

Lampiran A

Lampiran A :

Model Matematika Gilbert dan Logaritma pada bahasa pemrograman MATLAB

Implementation of the Gilbert Model

```
function PYkmod = GilbertModel(lossPattern)

p01 = sum(lossPattern.BurstLengths)/length(lossPattern.Pattern);

s = 0;
for i = 1:length(lossPattern.BurstLengths)
    s = s + (i-1)*lossPattern.BurstLengths(i);
end

nrLostPackets = 0;
for i = 1:length(lossPattern.BurstLengths)
    nrLostPackets = nrLostPackets + (i)*lossPattern.BurstLengths(i);
end

p11 = s/(nrLostPackets - 1);

p10 = 1 - p11;

P1 = p01/(p01 + p10);
P0 = p10/(p01 + p10);

PYkmod = zeros(1,length(lossPattern.BurstLengths));

for i = 1:length(lossPattern.BurstLengths)
    PYkmod(i) = (p11^(i-1))*p10;
end

end
```

Implementation of the Logarithmic Model

```
function PYkmod = LogModel(lossPattern)

nrLostPackets = 0;
for i = 1:length(lossPattern.BurstLengths)
    nrLostPackets = nrLostPackets + (i)*lossPattern.BurstLengths(i);
end

mean = nrLostPackets/sum(lossPattern.BurstLengths);

teta = fzero(@(x) myfun(x,mean), 0.5);

PYkmod = zeros(1,length(lossPattern.BurstLengths));

for i = 1:length(lossPattern.BurstLengths)
    PYkmod(i) = ((-1/log(1-teta))*teta^(i))/i;
end

end
```

where *myfun* is a Matlab function containing the desired mathematical function:

```
function f = myfun(x,c)

f = ((-1/log(1-x))*x)/(1-x) - c;

end
```

Lampiran B

Lampiran B :

Implementasi Pola Paket Data Model Matematika Gilbert dan Logaritma

Implementation of the Pattern from the Gilbert Model

```
function GilbertPattern = createGilbPattern(lossPattern)

gilbertPattern = zeros(1,length(lossPattern.Pattern));

p01 = sum(lossPattern.BurstLengths)/length(lossPattern.Pattern);

s = 0;
for i = 1:length(lossPattern.BurstLengths)
    s = s + (i-1)*lossPattern.BurstLengths(i);
end

nrLostPackets = 0;
for i = 1:length(lossPattern.BurstLengths)
    nrLostPackets = nrLostPackets + (i)*lossPattern.BurstLengths(i);
end

p11 = s/(nrLostPackets - 1);

p10 = 1 - p11;

gilbertPattern(1) = 0;           %we assume the first packet has been
    succesfully                 received

for i = 2:length(gilbertPattern)
    r = rand(1);
    if (gilbertPattern(i-1) == 0)
        if (r <= p01)
            gilbertPattern(i) = 1;
        else
            gilbertPattern(i) = 0;
        end
    end
    if (gilbertPattern(i-1) == 1)
        if (r <= p10)
            gilbertPattern(i) = 0;
        else
            gilbertPattern(i) = 1;
        end
    end
end
end
```

```
GilbertPattern = testPattern(gilbertPattern)
```

```
End
```

Implementation of the Pattern from the Logarithmic Model

```
function LogPattern = createLogPattern(lossPattern)
```

```
logPattern = zeros(1,length(lossPattern.Pattern));
```

```
PYkmod = LogModel(lossPattern);
```

```
PYkmodRel = PYkmod * (lossPattern.PercentLosses/100);
```

```
okmod = PYkmod*sum(lossPattern.BurstLengths);
```

```
logPattern(1) = 0;           %we assume the first packet has been successfully  
                           received
```

```
for i = 2:length(logPattern)
```

```
    if (sum(logPattern(1:i)) >= lossPattern.NrLostPackets)  
        break
```

```
    end
```

```
    if (logPattern(i) == 0)
```

```
        if (logPattern(i-1) == 1)
```

```
            logPattern(i) = 0;
```

```
        else
```

```
            r1 = rand(1);
```

```
            if (r1 > (lossPattern.PercentLosses/100))
```

```
                logPattern(i) = 0;
```

```
            else
```

```
                if ((r1 > sum(PYkmodRel)) && (r1 <= (lossPattern.PercentLosses/100)))
```

```
                    disp(['Gap probability at packet ' num2str(i)])
```

```
                end
```

```
                if (r1 <= PYkmodRel(1))
```

```
                    logPattern(i) = 1;
```

```
                else
```

```
                    for j = 2:length(PYkmodRel)
```

```
                        if ((r1 > sum(PYkmodRel(1:(j-1)))) && (r1 <= sum(PYkmodRel(1:j))))
```

```
                            if (j > (length(logPattern) - i + 1))
```

```
                                logPattern(i:length(logPattern)) = 1;
```

```
                            else
```

```
                                logPattern(i:(i+(j-1))) = 1;
```

```
                            end
```

```
                        end
```

```
                    end
```

```
                end
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
LogPattern = testPattern(logPattern)
```

```
end
```