Smart Packets: Applying Active Networks to Network Management

Beverly Schwartz et. al.

BBN Technologies

Presented by Jinghua Hu

09/17/2002

Outline

- Introduction
- Smart Packets System Architecture
- Descriptions of Major Components
  - Smart Packets Formats/Encapsulation
  - Programming Languages
  - Virtual Machine
  - Security Considerations
- Discussions

Introduction

- Concept of Active Networks
  - capsules carrying user injected programs
  - active nodes performing computations
- Goals of this paper
  - Apply active networks to network management
  - Architecture descriptions, design and implementation

Network Management Review

- Components
  - Management stations
  - Managed objects/devices
  - Network Management Protocol
  - Management Information Base (MIB)

SNMP Review

- SNMP
  - Management station exchanges data/control with managed devices by polling/trapping
  - SNMP PDU type
    - GetRequest
    - SetRequest
    - Response
    - InformRequest

Motivation

- More per-device processing power available for network management
- Polling from management stations is not efficient in large scale networks

- Thinking about applying Active Networks
  - Programmable managed nodes

System Architecture

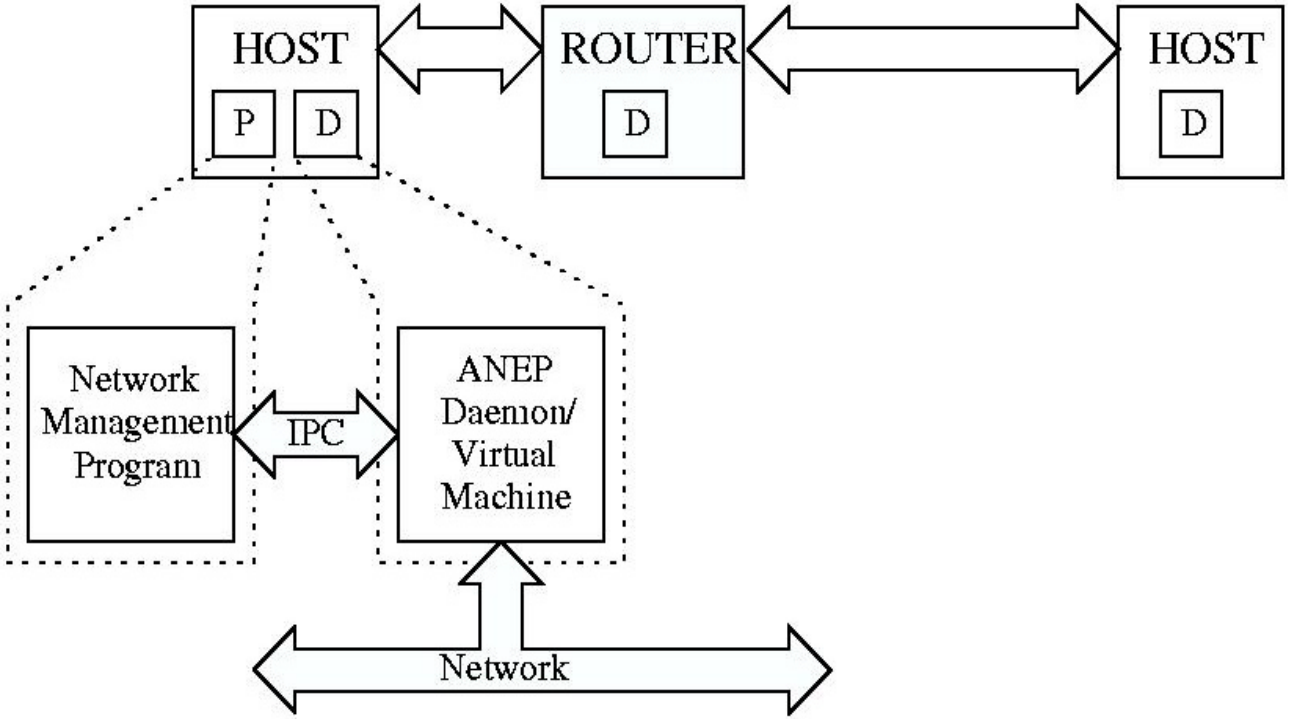

Fig. 1. IP and ANEP encapsulation

ANEP Daemon

- ANEP: Active Network Encapsulation Protocol
- ANEP Daemon
  - Injection point for smart packets
  - Reception point for smart packets
  - Performing execution of the received programs on virtual machine

Smart Packets Project

- Four Major Components
  - a specification for smart packet formats
  - a specification for programming languages
  - a virtual machine
  - a security architecture
- Design Principles
  - No persistent state
  - Program contained in a single packet

Part 1: Smart Packets

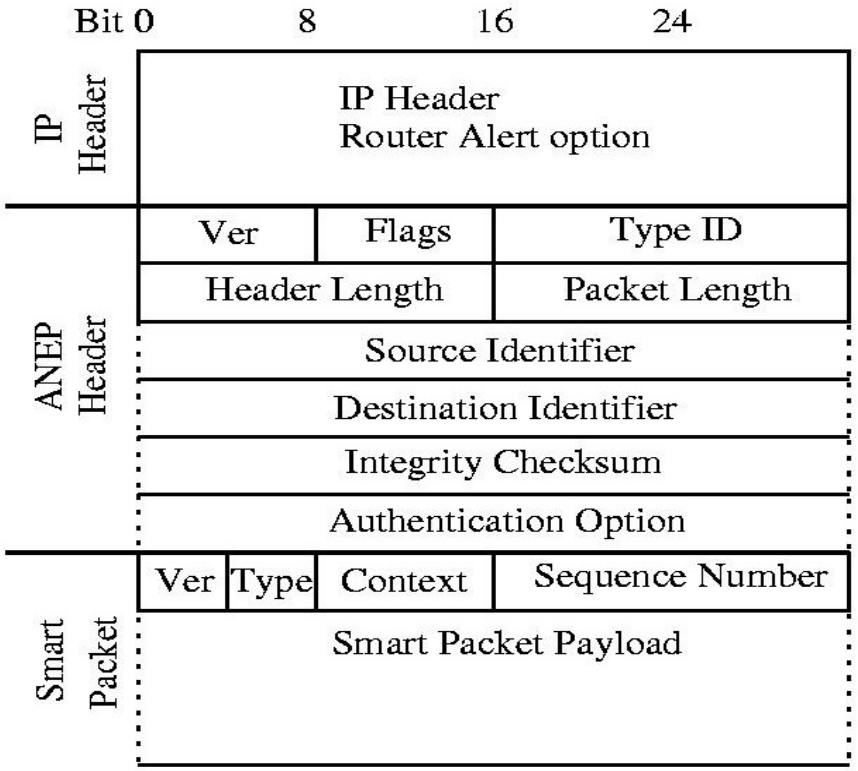

Fig. 2. A smart packet with IP and ANEP encapsulation

Smart Packet Formats

- Header
  - Version
  - Type
    - program packet (needs IP Router Alert Option)
    - data packet
    - error packet
    - message packet
  - Context: identifier for clients
  - Sequence number

Smart Packet Formats

- Payload
  - Carrying program/data/error/message
  - Baggage area
    - Allowing loading/unloading of data
    - NOT protected

ANEP encapsulation

- ANEP header
- ANEP authentication option

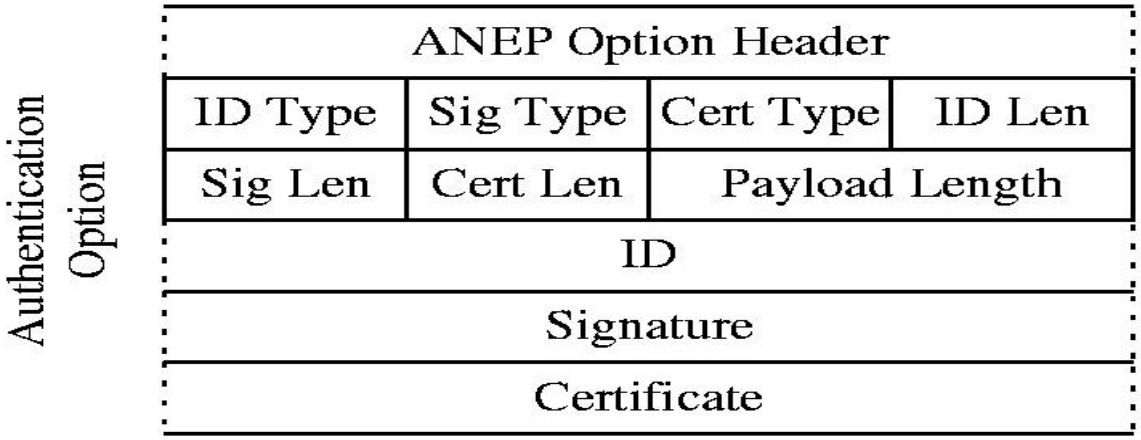

Fig. 3. ANEP Authentication Option

Summary of part 1

- A Smart Packet (header+payload) is encapsulated within an ANEP packet and then carried within IP
- Need to set “Router Alert” option in IP header

Part 2: Programming Languages

- Language Design Issues
  - Compact code size
  - Safety
  - Mobility
  - Support of special data types and operations for network management

Sprocket

- C++ style language
  - removal of unnecessary constructs
  - new features such as built-in types for packet, address, identifier and MIB added
  - support operations such as getting address, sending packet, retrieving header, querying MIB information, etc

Spanner

- Stack-based CISC Assembly Language
  - multi-clock complex instructions
  - compact code size
  - variable declarations
  - no access to memory
  - data stored either in variables or stack
  - branch and flow control
  - subroutines

Summary of part 2

- Statelessness favors compact code size
- High-level Language: Sprocket
- Assembly Language: Spanner
- Sprocket and Spanner are equivalent, while Spanner allows hand-optimization for a more compact size

Part 3: Virtual Machine

- Design Issues
  - feature set
  - security
- When a Program packet arrives, Daemon will
  - authenticate the sender identity
  - verify the data origin and data integrity
  - check if sender is authorized to run the program
  - fork a child process to run the virtual machine

Virtual Machine Implementation

- Spanner Virtual Machine
  - stack-based CISC architecture
  - conservatively handling of errors
  - aware of resource limits
  - resides on router's control processor
  - limited impact on router performance

Summary of part 3

- Virtual Machine is designed based on considerations of feature set, security and performance impact.

Part 4: Security Considerations

- Smart Packets: a security threat?
- Mechanisms to limit the threats:
  - limit on the creator of smart packet
  - authentication/authorization on data origin
  - data integrity check
  - restrict risky operations only to programs sent by authorized senders

Authentication/Authorization

- Public-key certificate for sender identification
- Digital signature for data integrity protection
  - protect ANEP header and entire smart packet, except ANEP packet length field and baggage of smart packet
- SNMPv3 Access Control database for authorization check

Summary of part 4

- Security issues addressed in the design
  - authentication/authorization
- Security challenges remain
  - part of the original packets is not protected
  - large certificates size vs. limited packet size
  - computation costs of verifying certificates

Experiences

- Examples
  - Retrieval of interface address and MTU
    - SNMP: two GET messages and two Response
    - Smart Packets: one Program packet and one Data packet
  - Traceroute
- Testbed experiments show that Smart packets network enables more efficient communications

Summary

- Contributions
  - Design and development of Smart Packets project
  - Programmable nodes provide more efficient communications and faster delivery of targeted network events
- Lessons Learned
  - Statelessness is a double-edged sword
  - Compact codes are valuable
  - IP is less extensible than believed
  - Security is challenging

References

- Kurose and Ross, Network Management. Computer Networks, Chap 8.
- RISC Architecture, http://cse.stanford.edu/class/sophomore-college/projects-00/risc/risccisc/
- Beverly Schwartz, Technical Memos, http://www.ir.bbn.com/~bschwart/