션 관리자는 RTSP를 통한 세션제어를 담당하고 Fail-Over를 대비하여 클라이언트 세션 요청 정보를 저장하고 있다. Fail-Over 관리자는 현재 세션의 RTSP 상태와 RTP 헤더 정보를 저장하고 있으며 서버에 대해 고장여부를 확인하기 위한 질의모듈을 포함한다. 또한, Fail-Over가 발생한 뒤 RTP 헤더정보를 수정하는 역할도 담당한다. 고장탐지 모듈은 RTP 패킷 도착간격을 계산하고 서버의 응답시간이 임계점을 초과하게 되면 Fail-Over 관리자에게 알려주는 역할을 한다. 본 논문에서는 이러한 임계점을 네트워크 상태에 따라 변동시켜 서버의 고장을 빠르게 탐지할 수 있도록 제안한다. 프록시는 집단으로 구성된 스트리밍 서버들에 대한 정보를 보관하고 클라이언트는 프록시를 통해 서버와 세션을 연결해야 한다.

본 연구는 한국과학재단에서 지원하는 특정기초연구사업으로 수행하였음 (과제번호: R01-2002-000-00141-0).

### 3. 동적 임계점에 기반한 Fail-Over 구조

다음은 서버의 고장탐지를 위한 동적 임계점에 대해 설명하고 스트리밍 환경에 적합한 Fail-Over 구조를 기술한다.

#### 3.1 동적 임계점

서버가 네트워크 상태 불량이나 프로세서의 고장 등으로 인해 서비스를 수행하지 않게 되면 고장으로 판단될 수 있다. 서버의 고장판단 시점은 응답 전송의 중단으로 확인 가능하지만 이것이 네트워크 상태가 악화되어 전송 지연으로 나타날 수 있기 때문에 완전히 중단될 때까지 기다릴 수 없다. 임계점을 두어 전송 지연 시간을 초과하게 되면 서버에게 질의를 보내어 서버의 고장여부를 확인하게 된다. 네트워크 상태가 나쁠 경우 최대 응답지연시간을 임계점으로 사용하게 되면 서버의 고장탐지 시간이 증가하게 된다[2][3]. 따라서, 네트워크 상태에 따라 변동되는 동적 임계점(DT)이 필요하다. 그림2는 동적임계점 DT를 구하기 위한 알고리즘이다.

$$DT_{i+1} = \begin{cases} DT_i - \frac{(DT_i - Cur_i)}{e^{\frac{1}{n(D_i - Cur_i)}}} & \text{if } (D_i > Cur_i), \\ Cur_i + \frac{MaxRDT - DT_i}{e^{\frac{1}{n(Cur_i - D_i)}}} & \text{if } (D_i \leq Cur_i) \end{cases}$$

$$\text{elseif } (Cur_i \geq MaxRDT) \begin{cases} DT_i - \frac{(DT_i - Cur_i)}{e^{\frac{1}{n(D_i - Cur_i)}}} \times \frac{1}{Cur_i} & \text{if } (D_i > Cur_i), \\ Cur_i + \frac{Cur_i}{e^{\frac{1}{n(Cur_i - D_i)}}} & \text{if } (D_i \leq Cur_i) \end{cases}$$

그림 2. 동적 임계점 알고리즘

서버가 응답하는 메시지 P<sub>i-1</sub>와 P<sub>i</sub>의 도착 시간차이를 Cur<sub>i</sub>라고 할 때 최대 패킷 도착 간격 MaxRDT는 MAX(Cur<sub>i-1</sub>, Cur<sub>i</sub>)가 된다. 이때 i번째 동적 임계점 DT<sub>i</sub>는 Cur<sub>i</sub>와 MaxRDT를 비교하여 결정된다. Cur<sub>i</sub>가 MaxRDT보다 작은 경우, 현재 네트워크상태가 안정되어 있다는 것을 의미하기 때문에 DT<sub>i+1</sub>의 값을 Cur<sub>i</sub>와 비교하여 변경하였다. Cur<sub>i</sub>가 MaxRDT를 초과하게 되는 경우, 네트워크상태가 좋지 않다는 것을 의미한다. 따라서, Cur<sub>i+1</sub>은 증가할 가능성이 높기 때문에 DT<sub>i+1</sub>의 증가폭이 Cur<sub>i</sub>를 기준으로 변경하였다.

속도와 정확도는 고장탐지의 효율성을 고려한 요소이다[3]. DT를 사용하게 되면 MaxRDT와 Cur<sub>i</sub>의 차이를 만큼 변동되기 때문에 효율적인 임계점을 찾을 수 있다. 그림 3은 패킷이 임계점 DT와 MaxRDT 시간 이내에 도착했을 때와 초과했을 때, 다음에 적용된 임계값의 차이를 보여준다. 그림 3의 (a)는 패킷이 두 임계점 시간 이내에 도착했을 경우로써, 다음 패킷 도착 시간에 적용될 임계점의 값이 MaxRDT보다 DT가 작기 때문에 고장탐지에 대한 속도가 증가하게 되는 것을 설명한다. 그림 3의 (b)에서는 Cur<sub>i+1</sub>이 MaxRDT를 초과하여 패킷이 도착하는 경우이다. Cur<sub>i+1</sub>을 다음 임계점에 반영하게 되면 DT는 증가하게 된다. 패킷 도착 간격이 증가했다는 것은 네트워크 상태가 좋지 않다는 것을 의미하여 Cur<sub>i+2</sub>도 증가 할 수 있기 때문에 증가된 DT는 MaxRDT보다 질의를 보낼 확률이 감소되는 것이다. 따라서 서버에게 질의를 적게 보냄으로써 DT의 정확도가

증가할 수 있게 된다.

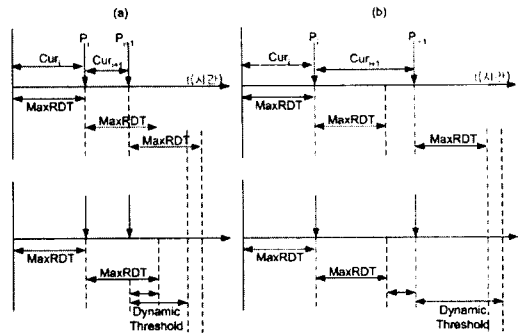


그림 3. 동적임계점과 최대 패킷 도착간격에 대한 비교

#### 3.2 Fail-Over 구조

스트리밍 환경에서 서버의 고장탐지는 효율성을 고려해 볼 때 정확성 보다는 속도가 우선되어야 한다. 스트리밍 환경에서 Fail-Over를 수행할 경우 새로운 세션을 설립하기 위해 소요되는 시간이 고장탐지 보다 많은 시간을 요구하기 때문이다.

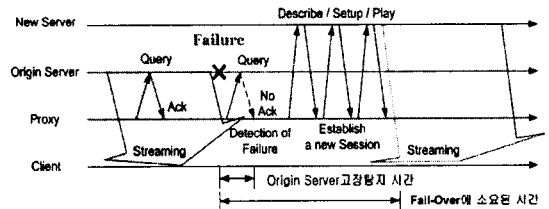


그림 4. 스트리밍에서의 Fail-Over 과정

그림 4는 스트리밍에서 서버에 대한 Fail-Over 과정을 나타내고 있다. Fail-Over 과정은 다음과 같다. 먼저 서버의 응답지연 시간이 임계점을 초과하게 되면 서버에게 질의를 보낸다. 질의에 대한 응답이 없게 되면 서버의 고장으로 판단하여 Fail-Over를 시작한다. 새로운 서버와 세션을 맺은 후 현재 저장된 세션 이후부터 새로운 서버에게 요청함으로써 Fail-Over가 완료된다.

그림 5는 멀티미디어 스트리밍 환경에서 Fail-Over에 대한 구조로써 Fail-Over는 프록시에서 서버의 고장탐지로부터 시작되는데, RTP 패킷 도착 지연 시간이 동적 임계점을 초과한 후 수행하게 된다. 현재 세션 상태를 확인하여 PLAY 상태일 경우에만 스트리밍 서버에 대한 질의를 보낸다. 질의는 RTSP의 "GET PARAMETER"를 이용하여 전송하고 스트리밍 서버의 응답이 오지 않을 경우 Fail-Over를 수행한다. 하지만, 서버의 응답이 왔을 경우 현재의 RTP 패킷 도착 지연시간이 클라이언트의 지연 감내 시간을 초과하게 되면 스트리밍 서버에 대한 Fail-Over를 강제적으로 수행한다. 이것은 스트리밍 서버의 네트워크가 정상적으로 작동하면서 서비스만 제공하지 않는 경우를 대비하기 위한 것으로 클라이언트에 대한 QoS를 보장한다. Fail-Over를 수행한 뒤 새로운 스트리밍 서버로부터 전송되는 패킷의 헤더는 기존의 서버에서 보낸 유류부하 형식, 순서, 시간 및 소스에 대한 정보를 동일하게 바꿔줘야 클라이언트가 지속적인 서비스를 받을 수 있다.

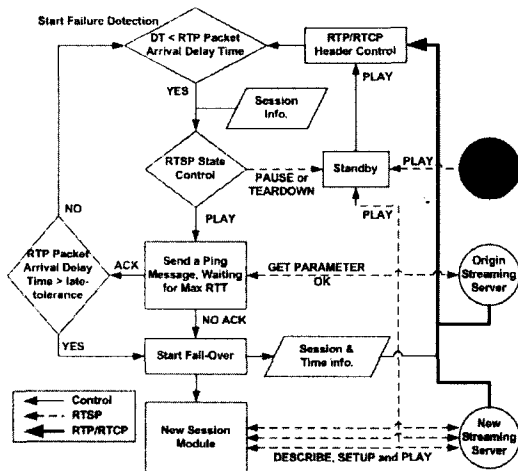


그림 5. 스트리밍 Fail-Over 구조

이다. 동적 임계점은 최대 패킷 도착 간격의 평균 0.091초보다 빠르며 질의를 보내게 되는 임계점 초과 횟수도 동적 임계점이 적었다. 따라서, 동적 임계점이 고장탐지의 효율성 측면에서 좋다는 것을 알 수 있다. 동적 임계점을 통해 Fail-Over를 실시한 결과, 클라이언트에서는 다른 서버와 세션을 재연결하지 않고 프록시를 통해 지속적으로 서비스를 받고 있다는 것을 확인하였다.

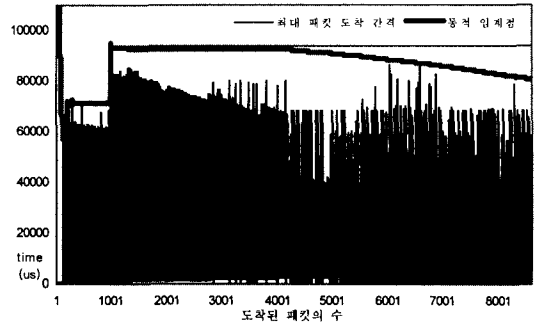


그림 7. 패킷 도착 간격에 따른 동적/정적 임계점 비교

#### 4. 구현 및 실험

동적 임계점을 이용한 Fail-Over에 대한 실험은 QTSS 4.1[6]와 QuickTime Player 4.0를 사용하였으며 프록시는 리눅스 환경에서 구축되었다. 고장탐지는 스트리밍 서버에서 RTP를 통해 전송되는 미디어 패킷이 존재하지 않으면 서버의 고장으로 판단하도록 구현하였다. 동적 임계점의 초기값은 클라이언트의 미디어 요청 정보를 이용하는데, 비디오 코딩 표준에 따른 비트 전송률을 응용하였다.

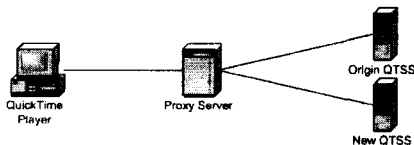


그림 6. Fail-Over 실험 환경

그림 6은 하나의 세션에 대해 두개의 서버를 이용하고 프록시 서버에서 스트리밍 서버에 대한 고장탐지와 Fail-Over를 수행토록 구성한 것이다. 동적임계점의 성능을 확인하기 위해 하나의 세션에 대해서 실험하였다. QTSS 4.1은 하나의 프로세스에서 여러 세션을 지원하고 있기 때문에 세션 하나에 대해 질의를 보냄으로써 프로세스와 네트워크의 고장을 동시에 탐지할 수 있다. 프록시 서버는 스트리밍 서버들에 대한 정보를 미리 저장하고 있다. Fail-Over은 서버의 네트워크 선을 임의적으로 차단시켜 클라이언트에서 지속적이며 투명하게 서비스를 받는지에 대한 실험을 수행하였다.

그림 7은 최대 패킷도착 간격과 동적 임계점의 변화를 나타내고 있는 실험 결과이다. 동적 임계점은 초기값 0.2초로 시작하여 118번째 패킷까지 감소하다가 패킷 도착시간이 최대가 될 때 증가하였다. 이후 7번의 증가와 감소를 반복하다 979번째 패킷부터는 감소하기 시작하여 네트워크 상태가 안정됨에 따라 동적 임계점이 감소되고 있는 모습을 보여 주고 있다. 반면, 정적 임계점으로 사용되는 최대 패킷 도착 간격은 초기 패킷 도착 간격의 증가로 인해 패킷 979번째 까지 8번 임계점을 초과하게 되는데 이후 약 0.094초에서 임계점이 고정되었다. 평균 패킷 도착 간격은 0.005초이며 동적 임계점의 평균은 0.088초

#### 5. 결론

본 논문에서 제안한 동적 임계점을 이용하면 서버의 고장을 빠르게 탐지할 수 있으며 잘못된 질의를 보내는 횟수도 최대 패킷 도착 지연시간을 이용할 때와 유사하여 고장탐지에 대한 효율적인 측면에서 효과적이라는 것을 확인하였다. 또한, 별도의 모니터를 사용하지 않고 스트리밍을 제어하는 프록시를 이용하여 서버의 고장을 탐지하였으며 서버와 클라이언트에 투명하게 작동하였다. Fail-Over는 단순히 클라이언트에서 지속적으로 서비스를 받는지 확인을 통해 수행하였지만, 향후 스트리밍 서버의 고장이 네트워크에서 발생되었는지 혹은 프로세스에서 발생되었는지 정확히 구분하기 위한 질의 설계와 다른 서버를 선택하기 위한 방법 및 여러 세션에 대한 효율적인 관리방법에 관한 연구를 추가적으로 수행할 필요가 있다.

#### 참고 문헌

- [1] A.Robertson, "Linux-HA Heartbeat System Design," Proceedings of the 4th Annual Linux Showcase & Conference, Atlanta USA, Oct 2000.
- [2] C.Fetzer, M.Raynal and F.Troneal,"An Adaptive Failure Detection Protocol," Dependable Computing, 2001 Proc. 2001 Pacific Rim International Symposium on, 2001 page 146-153.
- [3] I.Gupta, T.Chandra and G.Goldszmidt, "On Scalable and Efficient Distributed Failure Detectors," In proc. 20th Annual Symp. on Principles of Distributed Computing, page 170-179, ACM press 2001.
- [4] H.Schulzrinne, A.Rao and R.Lanphier, "RTSP : Real Time Streaming Protocol," RFC 2326, April 1998.
- [5] H.Schulzrinne, S.Casner, R.Frederick and V.jacobson, "RTP : A Transport Protocol for Real-Time Applications," RFC 1889, January 1996.
- [6] 'QTSS : Quick Time Streaming Server,' URL <http://www.apple.com/quicktime/products/qtss/>.