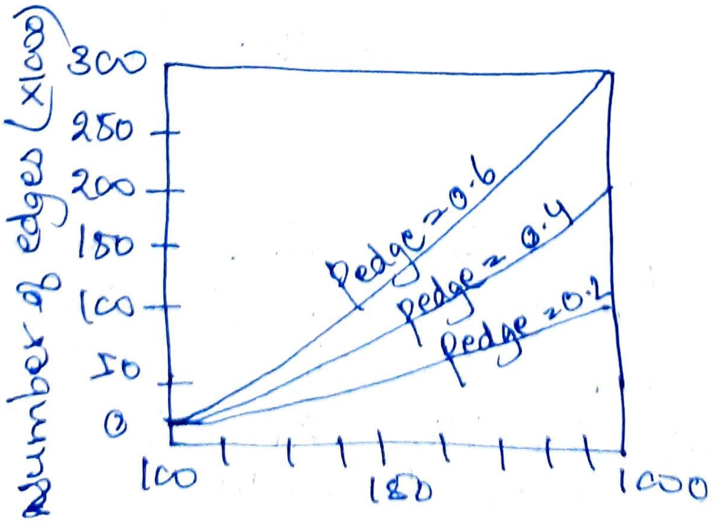
Arunima Singh Thakur, 180905218, Sec C, Roll no 31, Branch CSE, DS Assignment 2,

[illegible]

1) (b) *Flooding

- Essence – P simply sends a message m to each of its neighbours. Each neighbor will forward that message, except to P, and only if it had not seen m before.
- Performance – The more edges, the more expensive.
- Variation – Let Q forward a message with a certain probability Pflood, possibly even dependent on its own number of neighbors (i.e., node degree) or the degree of its neighbours.
- The size of a random overlay as function of the number of



* Epidemic protocols

⇒ Assume there are no write-write conflicts
- Update operations are performed at a single server
- A replica passes updated state to only a few neighbors.
- Update propagation is lazy, i.e, not immediate
- Eventually, each update should reach every replica.

<u>Two forms of epidemics:</u>–

★ <u>Anti-Entropy</u> – Each replica regularly chooses another replica at random, and exchanges state differences, leading to identical states at both afterwards.

★ <u>Rumor spreading</u> – A replica which has just been updated (i.e, has been contaminated), tells a number of other replicas about its update (contaminating them as well).

---

2) <u>Naming</u>

- Names are used to denote entities in a distributed system. To operate on an entity, we need to access it at an access point. Access points are entities that are named by means of an address
- A location-independent name for an entity E, is independent from the addresses of the access points offered by E.

<u>Identifiers</u>

- Pure name – A name that has no meaning at all; it is just a random string. Pure names can be used for comparison only.
- Identifier: A name having some

specific properties.

→ An identifier refers to at most one entity

→ Each entity is referred to by at most one identifier

→ An identifier always refers to the same entity (i.e. it is never reused).

• Observation

An identifier need not necessarily be a pure name, ie., it may have content.

---

1 a) ~~Logical~~ Commands:-

- MPI_bsend → Append outgoing messages to a local send buffer
- MPI_send → Send a message & wait until copied to local or remote buffer.
- MPI_ssend → Send a message and wait until transmission starts.
- MPI_sendrecv → Send a message and wait for reply.
- MPI_isend → Pass reference to outgoing message, and continue.
- MPI_issend → Pass reference to outgoing message, and wait until receipt starts.
- MPI_recv → Receive a message; block if there is none.
- MPI_irecv → Check if there is an incoming message, but do not block.

Logical commands:-

- MPI_LAN — Logical AND
- MPI_LOR - Logical OR
- MPI_LXOR - Logical Exclusive OR

MPI assumes communication takes place within a known group of processes. Each group is assigned an identifier. Each process within a group is also ~~to~~ assigned a (local) identifier. A (groupID, processID) pair therefore uniquely identifies the source or destination of a message, and is used instead of a transport-level address. There may be several, possibly overlapping groups of processes involved in a computation & that are all executing at the same time. At the core of MPI are messaging operations to support transient communication.