SAMTK: A Toolkit for Scalable Adaptive Adaptive Multicast Updates

samrg @ IETF71

Nobuo Kawaguchi
Nagoya University
Why SAMTK?

- To fill the gap between researchers and real world application developers.
- Share the common technologies for multi-point communications.
- Support Hybrid Configuration of multi-point communication protocols.
- Enable step-by-step deployment of hybrid ALM/XCAST/Multicast communication.
SAMTK: a Toolkit for SAM

- Platform for both SAM researchers and application developers.
- Common API for SAM communication protocols.
  - Group management.
  - Multi-layered communication.
- Ease of application development.
- C++, Qt, Multi-platform (Win / Mac / Linux / BSD)
Achievement
SAMTK Group Manager
Traffic Monitor

- Monitor the traffics over SAMTK
SAM Issues

• Group Management
  – Member Join/Leave, Scalability

• Traffic Management
  – How many packets can be sent

• Topology Management
  – How to route the traffic
SAM Issues

- **Group Management**
  - Member Join/Leave, Scalability

- **Traffic Management**
  - How many packets can be sent

- **Topology Management**
  - How to route the traffic

**SAMTK** manages here

**Plug-in** manages here
Design Choices of SAMTK

Lessons from XCAST6 deployment/development

• Keep it simple, stupid
  – Do not design Group Management
    • currently Web server (1000 line of php) based.
    • extensible design with XML.
    • P2P based could be introduced

• Multi-protocol support is important
  – not ideal IPv6 world

• Multi-platform & rich interface is important
  – to support many users.
Design Choices of SAMTK

Lessons from XCAST6 deployment/development

- Keep it simple, stupid
  - Do not design Group Management
    - currently Web server (1000 line of php) based.
    - extensible design with XML.
    - P2P based could be introduced

- Multi-protocol support is important
  - not ideal IPv6 world

- Multi-platform & rich interface is important
  - to support many users.
Design Choices of SAMTK

Lessons from XCAST6 deployment/development

• Keep it simple, stupid
  – Do not design Group Management
    • currently Web server (1000 line of php) based.
    • extensible design with XML.
    • P2P based could be introduced

• Multi-protocol support is important
  – not ideal IPv6

• Multi-platform & rich interface is important
  – to support many users.
Design Choices of SAMTK

Lessons from XCAST6 deployment/development

• Keep it simple, stupid
  – Do not design Group Management
  • currently Web server (1000 line of php) based.
  • extensible design with XML.
  • P2P based could be introduced

• Multi-protocol support is important
  – not ideal IPv6

• Multi-platform & rich interface is important
  – to support many users

Scalability for Number of Groups

ALM/ Overlay with Plugin

Qt / C++ implementation
SAMTK Architecture

SAMTK Group Web Server
(Apache / PHP)

HTTP/XML

SAM Applications

Group Interface
Group Management Module

Application Interface

SAMTK Core Module

Protocol Interface

XCAST Plugin

ALM Plugin

IPv4

XCAST6

OS (Windows / Mac / Linux / FreeBSD.. )
Application Interface

Interface from applications

- **SAMSocket**: Multipoint socket class
  - Separation of send socket and receive socket
  - Underlined protocol can be choose through plugin interface

- **GroupAddress**: Multipoint address class
  - Composed from member information
  - Currently identified through group URI
SAMSocket subclasses

• **SAMSendSocket**
  – UDP based interface
  – Socket for Multipoint send

• **SAMReceiveSocket**
  – Socket for Multipoint reception
  • management of several streams from different sources
  – Qt Slot/Signal based handling
Group Interface

• Several concepts of "Group" in SAM
  – Multicast group
  – Small group
  – Hybrid group... (Multicast - Unicast)

• We currently employ simple server-based group management.
  – will implement p2p group management.

• Every group/member is identified through GroupURI / MemberURI
Group Management Methods

- `getSAMGroupMemberList(GroupURI)`
- `getSAMGroupMember(MemberURI)`
- `getSAMGroupInfo(GroupURI)`
- `getSAMGroupAddress(GroupURI)`
- `addGroup(newGroupURI, path)`
- `deleteGroup(GroupURI)`
- `addMember(GroupURI)`
- `joinGroup(GroupURI, properties)`
- `deleteMember(MemberURI)`
- `setProperty(MemberURI, Key, Value)`
- `deleteProperty(MemberURI, Key, Value)`
Sample Group URI

http://group.samtk.org/ietf71

<?xml version="1.0"?>
<response>
  <request type="get"><result status="success"/></request>
  <group><group-ref id="159" uri="http://group.samtk.org/ietf71" member-num="2"/>
    <path>ietf71</path>
    <parent-group><group-ref id="1" uri="http://group.samtk.org/root" member-num="0"/>
    </parent-group><properties/>
    <subgroups/><members><member><member-ref id="2212" uri="http://group.samtk.org/member.php/2212"/>
      <group><group-ref id="159" uri="http://group.samtk.org/ietf71"/>
      <properties>
        <property key="Application">SAMVideoConf</property>
        <property key="port">32198</property>
        <property key="address">192.168.3.100</property>
        <property key="name">kawaguti</property>
      </properties>
    </member>
    <member><member-ref id="2213" uri="http://group.samtk.org/member.php/2213"/>
      <group><group-ref id="159" uri="http://group.samtk.org/ietf71"/>
      <properties>...
    </members>
  </group>
</response>
Protocol Interface

- Interface for protocol plugins

- Overlapping socket interface to Multi- Destination

- Multi-destination is passed by SAMGroupAddress
Protocol API ( = Plug-in Interface)

- setGroup(GroupId)
- writeDatagram(char *, int, GroupId)
- readDatagram(char *, int, HostId)

- bool hasPendingDatagrams()
- bind(port)

Currently, very simple style.
How to handle “Protocol Specific Information”

• Use Group Server extension field.
• Group Server has XML extension capability.
  – Attribute – Value Pair is currently used

• So if we utilize the group server information, we can deploy hybrid SAM network.
Simple SAMTK group

Node A → Node B → Node D → Node C → Node A

Group-0
Node A
Node B
Node C
Node D

Full Mesh Network by Multi-Unicast
Relay Node on SAMTK

• Transfer a packets between different Group / different Transport

• May transform media into different type
  – Bitrate conversion
  – Video composition / division
  – Audio composition
Multi-hop SAMTK group

Introduce “Relay Node”

Inter-Group Network by SAMTK

Group-0
- Node A
- Node B
- Node C

Node-C Relays
- Group-0
- Group-1

Group-1
- Node-C
- Node-D
Hybrid SAMTK group

Introduction of Relay Node

Group-0 (Multicast)
Node A
Multicast-group X

Node-A Relays
Group-0
Group-1

Group-1 (Unicast)

Node-C Relays
Group-1
Group-2

Group-2 (Xcast)
Node-C
Node-D
Node-E

Hybrid Network by SAMTK
Hand-made Hybrid SAM Network

• User can add each hybrid network via SAMTK Group Server.

• Might be a step for automatic hybrid network.
SAM Network through NAT/FW

Utilization of "Relay Node"

Global Internet

Node A

NAT/FW

Node B

NAT/FW

Node C

NAT/FW

Node D

Node A relays
- Group -0
- Group -1
- Group -2

Group-0
- Node-A
- Node-B

Group-1
- Node-A
- Node-C

Group-2
- Node-C
- Node-D

Automatic configuration of these groups is desired
Summary

• SAMTK enables quick test/deployment of a new multipoint communication protocols.
• Researchers take a benefit of real-world application based on SAMTK.
  – easy to compare with other protocols.
• Application developer take a benefit of multi-protocol implementation.
• We can deploy “Hybrid SAM Network” by introducing “Relay Node”.
Comments!

• Please send comments to kawaguti @ nagoya-u.jp

• http://sourceforge.net/projects/samtk

• Documentation
  – http://samtk.org
Towards a P2P Group Management

• We can use P2P Network for Group management.

• But it is still in early stage to consider about Group Management Protocols.