QoS Control in the Internet

Raouf Boutaba
School of Computer Science
University of Waterloo
E-mail: rboutaba@bbcr.uwaterloo.ca
Outline

- Multimedia Networking Applications
- Challenges and Solutions in IP Networks
- Audio/Video Compression in the Internet
- Streaming Stored Audio/Video, RTSP
- Real time (phone) over IP best effort
- RTP, RTCP, H.323
- Improving QoS in IP: Basic Principles
- Scheduling and Policing Mechanisms
- Integrated & Differentiated IP Services
- MPLS and traffic engineering
- Policy Based Networking for QoS control
- COPS extensions for distributed control
Multimedia Networking Appl’s

- Typically sensitive to delay, but can tolerate packet loss (would cause minor glitches that can be concealed)
- Data contains audio and video content ("continuous media"),
- Three classes:
  - Streaming stored audio and video
  - One to many streaming of real-time audio and video
  - Real-time interactive audio and video
Streaming Stored Audio/Video

Clients
- Request audio/video files (compressed) from servers
  Examples: Music, Movies
- Clients pipeline reception over the network and display

Delay
- From client request until display start can be 1 to 10 seconds
- Playback starts while client continues receiving file from server → Streaming

User interactivity
- User can control operation (similar to VCR): pause, resume, fast forward, rewind, etc.
Unidirectionnel Real Time

- Similar to existing TV and radio stations, but delivery on the network
  - Microsoft provides an Internet radio station guide

- One to many streaming
  - Many users who are simultaneously receiving the same real-time audio/video program
  - Efficient distribution through multicasting (however, today’s most one-to-many audio/video transmission in the Internet use separate unicast streams)

- Non-interactive, just listen/view

- Delay
  - From client click until playback start is up to 10’s seconds
Real-time Interactive Appl’s

✈ Internet Phone or video conference

✈ Delay

✈ More stringent delay requirement than Streaming and Unidirectional because of real-time nature

✈ Video: < 150 msec acceptable

✈ Audio: < 150 msec good, < 400 msec acceptable

✈ Interactivity

✈ One-to-many real-time is not interactive as users cannot pause or rewind transmission (hundreds others listening)

✈ Interactive in the sense that participants can orally and visually respond to each other in real time
Challenges

- TCP/UDP/IP suite provides best-effort, no guarantees on expectation or variance of packet delay
- Streaming applications delay of 5 to 10 seconds is typical and has been acceptable, but performance deteriorate if links are congested (transoceanic)
- Real-Time Interactive requirements on **delay** and its **jitter** have been satisfied by over-provisioning (providing plenty of bandwidth), what will happen when the load increases?...
- Most router implementations use only **FCFS** packet processing and transmission scheduling
Challenges (Cont)

To mitigate impact of “best-effort” protocols, we can:

- Use UDP to avoid TCP and its slow-start phase...
- Delay playback (~100 ms) to diminish network-induced jitter
- Buffer content at client and control playback to remedy jitter
- Adapt compression level to available bandwidth
- etc...
Solution Approaches in IP Networks

- Just add more bandwidth and enhance caching capabilities (over-provisioning)

- Need major change of the protocols:
  - Incorporate resource reservation (bandwidth, processing, buffering), and new scheduling policies
  - Set up service level agreements with applications, monitor and enforce the agreements, charge accordingly

- Need moderate changes:
  - Use small number of (possibly two) traffic classes for all packets and differentiate service accordingly
  - Charge based on class of packets (platinum, low-budget)
  - Network capacity is provided to ensure first class packets incur no significant delay at routers
Audio and Video Compression

Audio and video are digitized and compressed
- Need for digitization: Internet transmits bits
- Need for compression: uncompressed audio/video consumes tremendous amount of storage and bandwidth
- Compression removes inherent redundancies in digitized audio/video

Reduces amount of data by order of magnitude
- Example: A single image of 1024 pixels * 1024 pixels; each pixel encoded into 24 bits
  - Without compression:
    - Requires 3 MB of storage
    - Takes 7 min over a 64 Kbps link
  - With a modest compression (10:1 compression rate)
    - Requires 300 KB of storage
    - Takes less than 6 seconds
Audio Compression in the Internet

- **Pulse Code Modulation (PCM)**
  - Analog audio signal is sampled at some fixed rate; Value of each sample is an arbitrary real number
  - Each sample is rounded to one of a finite number of values ("quantization"). Number of quantization values is power of 2
  - Each quantization value is represented by a fixed number of bits. Bit representations of all samples are concatenated together form the digital signal

- **Examples of PCM encoding**
  - Speech: 8000 samples/sec; 8 bits per sample -> 64Kbps rate
  - Compact Disk: 44,100 samples/sec; 16 bits per sample -> rate of 705.6 Kbps for mono and 1.411 Mbps for stereo

- **Compression techniques used to reduce bit rate**
  - GSM (13 Kbps); G.729 (8 Kbps); G.723(6.4 and 5.3 Kbps)
  - MPEG layer 3 (MP3): 128 or 112 Kbps (near CD-quality music)
Video Compression in the Internet

**Principles**
- A video is a sequence of images; each image is displayed at constant rate, e.g. at 24 or 30 images per second.
- An uncompressed, digitally encoded image consists of an array of pixels; each pixel is encoded into a number of bits to represent luminance and color.
- Two types of redundancy in video (exploited in compression):
  - Spatial redundancy (within a given image; e.g. white space)
  - Temporal redundancy (repetition from image to subsequent image; e.g. two exactly same images)

**MPEG**
- MPEG 1, for CD-ROM quality video (1.5 Mbps)
- MPEG 2, for high quality DVD video (3-6 Mbps)
- MPEG 4, for object oriented video compression
- **H.261** (also popular in the Internet)
Streaming Stored Audio/Video

- Important and growing application due to reduction of storage costs, increase in high speed network access from homes, enhancements to caching and introduction of QoS in IP networks
- Audio/Video file is segmented and sent over either TCP or UDP, public segmentation protocol: **Real-Time Protocol (RTP)**
- User interactive control is provided, eg the public protocol **Real Time Streaming Protocol (RTSP)**
- **Helper Application**: displays content, which is typically requested via a Web browser; eg RealPlayer; typical functions:
  - Decompression
  - Jitter removal
  - Error correction: use redundant packets to be used for reconstruction of original stream
  - GUI for user control
Streaming From Web Servers

Audio: in files sent as HTTP objects

Video (interleaved audio and images in one file, or two separate files and client synchronizes the display) sent as HTTP object(s)

A simple architecture is to have the Browser requests the object(s) and after their reception pass them to the player for display
Streaming From Web Server (Cont.)

- Alternative: set up connection between server and player, then download
- Web browser requests and receives a **Meta File** (a file describing the object) instead of receiving the file itself;
- Browser launches the appropriate Player and passes it the Meta File;
- Player sets up a TCP connection with Web Server and downloads the file
Using a Streaming Server

- This gets us around HTTP, allows a choice of UDP vs. TCP and the application layer protocol can be better tailored to Streaming;
- Many enhancements options are possible (see next slide)
Options When Using a Streaming Server

- Use UDP, and Server sends at a rate (Compression and Transmission) appropriate for client; to reduce jitter, Player buffers initially for 2-5 seconds, then starts display.

- Use TCP, and sender sends at maximum possible rate under TCP; retransmit when error is encountered; Player uses a much large buffer to smooth delivery rate of TCP.
Real Time Streaming Protocol

❖ For user to control display: rewind, fast forward, pause, resume, etc...

❖ Out-of-band protocol (uses two connections, one for control messages (Port 554) and one for media stream)

❖ RFC 2326 permits use of either TCP or UDP for the control messages connection, sometimes called the RTSP Channel

❖ As before, meta file is communicated to web browser which then launches the Player; Player sets up an RTSP connection for control messages in addition to the connection for the streaming media
RTSP Operation
Meta File Example

<title>Twister</title>

<session>
  <group language=en lipsync>
    <switch>
      <track type=audio
        e="PCMU/8000/1"
        src = "rtsp://audio.example.com/twister/audio.en/lofi">
      <track type=audio
        e="DVI4/16000/2" pt="90 DVI4/8000/1"
        src="rtsp://audio.example.com/twister/audio.en/hifi">
        </switch>
    </track type="video/jpg">
    <track type="video/jpg"
      src="rtsp://video.example.com/twister/video">
  </group>
</session>
RTSP Exchange Example

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
   Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
   Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
   Session: 4231
   Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
   Session: 4231
   Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
   Session: 4231

S: 200 3 OK
Real Time (Phone) Over IP's Best Effort

- Internet phone applications generate packets during talk spurts
- Bit rate is 8 KBytes, and every 20 msec, the sender forms a packet of 160 Bytes + a header to be discussed below
- The coded voice information is encapsulated into a UDP packet and sent out; some packets may be lost; up to 20% loss is tolerable; using TCP eliminates loss but at a considerable cost: variance in delay; FEC is sometimes used to fix errors and make up losses
- End-to-end delays above 400 msec cannot be tolerated; packets that are that delayed are ignored at the receiver
- Delay jitter is handled by using timestamps, sequence numbers, and delaying playout at receivers either a fixed or a variable amount
- With fixed playout delay, the delay should be as small as possible without missing too many packets; delay cannot exceed 400 msec
Internet Phone with Fixed Playout Delay
Adaptive Playout Delay

- Objective is to use a value for p-r that tracks the network delay performance as it varies during a phone call.
- The playout delay is computed for each talk spurt based on observed average delay and observed deviation from this average delay.
- Estimated average delay and deviation of average delay are computed in a manner similar to estimates of RTT and deviation in TCP.
- The beginning of a talk spurt is identified from examining the timestamps in successive and/or sequence numbers of chunks.
Recovery From Packet Loss

- Loss is in a broader sense: packet never arrives or arrives later than its scheduled playout time.
- Since retransmission is inappropriate for Real Time applications, FEC or Interleaving are used to reduce loss impact.
- Simplest FEC scheme adds a redundant chunk made up of exclusive OR of a group of n chunks; redundancy is 1/n; can reconstruct if at most one lost chunk; playout time schedule assumes a loss per group.
- Mixed quality streams are used to include redundant duplicates of chunks; upon loss playout available redundant chunk, albeit a lower quality one.
- With one redundant chunk per chunk can recover from single losses.
Piggybacking Lower Quality Stream
Interleaving

- Has no redundancy, but can cause delay in playout beyond Real Time requirements
- Divide 20 msec of audio data into smaller units of 5 msec each and interleave
- Upon loss, have a set of partially filled chunks
Real-Time Protocol (RTP)

- Typically runs over UDP
- RTP viewed as a sub-layer of the transport layer
- RTP from an application developer perspective

![Diagram showing the layers of the network stack with RTP at the transport layer and UDP, IP, and Physical layer below.]

---

APNOMS’02, Jeju, Korea
RTP Packet

- RTP Provides standard packet format for real-time applications

- Specifies header fields below
**RTP Packet (Cont)**

- **Payload Type**: 7 bits, providing 128 possible different types of encoding; e.g., PCM, MPEG2 video, etc.

- **Examples:**

<table>
<thead>
<tr>
<th>Payload Type Number</th>
<th>Audio Format</th>
<th>Sampling Rate</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PCM mu-law</td>
<td>8 KHz</td>
<td>64 Kbps</td>
</tr>
<tr>
<td>1</td>
<td>1016</td>
<td>8 KHz</td>
<td>4.8 Kbps</td>
</tr>
<tr>
<td>3</td>
<td>GSM</td>
<td>8 KHz</td>
<td>13 Kbps</td>
</tr>
<tr>
<td>7</td>
<td>LPC</td>
<td>8 KHz</td>
<td>2.4 Kbps</td>
</tr>
<tr>
<td>9</td>
<td>G.722</td>
<td>8 KHz</td>
<td>48-64 Kbps</td>
</tr>
<tr>
<td>14</td>
<td>MPEG Audio</td>
<td>90 KHz</td>
<td>----</td>
</tr>
<tr>
<td>15</td>
<td>G.728</td>
<td>8 KHz</td>
<td>16 Kbps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Payload Type Number</th>
<th>Video Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Motion JPEG</td>
</tr>
<tr>
<td>31</td>
<td>H.261</td>
</tr>
<tr>
<td>32</td>
<td>MPEG1 Video</td>
</tr>
<tr>
<td>33</td>
<td>MPEG2 Video</td>
</tr>
</tbody>
</table>
RTP Packet (Cont)

- **Sequence Number**: 16 bits; used to detect packet loss.

- **Timestamp**: 32 bits; gives the sampling instant of the first audio/video byte in the packet; used to remove jitter introduced by the network; the timestamp is derived from a sampling clock at the sender.

- **Synchronization Source identifier (SSRC)**: 32 bits; identifies the source of a stream; each stream in an RTP session has a distinct SSRC; assigned randomly by the source when the new stream is started.
RTP Control Protocol (RTCP)

- Protocol specifies report packets exchanged between sources and destinations of multimedia information.
- Three reports are defined: Receiver reception, Sender, and Source description.
- Reports contain statistics such as the number of packets sent, number of packets lost, inter-arrival jitter.
- Used to modify sender transmission rates and for diagnostics purposes.
RTCP Bandwidth Scaling

- If each receiver sends RTCP packets to all other receivers, the traffic load resulting can be large.
- RTCP adjusts the interval between reports based on the number of participating receivers.
- Typically, limit the RTCP bandwidth to 5% of the session bandwidth, divided between the sender reports (25%) and the receivers reports (75%).
- Period for transmitting RTCP packets for a sender:

\[
T = \frac{\text{Number of senders}}{.25 \times .05 \times \text{session-bandwidth}} \quad \text{(average RTCP packet size)}
\]

- Period for transmitting RTCP packets for a receiver:

\[
T = \frac{\text{Number of receivers}}{.75 \times .05 \times \text{session-bandwidth}} \quad \text{(average RTCP packet size)}
\]
H.323

- A standard for real-time audio and video conferencing among end systems on the Internet
- H.323 end systems should be able to communicate with ordinary telephones
Improving QOS in IP Networks

- IETF groups are working on proposals to provide better QOS control in IP networks, i.e., going beyond best effort to provide some assurance for QOS.
- Work in Progress includes RSVP, Differentiated Services, and Integrated Services.
- Simple model for sharing and congestion studies:

![Diagram showing network model with nodes H1, H2, H3, H4, R1, R2, and a 1.5 Mbps link.]
Principles for QOS Guarantees

Consider a phone application at 1Mbps and an FTP application sharing a 1.5 Mbps link; bursts of FTP can congest the router and cause audio packets to be dropped; want to give priority to audio over FTP

PRINCIPLE 1: Marking of packets is needed for router to distinguish between different classes; and new router policy to treat packets accordingly
Applications misbehave (audio sends packets at a rate higher than 1Mbps assumed above);

PRINCIPLE 2: provide protection (isolation) for one class from other classes

Require Policing Mechanisms to ensure sources adhere to bandwidth requirements; Marking and Policing need to be done at the edges:
Principles for QOS Guarantees (Cont.)

- Alternative to Marking and Policing: allocate a set portion of bandwidth to each application flow; can lead to inefficient use of bandwidth if one of the flows does not use its allocation.

- **PRINCIPLE 3**: While providing isolation, it is desirable to use resources as efficiently as possible.
Principles for QOS Guarantees (Cont.)

- Cannot support traffic beyond link capacity
- **PRINCIPLE 4:** Need a Call Admission Process; application flow declares its needs, network may block call if it cannot satisfy the needs

![Diagram showing network topology and bandwidth allocation]
Summary

QoS for networked applications

- Packet classification
- Isolation: scheduling and policing
- High resource utilization
- Call admission
Scheduling And Policing Mechanisms

Scheduling: choosing the next packet for transmission on a link can be done following a number of policies:

- **FIFO**: in order of arrival to the queue; packets that arrive to a full buffer are either discarded, or a discard policy is used to determine which packet to discard among the arrival and those already queued.
Scheduling Policies

- **Priority Queuing**: classes have different priorities; class may depend on explicit marking or other header info, eg IP source or destination, TCP Port numbers, etc.
- Transmit a packet from the highest priority class with a non-empty queue
- Preemptive and non-preemptive versions
Scheduling Policies (Cont.)

- **Round Robin**: scan class queues serving one from each class that has a non-empty queue.

- **Weighted Fair Queuing**: is a generalized Round Robin in which an attempt is made to provide a class with a differentiated amount of service over a given period of time.
Policing Mechanisms

- Three criteria:
  - (Long term) **Average Rate** (100 packets per sec or 6000 packets per min??), crucial aspect is the interval length
  - **Peak Rate**: eg 6000 p p minute Avg and 1500 p p sec Peak
  - (Max.) **Burst Size**: Max. number of packets sent consecutively, ie over a short period of time
  - **Token Bucket mechanism**, provides a means for limiting input to specified Burst Size and Average Rate
Policing Mechanisms (Cont.)

- Bucket can hold \( b \) tokens; tokens are generated at a rate of \( r \) token/sec unless bucket is full of tokens.
- Over an interval of length \( t \), the number of packets that are admitted is less than or equal to \( rt + b \).
- Token bucket and WFQ can be combined to provide upper bound on delay:
Integrated Services

- An architecture for providing QoS guarantees in IP networks for individual application sessions
- relies on resource reservation, and routers need to maintain state info (Virtual Circuit??), maintaining records of allocated resources and responding to new Call setup requests on that basis
Call Admission

- Session must first declare its QOS requirement and characterize the traffic it will send through the network
- **R-spec**: defines the QOS being requested
- **T-spec**: defines the traffic characteristics
- A signaling protocol is needed to carry the R-spec and T-spec to the routers where reservation is required; RSVP is a leading candidate for such signaling protocol
- **Call Admission**: routers will admit calls based on their R-spec and T-spec and based on the current resource allocated at the routers to other calls

1. Request: specify
   - traffic (Tspec)
   - guarantee (Rspec)
2. Element considers
   - unreserved resources
   - required resources
3. Reply: whether or not request can be satisfied
**Integrated Services: Classes**

- **Guaranteed QOS**: this class is provided with **firm bounds** on queuing delay at a router; envisioned for hard real-time applications that are highly sensitive to end-to-end delay expectation and variance.

- **Controlled Load**: this class is provided a QOS **closely approximating** that provided by an unloaded router; envisioned for today’s IP network real-time applications which perform well in an unloaded network.
Differentiated Services

- Intended to address the following difficulties with Intserv and RSVP;
- **Scalability:** maintaining states by routers in high speed networks is difficult due to the very large number of flows
- **Flexible Service Models:** Intserv has only two classes, want to provide more qualitative service classes; want to provide 'relative' service distinction (Platinum, Gold, Silver, ...)
- **Simpler signaling:** (than RSVP) many applications and users may only want to specify a more qualitative notion of service

**Approach:**
- Only simple functions in the core, and relatively complex functions at edge routers (or hosts)
- Do not define service classes, instead provides functional components with which service classes can be built
**Edge Functions**

- At DS-capable host or first DS-capable router
- **Classification**: edge node marks packets according to classification rules to be specified (manually by admin, or by some TBD protocol)
- **Traffic Conditioning**: edge node may delay and then forward or may discard
Core Functions

- **Forwarding**: according to “Per-Hop-Behavior” or PHB specified for the particular packet class; such PHB is strictly based on class marking (no other header fields can be used to influence PHB)

- No state info to be maintained by routers!
Classification and Conditioning

- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP) and determine PHB that the packet will receive
- 2 bits are currently unused
- It may be desirable to limit traffic injection rate of some class; user declares traffic profile (e.g., rate and burst size); traffic is metered and shaped if non-conforming
Forwarding (PHB)

- PHB result in a different observable (measurable) forwarding performance behavior
- PHB does not specify what mechanisms to use to ensure required PHB performance behavior
- Examples:
  - Class A gets x% of outgoing link bandwidth over time intervals of a specified length
  - Class A packets leave first before packets from class B
- PHBs under consideration:
  - Expedited Forwarding: departure rate of packets from a class equals or exceeds a specified rate (logical link with a minimum guaranteed rate)
  - Assured Forwarding: 4 classes, each guaranteed a minimum amount of bandwidth and buffering; each with three drop preference partitions
Differentiated Services Issues

- AF and EF are not even in a standard track yet... research ongoing
- “Virtual Leased lines” and “Olympic” services are being discussed
- Impact of crossing multiple ASs and routers that are not DS-capable
- ...

...
MPLS

Multi-Protocol Label Switching:

- Hop-by-hop or source routing to establish labels
- Uses label native to the media
- Multi level label substitution transport
- Route at edge, switch in core
**MPLS Header**

- IP packet is encapsulated in MPLS header and sent down LSP
- IP packet is restored at end of LSP by egress router
- TTL is adjusted by default
**MPLS Header**

<table>
<thead>
<tr>
<th>Label</th>
<th>EXP</th>
<th>S</th>
<th>TTL</th>
</tr>
</thead>
</table>

- **Label**
  - Used to match packet to LSP
- **Experimental bits**
  - Carries packet queuing priority (CoS)
- **Stacking bit**
- **Time to live**
  - Copied from IP TTL
Forwarding Equivalence Classes

- **FEC** = “A subset of packets that are all treated the same way by a router”
- The concept of FECs provides for a great deal of flexibility and scalability
- In conventional routing, a packet is assigned to a FEC at each hop (i.e. L3 look-up), in MPLS it is only done once at the network ingress

Packets are destined for different address prefixes, but can be mapped to common path
Policy Based Networking (PBN)

- Based on policies
  - E.g., give administrators high priority when accessing the network resources

- Policies are Abstract, Goal oriented
  - "WHAT" instead of "HOW"
  - E.g.: Administrators (10.1.1.x) must have high priority (DSCP=6)
    - HOW approach: Remark 10.1.1.x traffic with DSCP=6
    - WHAT approach: Give high priority to Administrators

- The policy is still valid even if:
  - topology and/or service implementation changes
  - E.g., Administrators subnet is changed/expanded
  - Administrators do not need to be associated with a specific subnet!
PBN Architecture

Policy Editing tool

Directory Server

PDP

Other Services e.g., Event

PEP

Managed devices

PEP

PEP
Common Open Policy Service

The COPS BASE protocol [RFC 2784]

- Policy-related information exchange b/w the PDP to the PEP
- Determines the behavior of the entities, as far as the communication is concerned
- Does not define the semantics of the exchanged data
- Does not describe HOW this data is produced by the PDP or HOW this data will be interpreted by the PEP

COPS client-types

- Control different management areas (DiffServ, RSVP, accounting, Security, etc.)
- Each PEP implements one or more clients of various client-types
- Client-types are defined in separate documents (standard or vendor-specific)
- COPS-RSVP and COPS-PR are such clients
COPS Operation Modes

Outsourcing

1. INSTALL PEP
2. DEC REQ
3. PROCESS PDP

Provisioning

A. CHANGE PDP

B. DEC

C. INSTALL, UPDATE, DELETE CONFIGURATION DATA

Policies Current state

Device
**COPS-PR [RFC 3084]**

### Initialization
1. The PEP connects to the PDP, reports its capabilities/limitations and requests configuration data.
2. The PDP generates the initial policies according to the global policies and current network state.
3. The PDP sends initial policies.
4. The PEP stores these policies in its PIB. The data in the PIB determines the behavior of the device.

### Provisioning
A. The PDP detects changes.
B. The PDP sends commands that add, update or delete policies in the PIB.
C. The PEP updates its PIB.
**Policy Information Base**

- A tree of **PR**ovisioning **C**lasses (**PRCs**)  
- **PR**ovisioning **I**nstances (**PRIs**)  
- Policies can be constructed as a set of **PRIs**  
- **PIBs** are pre-defined  
- Different **PIBs** for different policing areas (Diffserv, Accounting, IP filtering, etc)
PIB Example

PIB: DiffServ Marking Example

If Packet Matches

( (DstAddress 128.1.1.1 or 128.1.1.2) and DSCP=6 )

Then

Mark with DSCP = 4

QoS IP Acl Definition Table (2.2.1.2)

<table>
<thead>
<tr>
<th>Idx</th>
<th>Acl ID</th>
<th>Ace ID</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>300</td>
<td>3</td>
<td>19</td>
</tr>
</tbody>
</table>

QoS Acl Target Table (2.1.3.2)

<table>
<thead>
<tr>
<th>Idx</th>
<th>Acl ID</th>
<th>Act ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>3</td>
</tr>
</tbody>
</table>

QoS Acl Action Table (2.1.3.1)

<table>
<thead>
<tr>
<th>Idx</th>
<th>DSCP</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>No</td>
</tr>
</tbody>
</table>

PRID | VALUE
--- | ---
2.1.3.2.1 | (100,2)
2.1.3.1.2 | (4,NO)
2.2.1.2.1 | (100,2,10)
2.2.1.2.2 | (100,1,11)
2.2.1.1.1 | (128.1.1.1,6)
2.2.1.1.2 | (128.1.1.2,6)
**COPS-PR policy examples**

**Policy 1:**

if traffic to IPs 128.1.1.1 or 128.1.1.2 has DSCP=4

then remark it with DSCP=6

<table>
<thead>
<tr>
<th>Event:</th>
<th>PDP-&gt;PEP DEC</th>
<th>PIB (@ PEP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEP connects</td>
<td>Install:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.3.2.1 ↳ (100,2)</td>
<td>2.1.3.2.1 ↳ (100,2)</td>
</tr>
<tr>
<td></td>
<td>2.1.3.1.2 ↳ (6,NO)</td>
<td>2.1.3.1.2 ↳ (6,NO)</td>
</tr>
<tr>
<td></td>
<td>2.2.1.2.1 ↳ (100,2,10)</td>
<td>2.2.1.2.1 ↳ (100,2,10)</td>
</tr>
<tr>
<td></td>
<td>2.2.1.2.2 ↳ (100,1,11)</td>
<td>2.2.1.2.2 ↳ (100,1,11)</td>
</tr>
<tr>
<td></td>
<td>2.2.1.1.1 ↳ (128.1.1.1,4)</td>
<td>2.2.1.1.1 ↳ (128.1.1.1,4)</td>
</tr>
<tr>
<td></td>
<td>2.2.1.1.2 ↳ (128.1.1.2,4)</td>
<td>2.2.1.1.2 ↳ (128.1.1.2,4)</td>
</tr>
</tbody>
</table>
**COPS-PR policy examples (Cont)**

**Policy 2:**

if traffic to engineers has DSCP=4 then remark it with DSCP=6

<table>
<thead>
<tr>
<th>Event:</th>
<th>PDP→PEP DEC</th>
<th>PIB (@ PEP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEP connects</td>
<td>&lt;NULL&gt;</td>
<td>&lt;EMPTY&gt;</td>
</tr>
<tr>
<td>No Engineer is logged</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Two Engineers log on at 128.1.1.1 and 128.1.1.2 | Install: 2.1.3.2.1 ≠ (100,2)  
2.1.3.1.2 ≠ (6,NO)  
2.2.1.2.1 ≠ (100,2,10)  
2.2.1.2.2 ≠ (100,1,11)  
2.2.1.1.1 ≠ (128.1.1.1,4)  
2.2.1.1.2 ≠ (128.1.1.2,4) | 2.1.3.2.1 ≠ (100,2)  
2.1.3.1.2 ≠ (6,NO)  
2.2.1.2.1 ≠ (100,2,10)  
2.2.1.2.2 ≠ (100,1,11)  
2.2.1.1.1 ≠ (128.1.1.1,4)  
2.2.1.1.2 ≠ (128.1.1.2,4) |
| if traffic to 128.1.1.1 or 128.1.1.2 has DSCP=4 then remark with DSCP=6 | Install: 2.1.3.2.1 ≠ (100,2)  
2.1.3.1.2 ≠ (6,NO)  
2.2.1.2.1 ≠ (100,2,10)  
2.2.1.2.2 ≠ (100,1,11)  
2.2.1.1.1 ≠ (128.1.1.1,4)  
2.2.1.1.2 ≠ (128.1.1.2,4) | 2.1.3.2.1 ≠ (100,2)  
2.1.3.1.2 ≠ (6,NO)  
2.2.1.2.1 ≠ (100,2,10)  
2.2.1.2.2 ≠ (100,1,11)  
2.2.1.1.1 ≠ (128.1.1.1,4)  
2.2.1.1.2 ≠ (128.1.1.2,4) |
| Engineer at 128.1.1.2 logs out if traffic to 128.1.1.1 has DSCP=4 then remark with DSCP=6 | Remove: 2.2.1.2.2  
2.2.1.1.2 | 2.1.3.2.1 ≠ (100,2)  
2.1.3.1.2 ≠ (6,NO)  
2.2.1.2.1 ≠ (100,2,10)  
2.2.1.1.1 ≠ (128.1.1.1,4) |
| An Engineer logs to 128.1.1.3 (similar to the first case) | Install: 2.2.1.2.2 ≠ (100,1,11)  
2.2.1.1.2 ≠ (128.1.1.3,4) | Similar to the first case |
**COPS-PR Shortcomings**

*Policy:*

if network is not congested then mark with DSCP=6 all traffic from 128.1.1.1

else (when network is congested), mark it with DSCP=4

Suppose the PIB supports policies of the form:

if packet matches X then set DSCP=Y

*PDP operation in COPS-PR:*

if network is not congested then ConfData1,

where ConfData1 implements “if packet matches 128.1.1.1 then set DSCP=6”

if network is congested then ConfData2,

where ConfData2 implements “if packet matches 128.1.1.1 then set DSCP=4”

*Shortcomings:*

❖ The PEP can only store and process limited types of policies
❖ The PDP communicates decisions instead of a decision-making process
❖ The PDP needs to be present in cases where this could be avoided
Meta-Policies

The PDP sends to the PEP the following rules (meta-policies):

- *if* (!C) *then* ConfData1,
  
  (“if packet matches 128.1.1.1 *then* set DSCP=6”)

- *if* (C) *then* ConfData2,
  
  (“if packet matches 128.1.1.1 *then* set DSCP=4”)

Also, the PDP

- Sends values for “C” according to the network state (congestion)
- Or, directs the PEP how to evaluate C (e.g., from its MIB)
**The small company example**

1. Internal LAN traffic is always allowed
2. The administrator can always access the Internet, whenever and from wherever he/she is logged in.
3. During overall congestion, traffic between the employee domain and the Internet is denied.
4. Internet can be accessed only during working hours (Monday to Friday, 9:00-17:00)

(Rule #1 has the highest priority, rule #4 the lowest)

“**overall congestion**”: evaluated based on the load @ the interfaces of Router A

PIB of Router A:  

<table>
<thead>
<tr>
<th>Prid</th>
<th>Src</th>
<th>SM</th>
<th>Dst</th>
<th>DM</th>
</tr>
</thead>
</table>

If  

((Source matches Srcaddr/Srcmask)  

&  

(Destination matches Destaddr/Destmask))  

then  

allow
Without Meta-policies

Events

08:59: No Admin. logged on
09:00: Start of working day
11:00: Congestion detected
11:05: No congestion
15:08: Congestion detected
15:11: Administrator logs on at X.Y.3.7
15:20: No congestion
17:00: End of working day
17:15: Administrator logs out
### With Meta-Policies

#### Events:

- **08:59**: No Admin. logged on
- **09:00**: start of working day
- **11:00**: congestion detected
- **11:05**: no congestion
- **15:08**: congestion detected
- **15:11**: administrator logs on at X.Y.3.7
- **15:20**: no congestion
- **17:00**: end of working day
- **17:15**: administrator logs out
Meta-Policies: Formal Definition

if (Condition) then {Actions}

Actions:
- pre-generated COPS-PR commands
- may contain parameters

Condition:
- a logical expression (encoded in a predefined way, e.g., according to an XML DTD)
- contains parameters

The PDP directs the PEP how to evaluate the parameters:
- MIB
- Value sent by the PDP
- Other (download script or code, LDAP, ...)
Scenarios

For each high-level policy, the PDP has the following options:

- **Scenario 1**: PDP sends a meta-policy. All (or some) parameters are evaluated by the PDP and sent to the PEP.
  - Less PDP processing
  - Scalability
  - Smaller messages
  - Robustness, efficiency

- **Scenario 2**: PDP sends a meta-policy. All parameters are evaluated by the PEP.
  - All previous
  - More distribution
  - Scalability, efficiency
  - Self dependency
  - Robustness, fault-tolerance

- **Scenario 3**: PDP sends a meta-policy. It monitors (some) parameters, but it directs the PEP how to evaluate them, in case of PDP absence.
  - All advantages of scenario 1
  - Fault-tolerance, robustness

- **Scenario 4**: PDP processes the policy and sends configuration data to the PEP according to COPS-PR (mainly for compatibility reasons)
Conclusion

Advantages of using Meta-policies:

 payout Efficiency
 payout Bandwidth: Similar commands do not need to be send to the PEP
 payout Some of the monitoring can be performed by the PEP
 payout PDP resources: CPU and memory savings (similar commands do not need to be re-generated or re-validated).

 payout Distribution
 payout Intelligence is distributed towards the PEPs
 payout Monitoring and Decision-Making are de-centralized

 payout Robustness
 payout Probability of PDP overload is reduced
 payout Less big DEC messages, which may get lost in a congested network, are exchanged

 payout Fault-tolerance
 payout Devices can operate correctly during larger periods of PDP absence
Conclusion (Cont)

Tradeoff:
Additional functionality vs increased complexity

- PDP: More complex algorithms
  - However, the PDP is already complex

- PEP: Increased CPU & memory requirements
  - The PDP may decide to send only a small number of selected meta-policies that will not overload the devices but they will increase the overall efficiency significantly
  - The backwards compatibility allows PEPs not to implement the extra functionality (no meta-policies)
References

Open Source


Internet Drafts


Publications

