Network Magic: Multicasting, UDP and IGMP

Some amazing feats that non TCP networking can accomplish.

Japan Linux Symposium, Tokyo 2009

Christoph Lameter <cl@linux-foundation.org>

Introduction

- High Performance Computing
- Performance
- The importance of event processing
- Events are a significant kernel problem.
- NUMA is mainstream.
- TCP vs. UDP
- Multicast vs Unicast
- OS UDP/Multicast problems
- Basic issues with OS and contemporary fast networking.
- Conclusion

The Magic

- Introduction
- Zap it: UDP is fast, latencies are eliminated
- Multicast can reach multiple recipients with a single message.
- Reach the unknown: Multicast can make communication indepedent of IP addresses.
- Switch self organization: IGMP protocol can organize a switch fabric
- State of the network
- Timing issues

Zap it over there. UDP speed.

- TCP usually takes hundreds of microseconds host to host.
- UDP sends trigger immediate NIC actions.
- UDP on 1G Ethernet takes less than 40 microseconds
- UDP via Infiniband Ethernet takes less than 15 microsecond
- Native Infiniband can signal in 2 microseconds.

Magical appears everywhere

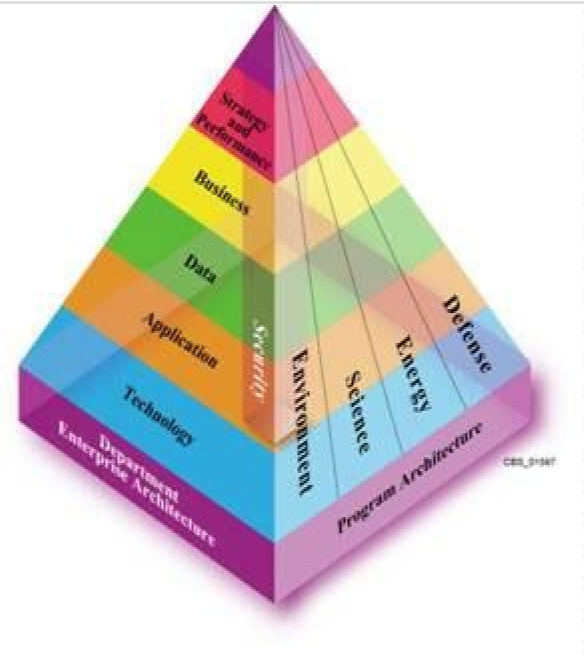
- UDP Multicast can send a single message that is received at multiple destinations.
- Minimal sender overhead
- Maximum exposure in minimal time.
- Multicast targeting is handled through subscriptions.
- Most economical way of using a network for information distribution.

Magical communication without destination addresses

- Multicast messages are send to Multicast addresses.
- Any subscriber can pick them up.
- Communication can occur entirely over multicast without the need to know the IP addresses of the endpoints.
- The broadcast domain defines the scope of how far information reaches.

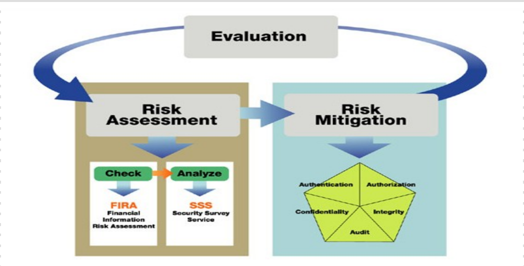
Magical Organization and Reorganization of network communication channels

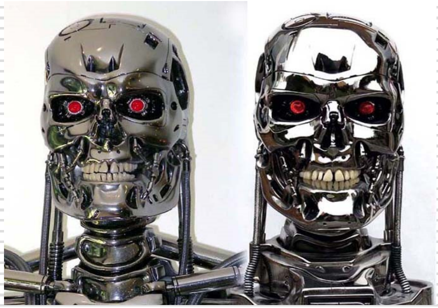
- Multicast channels are dynamically established.
- Behind the scenes the IGMP protocol is used to establish communication paths through the network fabric to endpoints.
- Fully transparent to the applications. Applications only indicate which Multicast channels they want to listen to.



A self aware network?

- Services can be advertised by machines via multicast.
- Detection of failing services is possible.
- Fallback is easy since no machine IP addresses are required for multicast.
- The network is self-healing and can react to shifting load conditions.
- Network nodes can track network events in a detailed way and react to events in a very fast way.
- Then the current state of the network becomes important. Knowledge about information flows and working services is essential and a application interacts with the network instead of interacting with endpoints (TCP model).



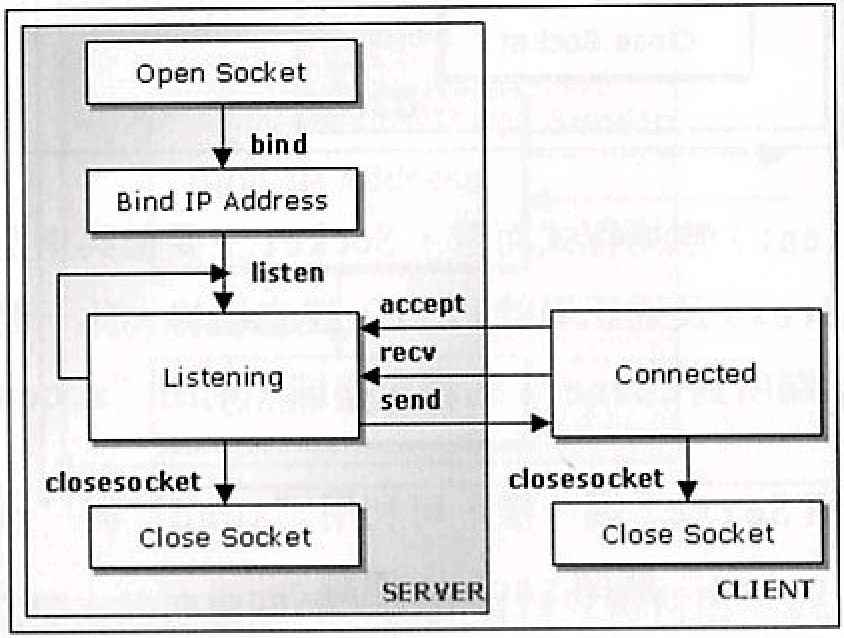


TCP vs. UDP

TCP	**UDP**
Connection oriented	Connectionless
Sending queues data	Send out immediately to the net
Large data volulmes	Small packets of information
Ordered	Unordered
Reliable	Delivery not guaranteed
Congestion management	Saturates links
Complex overhead and delays	Minimal processing overhead
Unlimited packet size	Max 64kbyte packet size
1-1 protocol	Multicast Capable
	Many senders, many receivers

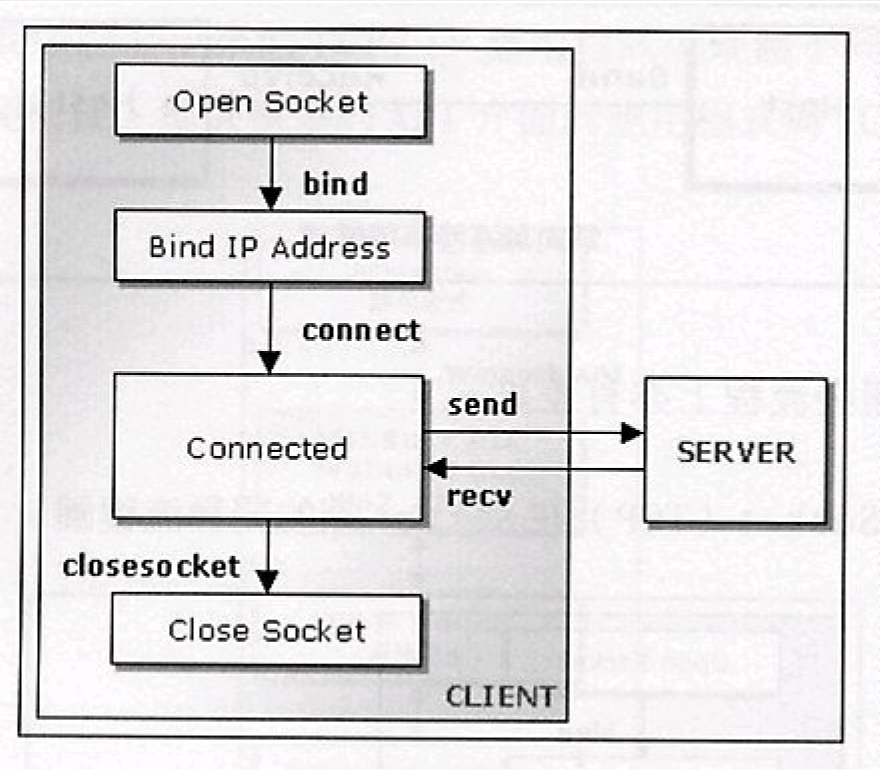
TCP Server Sample

- Listening socket is created
- Listen function creates data socket
- Only the data socket is unique 1-1 communication channel
- Arbitrary data lengths
- Complex and slow process.
- Significant network traffic to establish a connection



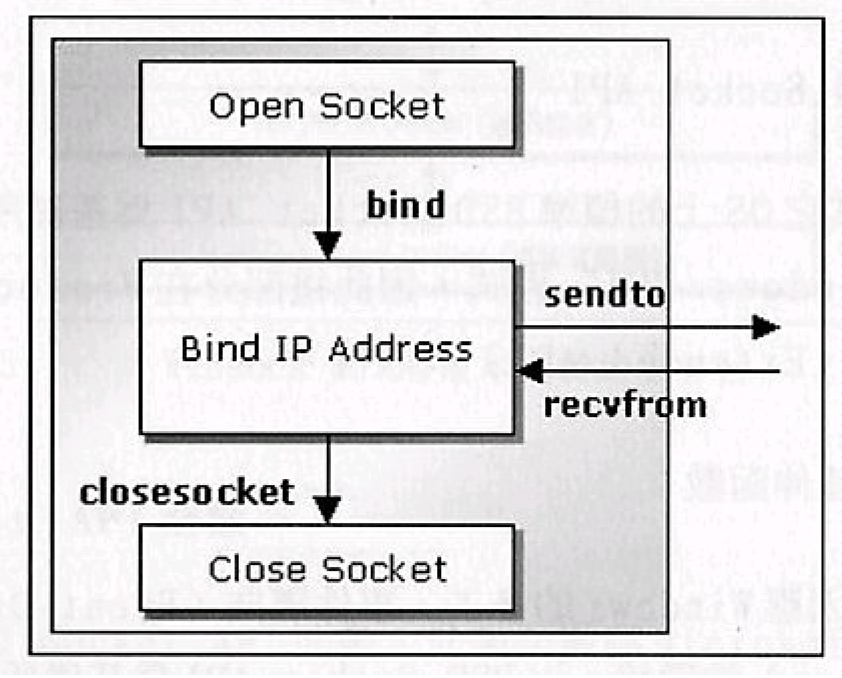
TCP client sample

- Reasonably simple to code
- OS does complex negotiation to establish a connection.
- Requirement to initiate and close connection



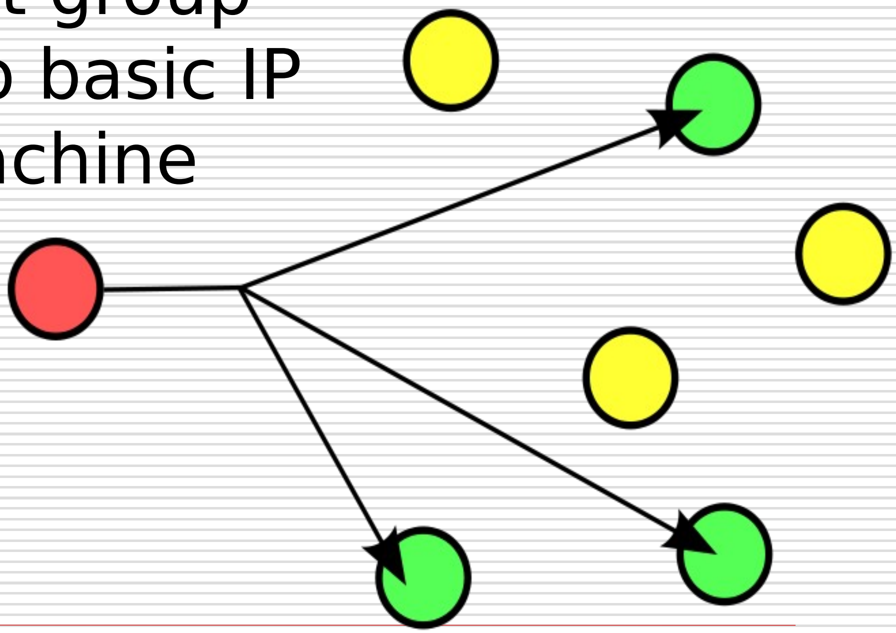
UDP Peer to peer

- Open a generic socket and bind it to generic address
- Single OS call sends to abitrary host
- Single OS call receives from arbitrary host.



Broadcast, Unicast, Multicast

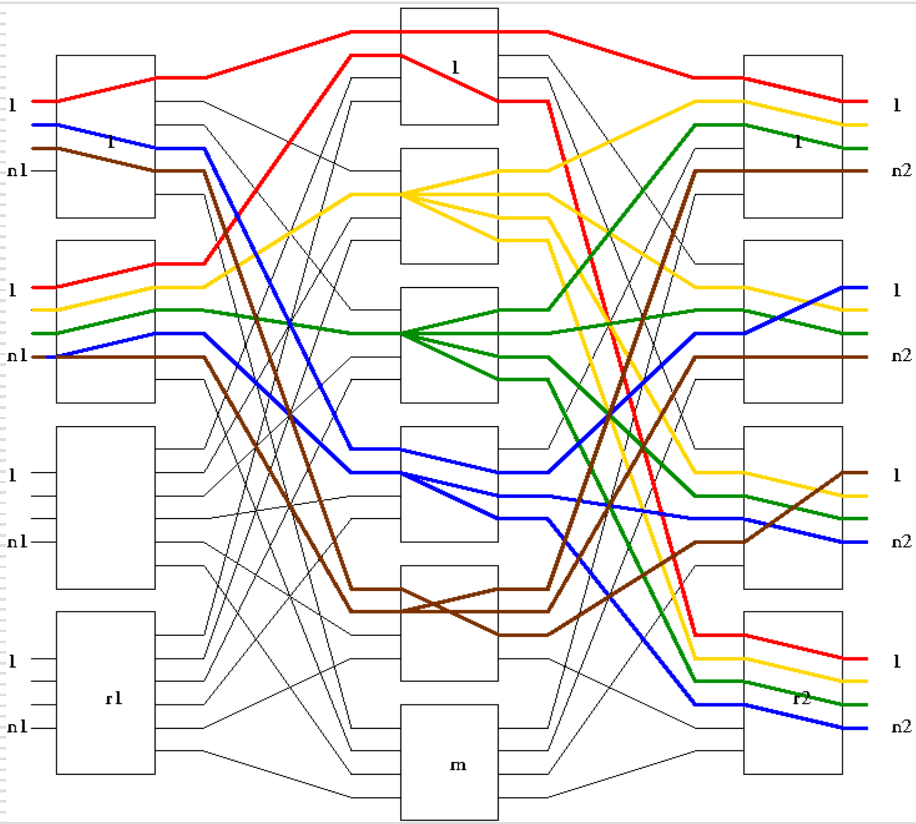
- Unicast: one sender, one receiver
- Broastcast: One sender all receiving
- Multicast: Send to a group of senders
- Must join the Multicast group
- Typically UDP but also basic IP
- Receivable by any machine
- Must limit scope



Multicast via UDP

- Communication channels are established in a broadcast domain
- Channel is given an IP address from a special range
- Any system in the broadcast domain can communicate to that channel using the IP
- Any system can listen to traffic on that broadcast domain via a special join command by the OS.
- Level 3 Multicast IP address (28 bits) is mapped to level 2 Mac address (23 bits).
- No congestion control: An out of control sender can bring down the network
- Multicast traffic is routed via network magic in the IP switch fabric (IGMPv2, IGMP snooping, PIM)

Multicast in a switch fabric



Multicast address space

- Class D 224.X.X.X -> 239.X.X.X (224.0.0.0/4)
- 224.0.0.X reserved for special uses (IANA managed)
- 233.A.A.X Public Multicast for autonomous systems
- 239.X.X.X local use
- Level 2 Multicast Addresses are 6 bytes. 3 have fixed values leaving 3 bytes. One bit of those is used for other functionality.
- Use low 23 bits for matching because NIC can only match those. Multiple MC addresses can have same Level 2 Mac address.
- NIC software discards wrong MC addresses.
- So really only 224.0.0.0/23 or so available

Layer 2 processing

- Multicast addresses have the form 01-00-5E-XX-XX-XX
- Switches route (or switch) traffic according to the lower 24 bits in the Mac address (bit 23 = 1 means reserved address)
- Switches listen to MC packets coming from a host, isolate the Multicast group and determine routing to the targets based on routing tables established via IGMP
- Most switches do store and forward for multicast packets (1G technology)
- Newer 10G switches can perform cut-through to multiple ports and thereby reduce latency.

IGMP (v 1 - 3)

- Establishes Multicast routing / switching
- Layer 3 protocol that determines layer 2 switching.
- Used to subscribe and unsubscribe from a MC address
- Switch can query hosts for a list of currently subscribed to Multicast addresses (necessary if the switch has lost state).
- Inter switch communication establishes routes
- IGMP v1 = Hosts can join. Time out mechanism
- IGMP v2 = Hosts can join and leave
- IGMP v3 = Source based Multicast subscriptions

ASM vs. SSM
Source based Multicasting

- ASM = IGMPv2 (224.0.0.0/4) [Any Source MC]
- SSM = IGMPv3 (232.0.0.0/8) [Source specific MC]
- SSM distinguishes MC channels based on source (S, G) vs. ASM (G)
- SSM allows subscription to multiple explicitly specified sources for a channel
- Multicast groups that are host specific. Different multicast domains
- SSM support in Linux was a recent addition before RH 4 was released. Generally stable in RH5 and later.

NIC Cards and Multicasting

- Ethernet NICs receive traffic for their MAC address. Plus broadcast traffic to FF:FF:FF:FF:FF:FF for ARP etc.
- MC reception requires reception on additional MAC addresses 01:00:5E:XX:XX:XX
- Nics have a table of Mac addresses that they listen to. Traffic to other addresses is discarded. Most can only listen to a few addresses. 15 is high (Broadcom)
- If a host subscribes to more than 13 broadcast channels then the NIC is put into *allmulti* mode in which all Multicast traffic is processed. OS begins to filter traffic.
- If switches only direct traffic of interest to the host (via IGMP etc) then useless packets will not be dropped by a host in *allmulti* mode.

Multicast in Linux

- Os tracks multicast subscription
- Support IGMP protocol to talk to switch fabric
- IGMP v2 support since 2.4
- IGMP v3 support since 2.6.7
- Display via *netstat -g*

Special Multicast Socket Options

- IP_ADD_MEMBERSHIP: Join group. Sends IGMP join
- IP_DROP_MEMBERSHIP: Leave group. Sends IGMP leave
- IP_MULTICAST_LOOP: Configure loopback behavior
- IP_MULTICAST_TTL: Time to live (scoping)
- IP_ADD_SOURCE_MEMBERSHIP: Subscribe to Multicast group from a specific IP source
- IP_DROP_SOURCE_MEMBERSHIP
- IP_BLOCK_SOURCE: Block multicast from IP
- IP_UNBLOCK_SOURCE

Linux UDP/Multicast issues

- Broken flow control to NIC. Network stack may drop UDP packets due to internal congestion.
- Dropped packets were not accounted until 2.6.32. Counter update was broken in 1999.
- Mysterious vanishing UDP packets phenomenon in deployed Linux base.
- Linux cannot sent UDP at line rates unless special measures are taken.
- Fix: Rely on throttling through SO_SNDBUF (socket output buffer). If SO_SNDBUF < size of data in packets bufferable by device then packet loss will not occur.
- Keep SO_SNDBUF small (<20k) in order to avoid network stack dropping packets if bursts occur.

Fundamental OS problems

- System calls take 5-10 microseconds
- Modern Fabrics can forward packets in 1 – 2 microseconds
- Must have kernel bypass methods to use capabilities of current hardware.
- Even with kernel bypass OS noise can have significant impact on event propagation.

Conclusion

- Multicast is a powerful mechanism that avoids having to know IP addresses for communication
- System can communicate on certain topics via MC addresses without knowing their identities.
- Organic network behavior
- Linux brokenness in this area has to be fixed.
- Fundamental OS noise and bypass issues ahead.