트래픽 데이터의 시계열 분석을 위한 데이터 마이닝 기법

김 철°, 이 도 헌
전남대학교 전산학과
전화 : 062-530-0110 / 핸드폰 : 011-9936-5688

Data Mining Technique for Time Series Analysis of Traffic Data

Cheol Kim, Doheon Lee
Dept. of Computer Science, Chonnam National University
E-mail : ckim@dbcore.chonnam.ac.kr

Abstract

This paper discusses a data mining technique for time series analysis of traffic data, which provides useful knowledge for network configuration management. Commonly, a network designer must employ a combination of heuristic algorithms and analysis in an interactive manner until satisfactory solutions are obtained. The problem of heuristic algorithms is that it is difficult to deal with large networks and simplification or assumptions have to be made to make them solvable. Various data mining techniques are studied to gain valuable knowledge in large and complex telecommunication networks. In this paper, we propose a traffic pattern association technique among network nodes, which produces association rules of traffic fluctuation patterns among network nodes. Discovered rules can be utilized for improving network topologies and dynamic routing performance.

I. 서론

네트워크에 대한 업무 의존도가 높아지고, 운용 범위가 크게 확장되면서, 신뢰성과 효율성을 갖춘 네트워크 구성 관리의 필요성이 크게 부각되었다. 네트워크 구성 관리(configuration management)란 네트워크 노드의 위상을 효율적으로 설정, 관리하는 것이다 [1]. 일반적으로 네트워크 위상을 최적화시키는 과정은 다음과 같다. 네트워크 위상 설계자가 경험적 알고리즘(heuristic algorithm)의 조합을 통해서 위상을 설계한 후 최적의 위상을 얻을 때까지 반복적으로 분석한다. 별(star), 원(ring)과 같은 특정 위상을 이용하여 최적의 위상을 찾는 방법[2]도 있고, 특정 위상에 기반을 두지 않는 방법도 있다. Perturbation기법은 후자의 대표적인 예로 branch exchange 방법, cut-saturation 방법, concave branch elimination 방법이 있다 [3]. 세 가지 중에서 cut-saturation 알고리즘이 가장 좋은 결과를 산출하며, 계산 비용이 다른 두 개보다 더 효율적이다. 대규모의 네트워크에 이와 같은 방법을 적용하기 위해서는 단순화(simplification)단계와 가정(assumption)단계를 걸쳐야 한다. 따라서 실질적으로 대규모의 네트워크에 적용하는 것은 어렵다. 최근 대규모의 네트워크에서 망 관리에 유용한 지식을 획득하기 위한 데이터 마이닝 기법이 연구되고 있다. 예를 들어 TASA(Telecommunication Network Alarm Sequence Analyzer)시스템은 네트워크 망에서 발생하는 알람들간의 상호 연관성을 밝혀 장애 관리를 하고 있다 [4,5,6,7].

본 논문에서는 네트워크 위상을 개선하기 위한 새로운 요소 기술로서, 네트워크 노드간 트래픽 패턴 연관성 기법을 제안한다. TASA시스템의 경우 알람들간의 상호 연관성을 추출한 반면에, 본 논문에서 제안한 네트워크 노드간 트래픽 패턴 연관성 기법은 각 노드의

트래픽 증감 패턴을 추출한 후, 노드간 트래픽 증감 패턴의 연관성을 찾는다. 각 네트워크 노드의 트래픽 상태 정보를 정기적으로 수집하고, 시간별 노드 트래픽 상태 정보를 타임 윈도우로 분할한다. 분할된 입력 자료에서 각 네트워크 노드의 트래픽 증감 패턴을 찾은 후, 노드간 트래픽 증감 패턴의 연관 규칙을 생성한다. 생성된 연관 규칙을 통해 노드의 트래픽 패턴 변화를 예측하고 연관성 있는 노드들을 재배치하거나 동적 라우팅을 통해서 패킷 전송로를 재 설정함으로써 네트워크 위상을 개선시킨다. 본 논문의 구성은 다음과 같다. 2장에서는 트래픽 상태, 트래픽 상태 이벤트, 트래픽 패턴, 노드간 트래픽 연관 규칙에 대해서 정의한다. 3장에서는 노드간 트래픽 패턴 연관 규칙 알고리즘 및 전체 노드간 트래픽 서열 정보에서 노드간 트래픽 패턴 연관 규칙을 추출하는 예를 제시한다. 마지막으로 4장에서는 결론을 내린다.

II. 노드간 트래픽 패턴 연관 규칙

2.1 트래픽 패턴 추출

정의 1. 트래픽 상태(Traffic State)

트래픽 상태는 네트워크 망에 존재하는 노드의 버퍼 상태를 나타낸다. 트래픽 상태는 노드의 패킷 포화율(packet saturation ratio)에 따라 <표1>과 같이 4단계로 정의한다.

표1. 노드의 트래픽 상태

단계	상태
1	노드의 버퍼가 75%이상 ~ 100%으로 채워져 있는 경우
2	노드의 버퍼가 50%이상 ~ 75%미만으로 채워져 있는 경우
3	노드의 버퍼가 25%이상 ~ 50%미만으로 채워져 있는 경우
4	노드의 버퍼가 25%미만으로 채워져 있는 경우

□

네트워크 노드의 트래픽 상태를 미리 정의하여 일반화함으로써 노드의 버퍼 변화에도 유연하게 대응할 수 있다.

정의 2. 트래픽 상태 이벤트(Traffic State Event)

트래픽 상태 이벤트는 (노드 ID, 트래픽 상태)로 정의한다. 노드 ID는 네트워크 망에 존재하는 노드 식별자이고, 트래픽 상태는 노드의 트래픽 상태를 나타낸다. □

<그림 1>은 트래픽 상태 이벤트 서열을 나타낸다.

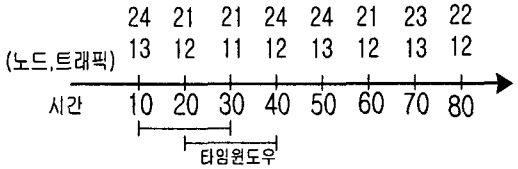

그림1. 트래픽 상태 이벤트 서열

정의 3. 트래픽 패턴(Traffic Pattern)

트래픽 상태 이벤트의 집합을 TS 라고 할 때, 트래픽 패턴 $TP = \langle t_1 \rightarrow t_2 \rightarrow \cdots\cdots \rightarrow t_k \rangle$, $t_i \in TS$ 라고 정의한다. 단, 트래픽 상태 이벤트 간에 순서가 존재하며, →는 시간의 흐름을 나타낸다. 또한 트래픽 패턴 안에 다른 트래픽 상태 이벤트의 개입을 허용하지 않는다. □

<그림 1>에서 각 노드별 트래픽 패턴을 찾기 위해서는 트래픽 상태 이벤트 서열을 일정 크기의 타임 윈도우(time window)로 분할해야 한다. 분할은 타임 윈도우 사이의 경계 데이터의 손실을 막기 위해서 중첩되게 분할한다. 만약 타임 윈도우를 30으로 분할하는 경우, 1번 노드 <3→2>패턴이 3번 발생되고, 2번 노드 <4→1>패턴이 2번 발견됨을 알 수 있다.

2.2 노드간 트래픽 패턴 연관 규칙

정의 4. 노드간 트래픽 연관 규칙(NTPAR)

트래픽 상태 이벤트의 집합 Σ의 원소로 구성된 서열들의 조합을 T , 노드를 N_i라고 할 때, 특정 시간 안에 발생하는 노드간 트래픽 패턴 연관 규칙(NTPAR)은 다음과 같이 표현한다.

$\{N_1.t_1, N_2.t_2, \cdots\cdots, N_{m-1}.t_{m-1}\} \rightarrow N_m.t_m$

□

규칙의 타당성 척도로서 지지도(support)와 신뢰도(confidence)를 다음과 같이 정의한다.

정의 5. 지지도(support)

트래픽 패턴의 지지도는 전체 트래픽 이벤트 서열(TE)을 특정 윈도우 크기만큼 중첩 되게 나눈 것 중 특정 노드간 트래픽 패턴이 그 안에 존재하는 비율이다.

$$Support(\alpha) = \frac{|w \in W(TE, win) | \alpha \leqslant w|}{|W(TE, win)|}$$

□

단, $W(TE, win)$는 전체 트래픽 이벤트 서열을 타임 윈도우로 분할한 집합을 나타낸다.
$|W(TE, win)|$ 는 집합의 원소 수를 나타낸다. win은 타임 윈도우의 크기이며, α 는 노드간 트래픽 패턴이다. $\leqslant$는 트래픽 패턴이 윈도우 안에서 포함됨을 의미한다.

정의 6. 신뢰도(confidence)

노드간 트래픽 패턴 연관 규칙(NTPAR)을 $\alpha \Rightarrow \beta$라고 할 때 α를 규칙의 조건부라 하고 β를 규칙의 결론부라 하면 조건부를 만족하는 트래픽 노드 패턴에 대해 결론부까지 동시에 만족하는 트래픽 노드 패턴의 비율을 의미한다. 단, β는 α의 superset이다.

$$Confidence(\beta) = \frac{Support(\beta)}{Support(\alpha)}$$

□

III. 노드간 트래픽 패턴 연관성 기법의 구현

3.1 노드간 트래픽 패턴 연관성 기법 알고리즘

전체 노드의 트래픽 상태 이벤트 서열 중 각 네트워크 노드의 트래픽 패턴을 찾은 후, 노드들의 트래픽 패턴간의 연관 규칙을 생성한다. 알고리즘은 <그림 2>와 같다.

```
(1)  Node_Traffic_Association( )
(2)  begin
       /* 전체 트래픽 상태 정보를 읽는다 */
(3)    NodeTrafficRead();
       /* 각각의 시간별 노드 트래픽 정보를 하나의
          아이템으로 추출한다. */
(4)    L1 = MakeFirstTrafficSet();
(5)    for (k = 2; Lk-1 ≠ ∅; k++) do
(6)    begin
         /* Lk-1 로부터 새로운 후보 노드간 트래
            픽 패턴을 만든다 */
(7)      Ck = MakeCandidateTrafficSet( Lk-1);
         /*후보 노드간 트래픽 패턴의 개수를
         센다*/
(8)      CountCandidateTrafficSet( Ck);
         /* Ck가 최소 지지도를 만족하는 경우
            Lk에 할당한다. */
(9)      Lk = FilteringTrafficSet( Ck);
(10)     PatternList[k] = Lk
(11)   end
(12)   index = 2;
(13)   while (PatternList[index] != NULL) do
(14)   begin
(15)     WriteRule(index);
(16)     index++;
(17)   end
(18) end
```

그림 2. 노드간 트래픽 패턴 연관 규칙 알고리즘

(3)에서 트래픽 상태 이벤트 서열을 읽고, 크기가 1인 후보 트래픽 집합, 즉 시간별 노드 트래픽 상태 정보를 하나의 트래픽 집합으로 (4)에서 생성한다. 생성된 트래픽 집합을 기반으로 (7)에서 후보 트래픽 집합을 만들고, (8)에서 후보 트래픽 집합의 지지도를 계산한 후, (9)에서 최소 지지도 이하인 후보 트래픽 집합을 제거한다. (10)에서 트래픽 패턴의 크기를 증가시킨 후 (7)~(9)과정을 반복한다. (13)~(17)에서 발견된 규칙을 출력한다.

3.2 노드간 트래픽 패턴 연관 규칙 도출

전체 트래픽 상태 이벤트 서열이 <그림 2>와 같이 존재한다고 가정한다.

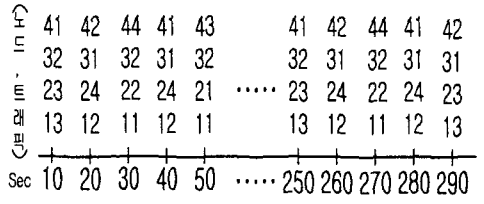

그림 2. 전체 트래픽 상태 이벤트 서열

만약 2분 30초안에 발생하는 노드간 트래픽 패턴 연관 규칙을 찾는 경우, 전체 트래픽 상태 이벤트 서열을 150의 타임 윈도우로 나누게 된다.

<그림 3>은 전체 트래픽 이벤트 서열을 타임윈도우로 나눈 입력자료이다.

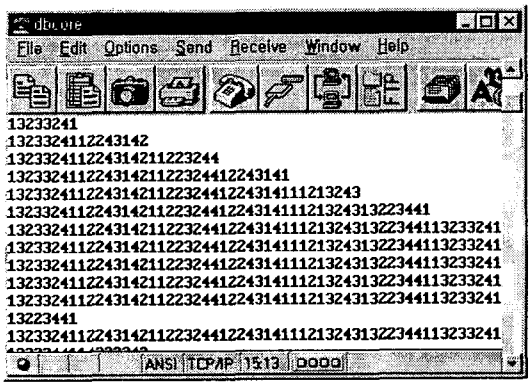

그림 3. 타임 윈도우로 나누어진 전체 트래픽 이벤트 서열

타임 윈도우로 나누어진 입력 자료를 가지고 실행을 시키면 <그림 4>와 같은 노드간 트래픽 연관 규칙을 생성한다. 단, 최소 지지도는 30으로 하였다.

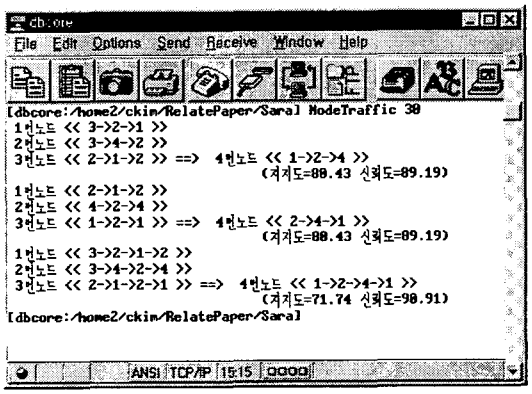

그림 4. 노드간 트래픽 연관 규칙

찾아진 노드간 트래픽 연관 규칙을 통해서 150초안에 1번 노드가 <<2→1→2>>로 변화하고, 2번 노드가 <<4→2→4>>로 변화하고, 3번 노드 <<1→2→1>>로 변화할 때, 4번 노드가 <<2→4→1>>로 변화되는 것을 예측할 수 있다. 이 규칙의 지지도는 80.43%이고, 신뢰도는 89.19%임을 알 수 있다. 결과를 바탕으로 3번 노드의 버퍼 상태와 4번 노드의 버퍼 상태가 서로 연관되어져 있음을 알 수 있다. 결국 3번 노드의 버퍼 성능을 향상시키거나, 3번 노드와 관련된 라우팅 테이블 정보를 재 설정함으로써 네트워크 망을 좀 더 효과적으로 구성할 수 있다.

Ⅳ. 결론

본 논문에서는 네트워크 구성을 개선시키기 위한 새로운 요소 기술로서, 데이터 마이닝 관점에서 네트워크 위상을 개선하는 네트워크 노드간 트래픽 패턴 연관성 기법을 제안하고 구현하였다. 각 네트워크 노드에 대한 트래픽 상태 정보를 수집하고, 시간별 노드 트래픽 상태 정보를 타임 윈도우로 분할하였다. 분할된 입력 자료에 노드간 트래픽 연관성 기법을 적용하여, 각 네트워크 노드의 트래픽 증감 패턴을 찾고, 노드의 트래픽 증감 패턴간의 연관 규칙을 생성하였다. 생성된 연관 규칙을 기반으로 노드의 트래픽 변화를 예측하였다.

본 논문에서 제안한 노드간 트래픽 패턴 연관 규칙은 연관성 있는 노드들을 재배치하거나 동적 라우팅을 통해서 패킷 전송로를 재 설정함으로써 네트워크 위상을 개선시킬 수 있다.

참고문헌

[1]. [ITU-T,1992i] ITU-T. Recommendation X.700: Management framework for Open Systems Interconnection(OSI) for CCITT applications, September 1992

[2] R. L. Sharma, "Network Topology Optimization" , *New York: Van Nostrand Reinhold, 1990.*

[3] Ricardo F. Garzia and Mario R. Garzia, "Network Modeling, Simulation, and Analysis." *New York: M. Dekker Inc., 1990*

[4]. TASA : Telecommunication Alarm Sequence Analyzer, or "How to Enjoy Faults in Your Network". In Proceedings of the 1996 IEEE Network Operations and Managment Symposium (NOMS '96), 520-529.

[5]. K. Hatonen, M. Klemettinen, H. Mannila, P. Ronkainen, and H. Toivonen. "Knowledge discovery from telecommunication network alarm databases." In 12th Int'l Conference on Data Engineering (ICDE'96), pp. 115 -- 122, New Orleans, Louisiana, February 1996.

[6]. Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. "Discovering frequent episodes in sequences." In Proc. of the Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95), Montreal, Canada, August 1995.

[7]. M. Klemettinen, H. Mannila, and H. Toivonen. "Rule discovery in telecommunication alarm data." Journal of Network and Systems Management, June/July 1998.