

네트워크가상환경에서 데드레커닝 알고리즘의 성능 해석 및 향상에 관한 연구

심광현⁰ 김종성

한국전자통신연구원 가상현실연구개발센터 (khshim, joskim)@etri.re.kr

A Study On Performance Analysis and Improvement of Dead-Reckoning Algorithm in Networked Virtual Environment

Kwang-Hyun Shim⁰ Jong-Sung Kim

VR Center, Electronics and Telecommunications Research Institute

요 약

본 논문에서는 온라인 게임과 같은 대규모 네트워크가상환경에서의 서버와 클라이언트간에 교환되는 정보의 대부분을 차지하는 공유객체의 이동에 대한 데드레커닝 알고리즘을 마코프 체인 (Markov Chain)을 이용하여 모델링하고 그것의 성능을 해석한다. 그리고 사용자측면에서는 거의 성능저하가 없으면서 전체적으로 네트워크부하를 줄여서 서버가 보다 많은 클라이언트를 처리해 줄 수 있도록 하기 위한 데드레커닝 문턱값의 제어기법들을 소개한다.

1. 서 론

네트워크가상환경(Networked Virtual Environment)이란 네트워크를 통해 분산되어 있는 데이터들을 기반으로 하여 컴퓨터에 의해 시뮬레이션 되는 가상의 세계를 의미한다[1]. 사용자들에 실제로는 멀리 떨어져 있더라도 네트워크 가상환경 안에서는 서로 실시간으로 상호작용을 할 수가 있다. 이러한 기술은 군사 시뮬레이션, 공동설계, 사이버 교육 등 다양한 분야에 응용[2],[3]되고 있으며 특히 최근 우리나라에서는 게임에 응용되어 많은 대규모 온라인게임(MMPOG)들이 개발되고 있다.

그러나, 이러한 시스템에서 하나의 서버는 수 백 또는 수 천명의 클라이언트들을 처리하여야 하므로 과도한 네트워크 부하가 걸리게 되는데 이를 줄이기 위한 방법들 중의 하나가 데드레커닝 (dead reckoning) 알고리즘이다 [4],[5]. 데드레커닝 알고리즘에서는 클라이언트들과 서버는 공유객체의 상태값의 오차(실제값과 예측값의 차)가 문턱값(threshold)이상일 때만 상대방에게 갱신파킷을 보낸다. 받는 쪽에서는 갱신파킷을 받기 전에는 가장 최근에 받은 갱신파킷을 가지고 계속해서 예측을 하게 된다. 이와 관련하여 많은 연구들이 진행되어 있어 왔으나 [2],[3],[6], 기존의 연구들에서는 문턱값이 고정되어 있다고 가정하고 있다. 문턱값은 갱신파킷의 발생주기와 비례하는데 일반적으로 가상환경 내에서 공유객체에게서 멀리 떨어져 있는 클라이언트보다는 가까이 있는 클라이언트에게 보다 자주 갱신파킷을 보내는 것이 합리적일 것이다. 이것은 클라이언트가 가까이 있는 공유객체일수록 자주 상호작용을 하기 때문이다.

이에 본 논문에서는 먼저 데드레커닝 알고리즘을 마코프 체인(Markov Chain)을 이용하여 모델링하여 그것의

성능을 해석하고 공유객체에서의 거리에 따라 문턱값을 제어하여 갱신파킷의 전송율에 차등을 줌으로써 전체적으로 네트워크 부하를 줄일 수 있는 기법을 제안한다.

2. 데드레커닝 알고리즘의 모델링

한편, 서버와 클라이언트간에 교환되는 네트워크 패킷의 대부분이 공유객체들의 이동에 따른 상태갱신파킷들이다. (온라인게임에서는 70-80%를 차지함) 이를 근거로 다음과 같이 모델을 가정한다.

- i) 일단 1차원 이동만을 가정한다. 2차원 또는 3차원의 경우의 비슷하게 확장이 가능할 것이다.
- ii) 상태 갱신 패킷은 공유객체의 위치와 이동방향에 대한 정보를 가진다. 이동시의 속도는 일정하다고 가정한다.
- iii) 충분히 작은 일정 단위시간 동안의 이동거리를 단위거리로 하며 모든 시간과 거리들은 이를 기본 단위로 하여 측정한다.
- iv) p 는 단위시간 후에 공유객체의 위치오차가 그대로 유지될 확률이고 q 는 오차가 증가(감소)할 확률이다. K 는 문턱값으로서 오차가 이 값이 되면 갱신파킷이 보내져서 오차는 다시 0 이 된다.

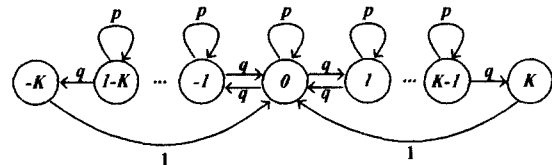


그림 1. 데드레커닝 알고리즘에 대한 마코프 체인

3. 데드레커닝 알고리즘의 성능해석

위의 모델의 근거로 데드레커닝 알고리즘의 성능을 해석하기 위해 먼저, $X(n)$ 을 시간 n 에서의 오차값이라고 가정하고 $y_i(n)$ 을 다음과 같이 정의한다.

$$y_i(n) = \Pr\{X(n) = i / |X(j)| < K, j = 0, 1, \dots, n-1\},$$

$$n = 0, 1, 2, \dots$$

그때, 다음과 같은 상태방정식을 얻을 수 있다.

$$y_i(n) = \begin{cases} p \cdot y_i(n-1) + q(y_{i-1}(n-1) + y_{i+1}(n-1)), & i = -K+2, -K+1, \dots, K-2 \\ p \cdot y_i(n-1) + q \cdot y_{i-1}(n-1), & i = K-1 \\ p \cdot y_i(n-1) + q \cdot y_{i+1}(n-1), & i = -K+1 \end{cases} \quad (1)$$

$$n = 1, 2, \dots$$

그리고 아래와 같이 정의하면

$$Y(n) = [y_{K-1}(n), y_{K-2}(n), \dots, y_{-K+2}(n), y_{-K+1}(n)]^T$$

$$n = 0, 1, 2, \dots$$

$$Y(0) = [0, \dots, 0, y_0(0) = 1, 0, \dots, 0]^T,$$

식 (1) 은 다음과 같이 표현될 수 있다.

$$Y(n+1) = M(n)Y(n) = \left(\prod_{i=0}^n M(i) \right) Y(0), \quad (2)$$

$$n = 1, 2, \dots$$

$$M(n) = \begin{cases} M_0, & n = 0, 1, \dots, K-1 \\ M_1, & n = K, K+1, \dots \end{cases}$$

여기서

$$M_0 = \begin{bmatrix} 0 & q & 0 & \dots & 0 & 0 \\ 0 & p & q & \dots & \vdots & 0 \\ 0 & q & p & \dots & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & q & 0 \\ 0 & \dots & 0 & q & p & 0 \\ 0 & 0 & \dots & 0 & q & 0 \end{bmatrix}, \quad M_1 = \begin{bmatrix} p & q & 0 & \dots & 0 & 0 \\ q & p & q & \dots & \vdots & 0 \\ 0 & q & p & \dots & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & q & 0 \\ 0 & \dots & 0 & q & p & q \\ 0 & 0 & \dots & 0 & q & p \end{bmatrix}$$

특히, 정수 $n < K$ 에 대해, 식 (2)는 다음과 같고

$$Y(n) = M_0^n Y(0), \quad n = 1, 2, \dots, K-1$$

그때, 다음과 같은 식을 얻을 수 있다.

$$y_i(n) = \sum_{j=0}^{\lfloor \frac{n-|i|}{2} \rfloor} \binom{n}{n-|i|-2j} \binom{|i|+2j}{j} p^{n-|i|-2j} q^{j+2j}$$

여기서 $\lfloor n \rfloor$ 는 n 이하의 최대 정수이다.

그리고, 정수 $n \geq K$ 에 대해서는

$$Y(n) = M_1^{n-K} Y(K-1), \quad n = K, K+1, \dots$$

계속해서, 시간 n 에서 오차가 최초로 문턱값에 도달할 확률은 다음과 같고

$$Z_K(n) = \Pr\{|X(n)| \geq K / |X(i)| < K, i = 0, 1, n-1\}$$

$$= q \cdot y_{K-1}(n-1) + q \cdot y_{-K+1}(n-1)$$

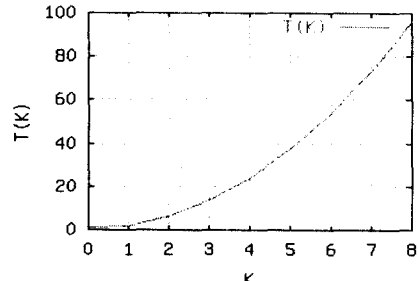
$$= 2q \cdot y_{K-1}(n-1)$$

문턱값이 주어졌을 때 상태갱신패킷의 평균 발생주기는 다음과 같이 구해진다.

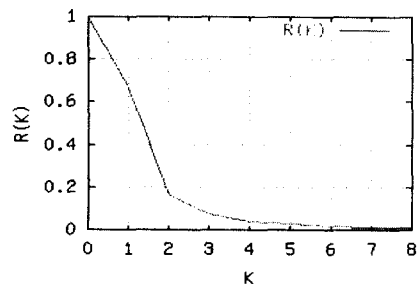
$$T(k) = \sum_{n=0}^{\infty} n \cdot Z_K(n) = 2q \sum_{n=K-1}^{\infty} (n+1) \cdot y_{K-1}(n)$$

마침내, 하나의 클라이언트에 대해 필요한 상태 갱신패킷의 평균 전송율을 다음과 같이 얻을 수 있다.

$$R(K) = \frac{1}{T(k)} = \frac{1}{2q \sum_{n=K-1}^{\infty} (n+1) \cdot y_{K-1}(n)} \quad (3)$$



(a)



(b)

그림 2: (a) 평균 문턱값 도달시간 = 평균 갱신패킷 발생주기, (b) 평균 갱신패킷 발생율 ($p=q=1/3$)

그림 2. 는 컴퓨터 통해 데드레커닝 문턱값의 변화에 따른 평균 문턱값 도달시간과 평균 갱신패킷 발생율을 계산한 것이다.

4. 데드레커닝 문턱값의 제어

이 절에서는 갱신패킷의 전송율을 줄여서 전체적으로 네트워크 부하를 줄이기 위한 문턱값의 제어기법을 소개한다. 문턱값은 거리에 따라 단순 증가함수이어야 한다. 는 조건 하에 문턱값의 제어식을 다음과 같이 정한다.

$$K(x) = (K_2 - K_1) \left(\frac{x}{X_A} \right)^m + K_1, \quad (4)$$

$$0 < x \leq X_A, \quad m \geq 0$$

여기서 K_1 은 최소 문턱값 (기존의 알고리즘에서 사용된 고정된 문턱값) 이고 K_2 는 최대 문턱값 (최대 허용 오차) x 는 공유객체와 클라이언트와의 거리이고 X_A 는 공유객체의 유효반경으로 이것을 넘으면 클라이언트에게 전혀 갱신패킷을 주고 받지 않는다.

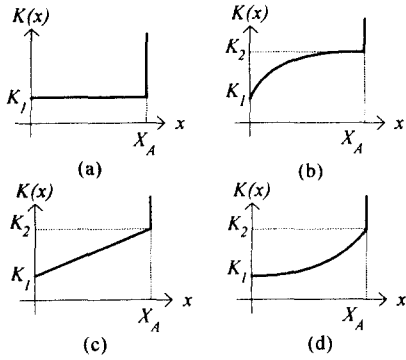


그림 3: 데드레커닝 문턱값을 제어식: (a) $m=0$, (b) $m=0.5$, (c) $m=1$, (d) $m=2$

일반적으로 그림 3의 제어식에서 (d) $m=2$ 의 경우가 가장 바람직하다고 할 수 있다. 이것은 가까이 클라이언트에게는 전송율을 크게 감소시키지 않으면서 상대적으로 분포가 많은 멀리 떨어져있는 클라이언트에 대한 전송율을 급격하게 감소시킴으로써 전체적인 부하를 크게 감소시킬 수 있기 때문이다.

식 (4)의 제어식을 사용할 때 하나의 공유객체가 하나의 클라이언트에게 보내야 하는 갠신패킷의 평균 전송율은 다음과 같이 구할 수 있다.

$$S = \int_0^{x_A} \frac{2\pi x}{\pi x_A^2} R(K(x)) dx \quad (5)$$

그림 4.는 문턱값을 제어하였을 때 거리에 따른 문턱값, 도달시간, 패킷 전송율의 변화를 컴퓨터로 계산한 결과들이다.

5. 결론

본 논문에서는 데드레커닝 알고리즘의 성능을 해석하였고 성능향상을 위한 데드레커닝 문턱값을 제어기법을 소개하였다. 이와 같이 거리에 따라 데드레커닝 문턱값을 조절하면 사용자측면에서의 성능은 큰 변화 없이 네트워크 부하를 줄일 수 있게 된다. 이것이 실제로 쓰일 때는 문턱값이 거리에 따라 연속적으로 제어 되기는 힘들기 때문에 불연속적으로 제어가 되어야 할 것이다.

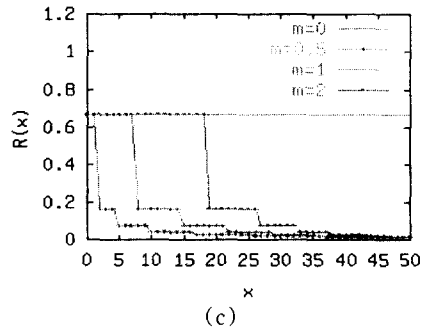
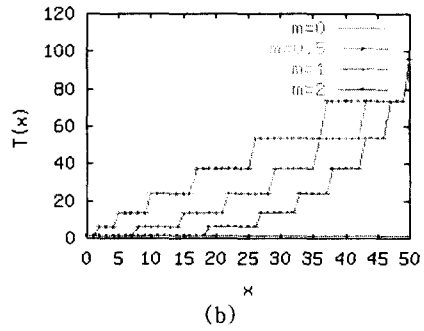
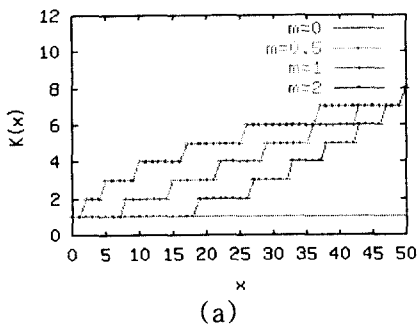


그림 4: (a) 데드레커닝 문턱값, (b) 평균 문턱값 도달시간, (c) 평균 갠신패킷 전송율, 여기서, $p=q=1/3$, $x_A=50$, $K_1=1$, $K_2=8$

참고문헌

- [1] S. Singhal and M. Zyda, Networked Virtual Environments: Design and Implementation. MA:Addison-Wesley, July 1999.
- [2] Deb Fullford, Distributed Interactive Simulation: It's Past, Present, and Future, Proc. of the 1996 Winter Simulation Conf. pp. 179-185, 1996.
- [3]. M. R. Macedonia, D. P. Pratt and etc., NPSNet: A Network Architecture For Large-Scale Virtual Environments, PRESENCE, vol. 3, no.4, pp. 265-287, 1994.
- [4]. 심광현, 외, "분산가상환경을 위한 네트워크 및 서버 기술", 정보과학회지, 제 19권, 5호, 2001년 5월.
- [5]. 양광호, 심광현, 외, "온라인 게임 서버의 기술 동향" 전자통신동향분석, 제 16권 제 4호 2001년 8월.
- [6]. S. Singhal, Effective remote modeling in large-scale distributed simulation and visualization environments, Ph.D. dissertation Dept. of Computer Science, Stanford University Aug. 1996.