Khongorzul Dashdondov, Yong-Ki Kim, Mi-Hye Kim, 2019. Performance analysis of the stop-and-wait automatic repeat request protocol under Markovian interruptions. *Frontiers of Information Technology & Electronic Engineering*, 20(9):1296-1306. https://doi.org/10.1631/FITEE.1700185

Performance analysis of the stop-and-wait automatic repeat request protocol under Markovian interruptions

Key words: Stop-and-wait ARQ protocol; Markovian interruptions; Poisson distribution; Buffer occupancy; Waiting time

Corresponding author: Mi-Hye KIM
E-mail: mhkim@cbnu.ac.kr
ORCID: http://orcid.org/0000-0001-5859-5471

Introduction

- Performance analysis of an integrated voice/data multiplexer using a stop-and-wait (SW) Automatic Repeat reQuest (ARQ) protocol is provided.
- Stop-and-wait (SW) ARQ: The transmitter ***sends a data packet and waits for the corresponding ACK/NACK*** before starting the next transmission.
- ARQ retransmission protocol is used for data packets
- Markov interruption is used for voice packet.
  - Voice:
    - Real time
    - Higher priority
    - On/OFF Markov process
  - Data:
    - Packet switching method
    - Non real-time
    - Lower priority (if no voice, then data transmission)

Main idea

- Packet (i) is in transmission if no blocking by the voice.
- Packet (i) is also copied into auxiliary buffer Qy:
  - If ACK signal arrives, Packet (i) is discarded.
  - If NACK signal arrives, Packet (i) is put into input buffer and retransmitted.

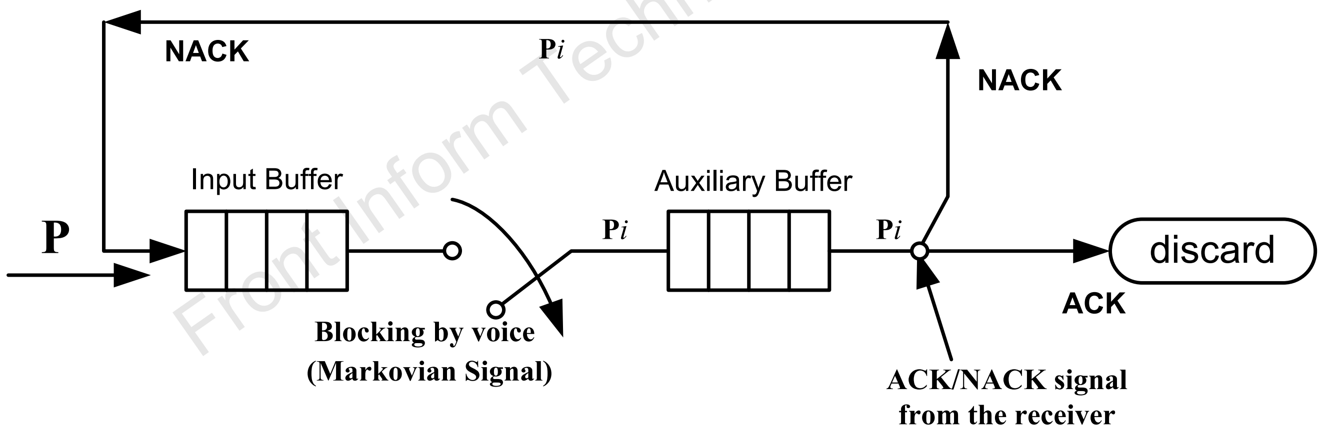

Fig. 1 System model

Main idea (Cont’d)

- The transmitter should check the ACK feedback signal before sending the next packet at the next available slot time.
- However, in the case of transmission error, the transmitter receives the NACK signal from the receiver and retransmits the current packet again at the next frame time.
- When buffer is empty, the transmitter may check the buffer. If the buffer is not empty, the transmitter may send out the packet at the next time slot.

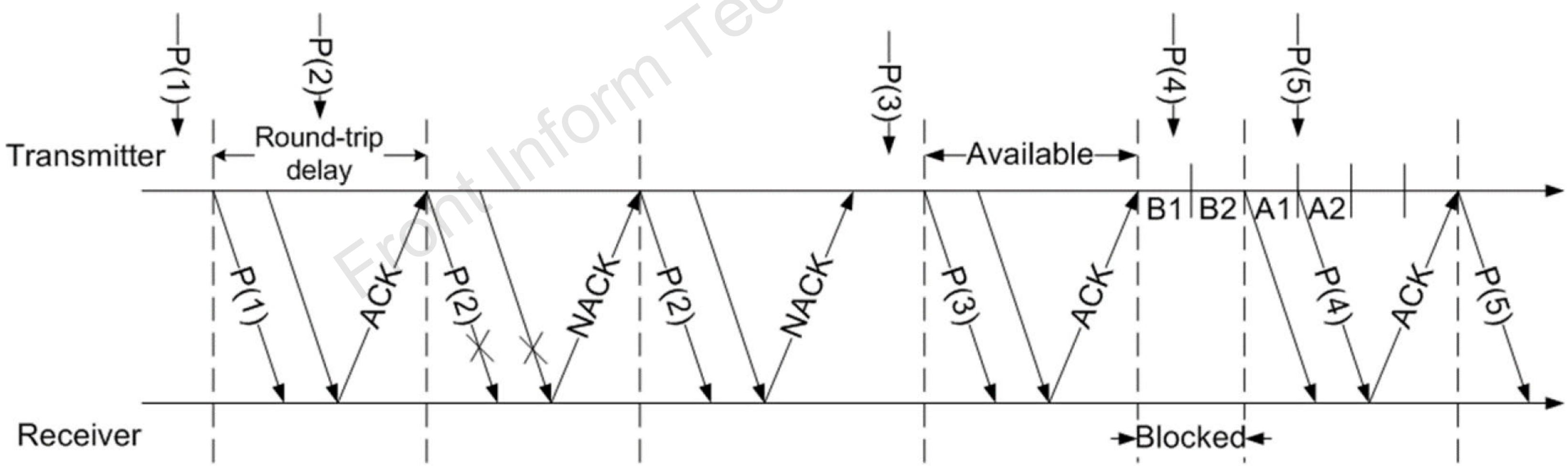

Fig. 2 Transmission diagram for the SW ARQ protocol (round trip time *r*=4 slots)

Method

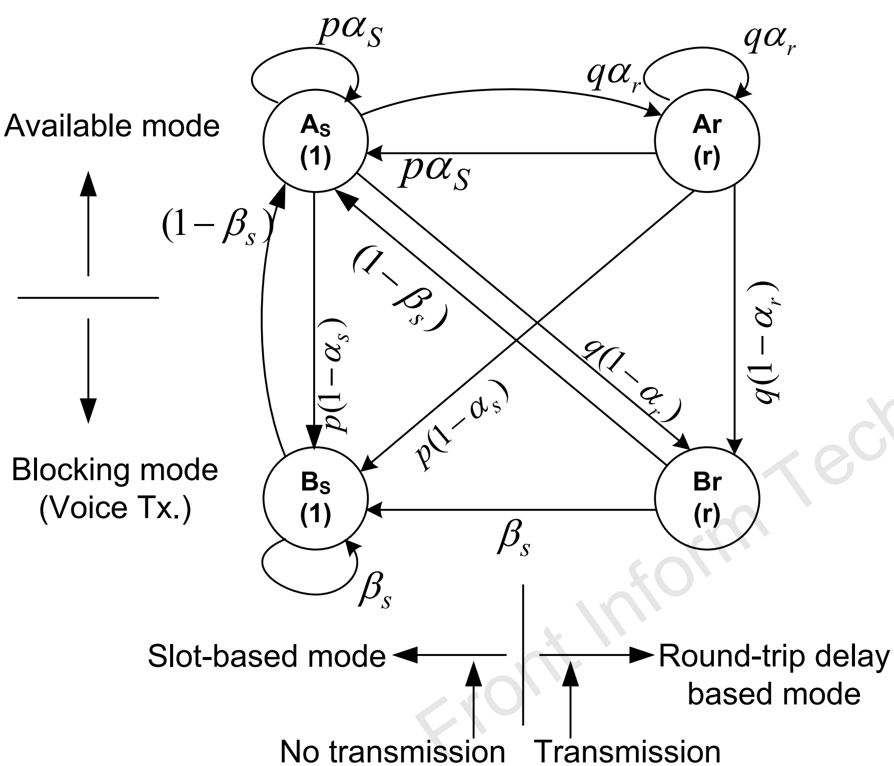

Fig. 5 Modeling of the SW protocol
State transition diagram of four state Markovian processes, considering the round-trip time and slot time for retransmission

The average buffer occupancy can be solved as follows:

$$N'_{SW}(1) = A_s'(1) + B_s'(1) + r\left\{A_r'(1) + B_r'(1)\right\} - \frac{r(r-1)}{2}\left\{A_r(1) + B_r(1)\right\} S'(1)$$

Then, from Little's law, the average waiting time is given by

$$W_{SW} = N'_{SW}(1)/\lambda$$

Numerical results

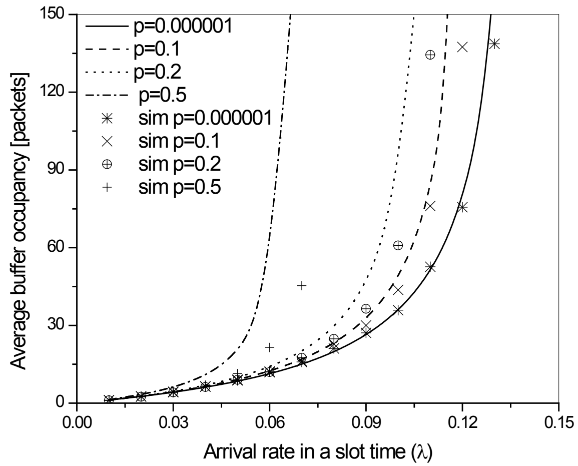

Fig. 6 The average buffer occupancy in SW protocol as a function of arrival rate with packet error probability as a parameter

$\alpha_s = 0.9972 \quad \beta_s = 0.9962$

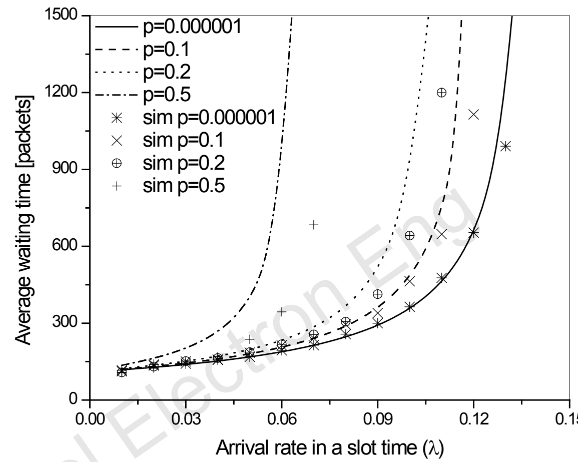

Fig. 7 The average waiting time in SW as a function of arrival rate with packet error probability as a parameter

$\alpha_s = 0.9972 \quad \beta_s = 0.9962$

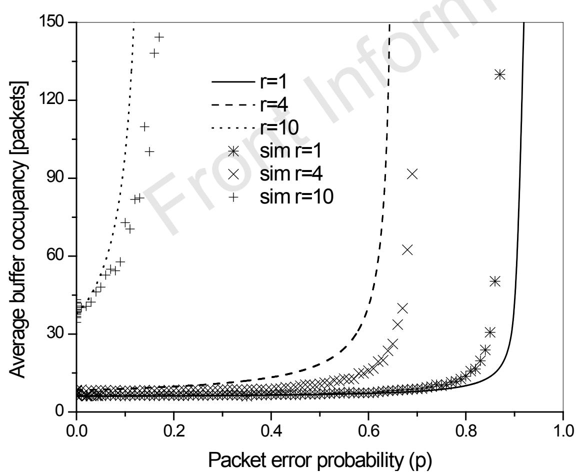

Fig. 8 The average buffer occupancy in SW as a function of packet error probability as parameters

$\lambda = 0.05, \alpha_s = 0.9972, \beta_s = 0.9962$

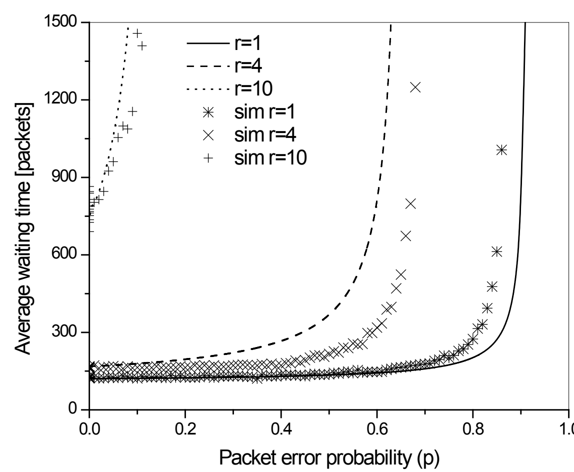

Fig. 9 The average waiting time in SW as a function of packet error probability as parameters

$\lambda = 0.05, \alpha_s = 0.9972, \beta_s = 0.9962$

Conclusions

- This work delivers models of commonly used schemes of SW ARQ.
- Combining the system states, the z-transform of buffer occupancy has been found, from which the average buffer occupancy and the corresponding average waiting time have been obtained.
- From the analytical results, the slot time has to be carefully picked out under the given packet error probability to prevent over buffer occupancy and waiting time.
- The closed-form solution for buffer occupancy is important when queueing theory is applied to data communications. The solution for buffer occupancy and queueing delay can be achieved.
- Eventually, the number of voice interruptions that can be accepted for the quality of service (QoS) when values among buffer occupancy AND/OR delay are specified can be calculated. Inversely, buffer occupancy AND/OR delays can be calculated for a specific number of voice signals. This allows for faster and more efficient data communication systems.