Spreadsheets in Education (eJSiE)

Volume 7 | Issue 3 Article 1

3-10-2014

Modelling Slotted-ALOHA Simulator Using Computational Spreadsheet Based e-Forms

Abdullah A. Basuhail
King Abdulaziz University, abasuhail@yahoo.com

Recommended Citation
Basuhail, Abdullah A. (2014) Modelling Slotted-ALOHA Simulator Using Computational Spreadsheet Based e-Forms, *Spreadsheets in Education (eJSiE)*: Vol. 7: Iss. 3, Article 1.
Available at: http://epublications.bond.edu.au/ejsie/vol7/iss3/1

Modelling Slotted-ALOHA Simulator Using Computational Spreadsheet Based e-Forms

Abstract

This paper demonstrates models that were designed and implemented to simulate slotted ALOHA multiple access computer network protocol. The models are spreadsheet-based simulating e-forms that were designed for students use in college level data communication and networking courses. Specifically, three models for simulating this protocol are fully implemented using spreadsheets. The features of these models are simplicity and quickness of implementation compared with other implementation techniques. These models assisted instructors to develop educational objects that in turn will help students for better understanding and exploring of the scientific concepts related to computer protocols by the aid of visual and interactive spreadsheet-based e-forms. Moreover, advanced spreadsheet techniques such as imagery integration, hyperlinks, conditional structures, conditional formats, and charts insetting, to simulate scientific notions that are taught to undergraduate students were exploited in these models. The models designing technique is characterized by simplicity, flexibility, and affordability. The technique can be applied and used in many disciplines of education, business, science, and technology. Generally, the developed computational e-forms can be used by instructors to illustrate topics in attractive fashions. In addition, students and learners can use the developed educational objects without instructor supervision in self-education or e-learning environments.

Keywords

computer networks, slotted-Aloha, spreadsheets, e-learning, interactive simulation

Modelling Slotted-ALOHA Simulator Using Computational Spreadsheet Based e-Forms

Abstract

This paper demonstrates models that were designed and implemented to simulate slotted ALOHA multiple access computer network protocol. The models are spreadsheet-based simulating e-forms that were designed for students use in college level data communication and networking courses. Specifically, three models for simulating this networking protocol are fully implemented using spreadsheets. The features of these models are simplicity and quickness of implementation compared with other techniques. These models assist instructors to develop educational objects that in turn will help students for better understanding and exploring of the scientific concepts related to computer protocols by the support of visual and interactive spreadsheet-based e-forms. Moreover, advanced spreadsheet techniques and features such as imagery integration, hyperlinks, conditional structures, conditional formatting, chart insertion, built-in formulas and functions to simulate scientific notions that are taught to undergraduate students were exploited in these models. The designing technique is characterized by simplicity, flexibility, and affordability. This implementation approach can be used as a framework for structuring similar electronic models related to many disciplines in education, business, science, and technology. Generally, the developed computational spreadsheet-based e-forms can be used by instructors to illustrate topics in attractive and professional fashions in e-teaching sessions. In addition, students and learners can use the developed educational models without instructor supervision in self-education or e-learning environments.

Keywords: slotted-ALOHA, computer networks, spreadsheets, interactive simulation, e-learning.

1. Introduction

Scientific problems of computational nature can be made more grasp to students by using modern computers. One of the common tools for simulation and realization of such problems is computer programs written in high-level programming languages, such as Java, C#, C++, etc. This paradigm of implementation requires knowledge and experience with programming techniques, and familiarity with visualization tools, as well as, dedicating time and budget for achieving this purpose.

Spreadsheets are powerful alternative computing tools offered for flexible use and rapid implementation and realization for computational tasks. They are used to store data in tables, perform mathematical and logical computations, and visualize data and results in various formats and styles.

Mostly, spreadsheets are used as implementing computational tools for topic or problem of study. Spreadsheet-based implementations assist students in learning and offer them the chance to try, test, and investigate several cases, trials, and experiments related to a designed problem. They also facilitate exploration of what-if scenarios and build a hands-on experience for scientific concepts and facts.

Modern electronic spreadsheets are used in various scientific fields. Their features and capabilities make them attractive tools to solve complex problems in attainable manners. They have been used in a wide range of engineering and technology applications [1], [2], [3], and [4]. They are used to introduce practical applications to undergraduate college students [5]. Some researchers reported the use of spreadsheets for simulation in diverse scientific fields [6], [7], and [8]. Other researchers used them for implementing, studying, and analysing systems and models [9]. Still some researchers used electronic spreadsheets for applying mathematics for computer processing purposes [10]. Spreadsheets also are used as one of the most suitable tools for the rudimentary engineering education [11]. They are also used in teaching computational thinking [12]. A program has been developed to assist chemistry students in understanding the mole concept using spreadsheets [13].

The technique of modelling and implementing systems and problems using spreadsheets can be practiced by instructors who are deficient or inexpert with the implementations using high-level computer programming languages and techniques or using sophisticated software packages.

Generally, the use of available tools, such as spreadsheets, can facilitate the design and implementation stages for non-expert developers of e-content [14]. Although there are some features in other electronic packages and tools which are superior to spreadsheets based designs, or they are favoured by some users, the popularity of the later one can make the development and use of spreadsheet-based designs to be a building process over previous owned expertise [14].

2. Spreadsheets

An electronic spreadsheet is a collection of cells tabulated into rows and columns. Each cell is numbered by a concatenation of a column alphabetical letter and a row number that represents the location of the cell. Spreadsheet cells can contain data of assorted types. A cell can also be formulated to compute simple or complex formulas and mathematical functions.

Modern spreadsheet packages are equipped with numerous built-in functions related to various scientific disciplines and applications. The use of those pre-programmed functions reduces some of the burden of programming, testing, and debugging required by the problem under consideration. This feature encourages developers to integrate designated functions in their implementations and in turn saves the effort of entering, verifying, and validating needed formulas and functions.

Visuals, such as pictures and computer graphics, can be integrated in spreadsheet-based designs. The use of these objects enhances the look-and-feel user interface for the considered topic or problem. It makes the final designed object to be more informative, attractive to students, and resembles the visualization of the real problem.

Another powerful spreadsheet feature is the ability of representing data using charts with wide variety of styles. This capability enhances displaying and observing relations among data collections. It also enables representing output results with the cost of few fingertip keystrokes during implementation of a computational e-form.

Other spreadsheets features, such as decision-making, sorting, filtering, hyperlinks, conditional formatting, and many other characteristics, facilitate implementing numerous powerful and flexible electronic computational spreadsheet-based educational objects that are used in various educational and training applications and fields, specifically, in e-learning, e-training, and e-teaching environments.

3. Slotted ALOHA Multiple Access Computer Network Protocol

In multiple access computer networks, stations are connected using a common communication link [15]. Control and coordination to access the link are needed for transmitting data frames among stations. However, if more than one station tries to send data frame during the same time slot with no mutual exclusion, a collision will occur and the data frames will be either destroyed or modified. In order to avoid access conflict or minimize it, a protocol must be imposed for transmission [16].

One of the classical protocols taught to college level students in data communication and networking courses is Slotted ALOHA multiple access protocol [15] and [16]. In this protocol, if a station generates only one frame and no other station generates a frame during this time, the frame will reach its destination successfully.

In the slotted ALOHA protocol, time is divided into slots of *T* sec. each. This time is adjusted to be enough to send an entire data frame. A station is forced to transmit only at the beginning of the time slot. In other words, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame.

There is possibility of collision if two stations try to send at the beginning of the same time slot. In that case, the transmission channel is carrying incorrect or destroyed data resulted from the collision.

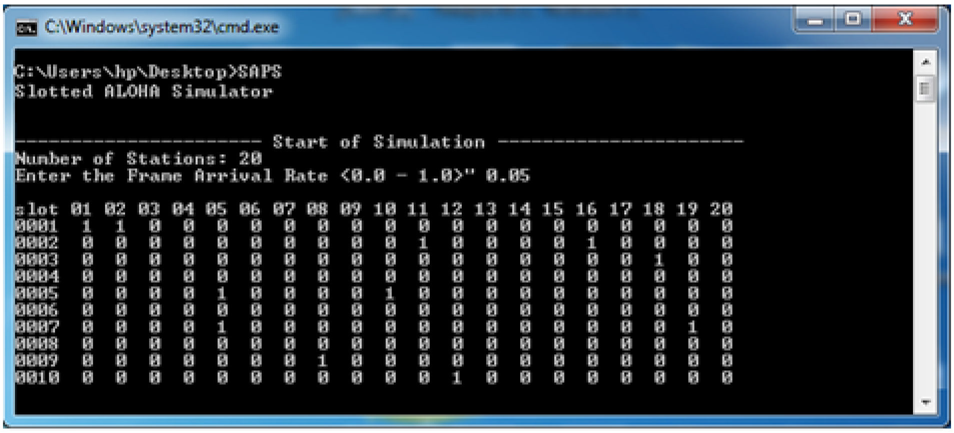
The average number of successful transmissions for slotted ALOHA protocol, known as throughput *S*, is computed as

$$S = G \text{ x } e^{-G} \tag{1}$$

where *G* is the average number of frames generated by the system during one frame transmission time. Theoretically, the maximum throughput S_{max} is 0.368, when *G* = 1. In other words, if frames are generated during transmission time period, then 36.8% of these frames reach their final destinations successfully. Collided frames are retransmitted at later time slots.

4. Simulating Slotted ALOHA Protocol Using Spreadsheets

During the teaching of data networks and communication courses, one of the tasks that I assigned to the students was examining and analysing the behaviour of some networking protocols. The students needed to simulate those protocols by writing computer programs using one of the high-level programming languages on their own. Usually, most of the students were faced with the obstacle of implementing such protocols due to the burden of designing and programming the problem from scratch. Moreover, in most of the cases, the realizations were performed as command line interface, CLI-based; in which the interaction was conducted through entering input data and collecting output results, as shown in Figure 1. Furthermore, for the students, design and implementation of protocol simulators using visualizations were even more complicated and time consuming compared with CLI designs.



```
C:\Users\hp\Desktop>SAPS
Slotted ALOHA Simulator


---------------------- Start of Simulation ----------------------
Number of Stations: 20
Enter the Frame Arrival Rate <0.0 - 1.0>" 0.05

slot 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20
0001  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0002  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0
0003  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0004  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0005  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
0006  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0007  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
0008  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0009  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
0010  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
```

Figure 1: CLI-based slotted ALOHA simulator

Due to their flexibility and attractive implementation features, several spreadsheet based models that helped students in performing analysis and exploration of slotted ALOHA protocol were designed and implemented. Those models offered the students better understanding and hands-on experience for related topics. The models also saved time of designing and programming stages for the realization of the protocols. That saved time was dedicated for more focus and investigation on the topics of study.

To facilitate the task of exploring random access computer network that applies slotted ALOHA protocol, three models for analysing computer network behaviour that applies this protocol were designed and implemented using spreadsheets.

The implementation of the models used several spreadsheet and graphical components and features. The components are categorized as either graphic, control, or cell. Any cell is crafted and formatted according to the purpose needed. Table 1 lists the elements exploited in designing and building the models implemented for the slotted ALOHA protocol simulators (SAPS).

Table 1: Components used in spreadsheet-based SAPS models

Component	Category	Use
Computer station	graphic-cell	toggle monitor colour
Channel cable	graphic-cell	detect channel state
Radio buttons	control	switch station: off/on
Spin button	control	set slots number
Frame	cell	detect attempt status
Station-slot frame	cell	detect slot state
Link	cell	network link state
Input textbox	cell	enter user input
Output textbox	cell	display output result
Legend	cell	explanatory text

4.1. Design Methodology

Building a computational e-form for educational purposes requires several stages to control the design. Specifically, these phases can be taken as guideline for developing e-Forms that range from simple to complex ones using spreadsheet tables.

First, the gridlines of the sheet are made invisible. Moreover, it is recommended to resize it to almost one screen size. Refilling the cells at locations where it is required so they become visible. The overall view of the e-form is improved by adding proper main explanatory titles and labels.

Next, inserting graphics and images to enhance the user interface of computational e-form designed for problem under consideration. It can make the design to be attractive, informative and self-used.

Controls, such as radio buttons and spin buttons, can also be incorporated in the design. They offer more flexibility and easiness when using developed spreadsheet-based e-Forms. The controls require programming and linking to the appropriate e-Form components for accurate functioning.

Mostly, a computational e-form includes one or more mathematical formulas and/or equations. The use and integration of built-in functions in a computational e-form helps in designing more rapid and accurate implementations with high reliability. In several cases formulas are written directly into cells. In some cases, VBA might be needed for embedding subroutines which will be called when needed in the e-Form. That is governed by the availability of suited built-in formulas and the complexity of the problem under consideration.

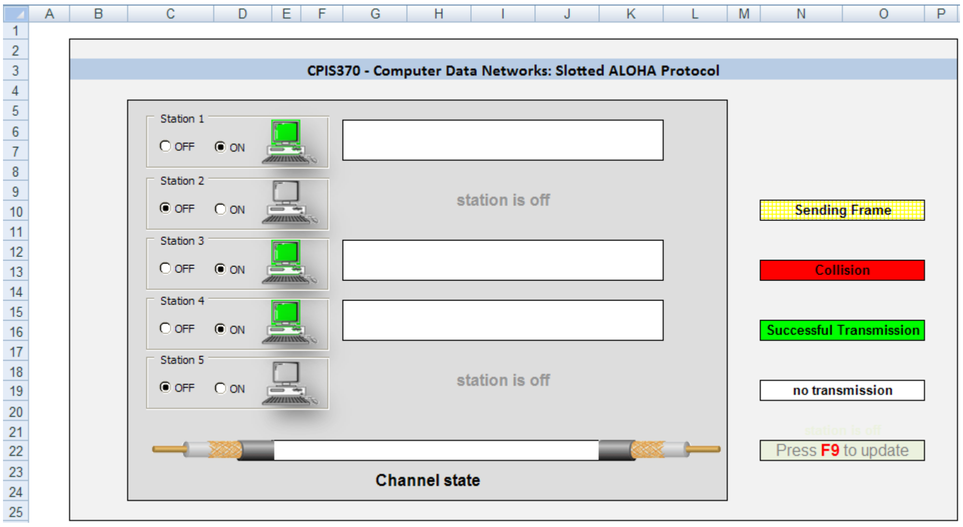
The designing methodology was used to implement several spreadsheet-based computational e-Forms. For the purpose of this paper, the following subsections will shed light on three interactive spreadsheet-based models that were designed and implemented for the slotted ALOHA protocol simulator. The developed models are used in computer data communication and networking courses.

5. SAPS: Model 1

The first computational spreadsheet-based model that was designed and implemented to visually simulate slotted ALOHA protocol. The model contained several components including spreadsheet elements. It was used to illustrate the successful and collision states of network transmissions.

5.1. Implementation

The implementation of this model used several spreadsheet and graphical components among the elements listed in Table 1. The implemented model is shown in Figure 2. The spreadsheet includes a collection of five stations that can send frames using slotted ALOHA protocol model 1.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

Station 1 ○ OFF ● ON

Station 2 ● OFF ○ ON station is off

Station 3 ○ OFF ● ON

Station 4 ○ OFF ● ON

Station 5 ● OFF ○ ON station is off

Channel state

🟨 Sending Frame

🟥 Collision

🟩 Successful Transmission

□ no transmission

Press F9 to update

Figure 2: SAPS model 1: simulator settings

Each station can either be switched "off" or "on" by toggling its corresponding radio button. The sheet is automatically programmed to display green glowing colour on the monitor of any switched-on station.

The rectangular cell to the right of each station represents the transmitting channel used to indicate the state of the station with regard to the attempt to transmit data frame.

Five legend-cells are added to the right of the exhibition as shown in Figure 2. These cells help the user in using the model. These cells assist in following-up with the resulted simulation actions and results.

The channel state cell shown at the bottom of Figure 2 is used to display the network state during data transmissions. It is programmed to indicate and visually detect the status of transmission in the network.

The stages for transmission by any switched-on network station in this model are performed by the following algorithmic steps:

1. Attempt transmissions:

```
for all stations
    if (station = "on") {
        toggle station monitor colour to green;
        attempt = generate random number binary (0/1);
        if (attempt = 1) {
           frame = 1;
           toggle station monitor colour to yellow;
        }
        else
           frame = 0;
}
```

2. Compute total signal carried by the channel:

$$signal = \sum_{i=1}^{n} frame_i$$

where n is number of stations.

3. Determine transmission channel state:

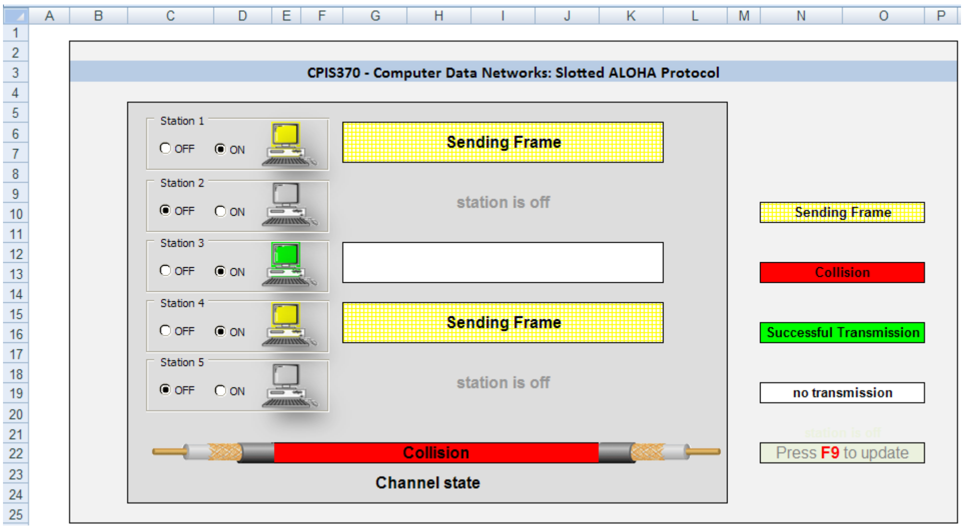
$$state = \begin{cases} sucessful & if\ signal = 1 \\ idle & if\ signal = 0 \\ collision & if\ signal > 1 \end{cases}$$

4. Toggle colours

```
if (state = successful) {
    toggle transmitting station monitor colour to green;
    toggle channel colour to green;
}
elseif (state = collision)
    toggle channel colour to red;
```

The first step for using this model is to setup the stations off/on buttons according to the desired simulation experiment. For example, as can be depicted from Figure 2, stations 1, 3 and 4 are switched "on", on the other hand, stations 2 and 5 are turned "off". The colour of any switched-on station is toggled to green. On the other hand, the monitor of any transmitting station is automatically toggled to yellow colour.

Figure 3 illustrates a situation of random access slot in which two stations, specifically; stations 1 and 4 are transmitting simultaneously. That slot resulted into collision. This is depicted by observing that the two stations, attempted to transmit at the same time slot without mutual exclusion. This situation is marked by the alarming red colour in the channel state cell and collision message is generated in this case.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

Station 1 ○ OFF ● ON — Sending Frame

Station 2 ● OFF ○ ON — station is off

Station 3 ○ OFF ● ON

Station 4 ○ OFF ● ON — Sending Frame

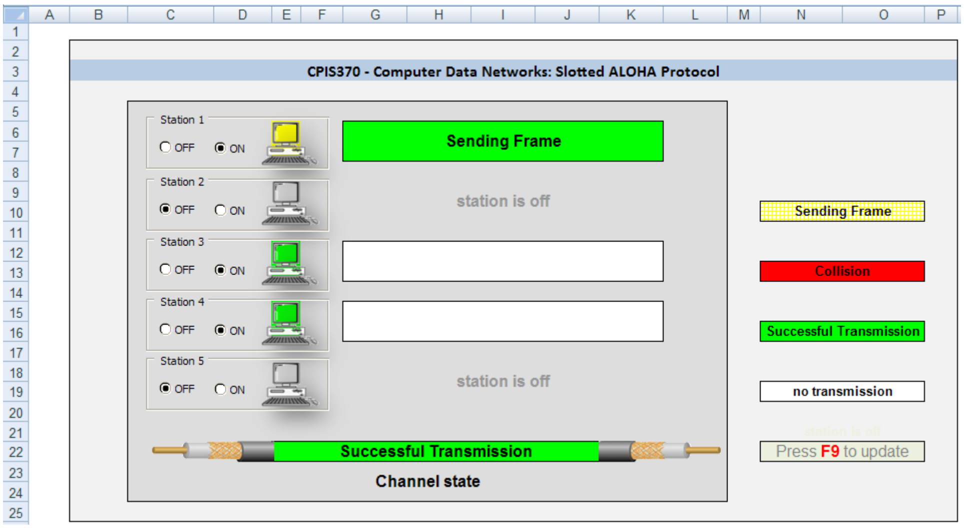
Station 5 ● OFF ○ ON — station is off

Collision

Channel state

Figure 3: SAPS model 1: collision slot

On the other hand, Figure 4 illustrates a case in which only one station is using the channel at a certain slot with mutual exclusion; specifically, station 1. This situation is resulted in successful transmission as this case is signified by automatic toggling of the channel state to green coloured cell with proper generated message.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

Station 1 ○ OFF ● ON — Sending Frame

Station 2 ● OFF ○ ON — station is off

Station 3 ○ OFF ● ON

Station 4 ○ OFF ● ON

Station 5 ● OFF ○ ON — station is off

Successful Transmission

Channel state

Figure 4: SAPS model 1: successful transmission slot

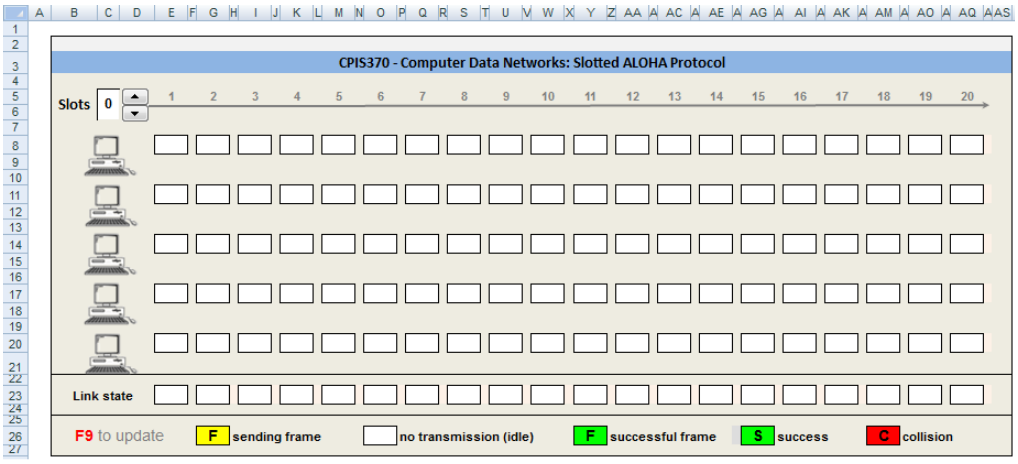
The SAPS model 1 simulator can run over and over by pressing the function key *F9*, in such case, a new simulation data is being generated. Generally, this model is used to anticipate the network successful transmission and collision states.

6. SAPS: Model 2

The second model was designed to study the transmission traffic during selected number of slots applied to a network that applies slotted ALOHA protocol. In this model, the stage of simulation can be set in order to study the behavior of the network over a period of time.

6.1. Implementation

The model used several spreadsheet and graphical components from the list in Table 1. The implemented model is shown in Figure 5. The spreadsheet includes a collection of stations that can send frames using slotted ALOHA protocol model 2. The stations can transmit data frames for a period of up to 20 slots. The number of slots to be used during simulation is set using spin button. The transmitted frames are shown as white cells at the intersections of the station-slot grid.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

Slots 0

Link state

F9 to update

F sending frame

no transmission (idle)

F successful frame

S success

C collision

Figure 5: SAPS model 2: simulator settings

The stages for transmission by any switched-on network station in this model are performed by the following algorithmic steps:

1. Attempt transmission: generate frames for all cells in station-slot grid

$$gframe = rand(0{,}1);$$

where *rand* is binary random number between 0 and 1 inclusive.

2. Label all cells in the station-slot grid:

$$cframe = \begin{cases} "F" & if\ gframe = 1 \\ "nil" & if\ gframe = 0 \end{cases}$$

3. For each simulation slot t

i. Determine total link signal:

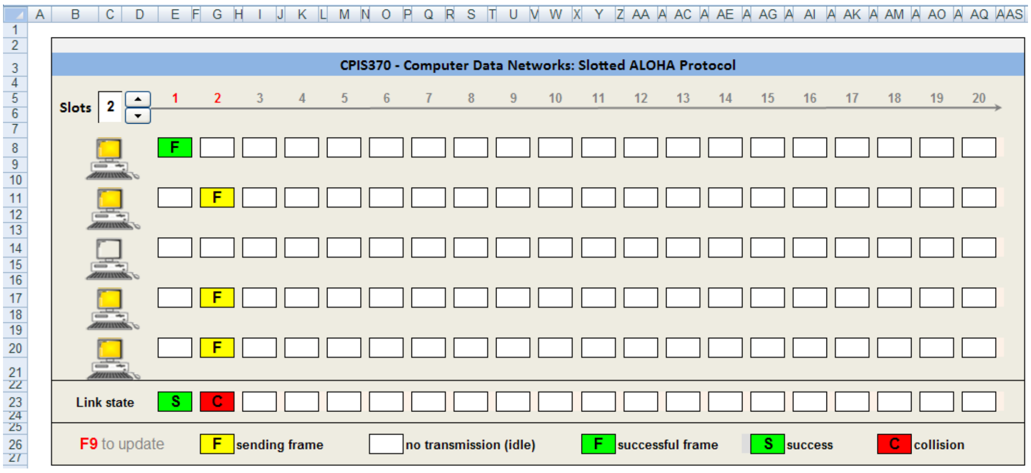
$$signal_t = \sum_{i=1}^{n} cframe_{it}$$

where n is the number of stations used in the simulation.

ii. Determine and label link state

$$state_t = \begin{cases} "S" & if\ signal_t = 1 \\ "" & if\ signal_t = 0 \\ "C" & if\ signal_t > 1 \end{cases}$$

The first step in using this model is to setup the desired number of slots to run the simulation. Figure 6 demonstrates a sample simulation run within a period of 2 slots of transmissions. As can be noticed from the figure, the stations that attempted to send frames are interactively glowing with yellow monitor colour. A successful transmission displays character *"S"* in the corresponding station-slot cell; on the other hand, a collision transmission is labeled with character *"C"*.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

Slots 2

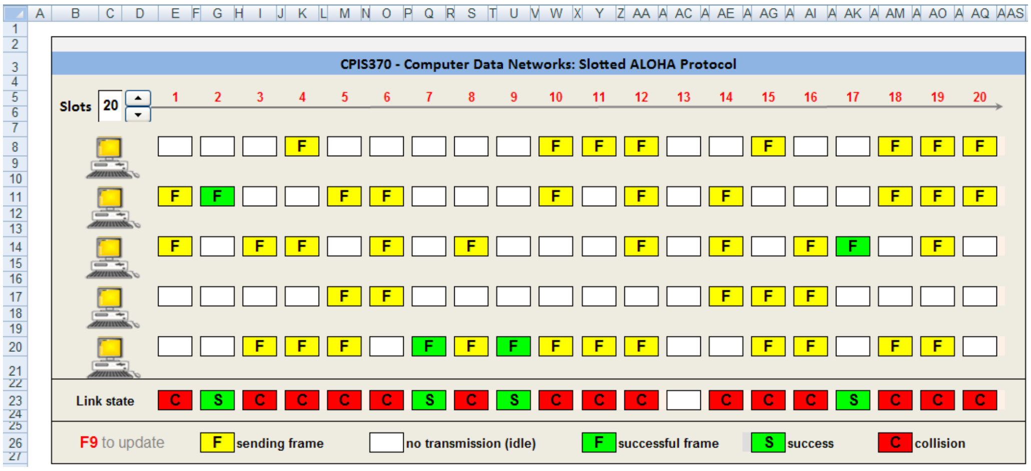
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	F																			
		F																		
		F																		
		F																		
Link state	S	C																		

F9 to update F sending frame □ no transmission (idle) F successful frame S success C collision

Figure 6: SAPS model 2: simulation using 2 slots

Students studying and analyzing the behavior of this protocol can simply run the simulator over and over by hitting the function key *F9*. Each time the simulation is run, new simulation data and states are generated and the situation is updated.

Figure 7 depicts exploiting all the possible slots to generate full output simulation data. It can be noticed that most of the slots resulted in collisions. This situation is in accordance with the theoretical behaviour of the slotted ALOHA networking protocol; specifically, the maximum throughput of a network that applies this protocol is within the range of 36.8%, which means that most of the transmitting slots will result into collision.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

Slots 20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
				F						F	F	F			F			F	F	F
	F	F			F	F				F		F		F				F	F	F
	F		F	F		F		F				F		F		F	F		F	
					F	F								F	F	F				
			F	F	F		F	F	F	F	F	F			F	F		F	F	
Link state	C	S	C	C	C	C	S	C	S	C	C	C		C	C	C	S	C	C	C

F9 to update F sending frame □ no transmission (idle) F successful frame S success C collision

Figure 7: SAPS model 2: simulation using 20 slots

This model can be used as hands-on experience for students to observe the actual traffic of the network within several time slots and to study the relation between the number of stations in the network and the state of transmission.

7. SAPS: Model 3

The third spreadsheet-based model that was designed and implemented for slotted ALOHA protocol used several components from the list in Table 1. This model is more interactive and informative than the two previous mentioned models, as the user can set and change some parameters related to the simulated network.

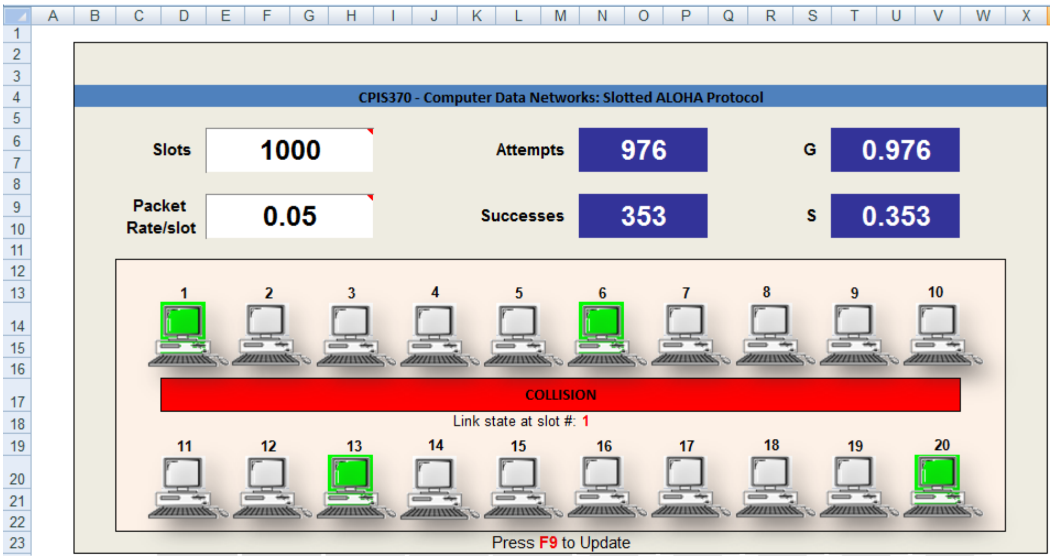
7.1. Implementation

In this model, the parameter values; specifically, the numerical values of desired number of slots to carry the simulation and packet rate are inputted in their designated cells. Accordingly, the sheet automatically computes and displays the simulation resulted output information; specifically, number of total attempts, number of successful transmissions, average number of frames generated, and network throughput.

The stages for transmission by any switched-on network station in this model are performed by the following algorithmic steps:

```
1. Initialize parameters:
    i. set number of slots;
    ii. set packet rate: α;
2. Attempt transmission
  for i = 1 to slots
     $backlogged_i = 0$;
     for j = 1 to stations {
              $frame_{ij} = \begin{cases} "F" & if\ rand(0/1) < \alpha \\ "nil" & if\ rand(0/1) \geq \alpha \end{cases}$
         $backlogged_i$ += $frame_{ij}$;
         attempts += $frame_{ij}$;
         if ($frame_{ij}$ = "F")
               toggle transmitting station monitor colour to green;
     }
              $state_i = \begin{cases} "success" & if\ backlogged_i = 1 \\ "collision" & if\ backlogged_i > 1 \end{cases}$
     if ($state_i$ = success)
           toggle link channel colour to green;
     elseif ($state_i$ = collision)
           toggle link channel  colour to red;
  }
3. Compute:
   load = attempts/slots;
   throughput = successes/slots;
```

The e-Form interactively displays the stations that attempted to send with green glowing colour monitors, as well as, the link state cell. Figure 8 illustrates a case of collision transmission. As can be noticed in this case, four stations are attempting to transmit simultaneously at the same time slot. The link state cell colour is turned into red to visually indicate the occurrence of collision.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

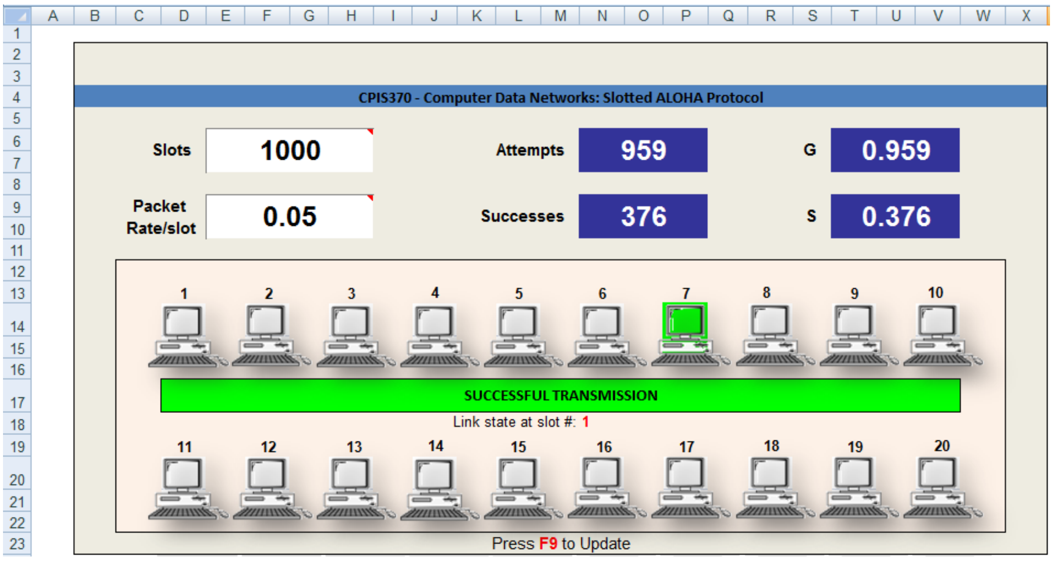
Slots 1000
Attempts 976
G 0.976
Packet Rate/slot 0.05
Successes 353
S 0.353

COLLISION
Link state at slot #: 1

Press F9 to Update

Figure 8: SAPS model 3: collision slot

Figure 9 demonstrates a case of successful transmission slot. As can be observed in this case, the link is used by only one transmitting station; precisely, station 7. The link state cell colour is turned into green to visually indicate the occurrence of successful transmission with a proper text message.



CPIS370 - Computer Data Networks: Slotted ALOHA Protocol

Slots 1000
Attempts 959
G 0.959
Packet Rate/slot 0.05
Successes 376
S 0.376

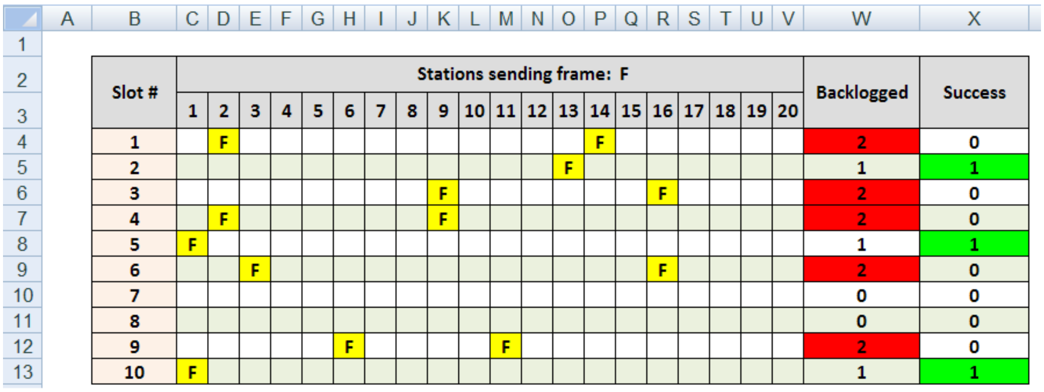
SUCCESSFUL TRANSMISSION
Link state at slot #: 1

Press F9 to Update

Figure 9: SAPS model 3: successful transmission slot

Simulation data can be extracted from this model. The total number of transmission attempts within the duration of 1000 slots is 959 attempts. The number of successful transmissions during this period is 376. The network throughput in this case is 37.6%.

The remaining simulation data and results for this model are generated and tabulated in a separate worksheet. This worksheet is updated each time a change is performed or the simulator is run. Figure 10 illustrates the first 10 out of 1000 slots of the simulation data which is generated by the SAPS model 3. This grid shows the traffic activity performed by all the network stations and their occurrences in the time domain.



Slot #	Stations sending frame: F																				Backlogged	Success
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
1		F												F							2	0
2													F								1	1
3									F							F					2	0
4		F							F												2	0
5	F																				1	1
6			F													F					2	0
7																					0	0
8																					0	0
9						F					F										2	0
10	F																				1	1

Figure 10: SAPS model 3: simulation data for 10 slots

Students can use this worksheet for further analysis and investigation. The spreadsheet automatically labels the slots of transmission for all the stations. The last two columns of the worksheet are automatically distinctly highlighted at the places of occurrences of successful and collision transmissions. Specifically, the backlogged slots at which collision occurs and the success slots resulted from situations when only one station attempts to transmit and uses the channel with mutual exclusion. These columns are marked for easy tracking. They have been done using conditional formatting feature.

7.2. Experimentation

The students in computer data communication and networking courses were asked to utilize the ready-to-use simulators to generate data traffic which was used to study and analyse the behaviour of a network that applied multiple access slotted ALOHA protocol for data transmissions among its nodes. The analysis was including the changes applied to some of the parameters and their effects and consequences on the network performance.

Table 2 summarizes example of the simulation data and results that were recorded and collected. The data was used to study the effect of changing the packet rate (α) on the overall throughput for a network that utilized SAPS model 3. The 2nd and the 3rd columns of Table 2 represent theoretical load (G), and throughput (S_{the}) for comparison purpose to the output throughput by the simulator (S_{sim}). A sample output data for one experiment is listed in the 4th column. The 5th column represents the absolute difference between the two throughput values. As can be noticed, the differences are very small values that falls in the range 0.000 - 0.028.

The output throughput values, as can be observed from Table 2, start at zero and then increases sharply to reach its maximum value which occurs at network load value $G = 1$, that is at packet rate value $\alpha = 0.05$, and then it gradually drops back towards zero.

Table 2: Effect of packet rate on throughput for SAPS network

α	G_{the}	S_{the}	S_{sim}	ΔS
0.00	0.00	0.000	0.000	0.000
0.02	0.40	0.268	0.296	0.028
0.04	0.80	0.359	0.354	0.005
0.05	1.00	0.368	0.362	0.006
0.06	1.20	0.361	0.353	0.008
0.08	1.60	0.323	0.308	0.015
0.10	2.00	0.271	0.251	0.020
1.12	2.40	0.218	0.193	0.025
1.14	2.80	0.170	0.161	0.009
1.16	3.20	0.130	0.138	0.008
1.18	3.60	0.098	0.097	0.001
0.20	4.00	0.073	0.053	0.020

Since the data transmission in this kind of network is random access, the frames generation is also random and it is variable during time. Table 3 lists 5 experiments for generating data frames during a period of 1000 slots for each sample and they are shown in columns 2 through 6. The last column of Table 3 gives the average value for all the five experiments performed in this analysis.

Table 3: Effect of packet rate on throughput for SAPS network - 5 experiments

α	S_1	S_2	S_3	S_4	S_5	$\bar{S}$
0.02	0.272	0.289	0.265	0.263	0.278	0.273
0.04	0.364	0.329	0.405	0.364	0.343	0.361
0.05	0.375	0.361	0.371	0.389	0.391	0.377
0.06	0.368	0.354	0.385	0.344	0.397	0.370
0.08	0.342	0.334	0.333	0.321	0.287	0.323
0.10	0.249	0.261	0.276	0.279	0.27	0.267
1.12	0.218	0.222	0.203	0.21	0.219	0.214
1.14	0.152	0.157	0.158	0.158	0.153	0.156
1.16	0.113	0.101	0.125	0.111	0.113	0.113
1.18	0.098	0.108	0.065	0.08	0.066	0.083
0.20	0.074	0.058	0.047	0.051	0.057	0.057

The relation between the packet rate and the average throughput compared to the expected theoretical output of transmissions for the network that applies the slotted ALOHA networking protocol is charted in Figure 11.

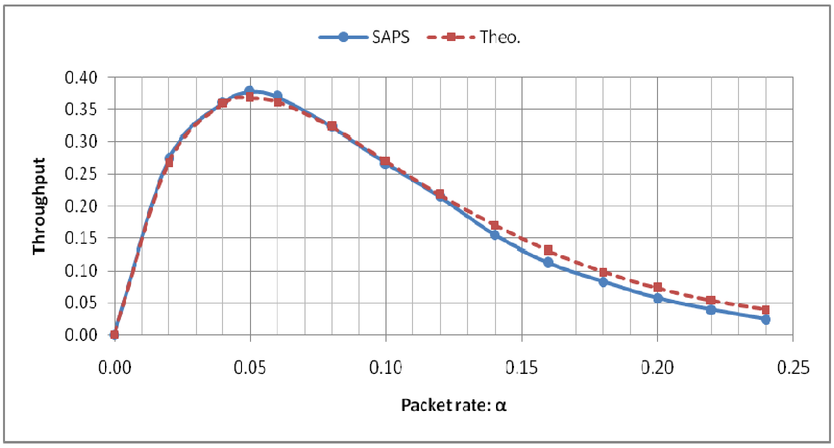


Figure 11: Network throughput measured using theoretical and simulation models

It can be concluded from this graph that the simulator output results, represented by the solid line curve, are almost coincide with the theoretical expected values, represented by the dotted line curve. The curves demonstrate the actual slotted ALOHA network behaviour affected by the packet rate. It starts to increase with small values of packet rate until it reaches a peak value, and then it decays slowly towards the zero value at high packet rates.

Many other experiments can also be performed and their output results can be analysed using the SAPS simulators. Students who are using these simulators need to concentrate only on studying and analysing the network behaviour and performance rather than concerning about programming and related debugging and testing tasks.

8. Conclusions

This paper demonstrated the design and implementation of three models for simulating slotted ALOHA protocol that was taught in computer communication and networking undergraduate courses. It also discussed some experiments with analysis performed using these models of simulators.

The models represent a promising alternative to the command line interfaced-based model which is time intensive with regard to implementation, and it lacks the visualization feature that helps students towards better grasping and quick visualizing of designated problems of study. Moreover, this designing method can relief the developers from many tedious tasks obligated by other implementation techniques, such as designing and writing programs using high-level computer programming languages.

The students who are given related problems to study and investigate showed better performance and response to examine and analyse computer network protocols using ready pre-programmed interactive simulator models rather than programming and using their own implementations. That was observed by the fewer number of submissions and completions of the assigned exercises and projects by students who submitted their final reports using their own models compared to almost full number of submissions and finishing when the students are given the interactive models.

The technique for implementing the models discussed here could be applied in other disciplines of science and technology. For example, numerous scientific problems of computational nature, such as concepts and laws in electricity, physics, etc. could be designed and implemented in similar fashions.

Some limitations of using spreadsheets-based e-Forms were faced during the design and implementation of the models. These limitations include fewer selections of animated interaction. These animations were mostly occur as appearance or hiding of text, or sensed as toggling of colours. Other drawback was experienced in the design of the models. That was due to the less flexibility when integrating graphical images within a computational e-form.

For future work, other computer network protocols will be implemented and investigated to develop more general platform for the design of spreadsheet-based computational e-forms. Students will be encouraged to follow the same design and implementation technique and adopt it for their implementations. Furthermore, spreadsheet sophisticated features, such as exploiting other forms control in the design and utilizing advanced VBA capabilities will also be incorporated for more flexibility and easiness of using the final developed educational spreadsheet-based e-Forms.

References

[1] Banta, A. (1998). Using Computer Spreadsheets in Engineering and Education. Frontiers in Education Conference Proceedings.

[2] Xu W., Liu Y., Koval, D., and Lipsett, M. (1999). Using Spreadsheet Software as a Platform for Power System Analysis. *IEEE Computer Applications in Power Journal*, vol. 12, pp: 41-45.

[3] Pearson, R. and Atkinson, G. (2003). Teaching Vacuum Technology Using Spreadsheet Calculations. Proceedings of the 15th Biennial University/ Government/ Industry Microelectronics Symposium.

[4] Alsagoff, S. (2010). Microsoft Excel as a Tool for Digital Forensic Accounting. IEEE International Conference on Information Retrieval & Knowledge Management, (CAMP).

[5] Svoboda, J. (1992). Using Spreadsheets in Introductory EE Courses, *IEEE Transactions on Education*, vol. 35, no. 3.

[6] El-Hajj, A. and Hazim, M. (1998). On Using Spreadsheets for Logic Networks Simulation. *IEEE Transactions on Education*, vol. 41, no. 4.

[7] Baicher, S., and Sherrington, J. (1996). Learning about Digital Signal Processing Using Spreadsheets and Simulation Software. *Journal of Engineering Science and Education*.

[8] El-Hajj, A., Kabalan, K., and Mahmoud, S. (2003). An Improved Spreadsheet Toolbox for Simulating Digital Networks. *IEEE Circuits and Devices Magazine*.

[9] Walter, D., and McMillan, M. (1990). A Spreadsheet Method for Studying Neural Networks. IEEE Proceedings of the 1990 Symposium on Applied Computing.

[10] Waldock, J. (2010). Applying Mathematics to Digital Image Processing using a spreadsheet, MSOR Connections, vol. 10, no. 3.

[11] Sakuma, F., and Matsui, M. (2007). Utilization of Spreadsheet Software as Engineering Education. Power Conversion Conference - PCC '07.

[12] Yeh, K., Xie, Y., and Ke, F. (2011). Teaching Computational Thinking to Non-computing Majors Using Spreadsheet Functions. 41st ASEE/IEEE Frontiers in Education Conference.

[13] Arifin, B., Ismail, H., and Yeop, U. (2012). The Mole Concept Using MS Excel. 2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE).

[14] Basuhail, A. (2009). Design and Implementation of e-Learning Content Using Simple and Obtainable Tools. First International Conference e-Learning and Distance Education.

[15] Tanenbaum, A., and Wetherall, D. (2011) Computer Networks, 5th Edition, Pearson Education Publishing Inc.

[16] Forouzan, B. (2007). Data Communication and Networking, 4th Edition, McGraw-Hill Publishing Company.