

Network Programming TDC 561

Lecture # 7: Multicast Network Programming

Dr. Ehab S. Al-Shaer

School of Computer Science & Telecommunication
DePaul University
Chicago, IL

Socket Options

- * Various attributes that are used to determine the behavior of sockets.
- * Setting options tells the OS/Protocol Stack the behavior we want.
- * Support for generic options (apply to all sockets) and protocol specific options.

Dr. Ehab Al-Shaer/Network Programming

Getting option values

Gets the current value of a socket option:

```
int getsockopt( int sockfd,  
               int level,  
               int optname,  
               void *opval,  
               socklen_t *optlen);
```

level specifies whether the option is a general option or a protocol specific option (what level of code should interpret the option) see P. 179 (Stevens).

Dr. Ehab Al-Shaer/Network Programming

Setting option values

Sets the current value of a socket option:

```
int setsockopt( int sockfd,
               int level,
               int optname,
               const void *opval,
               socklen_t optlen);
```

level specifies whether the option is a general option or a protocol specific option (what level of code should interpret the option).

Dr. Ehab Al-Shaar/Network Programming

General Options

- * Protocol independent options.
- * Handled by the generic socket system code.
- * Some options are supported only by specific types of sockets (SOCK_DGRAM, SOCK_STREAM).
- * This was just an overview
 - there are many details associated with the options described.
 - There are many options that haven't been described.

Dr. Ehab Al-Shaar/Network Programming

Summary of Socket Options

- * Socket Level
 - SO_SNDBUF, SO_RCVBUF, SO_KEEPALIVE, SO_BROADCAST, SO_REUSEADDR, SO_RESUEPORT
- * IP Level
 - IP_TTL, IP_MULTICAST_IF, IP_MULTICAST_TTL, IP_MULTICAST_LOOP, IP_ADD_MEMBERSHIP, IP_DROP_MEMBERSHIP
- * TCP Level
 - TCP_KEEPALIVE, TCP_MAXSEG, TCP_NODELAY
- * See Stevens' book P. 179

Dr. Ehab Al-Shaar/Network Programming

TCP socket options

- * **TCP_KEEPAIVE**: set the idle time used when **SO_KEEPAIVE** is enabled.
- * **TCP_MAXSEG**: set the maximum segment size sent by a TCP socket.
- * **TCP_NODELAY**: can disable TCP's Nagle algorithm that delays sending small packets if there is unACK'd data pending.

Dr. Ehab Al-Shaar/Network Programming

Multicasting Concept

- * **Unicast**: a packet is received by a *single* interface
- * **Broadcast**: a packet is received by *all* interfaces
- * **Multicast**: a packets is received by multiple interfaces (receivers) using a *single* local "transmit" operation

Dr. Ehab Al-Shaar/Network Programming

IP Multicast and the Mbone

- * **IP Multicast group addressing:**
 - Group level (process): IP mcast address + port
 - IP level: multicast address (224.x.x.x)
 - Ethernet level: IP mcast address mapping

Dr. Ehab Al-Shaar/Network Programming

IP Multicast and the Mbone

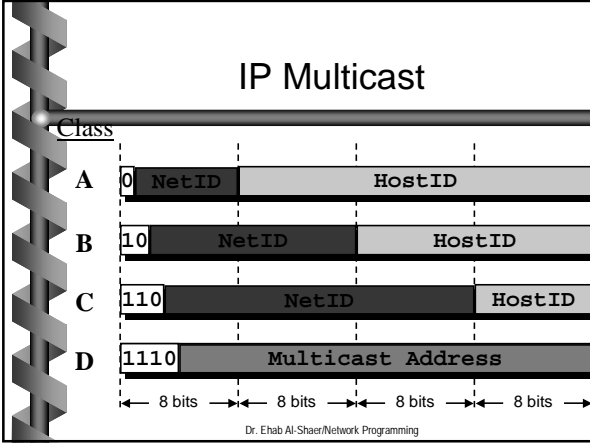
- * **Server-Oriented Multicast**
 - source sets up one-to-many multicast group
 - discourages dynamic groups
- * **Receiver-Oriented Multicast (Deering, 1991)**
 - senders need not be members (called open group)
 - Unlimited group size
 - no topological restrictions on membership
 - membership dynamic and autonomous
 - host groups may be transient or permanent (e.g., 224.0.0.1)

Dr. Ehab Al-Shaar/Network Programming

IP Multicast

- * host-group model
- * network-level; same packet format, different address
- * routers do all of the work
- * special IP addresses:
224.0.0.0 - 239.255.255.255
- * 28 bits => 268 million groups (plus scope)
- * ttl value limits distribution

Dr. Ehab Al-Shaar/Network Programming



IP Multicast in WAN

- * IGMP and Multicast Routing Protocols
- * Prune branches where no members and branches not on shortest paths

Dr. Ehab Al-Shaar/Network Programming

Mbone

- * Mbone = multicast backbone
- * virtual network overlaying Internet
- * needed until mcast capable routers deployed
- * IP in IP encapsulation

Dr. Ehab Al-Shaar/Network Programming

Mbone Protocols

- * **IP**
- * **UDP**: best effort
- * **RTP**: real-time transport
- * **RSVP**: resource reservation protocol
- * **SDP/SAP**: session description, announcement protocols

conf. control	audio	video	shared tools	session directory		
				SDP		
RSVP	RTP, RTCP		SAP	HTTP	SMTP	
UDP				TCP		
IP						
physical layer						

Dr. Ehab Al-Shaar/Network Programming

Session Protocols

- * **Session Description Protocol (SDP)**
 - used to describe the contents of mcast sessions
 - name, purpose, start time, duration
 - media (type, transport protocol, format)
 - how to receive media
- * **Session Announcement Protocol (SAP)**
 - mcast protocol for SDP
 - periodic transmission to known mcast address
 - frequency depends on other announcements and scope
- * **Session Directory (SDR)**
 - used to allocate multicast addresses to sessions
 - random allocation currently used
 - advertises multicast sessions uses SDP

Dr. Ehab Al-Shaar/Network Programming

Mbone Applications

- * **freeware**
 - vic (video), vat (audio), wb (whiteboard), NTE (Text Editor)
- * **commercial**
 - IP/TV - teleconferencing (Precept)

Dr. Ehab Al-Shaar/Network Programming

IP Multicast Data Structure

IP_ADD_MEMBERSHIP	struct ip_mreq	Join mcast group
IP_DROP_MEMBERSHIP	struct ip_mreq	Leave mcast group
IP_MULTICAST_IF	struct in_addr	Interface for mcast out
IP_MULTICAST_TTL	u_char	Set TTL for mcast out
IP_MULTICAST_LOOP	u_char	On/off loopback

```
struct ip_mreq {
    struct in_addr    imr_multiaddr /*class D mcast address */
    struct in_addr    imr_interface /* addr of local interface */
}
```

Dr. Ehab Al-Shaar/Network Programming

Multicast Client/Server Algorithm & Examples

* Algorithm

- UDP client/server algorithm
- use `setsockopt()` to join before bind or send

* Multicast Examples

- Simple SDR Example P. 505
- Send/Receive Multicast Example P. 510
- NTP Example P. 512
- SUN Example
 - See www.depaul.edu/~ealshaer/courses/tdc561

Dr. Ehab Al-Shaar/Network Programming
