ECS 132 HW 1

Rajat Mahesh Gupta

Question 2

Consider the ALOHA example, but now assume that initially, both nodes are active (X0 = 2)

a) What is the probability that X1 = 2?
b) What is the probability that X2=2 given X1=2. P(X2=2|X1=2)
c) What is the probability that X2=2 given X1=0. P(X2=2|X1=0)
d) Suppose it is known that X1 = X2. Find the probability that there were 0, 1 or 2 collisions during the first two epochs

```
set.seed(123)
# Function to simulate probabilities in the ALOHA example
sim <- function(p, q, nreps) {
  countx2eq2 <- 0
  countx1eq1 <- 0
  countx1eq2 <- 0
  countx1eq0 <- 0
  countx2eq2givx1eq2 <- 0
  countx2eq2givx1eq1 <- 0
  countx2eq2givx1eq0 <- 0
  count_coll_0 <- 0
  count_coll_1 <- 0
  count_coll_2 <- 0

  # Simulate nreps repetitions of the experiment
  for (i in 1:nreps) {
    numsend <- 0 # no messages sent so far

    # Simulate A and B's decision on whether to send in epoch 1
    for (j in 1:2) {
      # First simulate if the node is active and generates a message
      if (runif(1) < q) {

        # if it was initially inactive and now generates and sends a message
        if (runif(1) < p) {
          numsend <- numsend + 1
        }

      }
    }

    # Determine the state at the end of epoch 1
    if (numsend == 0) {
```

```
    X1 <- 0 # both nodes are inactive
  } else if (numsend == 1) {
    X1 <- 1 # one node is active
  } else {
    X1 <- 2 # both nodes are active
  }

  if (X1 == 2) {
    countx1eq2 <- countx1eq2 + 1
  } else if (X1 == 0) {
    countx1eq0 <- countx1eq0 + 1
  } else if (X1 == 1) {
    countx1eq1 <- countx1eq1 + 1
  }

  # Now simulate epoch 2
  numactive <- X1

  # If the node is inactive, it may become active with probability q
  if (numactive == 1) {
    if (runif(1) < q) {
      numactive <- numactive + 1
    }
  }

  if (X1 == 1 && runif(1) < q) {
    numactive <- numactive + 1 # inactive node generates a new message
  }

  # Determine the state at the end of epoch 2
  if (numactive == 1) {
    if (runif(1) < p) {
      X2 <- 0 # the active node sends successfully
    } else {
      X2 <- 1 # the active node remains active
    }
  } else { # numactive = 2
    numsend <- 0
    for (k in 1:2) {
      if (runif(1) < p) {
        numsend <- numsend + 1
      }
    }
    if (numsend == 1) {
      X2 <- 1 # one node sends successfully
    } else {
      X2 <- 2 # both nodes are active
    }
  }

  # Tally results
  if (X1 == 2) {
    countx2eq2 <- countx2eq2 + 1
```

```
    }

    if (X1 == 2) {
      if (X2 == 2) {
        countx2eq2givx1eq2 <- countx2eq2givx1eq2 + 1
      }
    } else if (X1 == 0) {
      if (X2 == 2) {
        countx2eq2givx1eq0 <- countx2eq2givx1eq0 + 1
      }
    } else if (X1 == 1) {
      if (X2 == 2) {
        countx2eq2givx1eq1 <- countx2eq2givx1eq1 + 1
      }
    }

    if (X1 == 0) {
      if (X2 == 2) {
        countx2eq2givx1eq0 <- countx2eq2givx1eq0 + 1
      }
    }

    # Check collisions during the first two epochs (X0 = 2)
    if (X1 == X2) {
      if (X1 == 0) {
        count_coll_0 <- count_coll_0 + 1
      } else if (X1 == 1) {
        count_coll_1 <- count_coll_1 + 1
      } else if (X1 == 2) {
        count_coll_2 <- count_coll_2 + 1
      }
    }
  }

  # Print results
  cat("P(X1 = 2):", countx1eq2 / nreps, "\n")
  # cat("P(X2 = 2):", countx2eq2 / nreps, "\n")
  cat("P(X2 = 2 | X1 = 2):", countx2eq2givx1eq2 / countx1eq2, "\n")
  cat("P(X2 = 2 | X1 = 0):", countx2eq2givx1eq0 / countx1eq0, "\n")
  cat("P(C1=C2=0 | X1=X2):", count_coll_0/nreps, "\n")
  cat("P(C1=C2=1 | X1=X2):", count_coll_1/nreps, "\n")
  cat("P(C1=C2=2 | X1=X2):", count_coll_2/nreps, "\n")
}

# Example usage
nreps <- 100000 # Number of repetitions
p <- 0.4      # Probability that an active node sends
q <- 0.8     # Probability that an inactive node generates a new message

# Call the simulation function
sim(p, q, nreps)
```

```
## P(X1 = 2): 0.10381
```

```
## P(X2 = 2 | X1 = 2): 0.5170022
## P(X2 = 2 | X1 = 0): 1.038958
## P(C1=C2=0 | X1=X2): 0
## P(C1=C2=1 | X1=X2): 0.21212
## P(C1=C2=2 | X1=X2): 0.05367
```