Welcome to this course! My name is Jasleen Kaur and I'm the instructor for this course.

We'll spend today's class talking about the course outline, the course requirements, and your first homework. But before we do that, I'd like for us all to spend some time and introduce ourselves. So if each of you can say 3 things: (1) what is your name, (2) what is your major and year, and (3) what is it that you expect to learn from this course.

So let me begin: my name is Jasleen and I conduct research in the design and analysis of networks and distributed systems. And I'm looking forward to a term of exciting ideas with all of you.

How many have NOT had socket programming?

What do you think is the "Internet"?
Multicast: one sender to many receivers

• **Multicast**: act of sending datagram to multiple receivers with single “transmit” operation

• **Question**: how to achieve multicast?

Network multicast:
- Routers actively participate in multicast, making copies of packets as needed and forwarding towards multicast receivers

Multicast group concept: use of indirection
- Host addresses IP datagram to multicast group
- Routers forward multicast datagrams to hosts that have “joined” that multicast group

Internet Multicast Service Model

[Diagram showing network multicast with IP addresses and multicast groups]

[Diagram showing multicast group concept with host addresses and indirection]
Multicast via Indirection: why?

- Naming and forwarding in IP tailored for point-to-point communication
- Indirection
  » Provides flexible naming
  » Decouples sender from receivers (and their joins and leaves)

Mobility and Indirection

How do you contact a mobile friend?

Consider friend frequently changing addresses, how do you find her?

☐ search all phone books?
☐ call her parents?
☐ expect her to let you know where he/she is?
Mobility and Indirection:

- Situation:
  - mobile node moves from network to network
  - correspondents want to send packets to mobile node

- Two approaches:
  - *Indirect routing*: communication from correspondent to mobile goes through home agent, then forwarded to remote
  - *Direct routing*: correspondent gets foreign address of mobile, sends directly to mobile
**Mobility: more vocabulary**

- **permanent address**: remains constant (e.g., 129.41.0.0/16)
- **care-of-address**: address in visited network (e.g., 192.129.3.24)
- **visited network**: network in which mobile currently resides (e.g., 70.129.10.24)
- **foreign agent**: entity in visited network that performs mobility functions on behalf of mobile
- **correspondent**: wants to communicate with mobile

---

**Mobility: registration**

End result:
- ♦ foreign agent knows about mobile
- ♦ home agent knows location of mobile
Mobility via Indirect Routing

Indirect Routing: comments

- Mobile uses two addresses:
  - Permanent address used by correspondent (hence mobile location is transparent to correspondent)
  - Care of address used by home agent to forward datagrams to mobile
- Foreign agent functions may be done by mobile itself
- Triangle routing: correspondent-home-network-mobile
  - Inefficient when correspondent, mobile are in same network
**Indirect Routing: moving between networks**

- Suppose mobile user moves to another network
  - registers with new foreign agent
  - new foreign agent registers with home agent
  - home agent updates care-of-address for mobile
  - packets continue to be forwarded to mobile (but with new care-of-address)

- Mobility, changing foreign networks transparent; ongoing connections can be maintained!
**Mobility via Direct Routing: comments**

- overcomes triangle routing problem
- non-transparent to correspondent: correspondent must get care-of-address from home agent
  - what happens if mobile changes networks?

**Mobile IP**

- RFC 3220
- Has many features we’ve seen:
  - home agents, foreign agents, foreign-agent registration, care-of-addresses, encapsulation (packet within a packet)
- Three components to standard:
  - agent discovery
  - registration with home agent
  - indirect routing of datagrams
Mobility via indirection: why indirection?

- Transparency to correspondent
- “Mostly” transparent to mobile (except mobile must register with foreign agent)
- Transparent to routers, rest of infrastructure
  - practical concern if excess filtering is in place in foreign networks
  - sinbad source IP address of mobile is its home address: spoofing?

An Internet Indirection Infrastructure

Motivation:

- Today’s Internet is built around point-to-point communication abstraction:
  - send packet “p” from host “A” to host “B”
  - one sender, one receiver, at fixed and well-known locations
- ... not appropriate for applications that require other communications primitives:
  - multicast (one to many)
  - mobility (one to anywhere)
  - anyone (one to any)
- We’ve seen indirection used to provide these services
  - idea: make indirection a "first-class object"
Internet Indirection Infrastructure (I3)

- Change communication abstraction: instead of point-to-point, exchange packets by name
  - each roobot has an identifier ID
  - to receive packet with identifier ID, receives R stores trigger (ID, R) in network
  - triggers stored in network overlay nodes

Service Model

- API
  - sendPacket(p);
  - insertTrigger(t);
  - removeTrigger(t) // optional

- Rest of system service model (like IP)

- Triggers periodically refreshed by end hosts
  - Q: What is this approach called?

- Reliability, congestion control, flow-control implemented at end hosts, and trigger-storing overlay nodes
Discussion

- Trigger is similar to routing table entry
- Application-level overlay infrastructure
  - Essentially: application layer publish-subscribe infrastructure
- Unlike IP, end hosts control triggers, i.e., end hosts responsible for setting and maintaining "routing tables"
- Provide support for
  - mobility
  - multicast
  - anycast
  - composite services
Such an indirection layer would support a large number of services. For example, to achieve mobility an end host needs only to update its trigger with the new address when it moves from one subnet to another.

Multicast is straightforward to achieve. The only difference between multicast and unicast is that in the case of multicast there are more than one hosts inserting triggers with the same ID.

Mobility
- Receiver updates its trigger as it moves from one subnet to another
  - mobility transparent to sender
  - location privacy

Multicast
- Unifies multicast and unicast abstractions
  - multicast: receivers insert triggers with same ID
- Application naturally moves between multicast and unicast, as needed
  - “impossible” in current IP model
Finally, IL supports composable services, i.e., performing on the fly transformation such as transcoding on the data packets as they travel through the network. To achieve this we replace the packet ID with a stack of IDs, where each identifier excepting the last one identifies a transformation to be applied on packets. The advantage of this solution versus proposed solutions is that you don’t need to find and configure the path, (you just insert the IDs in the proper order). Load balancing and robustness are easy to achieve. Just have more servers implementing the same operations. If one fails, the other one will take transparently over.

Anycast
- Route to any one in set of receivers
- Receiver i in anycast group inserts same ID, with anycast qualifications
- Route to receiver with best match between a and \( \eta \)

Composable Services
- Use stack of IDs to encode successive operations to be performed on data (e.g., transcoding)
- Don’t need to configure path between services
Finally, IL supports composable services, i.e., performing on-the-fly transformation such as transcoding on the data packets as they travel through the network. To achieve this we replace the packet ID with a stack of IDs, where each identifier excepting the last one identifies a transformation to be applied on packets.

The advantage of this solution over previous proposed solutions is that you don’t need to find and configure the path; you just insert the IDs in the proper order. Load balancing and robustness are easy to achieve. Just have more servers implementing the same operations. If one fails, the other one will take transparently over.
Discussion of I3

- How would receiver signal ACK to sender? what is needed?
- Does many-to-one fit well in this paradigm?
- Security, snooping, information gathering: what are the issues?
- In-network storage to handle disconnection?

Indirection: Summary

- We’ve seen indirection used in many ways:
  - multihop
  - mobility
  - Internet indirection

- Uses of indirection:
  - sender does not need to know receiver id:
    - do not want sender to know intermediary identities
    - elegant
    - transparency of indirection is important
    - performance: is it more efficient?
    - security: important issue for I3