제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

경량 RMON AGENT 시스템 개발

이준형*, 박진원*, 김명균*
*울산대학교 컴퓨터정보통신공학부
comfuny@cnlab.ulsan.ac.kr, zwsonic@cnlab.ulsan.ac.kr, mkkim@mail.ulsan.ac.kr

Development of Light-weight RMON Agent System

Joon-Hyung Lee*, Zin-Won Park*, Myung-Kyun Kim*
*School of Computer Engineering & Information Technology, University of Ulsan

요 약

RMON Agent 시스템은 각각의 서브 네트워크에 하나씩 있으며, 해당 네트워크에서의 패킷들을 수집, 분석하여 네트워크관리에 필요한 정보를 추출한다. 이러한 네트워크 관리에 필요한 프로토콜별 또는 호스트별 패킷양에 대한 관리정보는 RMON의 MIB에 정의되어 있다. RMON Agent 시스템은 해당 네트워크상의 모든 패킷을 수집, 분석해야 하므로 장비의 성능이 떨어질 경우에는 대규모 네트워크에서의 패킷 분실 우려가 있다. 본 논문에서는 RMON Agent의 대규모 네트워크에서의 패킷분실을 최소화하고 효율적인 관리정보를 수집할수 있도록 설계하였다.

1. 서론

오늘날 네트워크 기반 시스템이 점점 커지고 대규모화 됨에 따라서 네트워크의 환경은 점점 복잡해지고 어려워진 반면, 상호 연결성은 더욱 강화되어 네트워크에 대한 신속하고 효율적인 관리의 필요성이 요구되고 있다. 이러한 상황은 기존의 SNMP 기반의 네트워크 관리 시스템의 한계를 넘어선 관리 형태를 필요로 한다. 기존의 SNMP기반의 네트워크 관리 시스템은 각 Agent가 탑재된 장비가 자신의 처리결과만을 보유하고 각 Agent가 수집한 정보를 중앙 Manager로 가져와서 분석하는 방식이다. 따라서 네트워크 트래픽이 중앙 호스트로 집중되게 되면서 전체 네트워크에 부담을 주게 된다. RMON Agent 시스템은 이러한 중앙관리에 따른 호스트의 부담을 줄여주고, 한 세그먼트 전체에서 발생하는 트래픽을 파악하게 해준다. 즉, 전체 발생 트래픽, segment에 연결된 각 호스트의 트래픽, 호스트간의 트래픽 발생현황을 알려준다. RMON Agent가 수집해야 할 관리 정보들은 RMON MIB에 정의되어 있다. 그러나 RMON은 MAC계층 까지만 분석이 가능하므로 프로토콜별 트래픽 또는 네트워크에 영향을 주는 어플리케이션이 어떤 것인지 알 수 없다. 이런 점을 보완하기 위해서 등장한 것이 RMON2 이다. RMON2에 추가된 것은 프로토콜별 현황, 네트워크 계층의 호스트별 트래픽 등이 있다. 호스트간의 트래픽양을 파악함으로써 프로토콜별, 어플리케이션별 네트워크 트래픽 점유율을 알 수 있게 된다 [1, 2]. 그러나 RMON Agent 시스템은 네트워크 상의 모든 패킷을 수집, 분석해야 하므로 수집해야 할 관리정보가 많은 경우, 모든 RMON의 기능을 제공하기에는 RMON이 탑재된 장비 본래의 기능에 악영향을 줄수 있으며, 초당 수천 개의 패킷이 전송되는 큰 규모의 네트워크에서는 패킷을 분실할 우려가 있다.

본 논문에서는 대규모 네트워크에서의 패킷 분실율을 최소화 하면서 패킷을 수집, 분석할 수 있는 RMON Agent 시스템을 설계, 구현하였다. 수집하는 관리정보는 RMON2의 MIB를 기반으로 하며, 수집하는 정보는 RMON Agent가 탑재된 하드웨어의 성능에 따라 유연하게 조정 가능하도록 설계하였다. 또한 Agent 시스템의 패킷분석에 따른 로드를 최소

화 하면서 패킷을 놓치는 것을 방지하도록 설계하였으며, 수집정보의 중복을 최소화하기 위한 RMON 관리정보 구조를 설계하였다.

관리정보에 대한 접근과 처리가 빠르고 용이하도록 메모리 상에 관리정보를 유지하도록 하였으며, 관리정보에 대한 접근 및 처리가 용이하도록 XML 포맷을 사용하였고 Agent와 Manager간의 메시지 송수신은 XML이 이용된다.

2장에서는 Agent가 관리해야 할 관리정보에 대해서 기술하고 있으며, 3장에서는 본 Agent 시스템의 구조와 RMON Agent 관리정보에 대해서 기술한다. 4장에서는 구현된 RMON Agent에 대해서 기술하고, 5장에서는 결론 및 향후계획에 대해서 기술한다.

2. RMON Agent 관리정보

RMON Agent가 관리해야할 관리정보는 RMON의 MIB와 RMON2의 MIB에 정의되어 있으며, RMON의 MIB는 2계층인 MAC 주소에 기반 하여 패킷을 분석 및 관리정보 수집을 하고 있으며, RMON2의 MIB는 3계층이상 응용계층까지의 호스트별 또는 프로토콜별 패킷을 수집 및 분석한 관리정보를 유지한다. 본 논문에서는 관리자가 이해하기 편리한 상위계층을 지원하는 MIB2를 기반으로 RMON Agent를 구현하였다. RMON2의 MIB는 9개의 그룹으로 나뉘어서 관리되며 하드웨어의 성능에 따라서 관리자가 임의로 선택하여 구현가능하다. MIB는 테이블 형태로 이루어져 있으며 각 그룹은 제어테이블과 데이터테이블로 구성되어 있다. 제어테이블은 RMON Agent 시스템이 수집 분석을 위한 파라미터를 설정하는 역할을 하고, 제어테이블의 파라미터를 기반으로 데이터를 수집하여 데이터 테이블에 저장된다. RMON Agent 시스템이 수집 MIB 관리정보 그룹은 다음과 같다.

■ Protocol Dir Group : Agent가 모니터링하고 추가, 삭제, 구성할수 있는 프로토콜에 대해 정의하고 있다. 프로토콜을 나타내는 protocolDirTable과 최종갱신시간을 나타내는 protocolDirLastChange로 구성된다.

■ Protocol Distribution Group : 네트워크 세그먼트에서의 프로토콜별 데이터 전송정보를 정의한다. 프로토콜별 수집에 대해서 설정하는 제어테이블인 protocolDistControlTable과 이 테이블에 의해서 수집된 통계정보를 저장하는 protocolDistStatsTable로 구성되어 있다.

■ Address Map Group : 서브네트워크 상에서의 각 호스트별 MAC 주소와 와 네트워크주소간의 사상정보를 저장한다. 추가된 호스트와 삭제된 호스트 수를 나타내는 addressMapInserts 와 addressMapDeletes, 데이터수집에 대한 제어테이블인 addressMap ControlTable과 이 제어테이블에 의해서 수집된 주소사상정보를 저장하는 addressMapEntry가 있다.

■ Network Layer Host Group : 각 호스트별 송수신 정보에 대한 통계를 정의한다. 데이터 수집기능을 설정하기 위한 hlHostControlTable과 수집된 정보를 저장하는 nlHostTable로 구성된다. 또한 hlHostControlTable은 Application Layer Host의 제어테이블로도 이용된다.

■ Application Layer Host Group : 앞의 Network Layer Host Group과 같은 기능을 수행하지만 프로토콜별로 수집된다는 점이 다르다. hlHostControl Table에 의해서 수집된 정보는 alHostTable에 저장된다.

■ Network Layer Matrix Group : 서브네트워크 상의 각 호스트별 송수신량에 대한 통계정보를 저장한다. 수집 기능을 위한 제어테이블인 hlMatrix ControlTable과 통계정보를 저장하는 nlMatrixSD Table, nlMatrixDSTable이 있다.

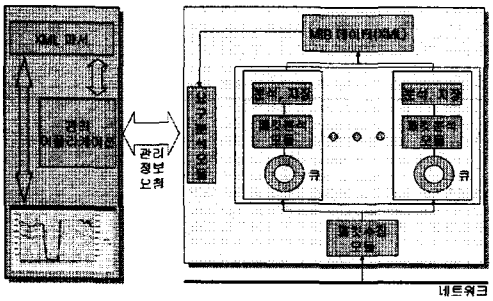
■ Application Layer Matrix Group : 각 호스트별 프로토콜별 전송 통계정보를 정의한다. 제어테이블은 hlMatrixControlTable을 사용하며, 통계정보를 저장하기 위한 alMatrixSDTable 과 nlMatrixDS Table이 있다.

그 외의 그룹들은 수집정보의 중복을 최소화 하거나 Agent의 로드를 최소화하기 위해서 포함시키지 않았다.

3. RMON Agent 시스템의 및 관리정보 구조

3.1 RMON Agent 시스템의 구조

<그림 1>은 RMON Agent 시스템의 전체적인 구조를 나타낸다. 패킷수집모듈은 네트워크로부터 수집한 패킷을 각각의 패킷분석 모듈의 큐에 저장하는 역할을 하며, 패킷분석모듈은 큐에 저장된 원시패킷을 분석하여 메모리상의 관리정보 데이터 구조에 추가한다. 관리정보 데이터는 동적으로 추가, 삭제, 변



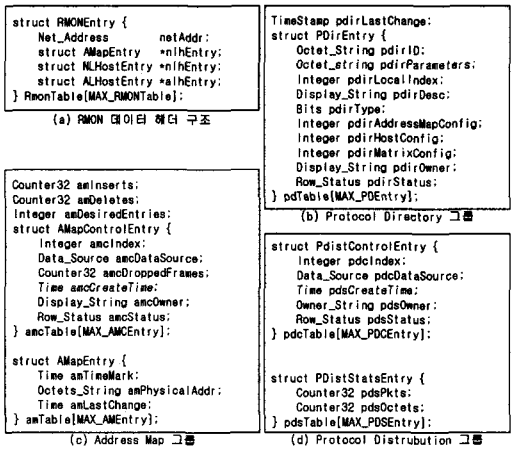
<그림 1> RMON Agent 시스템 구조

경이 가능하다. 관리자가 관리정보를 요청하게 되면 요청정보를 Agent로 보내게 되고 Agent는 각 패킷 분석모듈이 메모리 상에 유지하고 있는 관리정보를 모두 수집하여 XML DOM을 통하여 XML 파일로 변환하고 이것을 관리자에게로 전송하게 되면 관리 어플리케이션은 전송받은 XML 데이터를 XML Parser를 통하여 분석하여 관리정보 또는 그래프로 보여주게 된다. 패킷수집모듈과 패킷분석모듈 및 요구분석 모듈은 각각 스레드로 동작하여 패킷분석에 따른 로드시 패킷 손실을 최소화 하였다. 하드웨어 성능이나 네트워크 성능에 따라서 패킷분석모듈의 스레드의 수나 관리 어플리케이션의 정보요청주기를 유연하게 조정함으로써 누적되는 패킷의 수를 최소화 한다.

3.2 관리정보 구조

본 RMON Agent 시스템에서의 관리정보 데이터는 중복을 최소화 하고 빠른 처리가 가능하도록 설계 되었으며 메모리 상에서 유지하여 관리된다. <그림 2>와 <그림 3>은 RMON Agent 시스템에서 유지하는 관리정보 데이터 구조를 나타낸다.

<그림 2(a)>의 RmonTable은 전체의 RMON 관리정보를 위한 헤더테이블로 MIB의 addressMap, nlHost, alHost에 해당하는 포인터를 가지고, 관리정보의 중복을 최소화하기 위해서 네트워크 주소를 저장한다. 새로운 호스트 주소의 패킷이 수집되면 새로운 RmonTable 항목이 생성되며, 주소의 값으로 hash된 값을 할당하여 중복을 피한다. <그림 2(b)>는 Protocol Directory 그룹을 나타내는 것으로, 각 프로토콜별로 하나의 항목을 갖는다. <그림 2(d)>는 Protocol Distribution 그룹을 나타내며 해당 프로트콜에 대한 통계치 정보를 저장한다. <그림



<그림 2> RMON 관리정보 데이터 구조(1)

2(c)>는 Address Map 그룹을 나타낸다. amcTable은 RMON Agent 시스템이 모니터링 하는 네트워크 하나당 하나의 항목을 갖는데, 본 RMON Agent 시스템은 하나의 서브네트워크에서의 분석을 수행하므로 하나의 항목만을 가진다.

<그림 3(a)>는 Host 그룹에 대한 테이블로써 제어테이블인 nlmcTable과 두개의 데이터 테이블인 nlhTable과 alhTable로 이루어져 있다. nlhTable은 네트워크 주소에 기반 한 트래픽 정보를 수집하고 alhTable은 네트워크 주소와 트랜스포트 계층 정보를 수집한다. <그림 3(b)>는 Matrix 그룹에 대한 정보를 나타낸다. 제어테이블 역할을 하는 nlmcTable과 두개의 nlmSDTable과 almSDTable이 있다. Host 그룹과 마찬가지로 nlmSDTable은 네트워크 주소기반 정보를 수집하고 almSDTable은 네트워크와 트랜스포트, 응용계층의 정보를 수집한다. 두개의 테이블 모두 (송신자, 수신자) 순서로 인덱싱 되며, (수신자, 송신자) 순서로 인덱싱 되는 nlmDSTable과 almDSTable은 시스템의 부하를 줄이기 위해서 구현하지 않았다.

4. RMON Agent 시스템의 구현

본 RMON Agent 시스템은 패킷 수집 및 분석을 담당하는 Agent와 관리용 프로그램인 Manager 로 나누어져 있다. Agent는 윈XP 환경의 Visual C++ 7.0을 이용하여 개발하였으며, 패킷 수집을 위해서 Winpcap 3.0 Beta를 이용하였다.. <그림 4>는 해당 서브네트워크에서의 실시간 트래픽 현황을 보여

```
struct NLHostControlEntry {
  Integer nlhcIndex;
  Data_Source nlhcDataSource;
  Counter32 nlhcNlDroppedFrames;
  Counter32 nlhcNlInserts;
  Counter32 nlhcNlDeletes;
  Integer nlhcNlMaxDesiredEntries;
  Counter32 nlhcAlDroppedFrames;
  Counter32 nlhcAlInserts;
  Counter32 nlhcAlDeletes;
  Integer nlhcAlMaxDesiredEntries;
  Display_String nlhcOwner;
  Row_Status nlhcStatus;
} nlhcTable [MAX_NLHCEntry];

struct NLHostEntry {
  Time nlhTimeMark;
  Counter32 nlhInPkts;
  Counter32 nlhOutPkts;
  Counter32 nlhInOctets;
  Counter32 nlhOutOctets;
  Counter32 nlhOutMacNonUnicast;
  Time nlhCreateTime;
  struct NLMatrixSDEntry *pNLM;
} nlhTable[MAX_NLHEntry];

struct ALHostEntry {
  Time alhTimeMark;
  Counter32 alhInPkts;
  Counter32 alhOutPkts;
  Counter32 alhInOctets;
  Counter32 alhOutOctets;
  Time alhCreateTime;
  struct ALHostEntry *pALH;
  struct ALMatrixSDEntry *pALM;
} alhTable[MAX_ALHEntry];
```

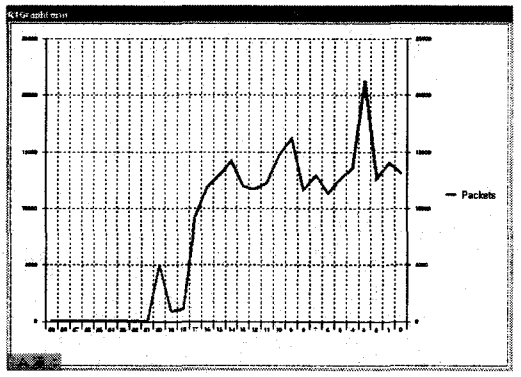
(a) Host 그룹

```
struct NLMatrixcontrolEntry {
  Integer nlmcIndex;
  Data_Source nlmcDataSource;
  Counter32 nlmcNlDroppedFrames;
  Counter32 nlmcNlInserts;
  Counter32 nlmcNlDeletes;
  Integer nlmcNlMaxDesiredEntries;
  Counter32 nlmcAlDroppedFrames;
  Counter32 nlmcAlInserts;
  Counter32 nlmcAlDeletes;
  Integer nlmcAlMaxDesiredEntries;
  Display_String nlmcOwner;
  Row_Status nlmcStatus;
} nlmcTable [MAX_NLHCEntry];


struct NLMatrixSDEntry {
  Time nlmSDTimeMark;
  Net_Address nlmSDDestAddr;
  Counter32 nlmSDPkts;
  Counter32 nlmSDOctets;
  Time nlmSDCreateTime;
  struct NLMatrixSDEntry *pNLM;
} nlmSDTable [MAX_NLMSDEntry];


struct ALMatrixSDEntry {
  Time almSDTimeMark;
  Net_Address almSDDestAddr;
  Counter32 almSDPkts;
  Counter32 almSDOctets;
  Time almSDCreateTime;
  struct ALMatrixSDEntry *pALM;
} almSDTable [MAX_ALMSDEntry];
```

(a) Matrix 그룹

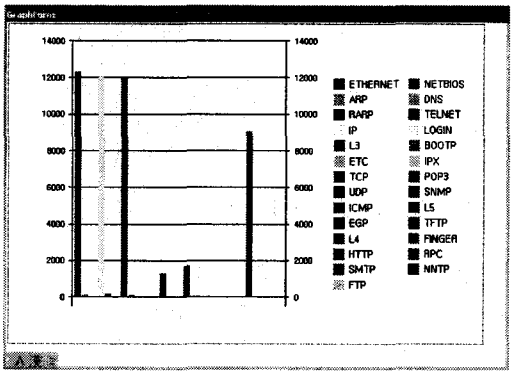
<그림 3> RMON 관리정보 데이터 구조(2)

주는 화면이다.



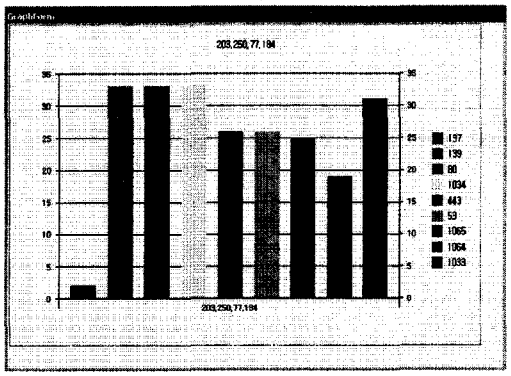
<그림 4> 네트워크 실시간 트래픽양

<그림 5>는 해당 서브네트워크에서의 프로토콜별 트래픽현황을 보여주는 화면이다.



<그림 5> 각 프로토콜별 트래픽양

<그림 6>은 하나의 호스트에 대한 다른 호스트들로부터의 연결 상황 및 트래픽 현황을 보여주는 화면이다.



<그림 6> 호스트별 포트별 트래픽양

5. 결론 및 향후 연구

본 논문에서는 트래픽 양이 많은 대규모 네트워크에서 패킷 분실을 최소화 하면서 트래픽을 분석할 수 있는 RMON Agent의 개발에 대해서 기술하였다. 본 RMON Agent는 패킷 분석도중 분실되는 패킷을 최소화하기 위하여 패킷수집모듈, 패킷분석모듈, MIB 데이터 생성모듈을 모두 분리하여 각각의 스레드로 처리하였고, 네트워크 및 하드웨어의 성능에 따라서 패킷분석 및 저장모듈의 수를 조정할 수 있도록 하였다. 또한 관리정보의 구성을 중복을 최소화하도록 구성하였으며, 관리정보의 빠른 처리를 위하여 메모리 상에 유지하도록 하였다.

RMON Agent 시스템은 네트워크 상의 모든 패킷을 수집, 분석하여 관리자에게 제공함으로써 네트워크에 대한 이상 징후나 침입에 대한 탐지기능을 제공한다. 따라서 본 연구실에서는 현재 구현된 RMON Agent 시스템을 네트워크 기반 침입탐지 시스템으로 사용하기 위한 연구를 진행 중이다. 이를 위해서 현재 RMON 관리정보에 대한 확장과 이상 징후 발견 시 관리자에게 통보하는 기능을 추가할 것이다.

참고문헌

[1] S. Waldbusser, Remote Monitoring Management Information Base (RFC 1757), Feb. 1995.

[2] S. Waldbusser, Remote Monitoring Management Information Base Version 2 (RFC 2021), Jan. 1997.