UNIVERSITY OF
Waterloo

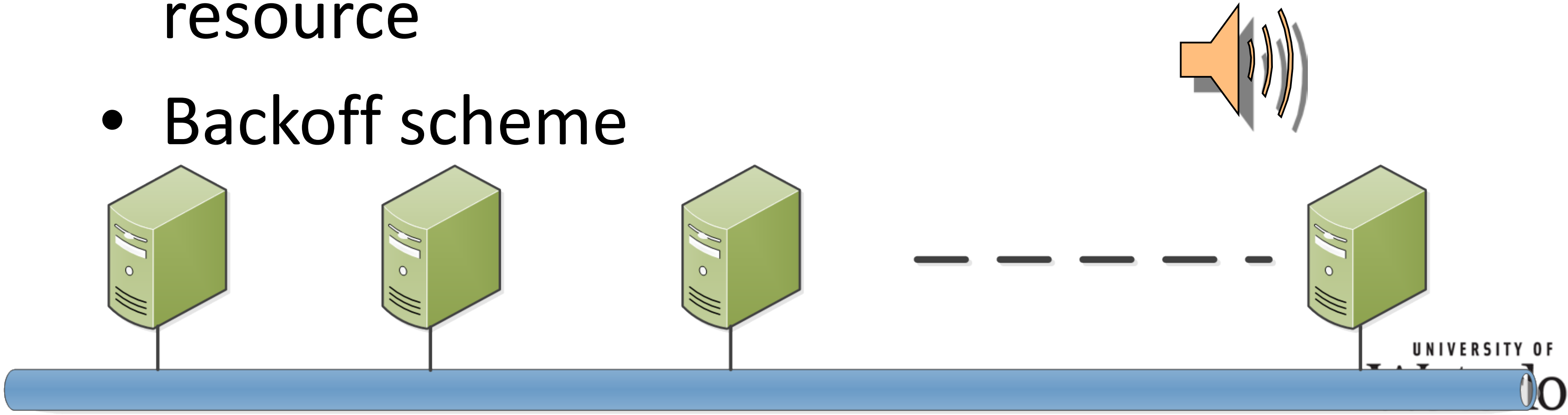
# ECE 358-Tutorial

## Project 2:

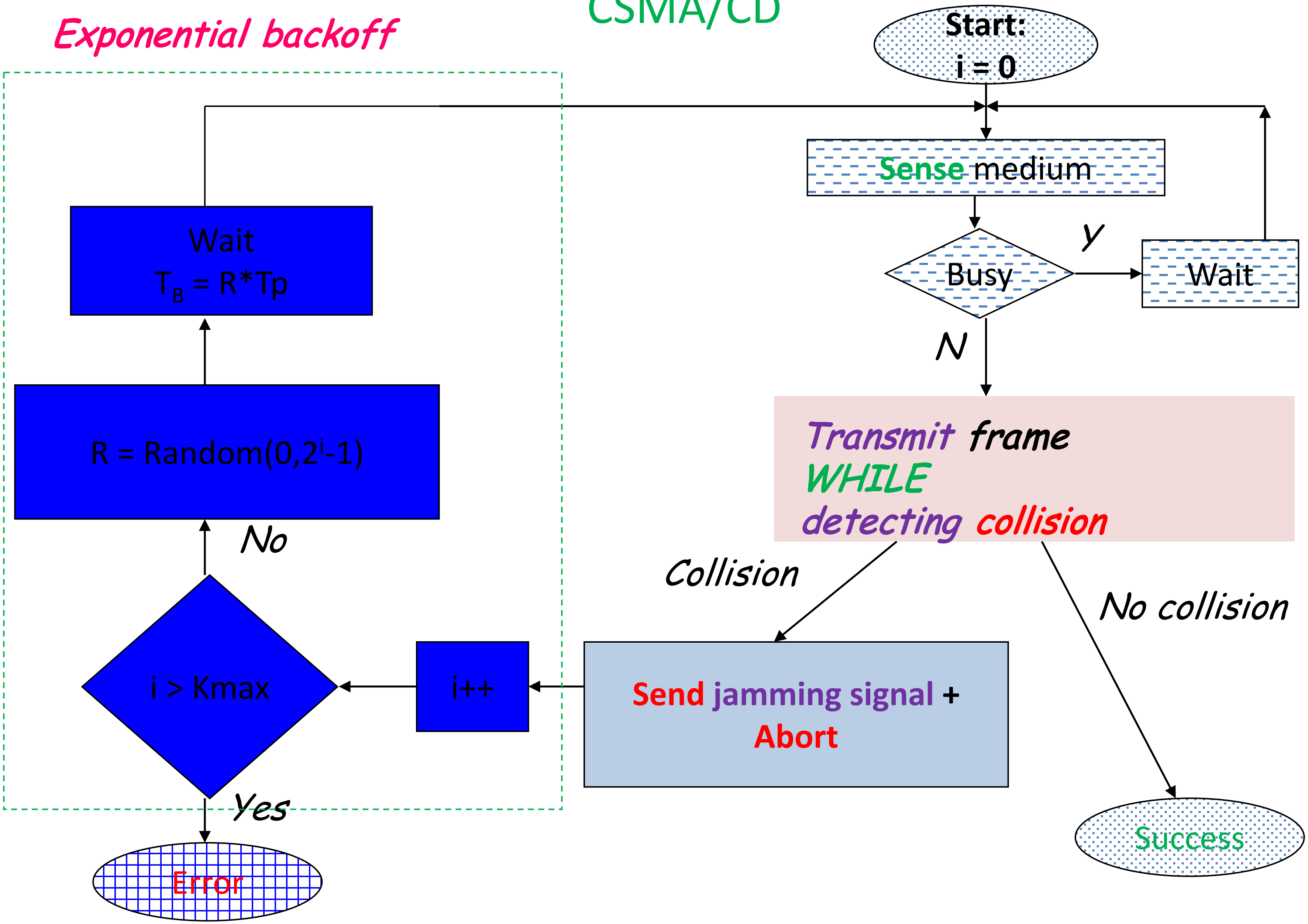
## CSMA/CD Simulation

# Background – CSMA/CD

- MAC protocol used in wired networks – Ethernet
- Sensing medium before transmitting
- Can detect ongoing collision – Reduce wasting resource
- Backoff scheme



# CSMA/CD


Exponential backoff
Start:
i = 0
Sense medium
Busy
Y
Wait
N
Transmit frame
WHILE
detecting collision
Collision
No collision
Send jamming signal +
Abort
Success
i++
i > Kmax
No
Yes
Error
R = Random(0,$2^i$-1)
Wait
$T_B$ = R*Tp

# CSMA/CD

- Medium sensing is done for **96** bit-times.

- Jamming signal length is **48** bits. Jamming signal creates enough energy on the medium for collision detection.

- Tp is equated with 512 bit-times.

- “i” saturates at 10.

# Simulation Overview (1)

- View each node as an individual queue (similar to M/D/1 queue)
- Use knowledge and/or some codes of proj. 1.
- Packet arrival at each queue is Poisson process.
  - See this in proj. 1.
- Packet service time is deterministic (L/W)
- One transmission can be heard by all others
- Performance evaluation

# Simulation Overview (2)

- Differences from proj.1:
  - Implement multiple queues (class-objects)
  - Implement collision detection
  - Packet service may cease before done (due to collision detected)
  - Implement backoff scheme

# Parameters (1)

- N: Number of stations. N is variable; increase it until the throughput performance shows stable curve ->$N_{max}$.
- A: Data packets arrival rate. A is the parameter of an exponential distribution (See proj. 1).
  Reference value: 10/20/50 packets/sec.
- W: Speed of the LAN.
  Reference value: 10/100 Mbps.
- L: Packet length.
  Reference value: 1500 bytes.

# Parameters (2)

- S: Signal propagation speed in the medium. $S=2*10^8$ m/s.
- Tp: 512bits/W.
- Waiting time: Used when medium is sensed busy. Random chosen from (0,Tp).
- Maximum retry count: $i_{max}$ =10
- Tick duration: Should be small. Suggest 10 ns or shorter.

# Assumptions (1)

- Assumptions may make your simulation easier, however, more inaccurate. So, make reasonable assumptions!
- Your simulation strongly depends on your assumptions!
- Make any assumption you need. Here are some :
  - The medium is error-free. The received packets are without any error.
  - One packet has one destination (randomly chosen or preset). However, it can be heard by all stations in the network.

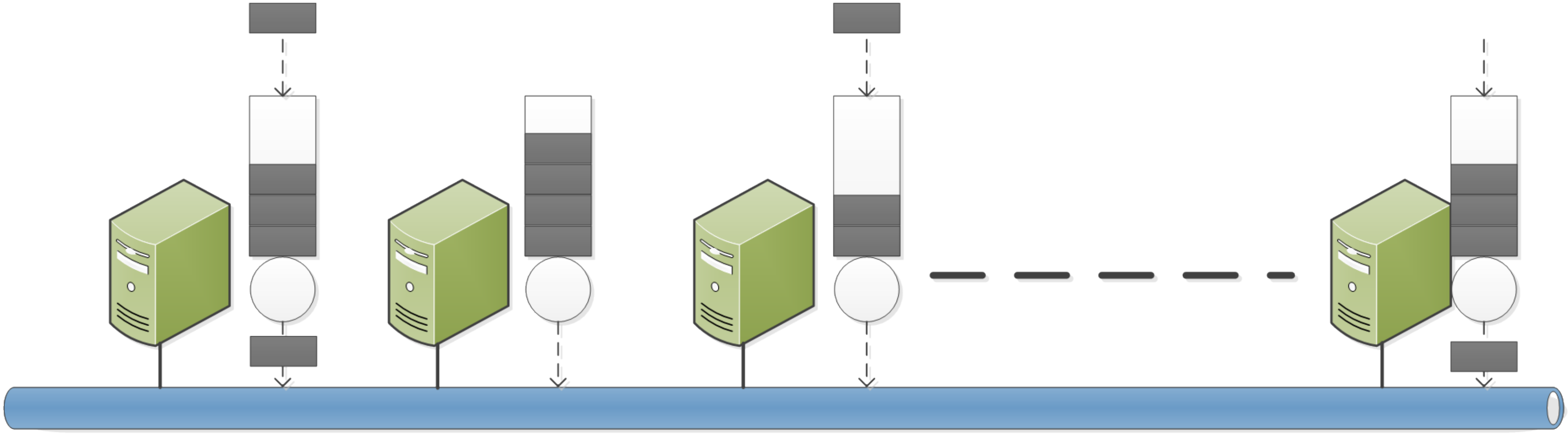
# Assumptions (2)

- What is the topology you use? (Bus/Ring/Star/Tree)
- What is the distance between two neighbors? Are all the distances identical?
- We can assume an instantaneous medium sensing . It may cost only one tick.
- Is the number of queue buffer finite/infinite? And if finite, what is it?
- Any other assumptions.

# Simulation Process

```
/* Simulation Process*/
/* Discrete Event simulation*/
for (int i = 0; i < TICKS; i++)
{
        /* In each tick, you should do something to each node*/
        /* Note that in a tick, different operations may be done to
           different nodes, depending on their states, e.g.,
           transmitting, idle, backoff and so on. */
        /* Carefully design your operations.
           Better draw a  flowchart first. */
}
```

# Details (1)



Simulate stations as multiple queues!

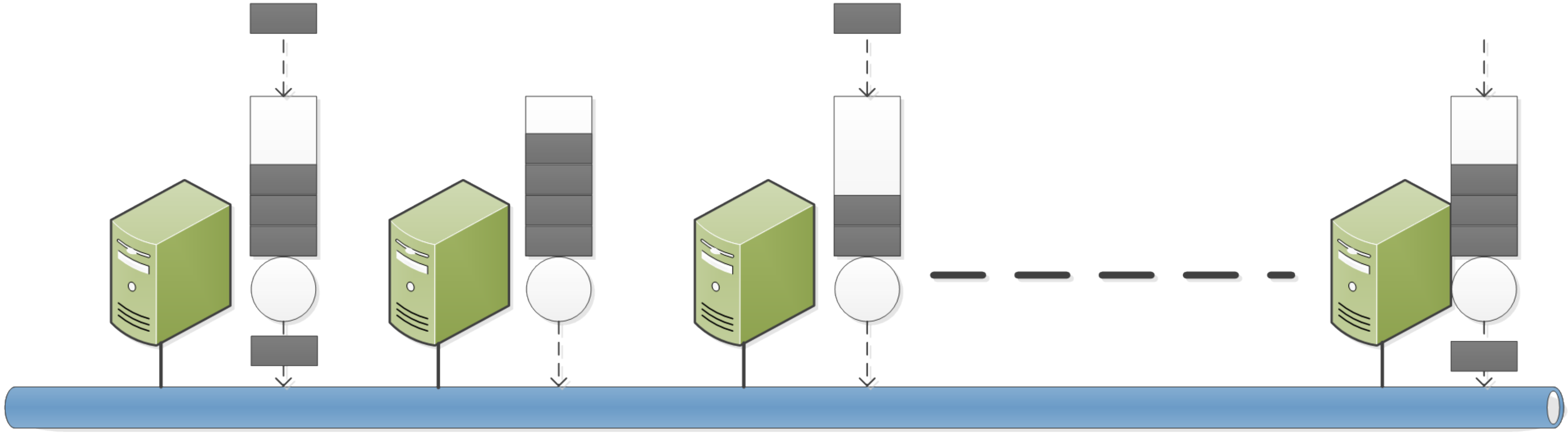
# Details (2)

- Packet arrival
  - Inter-arrival time: exponential distribution (A)
  - enqueue(): add a packet into queue. If the queue buffer is finite (M/G/1/K), you should check whether the queue is full in the current tick.

# Details (3)

- Collision detection
  - See the details in lecture notes.
  - Closely related to the propagation delay (distance between nodes).
  - A and B sense the medium as idle at the same time and start transmission. When A's signal propagates to B, and B has not finish transmission yet, a collision is detected at B. Same at A.

# How is collision detected?



Propatation delay $i$ ticks Propatation delay $i$ ticks

A transmits at
$[n, n+m]$ ticks

If B transmits at
$[n+i, n+m+i]$ ticks,
B detects a collision

When collision detected, cease current transmitting and backoff/discard the packet

# Details (4)

- Exponential backoff scheme
  - See the details in lecture notes.
  - Backoff time $T_B$= Tp*R.
  - R: An integer randomly chosen from $[0,2^i-1]$.
  - Each backoff, i++. If i>10, the packet is discarded.
  - When a packet is successfully transmitted or discarded, i<-0.

# Performance Evaluation

- Throughput
  - In your simulation, count the number of packets successfully received -> M.

$$\text{Throughput} = \frac{M \cdot L}{\text{Total simulation time}}$$

- Delay
  - ‘Different loads’ means different N.

$$\text{Average delay} = \frac{\sum_i^M \text{delay of } i^{th} \text{ packet}}{M}$$