# Distributed Systems

# Communication

Rik Sarkar
James Cheney

University of Edinburgh
Spring 2014

# Types of networks

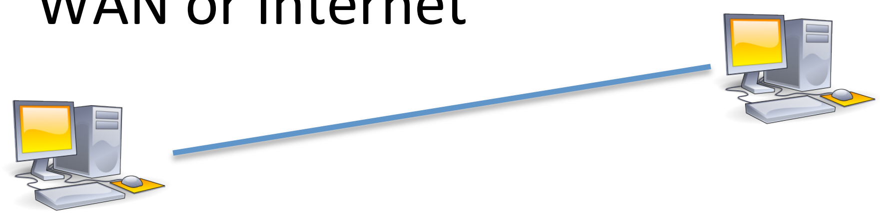
- Local area networks (Ethernet)
- Wide area networks
- Wireless LANs (WiFi)
- Wireless WANs (Cellular networks, 3G, 4G etc)
- Internetworks – comprising of many LANs, WANs etc connected together, allowing communication between computers. E.g. Internet.

# Packets

- Networks communicate data in messages of fixed (bounded) size – called packets
- More data requires more packets
- Number of messages or packets transmitted is a measure of communication used

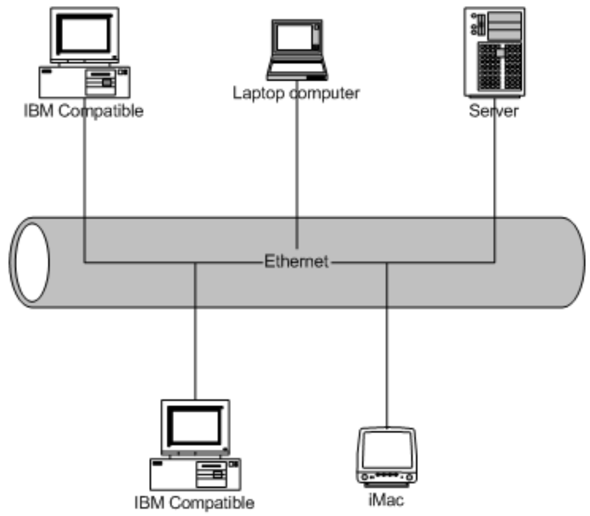
# Types of Communications

- **Point to point communication**
  - Message goes from one computer to another computer
  - There must be a connecting link
    - E.g. A LAN wire directly connecting the two
    - Or, the two computers are in the same LAN
    - Or they are in different LANs, but connected through WAN or Internet



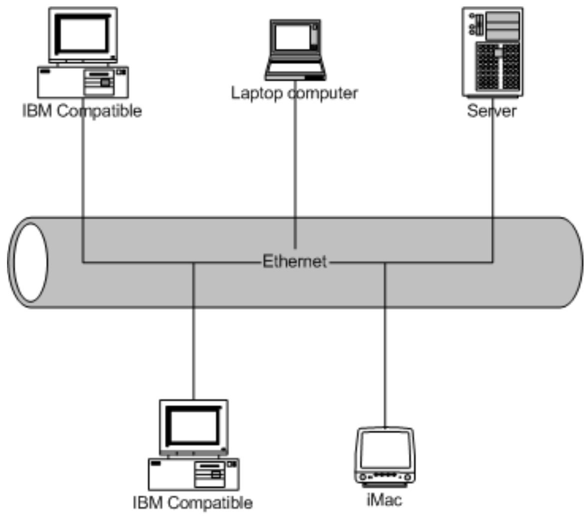
# Types of Communications

- **Broadcast**
  - Message goes from one computer to *all* other computers (restricted to some set)
    - For example, all other computers in the LAN, or some other system in consideration
  - Ethernet LAN is a broadcast medium
    - All computers are connected to a wire. They transmit messages on the wire and all can receive
  - Wireless LAN (WiFi) is a broadcast medium
    - Electromagnetic waves is the common medium



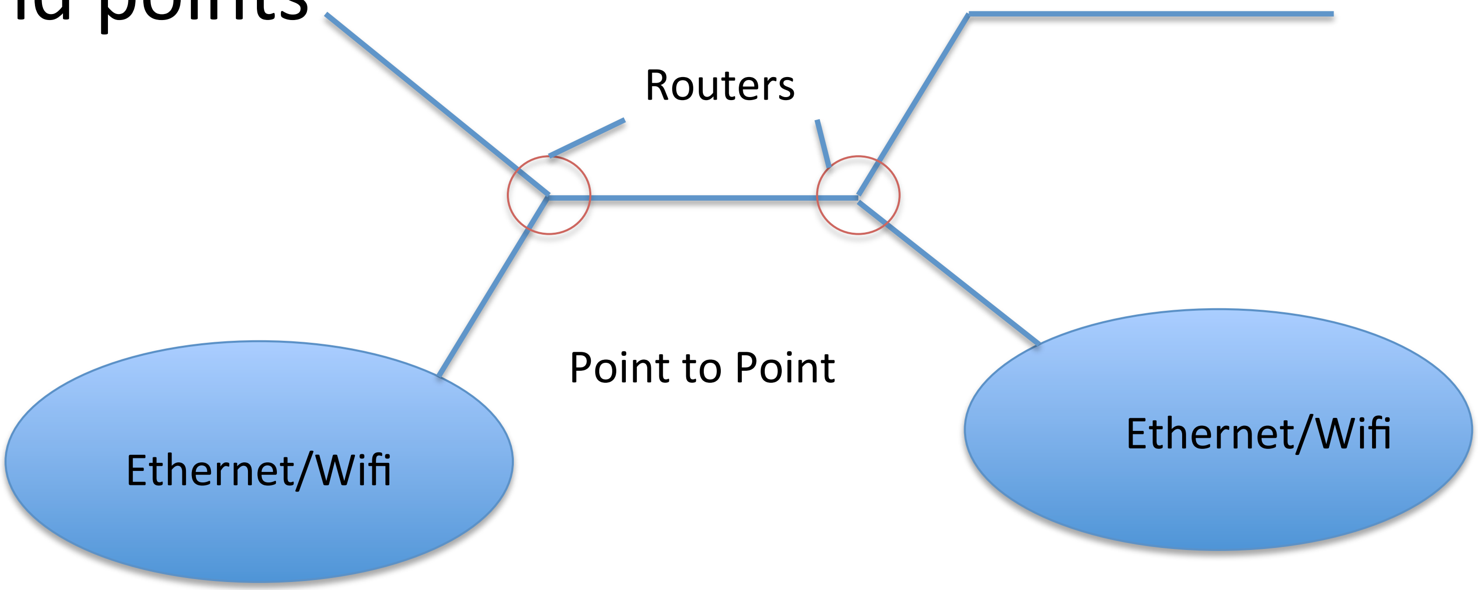

# Types of Communications

- **Broadcast**
  - Useful when message has to be sent to all computers
    - E.g. Multiplayer games, streaming a live video etc
  - Can be used to achieve point to point communication
    - Send message, with id of the receiver. Everyone else rejects it.
    - Does not scale well when there are many nodes in the system.
    - When Many pairs of nodes try to communicate using the same medium, messages clash.




# Real life networks

- Point to point for long distance & internetworking
- Broadcast for local, short range at the LAN end points

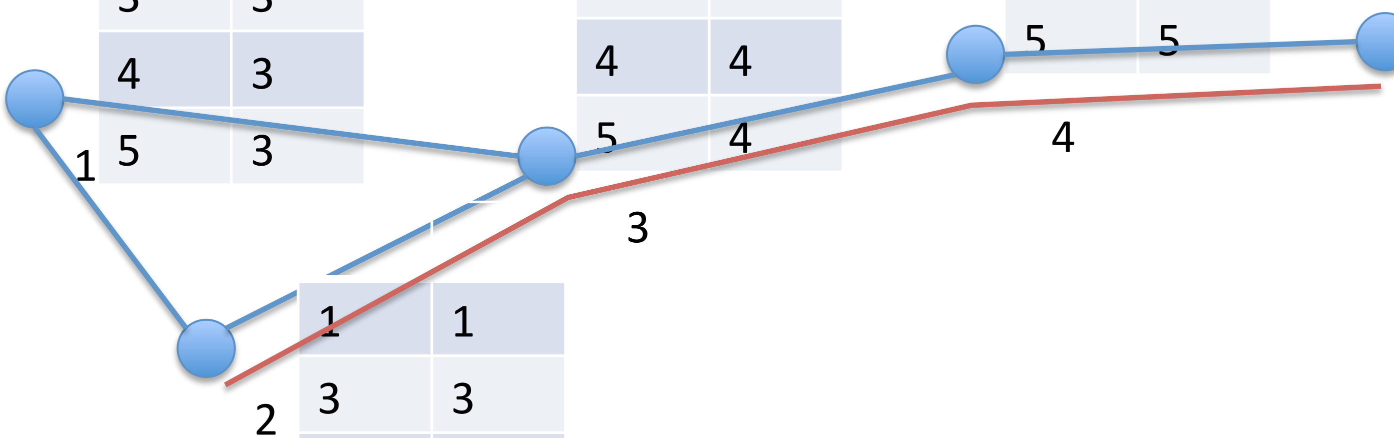


# Medium access in LANs

- The difficulty of using broadcast
  – If more than one node transmits, packets collide and both messages get garbled
  – MAC protocols ensure that when one node is transmitting, others keep quiet
  – If there is still collision, message is retransmitted

# Routing

- Finding a path in the network
- Every node has a routing table




Node 1:

| | |
|---|---|
| 2 | 2 |
| 3 | 3 |
| 4 | 3 |
| 5 | 3 |

Node 2:

| | |
|---|---|
| 1 | 1 |
| 3 | 3 |
| 4 | 3 |
| 5 | 3 |

Node 3:

| | |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 4 | 4 |
| 5 | 4 |

Node 4:

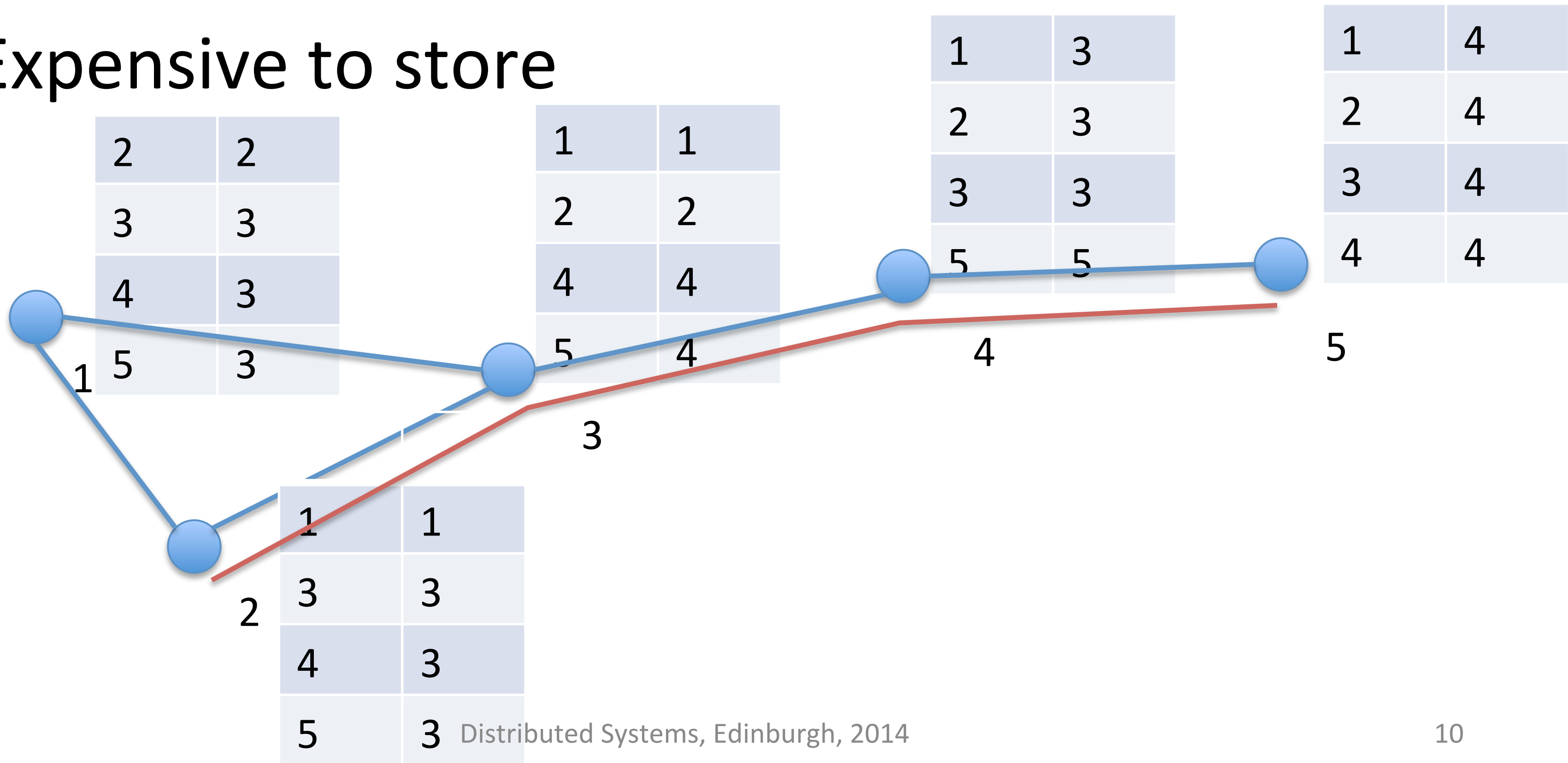
| | |
|---|---|
| 1 | 3 |
| 2 | 3 |
| 3 | 3 |
| 5 | 5 |

Node 5:

| | |
|---|---|
| 1 | 4 |
| 2 | 4 |
| 3 | 4 |
| 4 | 4 |

# Routing

- Finding a path in the network
- Every node has a routing table Size n-1
- Expensive to store




Node 1:

| | |
|---|---|
| 2 | 2 |
| 3 | 3 |
| 4 | 3 |
| 5 | 3 |

Node 2:

| | |
|---|---|
| 1 | 1 |
| 3 | 3 |
| 4 | 3 |
| 5 | 3 |

Node 3:

| | |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 4 | 4 |
| 5 | 4 |

Node 4:

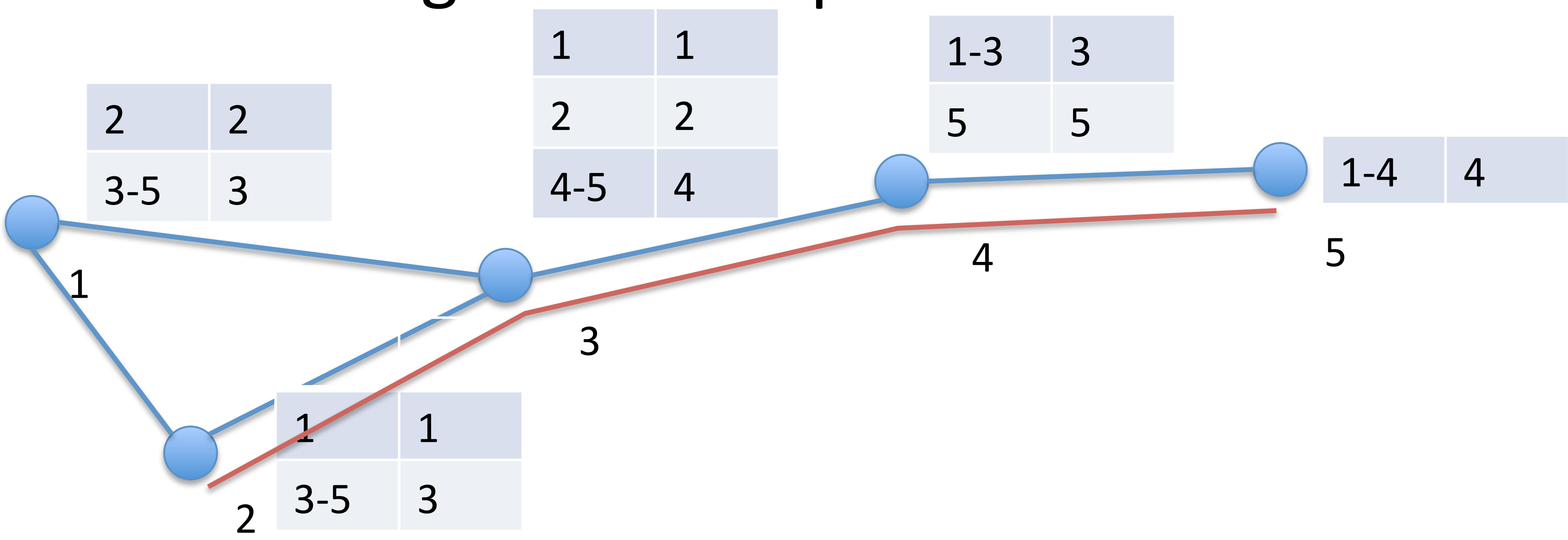
| | |
|---|---|
| 1 | 3 |
| 2 | 3 |
| 3 | 3 |
| 5 | 5 |

Node 5:

| | |
|---|---|
| 1 | 4 |
| 2 | 4 |
| 3 | 4 |
| 4 | 4 |

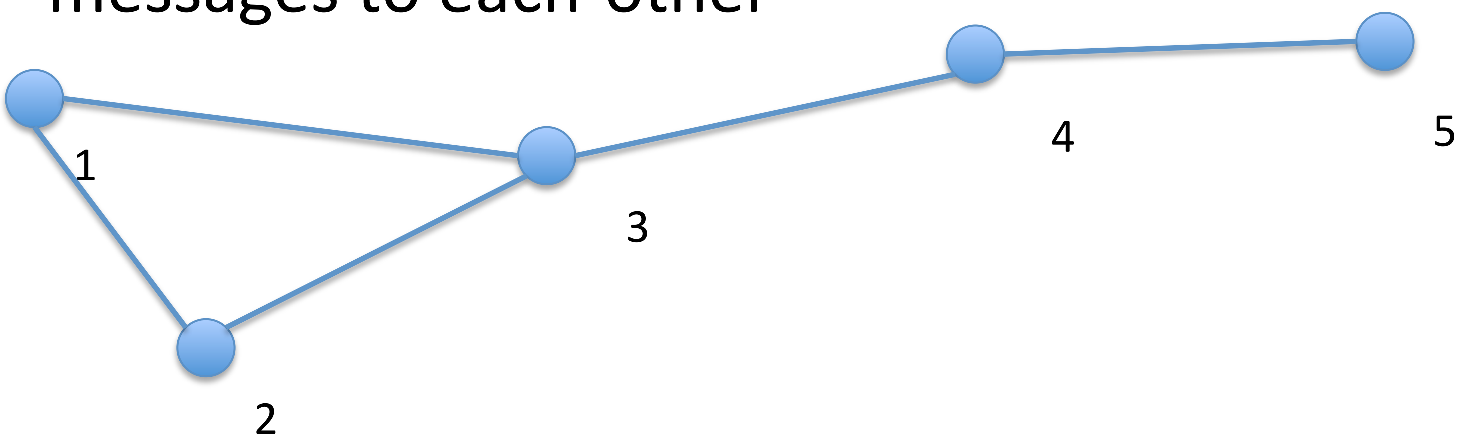
# Routing

- Smaller routing tables by combining addresses
- Used in IP (Internet) routing
- Smaller routing tables are preferable



# Netowrks as graphs

- Note:
  - Networks are usually drawn as graphs
  - Vertices are nodes/computers
  - Edge means these nodes can directly send messages to each-other



# An announcement..

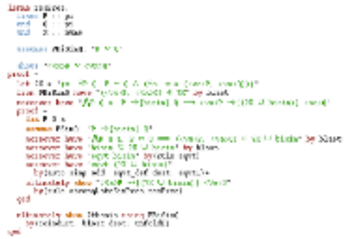
If you are interested in doing research
on
Distributed Systems
or related areas

lfcs | Laboratory for Foundations of Computer Science

cisa | Centre for Intelligent Systems and their Applications

icsa | Institute for Computing Systems Architecture

epcc



# PhD in Pervasive Parallelism

- Program
- Specify
- Design
- Verify
- Optimize
- Apply

http://pervasiveparallelism.inf.ed.ac.uk

intel · Agilent Technologies · Microsoft · altran · criticalblue (Accelerating Embedded Software) · Qualcomm · IBM · codeplay · ACE · Samsung · ARM · freescale · Synopsys · wolfson microelectronics · Oracle · amazon

- Contact James Cheney for:
  - Database or web programming languages
  - Data synchronization

- Contact Rik Sarkar for:
  - Algorithms for distributed computing
  - Sensor networks
  - Mobile networks

# MS/UG/MInf Projects in distributed Systems

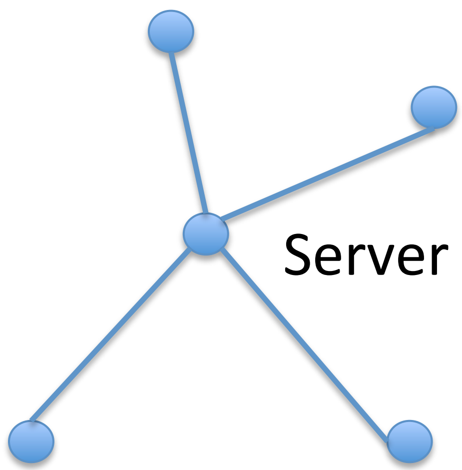
- Contact us to discuss more

# Communication cost

- A distributed computation should be *efficient*
  - Should use few messages
- Cost of a distributed computation:
  - Number of messages transmitted

# Example 1

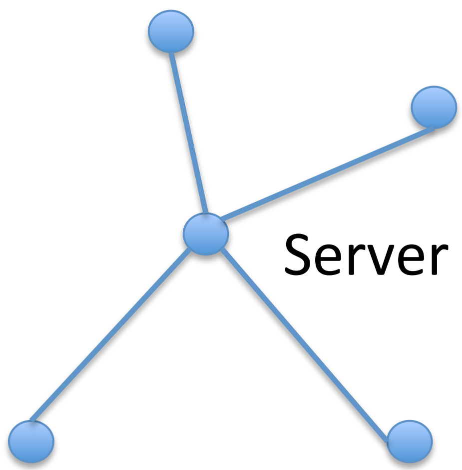
- A simple distributed computation:
  - Each node has stored a numeric value
  - Compute the total of these numbers




How many messages does it take?

# Example 1

- A simple distributed computation:
  - Each node has stored a numeric value
  - Compute the total of the numbers

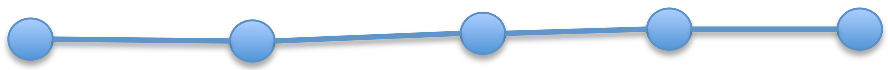



4

# Example 2

- A simple distributed computation:
  - Each node has stored a numeric value
  - Compute the total of the numbers

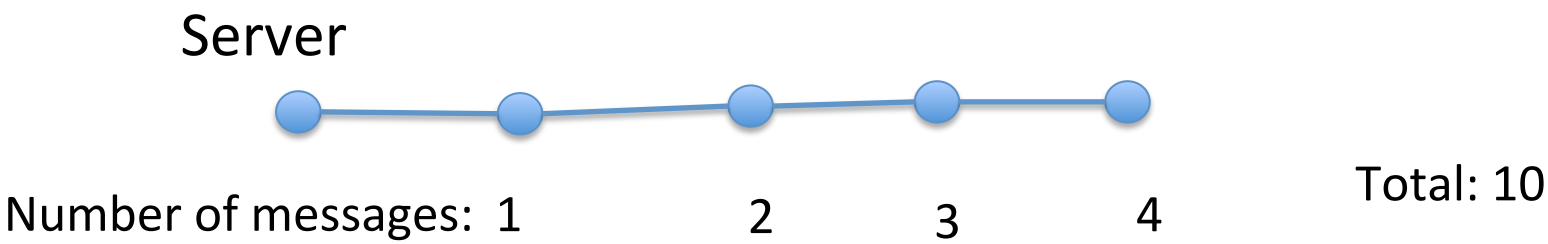
Server



How many messages does it take?

# Example 2

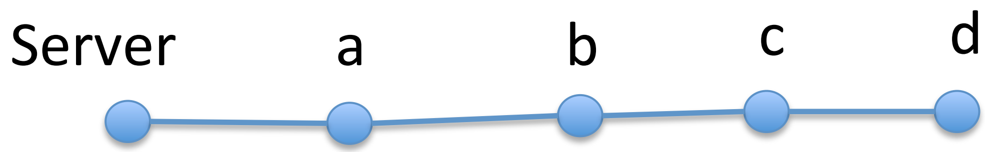
- A simple distributed computation:
  - Each node has stored a numeric value
  - Compute the total of the numbers



- Complexity may depend on the Network

# Example 2

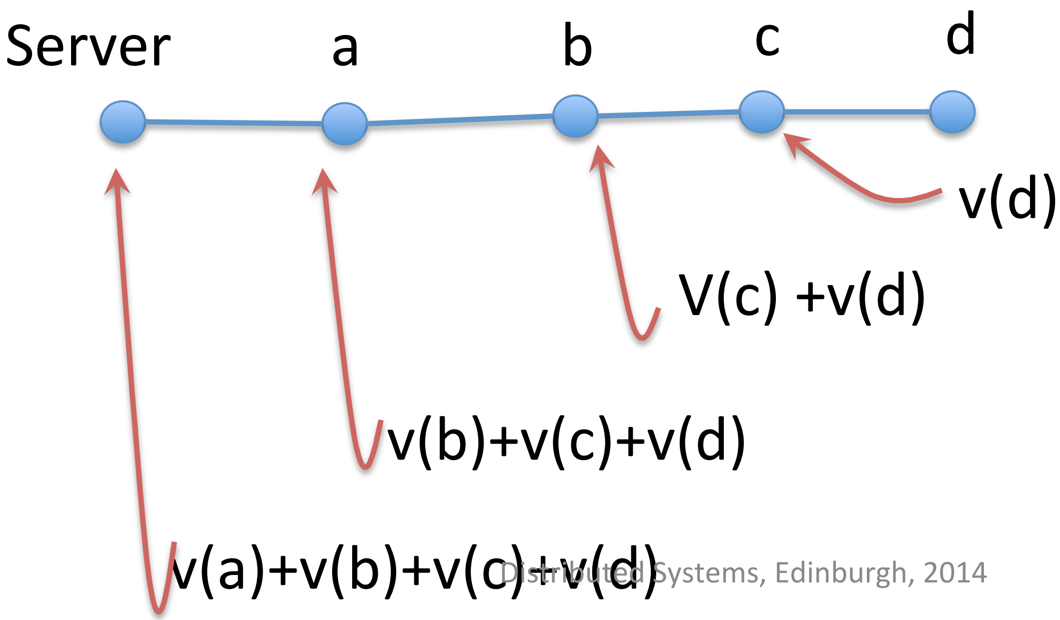
- A simple distributed computation:
  - Each node has stored a numeric value
  - Compute the total of the numbers




Can you find a better, more efficient way?

# Example 2

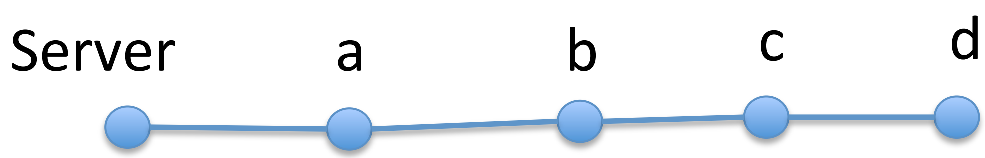
- A simple distributed computation:
  - Each node has stored a numeric value
  - Compute the total of the numbers




Cost: 4 messages

# Example 2

- A simple distributed computation:
  - Each node has stored a numeric value
  - Compute the total of the numbers




More generally, if there were n nodes, this would cost n messages

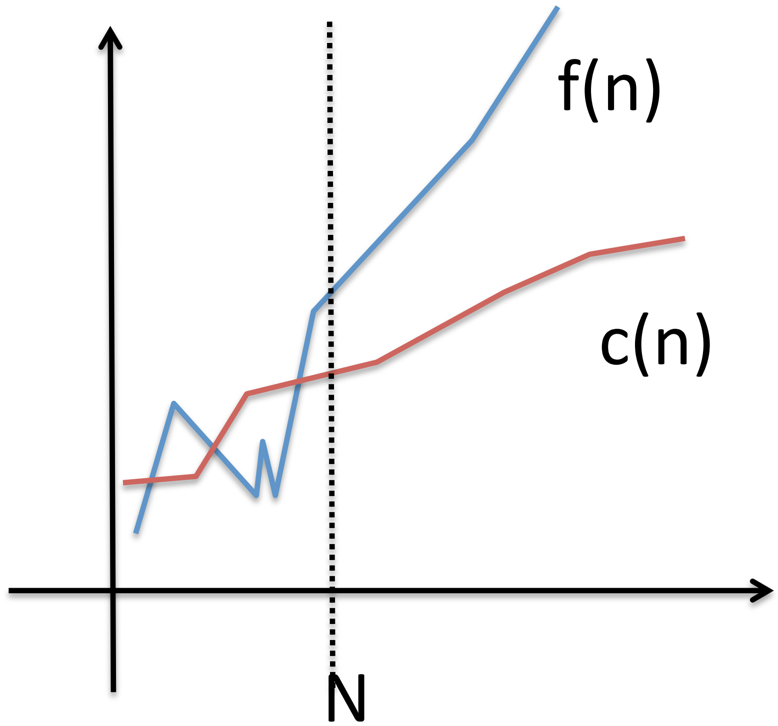
# Communication complexity

- Used to represent communication cost for general scenarios
- Called Communication Complexity or Asymptotic communication complexity

- Use big oh notation: $O$

# Big oh notation

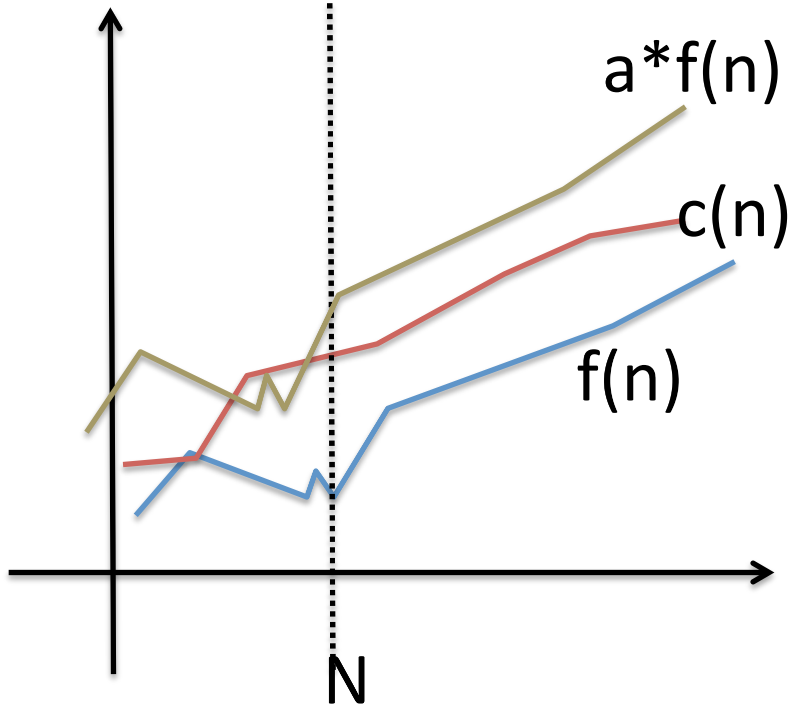
- For a system of n nodes,
- Communication complexity c(n) is O(f(n)) means:
  - There are constants a and N, such that:
  - For n>N: c(n) < a*f(n)




Allowing some initial irregularity, ‘f(n)’ can be seen as a bound on ‘c(n)’

# Big oh – upper bounds

- For a system of n nodes,
- Communication complexity c(n) is O(f(n)) means:
  - There are constants a and N, such that:
  - For n>N: c(n) < a*f(n)




Allowing some initial irregularity, ‘c(n)’ is not bigger than a constant times ‘f(n)’

In the long run, c(n) does not grow faster than f(n)

# Examples

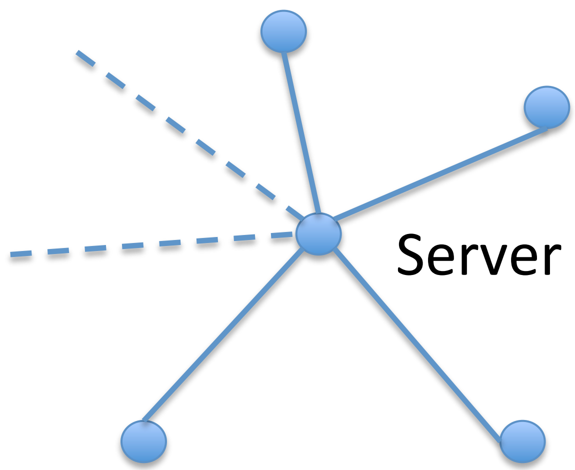
- $3n = O(?)$
- $n^2/5 = O(?)$
- $10\log n = O(?)$
- $2n^3+n+200 = O(?)$
- $15 = O(?)$

# Examples

- $3n = O(n)$
- $n^2/5 = O(n^2)$
- $10\log n = O(\log n)$
- $2n^3+n+200 = O(n^3)$
- 15 or any other constant= $O(1)$

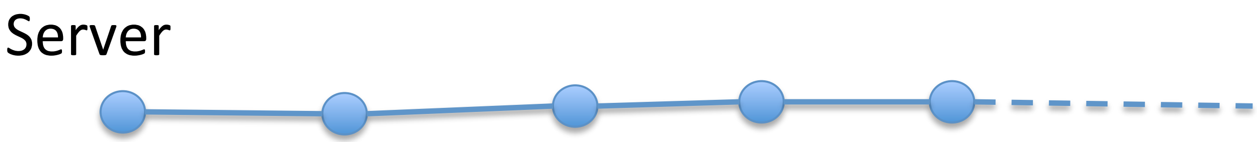
# Example 1

- ‘Star’ network
- Computing sum of all values
- Communication complexity: O(n)



# Example 2a

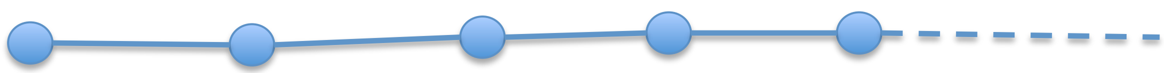
- ‘Chain’ topology network
- Simple protocol where everyone sends value to server
- Communication complexity: $1+2+\ldots+n = O(n^2)$



# Example 2b

- ‘Chain’ network
- Protocol where each node waits for sum of previous values and sends
- Communication complexity: 1+1+…+1 = O(n)

Server

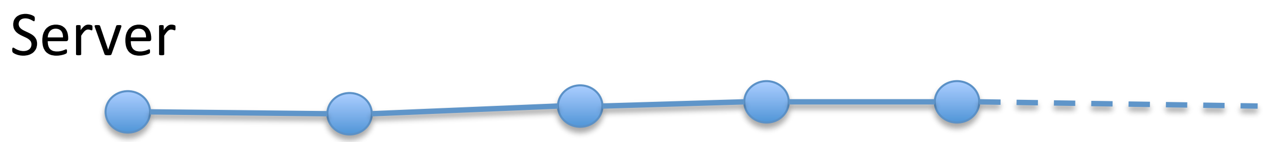


# Time complexity

- How much time does the computation take?
- Assume each transmission takes 1 unit time

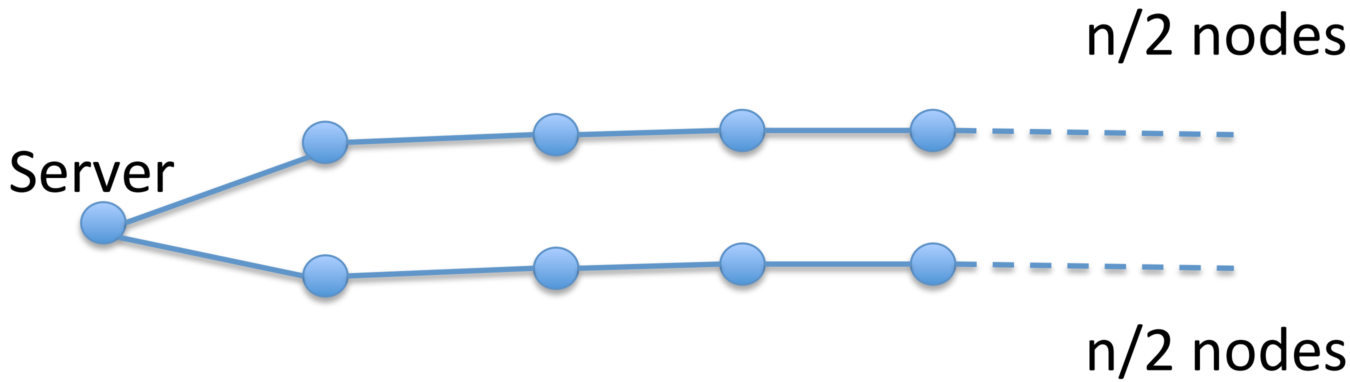
# Example 2b

- ‘Chain’ topology network
- Protocol where each node waits for sum of previous values and sends
- Time complexity: 1+1+…+1 = O(n)



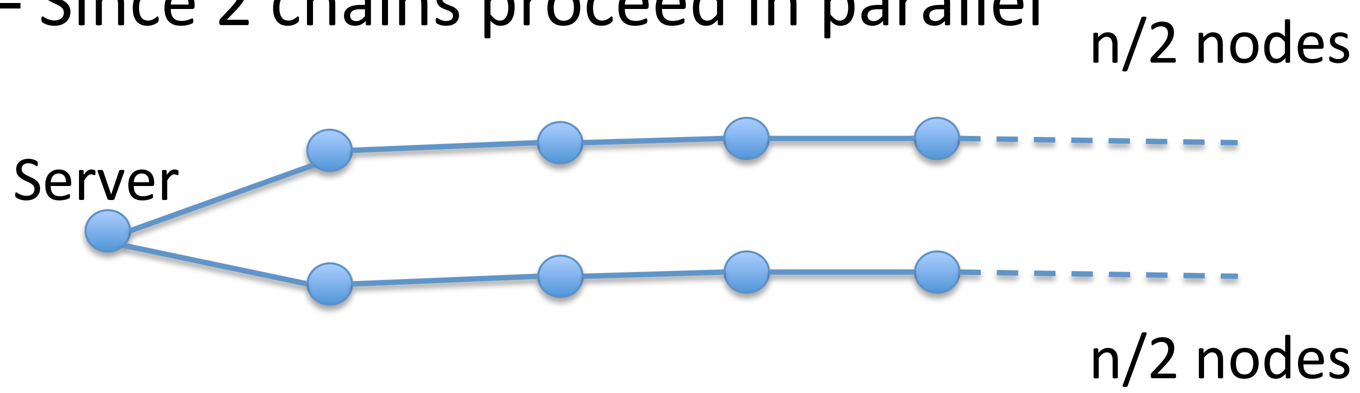
# Example 3

- 'Chain' network
- Protocol where each node waits for sum of previous values and sends
- Communication complexity: 1+1+…+1 = O(n)



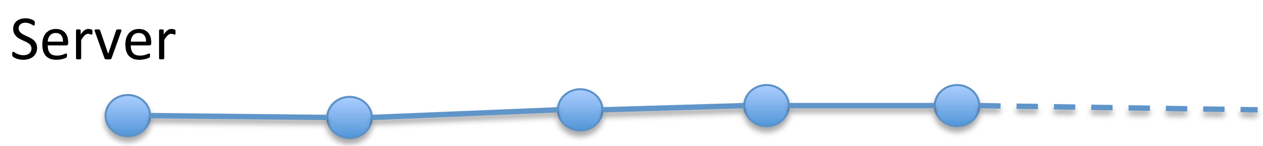
# Example 3

- 2 Chains network
- Protocol where each node waits for sum of previous values and sends
- Communication complexity: 1+1+…+1 = O(n)
- Time complexity: n/2 = O(n)
  - Since 2 chains proceed in parallel



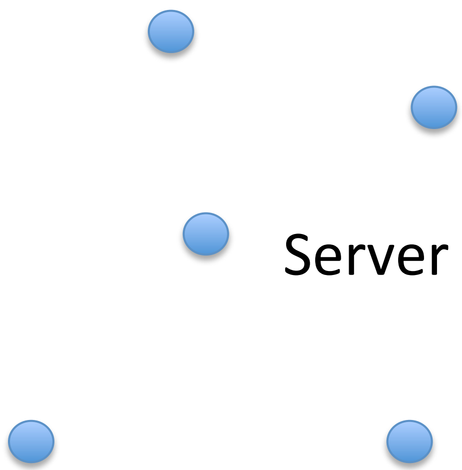
# Example 4

- What if the server has to send a message to all nodes?
  - Star : O(n)
  - Chain (naive) : O($n^2$)
    - Route to each node
  - Chain (smarter) : O(n)
    - Each node sends to its neighbor



# Example 4

- What if the server has to send a message to all nodes?
  - And the communication is broadcast?
  - O(1)

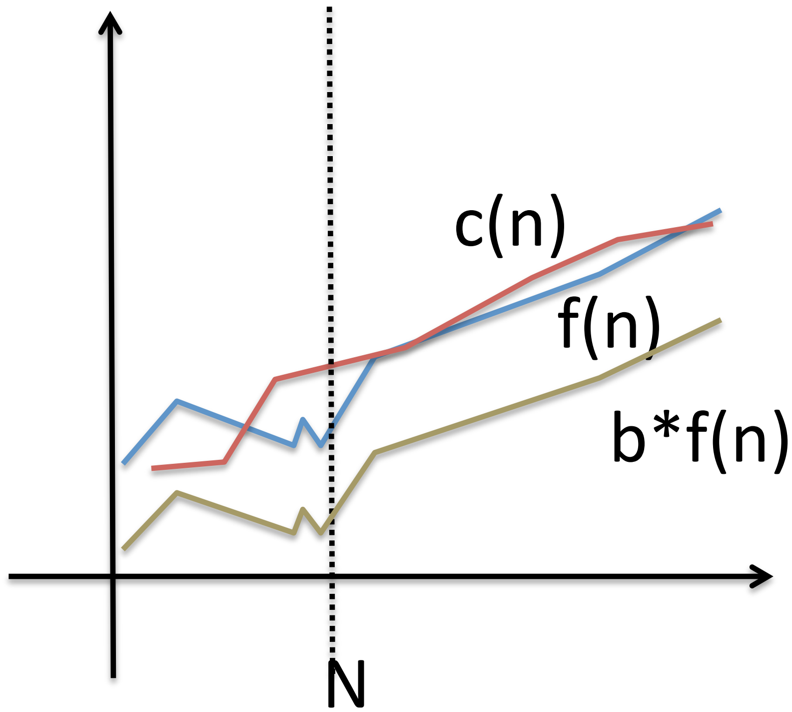


# Observation

- Suppose c(n)=n
  - Then c(n) is O(n) and also O($n^2$)
  - Although, when we ask for the complexity, we are looking for the tightest possible bound, which is O(n)

# Big Ω – lower bounds

- For a system of n nodes,
- Communication complexity c(n) is Ω(f(n)) means:
  - There are constants a and N, such that:
  - For n>N: b*f(n) < c(n)

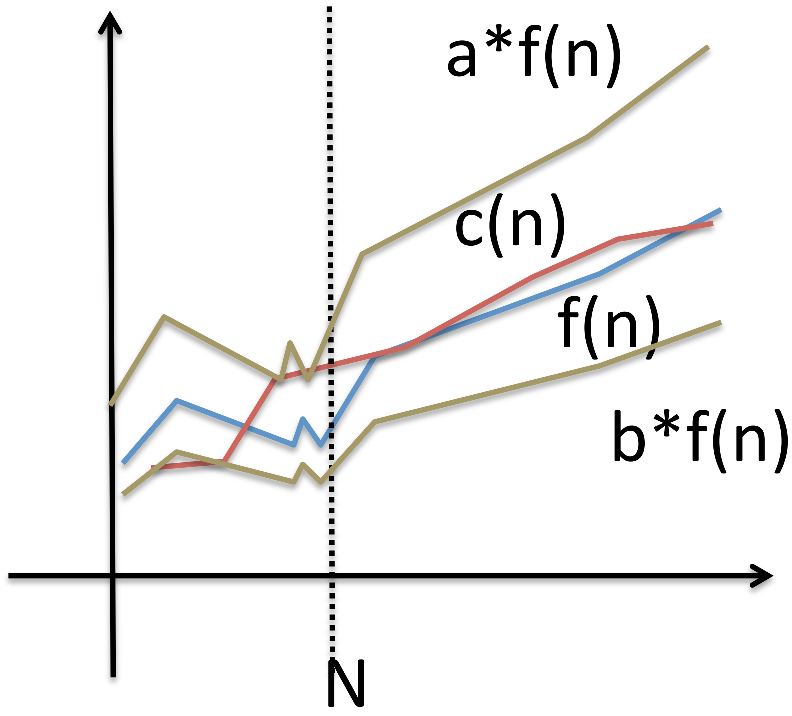



Allowing some initial irregularity, 'c(n)' is not smaller than a constant times 'f(n)'

In the long run, f(n) does not grow faster than c(n)

# Big θ – tight bounds: both O and Ω

- For a system of n nodes,
- Communication complexity c(n) is θ(f(n)) means:
  - There are constants a,b and N, such that:
  - For n>N: b*f(n)<c(n)<a*f(n)




Allowing some initial irregularity, c(n) and f(n) are Within constant factors of each other.
In the long run, c(n) grows at same rate as f(n), upto constant factors.

# In our examples

- Star network:
  - Complexity is $\Theta(n)$ (both $O(n)$ and $\Omega(n)$)
  - It does not take any more than a constant times n messages, it also does not take any less!
- Chain network:
  - Complexity $\Theta(n)$ or $\Theta(n^2)$ depending on algorithm