CS 525M – Mobile and Ubiquitous Computing Seminar

IEEE 802.11 MAC Protocol

Fragmentation or Error Correction?

By Choong-Soo Lee

Introduction

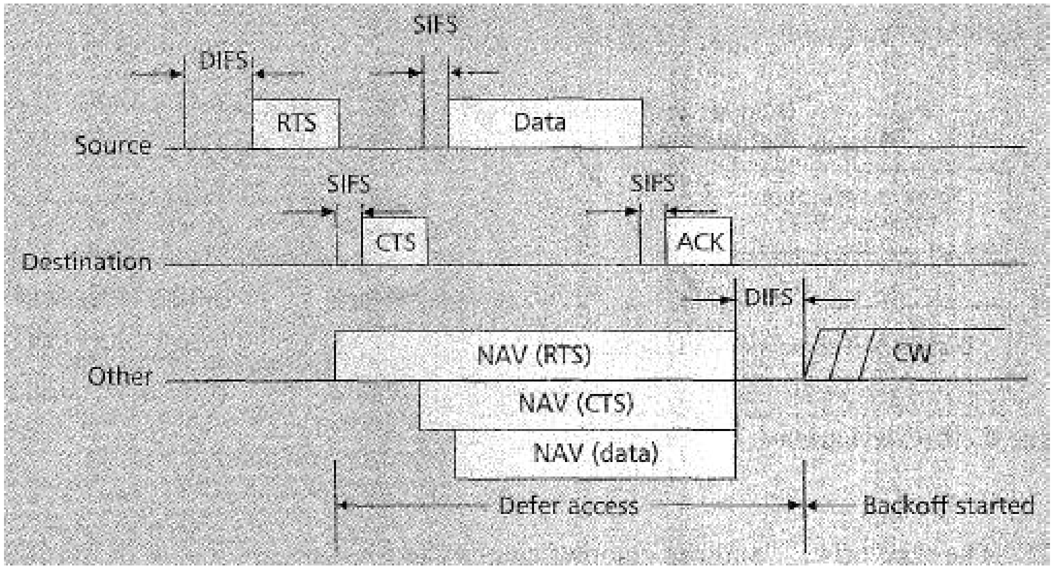
- IEEE 802.11 MAC Protocol
  - 4 way handshake (RTS/CTS/DATA/ACK)
  - A reliable MAC layer
    - Supports retransmissions
    - Hides loss information to upper layers such as TCP
  - Side effect
    - Increases delays of end-to-end connections

Outline

- Introduction
- Background
- Mathematical Analysis
- Simulation Validation
- Conclusion

Fragmentation

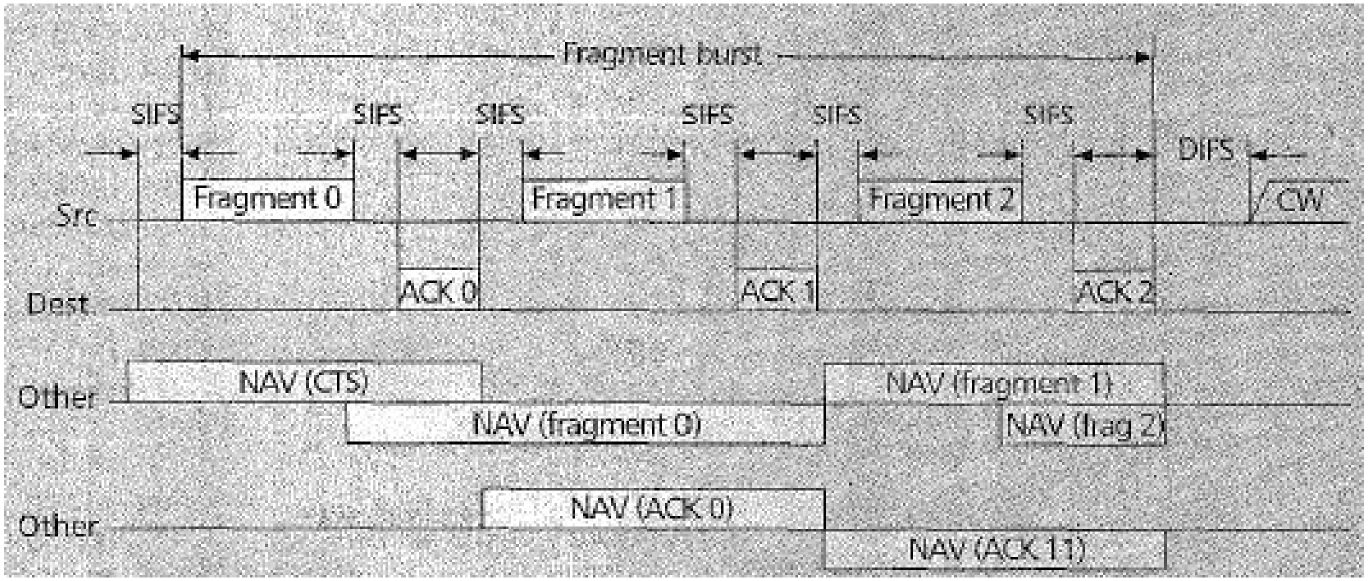
- Regular 4-way handshake
  - RTS / CTS to reserve
  - DATA / ACK to transmit



[Crow97]

Fragmentation

- 4+ way handshake
  - RTS / CTS to reserve
  - Multiple DATA / ACK to transmit



[Crow97]

Fragmentation

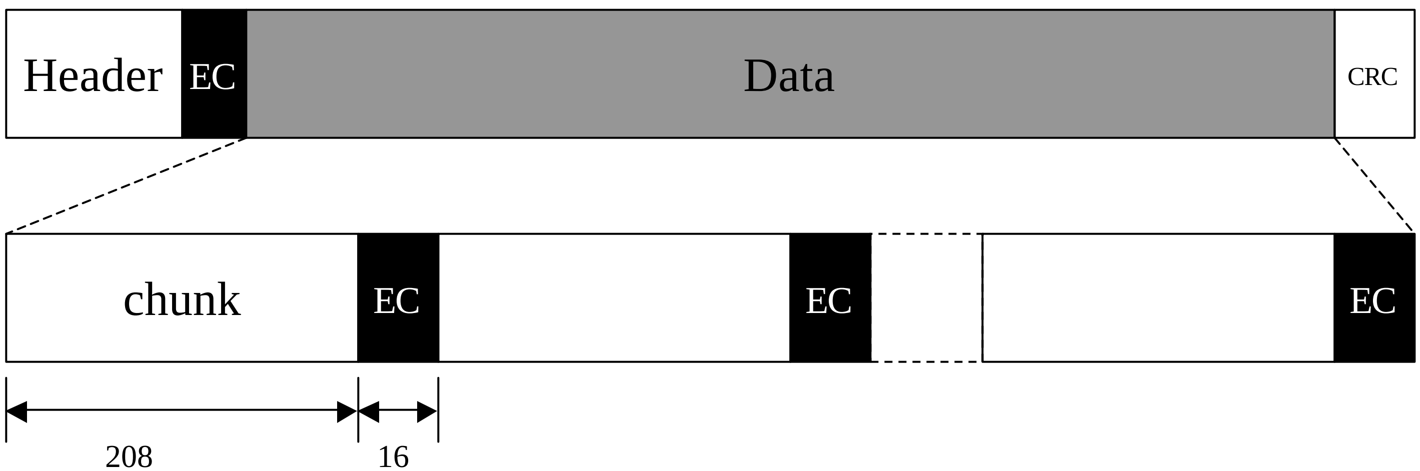
Low Error Rate	High Error Rate
No fragmentation	Fragmentation
There is rarely any need to retransmit frames. Fragmentation only introduces unnecessary overhead.	It is much faster to retransmit small fragments than an entire frame.

Error Correction

- The source adds a certain amount of redundancy to the data before transmitting.
- The destination repairs the data with the redundant information if errors have occurred.
- Reed-Solomon
  - One of the most popular error correction method
  - Redundancy on a symbol basis
  - With n symbols of redundancy, it can fix up to n / 2 symbols of error.

Error Correction

- IEEE 802.11e Group
  - Concerned with real-time traffic
    - Multimedia, online games, etc
  - A proposal to include error correction to MAC frames



Fragmentation

Low Error Rate	High Error Rate
No Error Correction	Error Correction
There is rarely any errors so it is a waste to calculate the necessary redundancy and send the overhead.	Error correction can help prevent retransmissions by repairing the frames.

What then?

- We know each method can help reduce the delay in the MAC layer.
- Questions
  - Can we use them together?
  - If so, what combination gives the best performance?

Outline

- Introduction
- Background
- Mathematical Analysis
- Simulation Validation
- Conclusion

Wireless Errors

- It is known that the wireless errors are usually bursty. There are mathematical models for wireless errors but they are more useful for simulation purposes.
- IID (Independent, Identical Distribution)
  - We assume that there is a fixed bit error rate and that this error rate is identical for each bit and independent of other bits.

Probabilities of Error

- RTS and CTS exchange

$$p_{rts}(0) = (1 - p_b)^{s_{rts}} (1 - p_b)^{s_{rts}}$$

- DATA and ACK exchange
  - No Error Correction

$$p_{data}(0) = (1 - p_b)^{s_f} (1 - p_b)^{s_a}$$

  - Error Correction

$$p_{cb} = \sum_{i=0}^{s_e} \binom{s_c + s_{oh}}{i} (1 - p_{be})^{s_c + s_{oh} - i} \, p_{bc}{}^{i}, \quad p_h = \sum_{i=0}^{4} \binom{\frac{s_h}{8} + 8}{i} (1 - p_{be})^{\frac{s_h}{8} + 8 - i} \, p_{be}{}^{i}$$

$$p_{eb} = 1 - (1 - p_b)^8, \quad p_{data}(0) = p_h \, p_{nc}{}^{n_c}$$

- Common Part : $p_x(i) = (1 - p_x(0))^i \, p_x(0)$

Service Times

- RTS / CTS Exchange

$$t_{rts}(0) = t_{rts} + t_{cts} + 2t_p + t_s$$

$$t_{rts}(i) = i\left(t_{rts} + t_r\right) + t_{rts}(0) + \sum_{k=1}^{i} t_b(k)$$

- DATA / ACK Exchange

$$t_{data}(0) = t_f + t_a + 2t_p + t_s$$

$$t_{data}(i) = i\left(t_f + t_r\right) + t_{data}(0) + \sum_{k=1}^{i} t_b(k)$$

Expected Service Time

$$t_{rts} = \sum_{i=0}^{r} p_{rts}(i) t_{rts}(i), \quad t = t_d + t_{rts} + t_s + t_{data} + t_d + t_b(0)$$

- Fragmentation

$$t_{data} = \sum_{i=0}^{r} p_{data}(i) t_{data}(i)$$

- No Fragmentation

$$t_{data} = \sum_{i(0)=0}^{r} \sum_{i(1)=0}^{r} \cdots \sum_{i(n-1)=0}^{r} \left(p_{data}(i(0)) p_{data}(i(1)) \ldots p_{data}(i(n-1)) \right) \left((n-1) t_s + \sum_{j=0}^{n-1} t_{data}(i(j)) \right)$$

$$t_{data} = n t_f p_s^{\,n-1} + (n-1) t_s p_s^{\,n}$$

- Finally: $t = t_d + t_{rts} + t_s + t_{data} + t_d + t_b(0)$

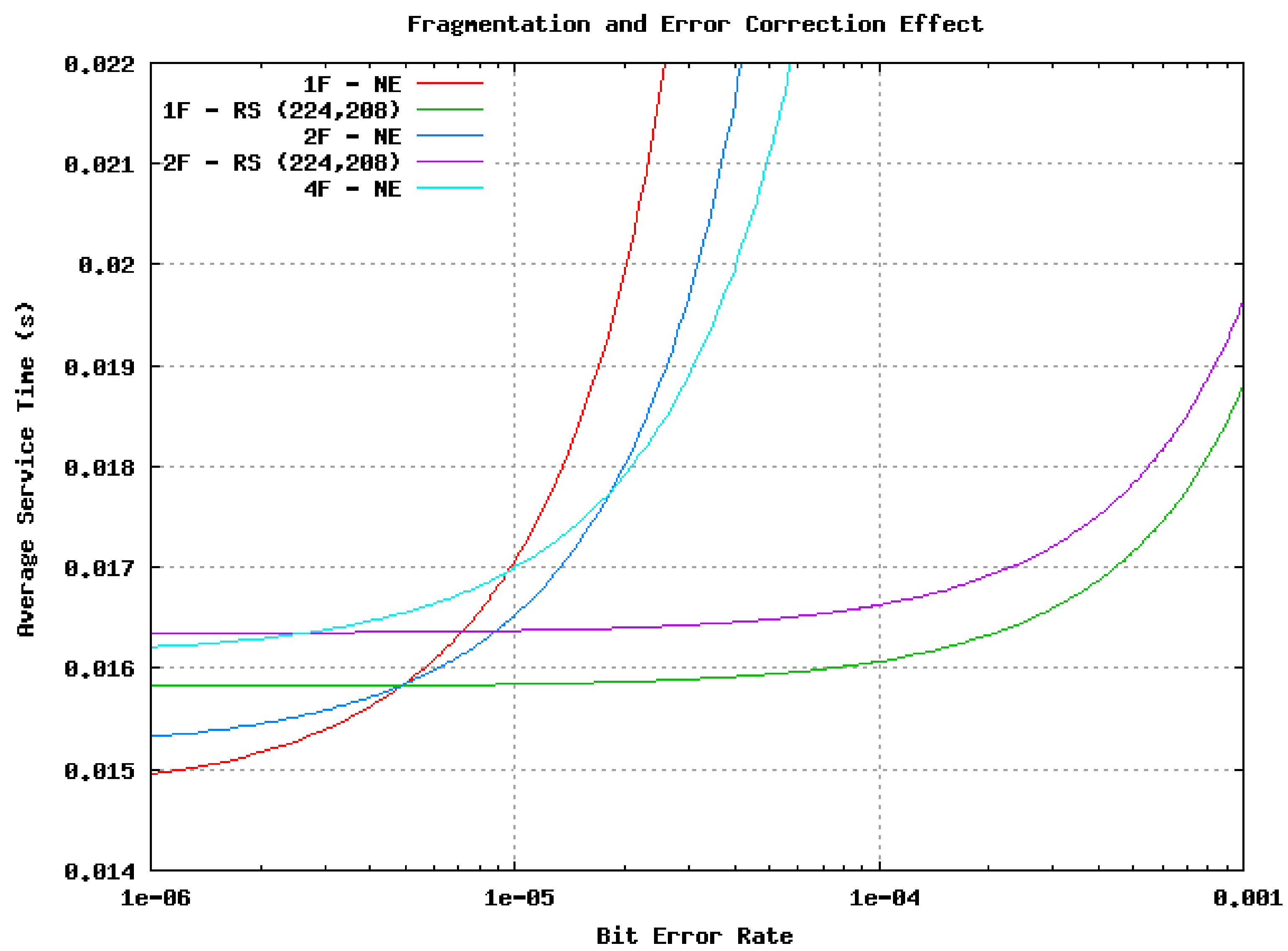
Outline

- Introduction
- Background
- Mathematical Analysis
- Simulation Validation
- Conclusion

Simulation

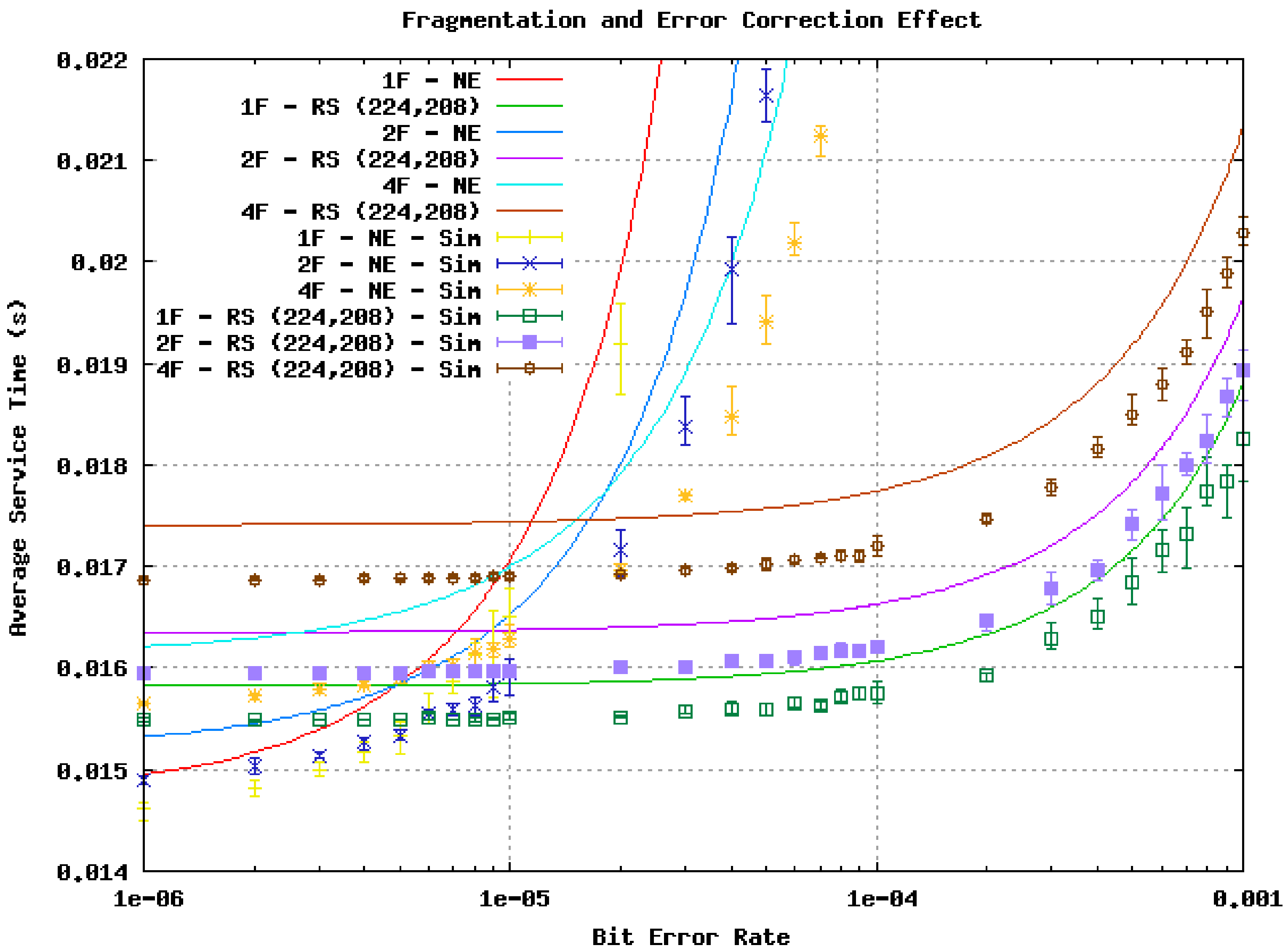
- Introduces the IID Error Module for generating errors
- Written in C++
- Modular in order to add different error modules for future research

Graph



Fragmentation and Error Correction Effect
1F - NE
1F - RS (224,208)
2F - NE
2F - RS (224,208)
4F - NE
Average Service Time (s)
Bit Error Rate
0.022
0.021
0.02
0.019
0.018
0.017
0.016
0.015
0.014
1e-06
1e-05
1e-04
0.001

Graph



Fragmentation and Error Correction Effect
Average Service Time (s)
0.022
0.021
0.02
0.019
0.018
0.017
0.016
0.015
0.014
1e-06
1e-05
1e-04
0.001
Bit Error Rate
1F - NE
1F - RS (224,208)
2F - NE
2F - RS (224,208)
4F - NE
4F - RS (224,208)
1F - NE - Sim
2F - NE - Sim
4F - NE - Sim
1F - RS (224,208) - Sim
2F - RS (224,208) - Sim
4F - RS (224,208) - Sim

Outline

- Introduction
- Background
- Mathematical Analysis
- Simulation Validation
- Conclusion

Conclusion

- Summary
  - Established a mathematical model to calculate the expected service time of a data frame between two nodes
  - Coded a simulator to validate the mathematical model to a certain extent
- Analysis
  - It looks like error correction overhead is reasonable to improve delay performance.

Future Work

- Enhance Mathematical Model
  - Introduce the cost of encoding/decoding error correction overhead.
- More simulation
  - Use a more realistic error model to introduce errors
- Other costs
  - Use power consumption
  - Come up with a metric to balance performance and power consumption