NETCONF기반 구성관리를 위한 성능향상 방법

Performance Improvement Methods for NETCONF-based Configuration Management

December 21, 2005

Sun-Mi Yoo
sunny81@postech.ac.kr

Distributed Processing and Network Management Lab
Dept. of Computer Science & Engineering, POSTECH

Contents

- 서론
- 관련연구
- NETCONF의 효율적 사용방안
- XCMS 시스템 개발
- 성능 검증
- 결론
- 향후과제

서론 (1)

- 연구 배경
  - 네트워크의 규모가 커짐에 따라 다양한 종류의 장치들에 대한 원격 구성관리가 요구됨
  - IETF의 NETCONF WG은 효율적인 구성관리를 위해 NETCONF 표준안들을 제시해 옴
  - 표준화로 구성관리의 상호연동성 문제가 해결될 수 있음
  - 다수의 장치들을 대상으로 수행하는 구성관리는 성능 측면의 효율성도 중요함

- 연구 동기
  - 표준화 초기단계인 NETCONF는 효율적인 측면에서의 기술적 검토가 많이 부족한 상태
  - NETCONF 표준에 따르면서 효율적으로 구성관리를 실행할 수 있는 방법에 대한 자료가 없음
  - NETCONF 표준에 대한 성능측정 및 분석결과, 효율적 사용방안 부재

서론 (2)

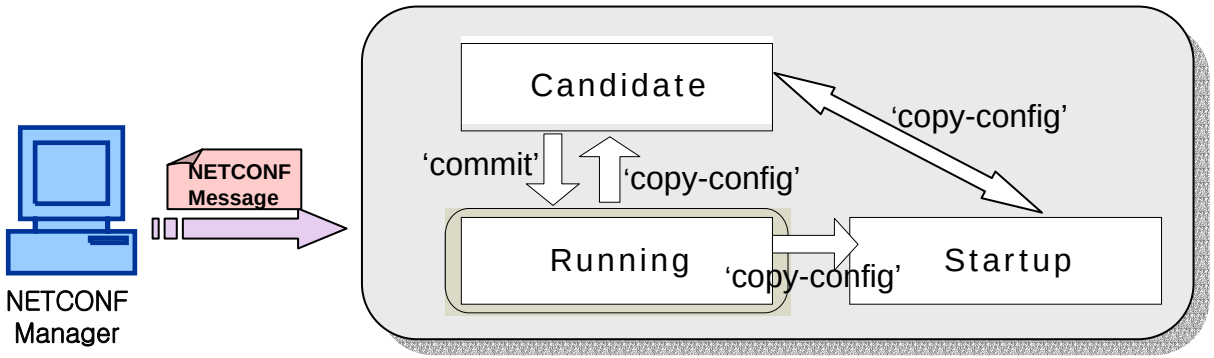
- 연구 목표

1. NETCONF 표준의 성능을 향상하여 구성관리의 효율성을 높일 수 있는 방안 제시
   - ✓ NETCONF의 각 계층별로 성능을 향상 시키는 방안 제시 및 성능 측정
   - ✓ NETCONF 표준을 준수하면서 성능을 향상 시킬 수 있는 방법과 표준에는 포함되지 않지만 성능을 향상 시킬 수 있는 방법
2. 응답시간, 네트워크 사용량, 시스템 자원 소모량 측정 및 해당 변수들을 최소화 시킬 수 있는 방안 제시
3. 성능 향상 방법을 NETCONF 시스템에 적용시키는 방법 제시
   - ✓ 해당 방법들이 적용된 XCMS 구조 제시

관련연구 (1)

- NETCONF(NETwork CONfiguration)
  - 네트워크 구성관리를 위해 구성 관리 프로토콜을 표준화하고 있는 IETF Working Group
  - NETCONF 프로토콜은 장치들간의 상호운영성을 높이기 위해 XML기반으로 정의됨
  - NETCONF는 구성 관리 서비스를 RPC 통신방법으로 제공
  - SOAP over HTTP, SSH, BEEP을 통신 프로토콜로 사용

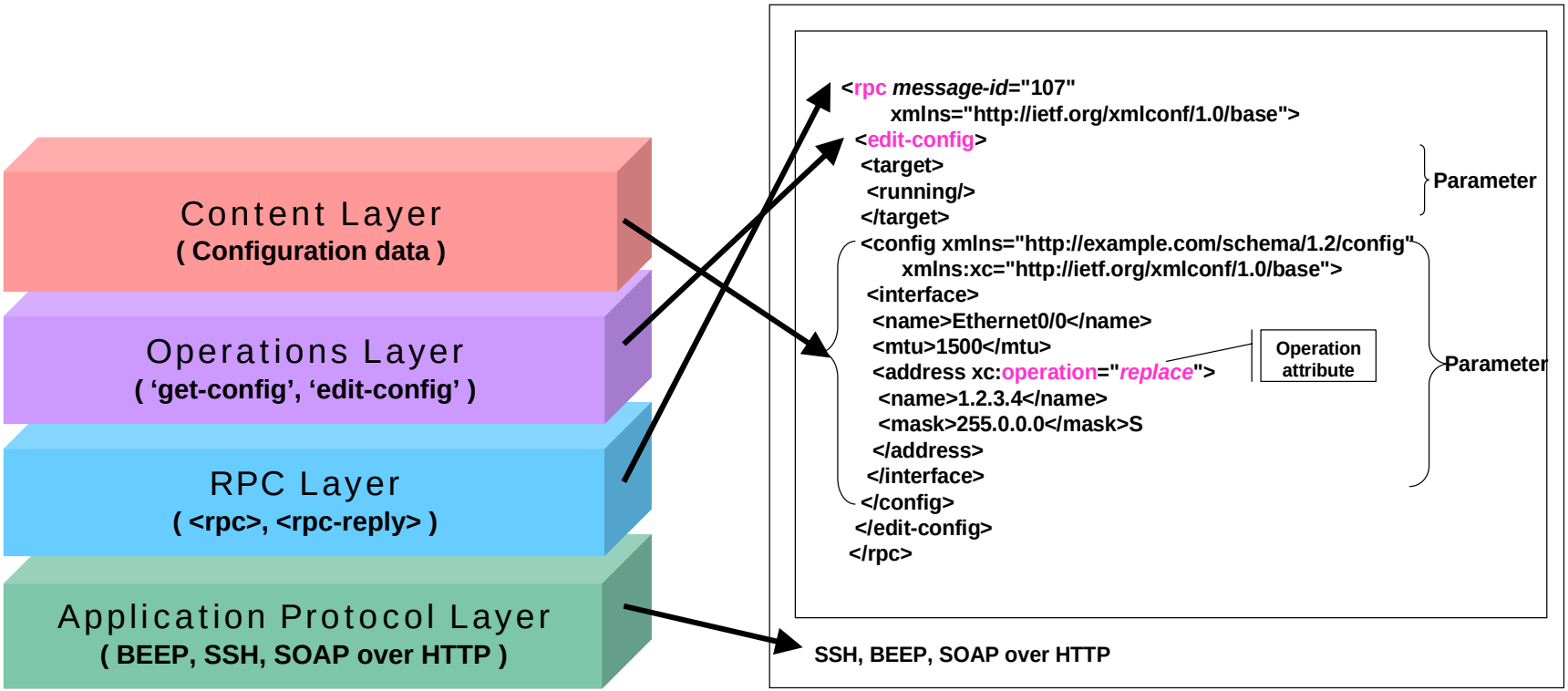
- NETCONF Configuration Model
  - NETCONF는 한 개 이상의 configuration datastore를 요구
  - configuration datastore에서 NETCONF 오퍼레이션 수행



DP&NM Lab
Postech, Korea

관련연구 (2)

- NETCONF 계층
  - NETCONF는 개념적으로 4개의 계층으로 나뉘어 짐
  - 각 계층들은 서로 독립적임

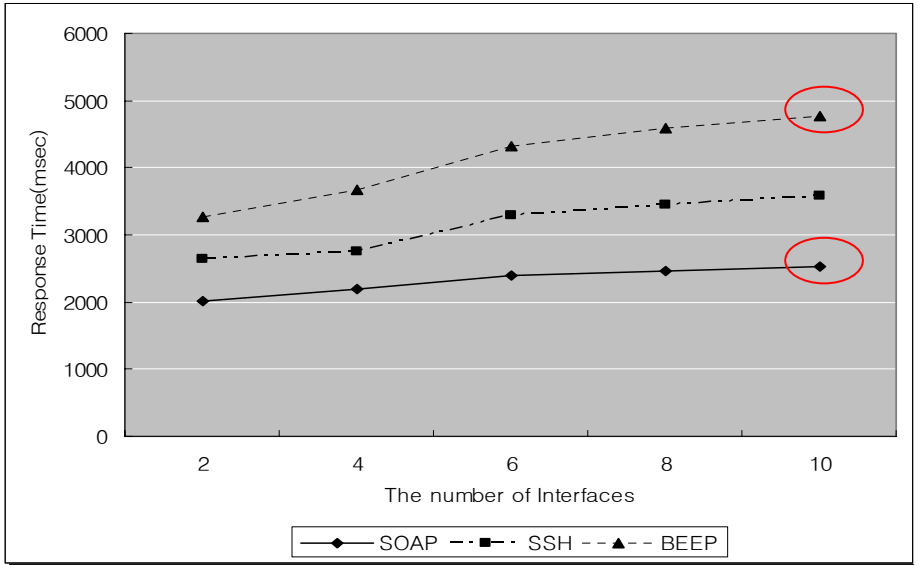


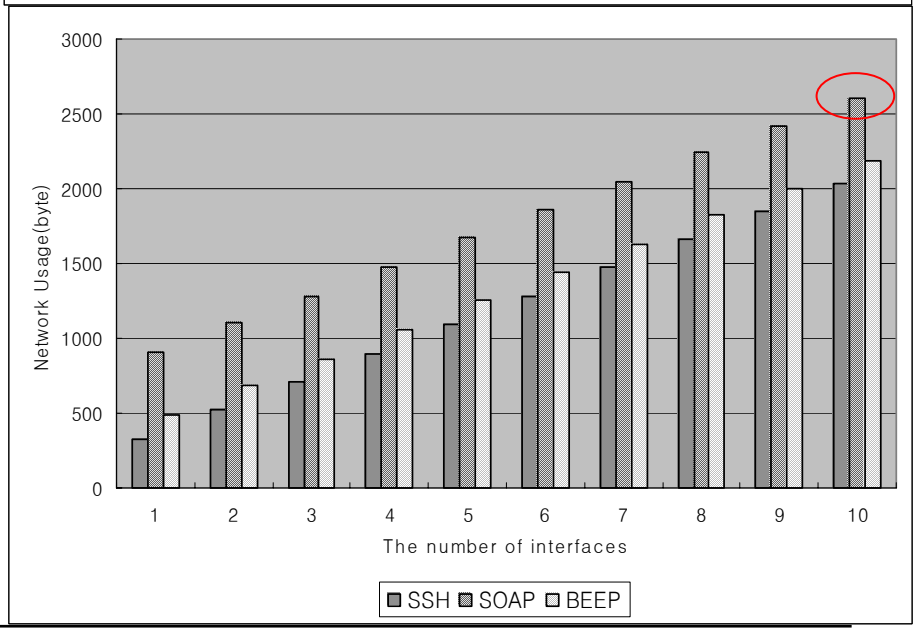
관련연구 (3)

- 네트워크 관리 성능 연구
  - SNMP 와 웹 서비스 기반 네트워크 관리 성능 비교 연구
    - ✓ A. Pras et al [IEEE eTNSM, Vol 1, No 2, 2004]
    - ✓ SNMP를 이용한 관리 시스템과 웹 서비스를 이용하는 관리 시스템의 성능을 비교
    - ✓ 웹 서비스를 기반으로 수행되는 관리시스템의 프로토타입 구현
    - ✓ 관리 대역폭과 응답시간을 측정
    - ✓ 많은 장치의 값을 읽어오는 경우, 웹 서비스가 훨씬 좋은 성능을 보임
  - XML 기반 구성관리 성능 연구
    - ✓ A. E. Nikolaidis et al [IEEE Comm. Mag., Volume 43, Issue 5, 2005]
    - ✓ 홈 네트워크 장비들을 관리하는 구성 관리 시스템의 성능
    - ✓ XML 기반 구성관리의 성능 향상을 위해 압축기법 사용을 제시
      - ➢ Lempel-Ziv 압축 알고리즘
    - ✓ 구성정보 데이터 크기변화에 따른 네트워크 사용량과 응답시간을 측정

Application Protocol Layer (1)

- NETCONF는 다수의 세션(session)과 전송 프로토콜 제공
  - SOAP over HTTP, SSH, BEEP
- 전송 프로토콜별 성능제시 필요
- 성능 분석
  - SOAP 프로토콜의 응답시간이 가장 적음
  - SOAP 프로토콜의 네트워크 사용량이 가장 많음
  - 네트워크 사용량과 프로토콜별 통신방식을 고려 해야함





DP&NM Lab
Postech, Korea

Application Protocol Layer (2)

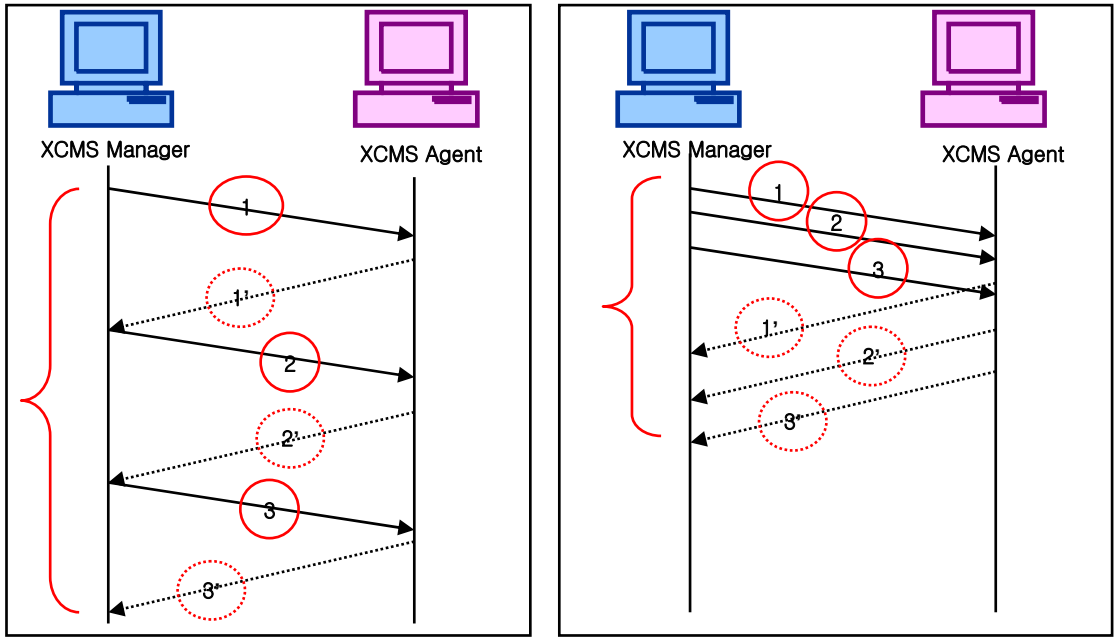
- 문제점
  - 효율적인 구성관리는 네트워크에 적은 영향을 주어야 함
  - NETCONF의 전송 프로토콜을 효율적으로 사용하는 방법이 제시된 자료가 없음
  - NETCONF는 XML 기술을 기반으로 동작함
    - ✓ 적은 데이터에도 태그, 전송 프로토콜 헤더메세지가 추가 됨
  - 구성정보의 양이 큰 경우에는 네트워크 사용량 뿐만 아니라 응답시간에도 영향을 준다
  - 전송하는 데이터의 양을 줄일 수 있는 방안 제시가 필요함

- 방법 제시
  - XML 메세지의 크기를 줄이는 방법들
    - ✓ Ex> binary XML, ZLIB, Lempel-Ziv
  - 전송하려는 NETCONF 메세지에 압축 기법 적용
    - ✓ 반복되는 데이터가 많은 NETCONF 메세지에 적합

DP&NM Lab
Postech, Korea

RPC Layer (1)

- NETCONF 요청 메세지와 응답 메세지를 나타내는 방법
- <rpc>와 <rpc-reply>는 일대일 통신
- pipelining 통신 방법
  - NETCONF의 수행시간을 줄이기 위한 방안
  - 에이전트가 다른 오퍼레이션 수행 시 매니저는 다른 요청 메시지 전송
  - 에이전트는 전송받은 순서대로 수행해야 함

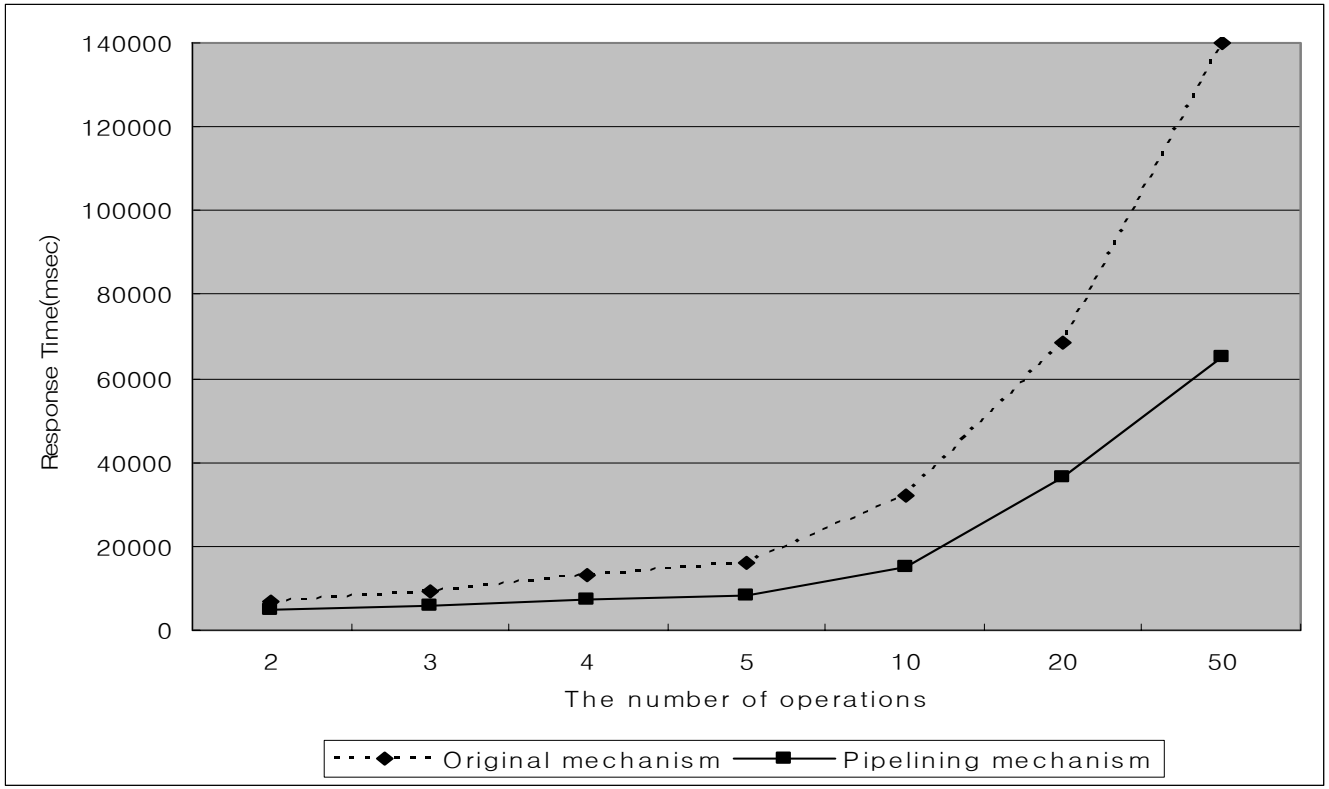


(a) No pipelining (b) Pipelining

DP&NM Lab
Postech, Korea

RPC Layer (2)

- 수행내용
  - Interface의 값을 읽어오는 'get-config' 오퍼레이션 수행
  - 명령어 개수에 따른 변화량들을 측정하기 위해 동일한 명령어 반복 수행



DP&NM Lab
Postech, Korea

RPC Layer (3)

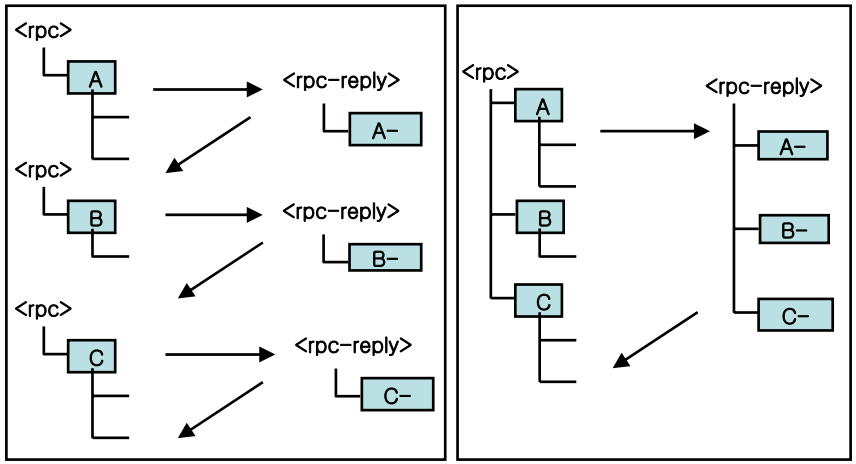
- 문제점
  - NETCONF 오퍼레이션간에는 연관 관계를 지님
    - ✓한개의 오퍼레이션을 완전하게 마치기 위해서는 다른 오퍼레이션의 수행이 요구됨
  - Pipelining 방식 사용시 네트워크 사용량이 증가할 수 있음
  - Pipelining의 이점을 지니며, 네트워크 사용량을 감소시키는 방법 제시 필요

- 방법 제시
  - Multi Command 통신방법
  - 하나의 <rpc>메세지에 연관관계가 있는 여러개의 오퍼레이션을 전송
    - ✓예> 'edit-config'를 <candidate>에 수행 후, 'commit' 수행 필요
  - 에이전트는 메세지내의 오퍼레이션 순서대로 실행
  - 중간에 오류 발생시 수정된 작업들은 rollback 처리

DP&NM Lab
Postech, Korea

RPC Layer (4)

- 효율성 향상
  - 한개의 <rpc>요청에는 단 한개의 <rpc-reply> 메세지 전송
  - 여러 장치들에 대해서 구성관리 작업 수행시 효율적
  - 네트워크 사용량 수행시간 모두 줄이며 성능 향상
  - Rollback 기능 요구



(a) Original mechanism (b) Multi command mechanism

Original NETCONF Request Messages	Multi command NETCONF Request Message
<?xml version='1.0'?> <rpc message-id='1'> <edit-config> <target> <candidate/> </target> <config> <interfaces> <iface operation="merge"> <name>eth1</name> <type>ethernet</type> </iface> </interfaces> </config> </edit-config> </rpc>	<?xml version='1.0'?> <rpc message-id='1'> <edit-config> <target> <candidate/> </target> <config> <interfaces> <iface operation="merge"> <name>eth1</name> <type>ethernet</type> </iface> </interfaces> </config> </edit-config> <commit/> <get-config> <source> <running/> </source> </get-config> </rpc>
<?xml version='1.0'?> <rpc message-id='1'> <commit/> </rpc>	
<?xml version='1.0'?> <rpc message-id='1'> <get-config> <source> <running/> </source> </get-config> </rpc>	

Operations Layer (1)

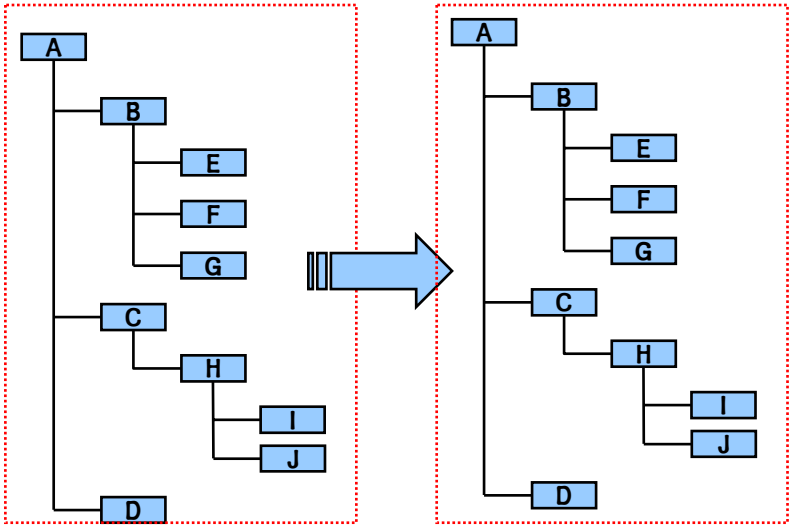
- 'copy-config'의 비 효율성
  - 큰 구성정보에서 아주 적은 부분만을 복사하려 할 때, 전체 데이터를 복사해야 하는 시간적 오버헤드 발생
  - 다른 장치로의 구성 정보 복사 및 특정 부분만을 반영하고 싶을 때, 수행 할 수 있는 방법이 존재하지 않음
  - <candidate>에 복사 후 'commit' 수행 시 전체 내용을 모두 반영
  - 여러대의 매니저로부터 관리가 되는 경우에는 충돌 발생 가능

- 방법제시
  - 'copy-config'에 filtering 방식 적용
  - Filtering을 사용하지 않는 경우 'get-config'에서와 동일하게 동작
    - ✓ No filtering인 경우에는 전체 구성 정보를 대상으로 수행

Operations Layer (2)

```
<?xml version='1.0'?>
<rpc message-id='1'>
  <copy-config>
    <source>
      <running/>
    </source>
    <target>
      <candidate/>
    </target>
  </copy-config>
</rpc>
```

<Request Message>

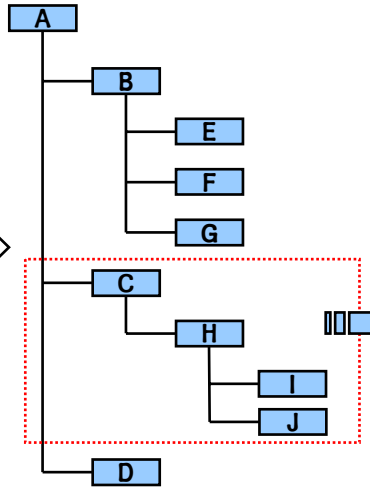


<Running configuration data tree>

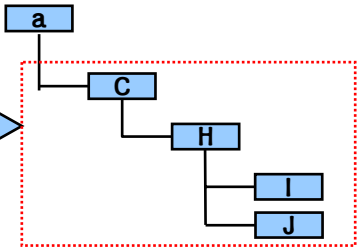
<Candidate configuration data tree>

```
<?xml version='1.0'?>
<rpc message-id='1'>
  <copy-config>
    <source>
      <running/>
      <filter type='subtree'>
        <C/>
      </filter>
    </source>
    <target>
      <candidate/>
    </target>
  </copy-config>
</rpc>
```

<Request Message>



<Running configuration data tree>



<Candidate configuration data tree>

Contents Layer (1)

- 구성정보들을 XML형태로 표현하는 기능
- 구성정보들의 속성 및 구조는 각 구현물들에 의존적임
- XPath vs Subtree filtering
  - XPath
    - ✓ 표현 방식이 다소 복잡하고 어려움
    - ✓ DOM에 모든 메세지를 올려서 파싱하는 방식으로 수행 됨
  - Subtree filtering
    - ✓ NETCONF에서만 사용되는 필터링 방식
    - ✓ 규칙이 아직까지는 명확하지 않고 배우기 어려움
    - ✓ 사용 예제가 많이 부족함
  - 두 방식의 기능적 차이점 이외에 성능적 차이도 중요함

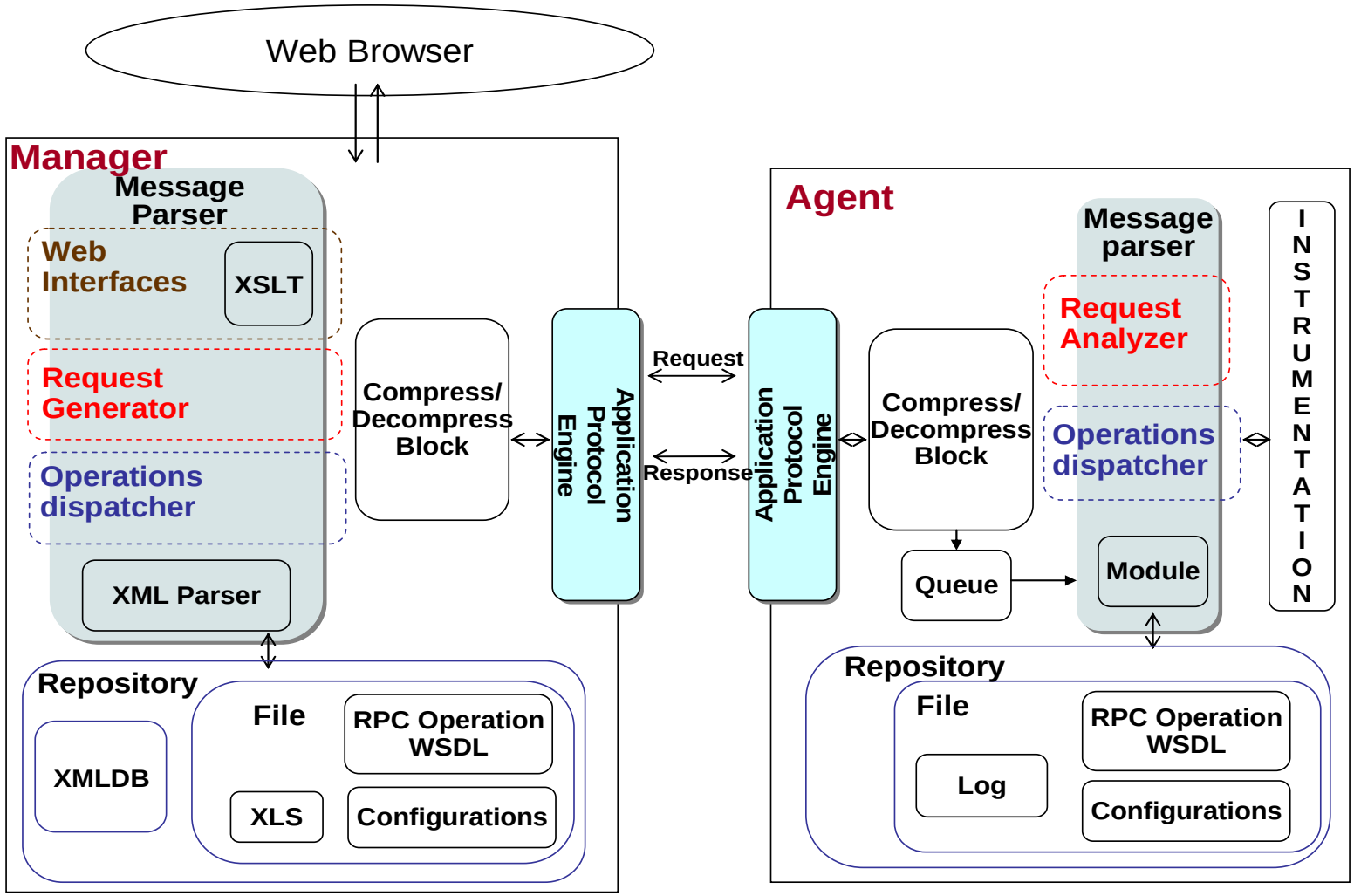
Contents Layer (2)

- 실험방법
  - Merge하는 경우와 간단하게 필터링을 요구하는 경우

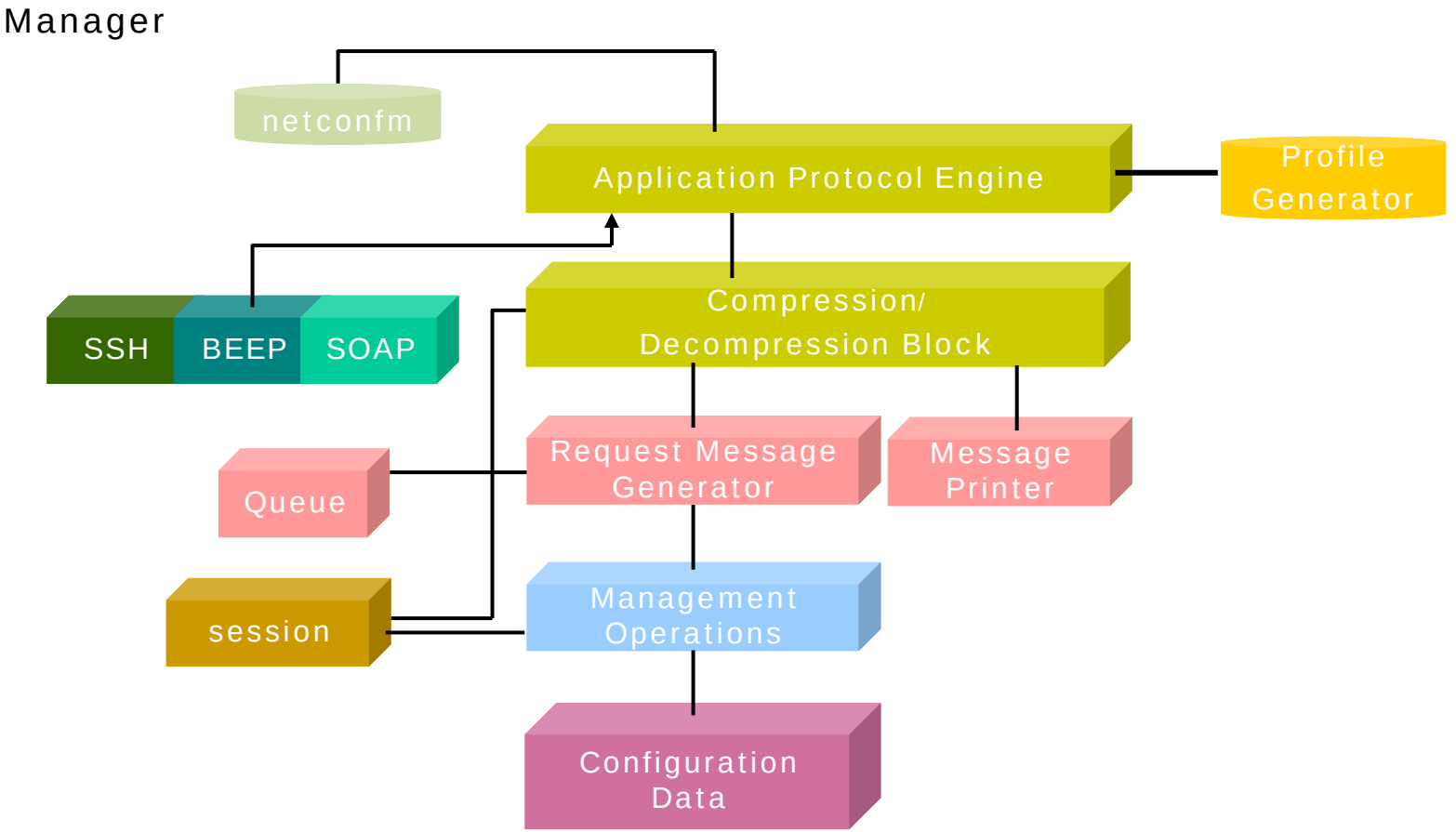
```
<?xml version="1.0"?>
<interfaces>
 <iface>
  <name>eth0</name>
  <address>141.223.82.1</address>
  <netmask>255.255.255.0</netmask>
  <bcast>255.255.255.255</bcast>
  <mtu>1500</mtu>
 </iface>
 <iface>
  <name>eth1</name>
  <address>141.223.82.2</address>
  <netmask>255.255.255.0</netmask >
  <bcast>255.255.255.255</bcast>
  <mtu>1000</mtu>
 </iface>
</interfaces>
```

	Subtree Filtering	XPath
Request Message_1 (Using merge)	<filter type="subtree"> <interfaces> <iface> <name>eth0</name> </iface> <iface> <mtu>1000</mtu> <name/> </iface> </interfaces> </filter>	<filter type="xpath"> //name[.='eth0']|//mtu[.='1000'] </filter>
Processing Time	**2.085 ms**	**1.901 ms**
Memory Usage	**1816 KB**	**1828 KB**
Request Message_2 (Using simple)	<filter type="subtree"> <interfaces> <iface> <name/> </iface> </interfaces> </filter>	<filter type="xpath"> //name </filter>
Processing Time	**1.716 ms**	**1.954 ms**
Memory Usage	**1812 KB**	**1828 KB**

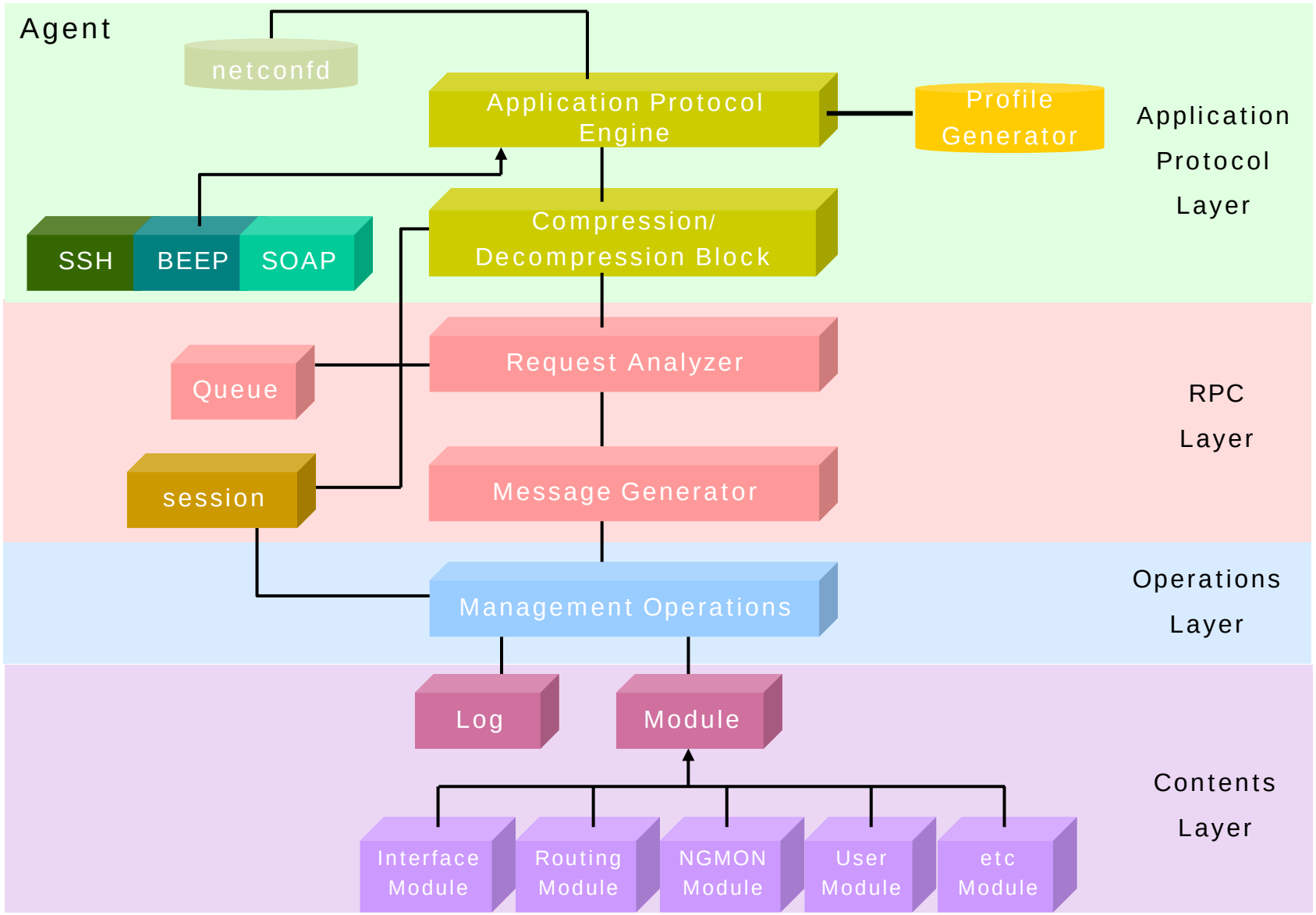
XCMS 구조


Web Browser
Manager
Message Parser
Web Interfaces
XSLT
Request Generator
Operations dispatcher
XML Parser
Compress/ Decompress Block
Application Protocol Engine
Request
Response
Repository
XMLDB
File
RPC Operation WSDL
XLS
Configurations
Agent
Application Protocol Engine
Compress/ Decompress Block
Queue
Message parser
Request Analyzer
Operations dispatcher
Module
INSTRUMENTATION
Repository
File
RPC Operation WSDL
Log
Configurations

XCMS Manager



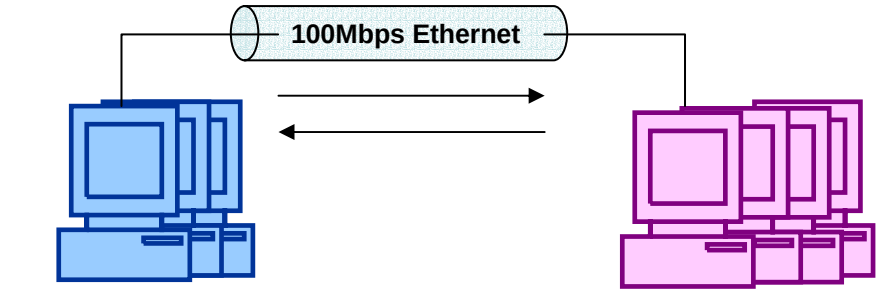
XCMS Agent



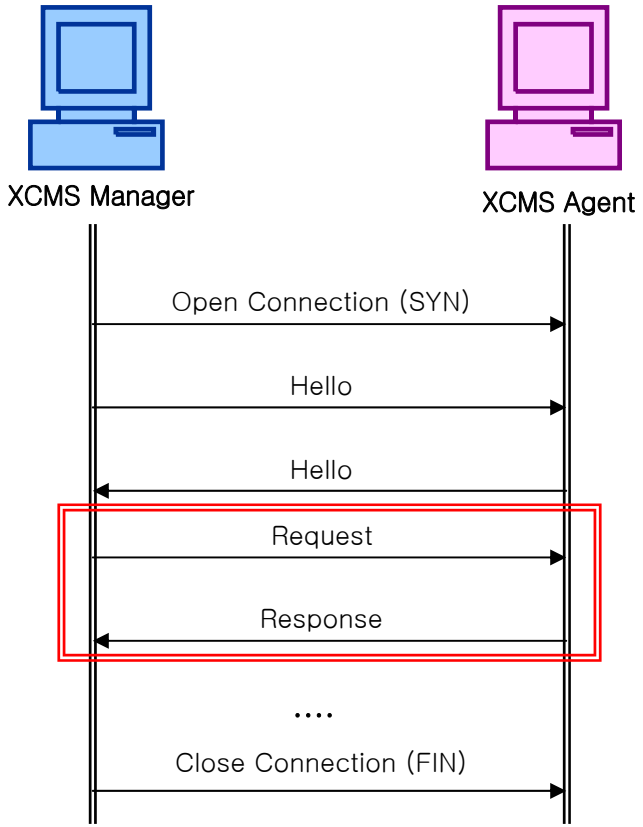
DP&NM Lab
Postech, Korea

성능측정 방법

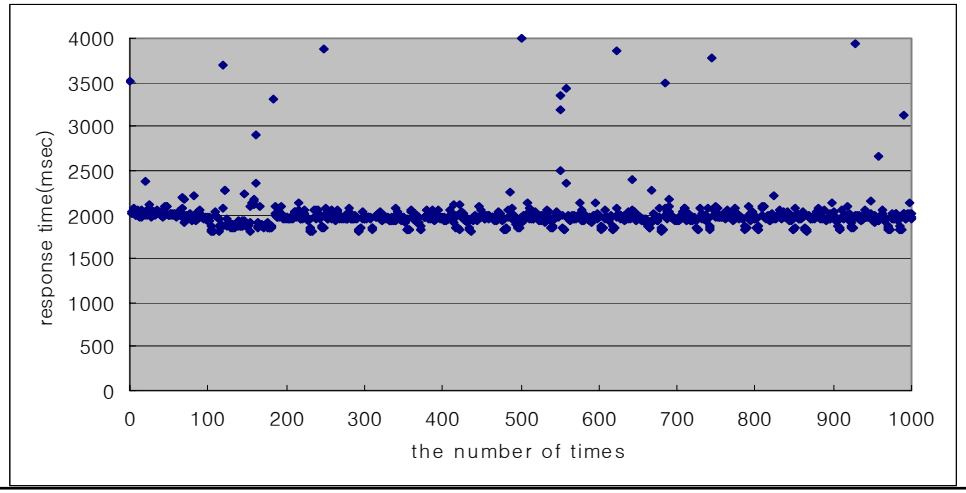
실험 환경



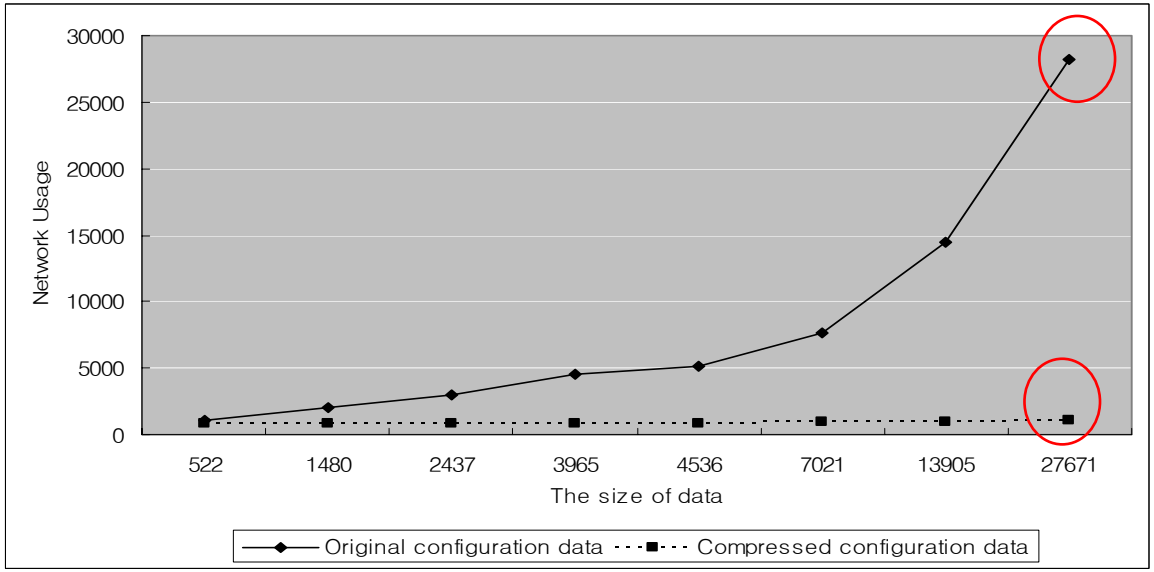
성능 측정

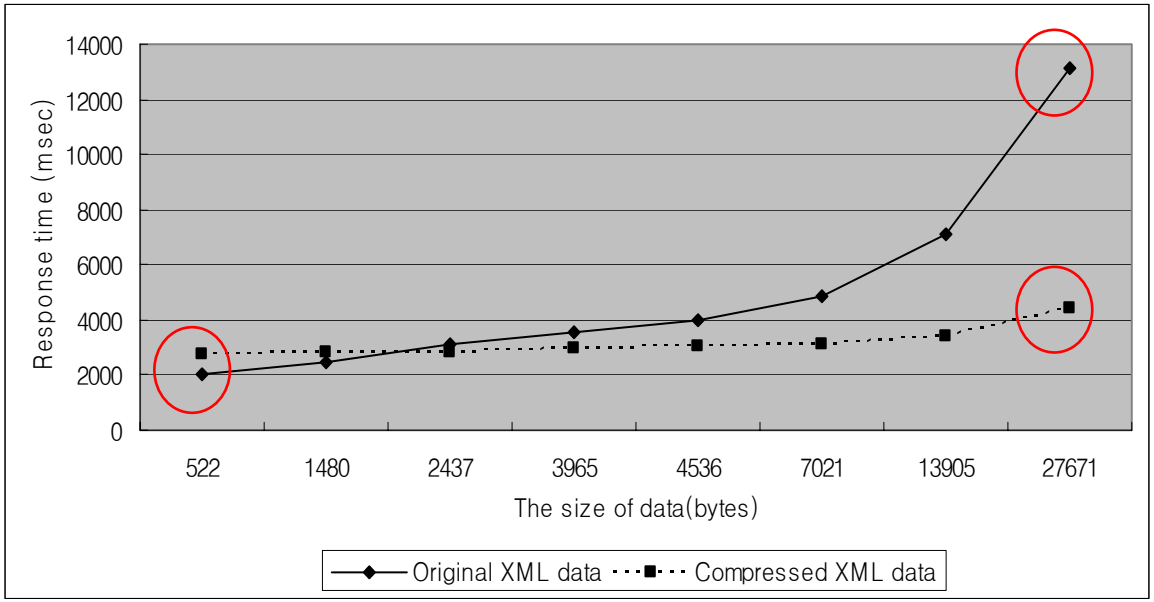


실험 정확성

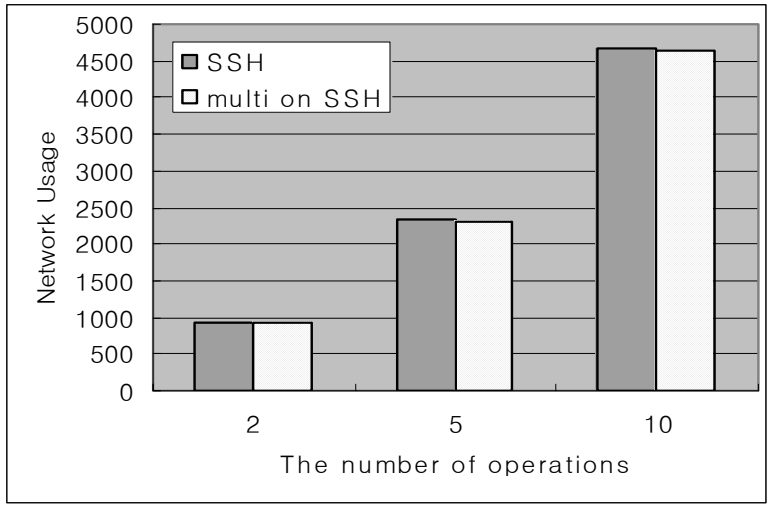


성능 측정 (1)

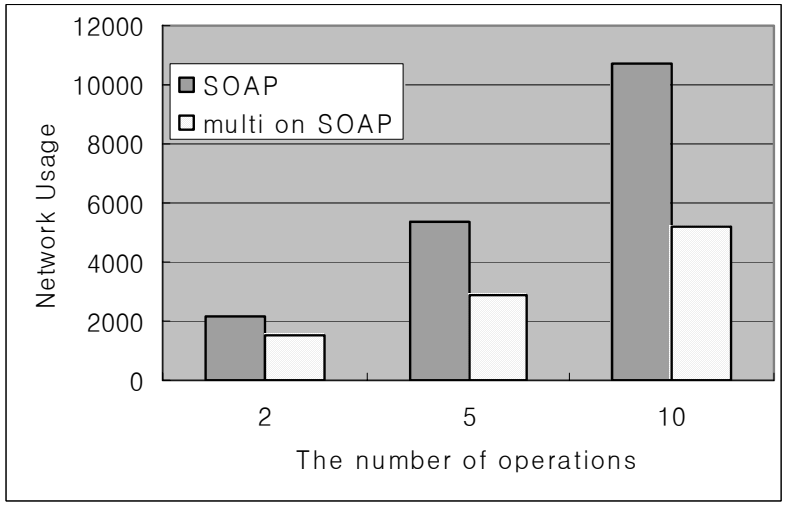

30000
25000
20000
15000
10000
5000
0
Network Usage
522
1480
2437
3965
4536
7021
13905
27671
The size of data
Original configuration data
Compressed configuration data


14000
12000
10000
8000
6000
4000
2000
0
Response time (msec)
522
1480
2437
3965
4536
7021
13905
27671
The size of data(bytes)
Original XML data
Compressed XML data

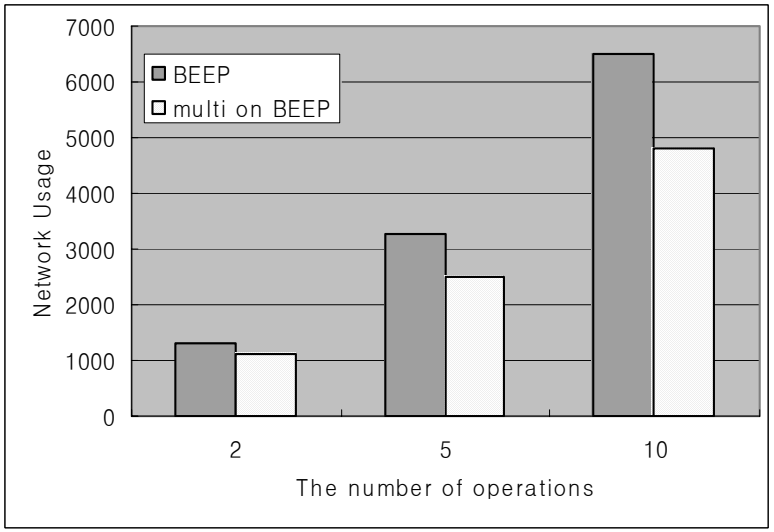
성능 측정 (2)



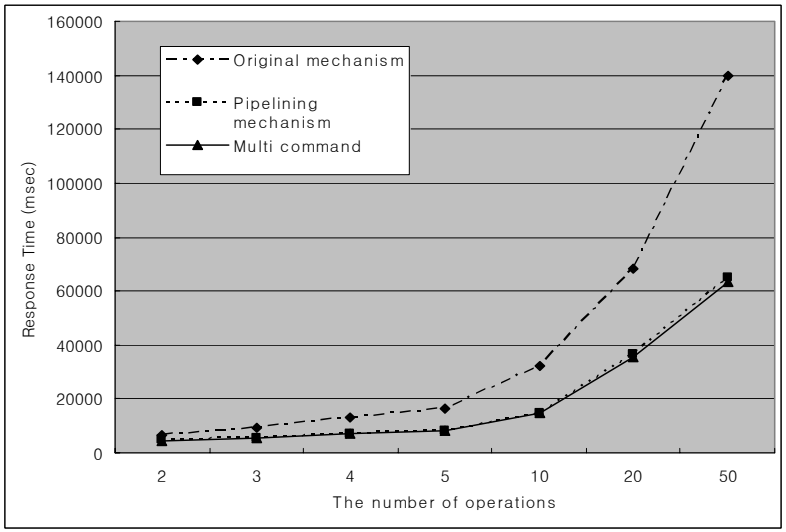
(a) SSH protocol network usage



(b) SOAP over HTTP protocol network usage



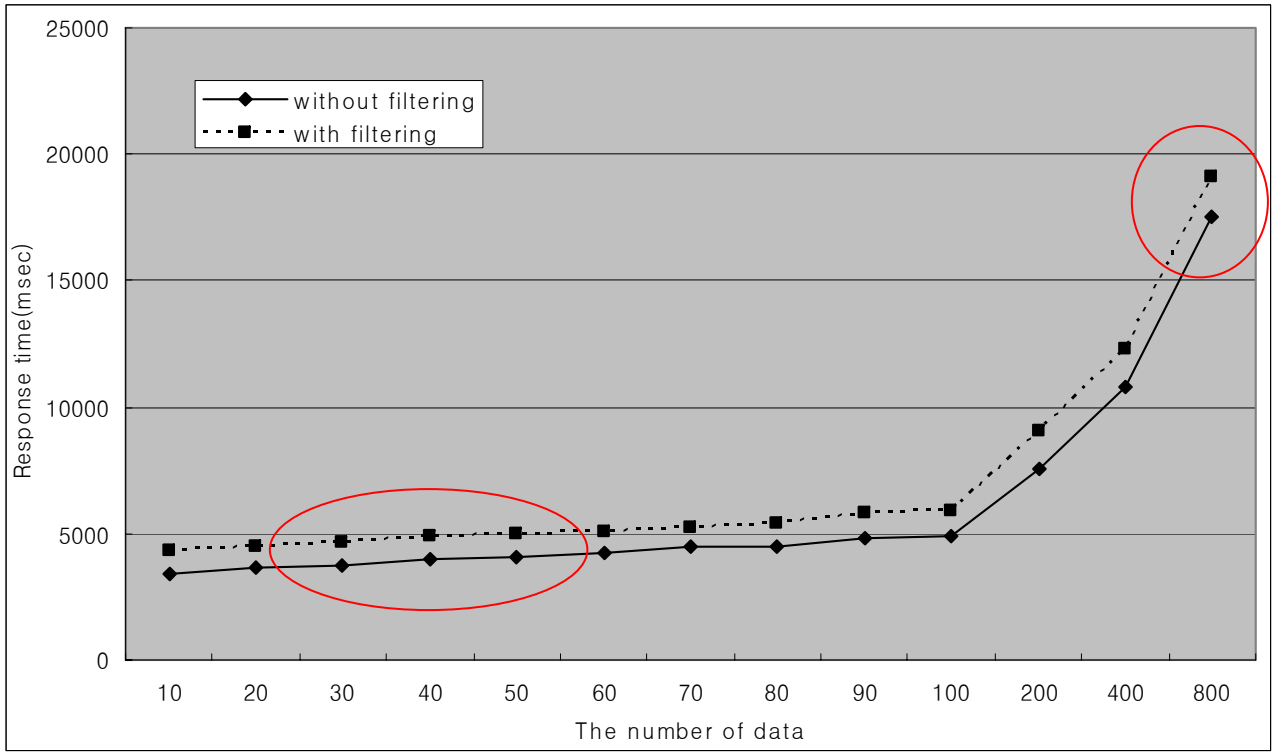
(c) BEEP protocol network usage



Multi command 응답시간

성능 측정 (3)

- 데이터 크기변화에 따른 'copy-config' 수행 시간



DP&NM Lab
Postech, Korea

결 론

- NETCONF 프로토콜의 효율성 검증
  - 수치적인 결과 값을 제시함
- NETCONF 표준에 대하여 성능을 향상시킬 수 있는 방법 제시
  - NETCONF 시스템 개발하는데 참고자료 제공
- 계층별 접근으로 다양한 환경에서의 성능 향상방법 제시
- 구성관리 성능에 대한 연구자료
- XCMS에만 국한된 결과 값?
  - 성능측정의 결과 값은 XCMS의 구현방식에 의존적
  - 수치적 차이는 있겠지만, 상대적 비교결과는 의존적이지 않다

향후과제

- 리눅스 시스템이 아닌 실제 동작중인 장치에 적용
- 제시한 방법들 표준화 반영
- 다른 구현물과의 성능 비교분석

DP&NM Lab
Postech, Korea

Q&A