Performance analysis of general backoff protocols

Andrey Lukyanenko and Andrei Gurtov

***Abstract*—In this paper, we analyze backoff protocols, such as the one used in Ethernet. We examine a general backoff function (GBF) rather than just the binary exponential backoff (BEB) used by Ethernet. Under some mild assumptions we find stability and optimality conditions for a wide class of backoff protocols with GBF. In particular, it is proved that the maximal throughput rate over the class of backoff protocols is a fixed function of the number of stations (N) and the optimal average service time is about Ne for large N. The reasons of the instability of the BEB protocol (for a big enough input rate) are explained.**

Additionally, the paper introduces novel procedure for analyzing bounded backoff protocols, which is useful for creating new protocols or improving existing, as no protocol can use unbounded counters.

***Index Terms*—Ethernet, backoff protocol, contention resolution, stability, optimality, queuing theory.**

I. Introduction

ETHERNET was developed in 1973 by Bob Metcalf and David Boggs at the Xerox Palo Alto Research Center. Nowadays, it is the most popular local area network due to the ease in maintenance and the low cost. The principle of Ethernet is that all stations are connected to the same shared medium through transceivers. Whenever a single station wants to send a message, it simply broadcasts it to the medium. When at some moment of time there are two or more messages in the medium, they interfere, and none of them can be received by any station. To deal with unnecessary collisions, a resolution protocol was developed (see [7]). It has the following mechanisms:

1) *Carrier detection.* This mechanism lets stations know when the network has a message. If any station senses that there is a phased encoded signal (continuous) in the network, then it will defer the time of its own transmission until the channel becomes empty.
2) *Interference detection.* Each station listens to the channel. When it sends a message it is continuously comparing the signal that has just been sent and the signal in the network at the same moment of time. If these signals have different values, the message is considered to be corrupted. Here we introduce the round trip time: this is the time during which the signal propagates from one end of the network to the other and back.
3) *Packet error detection.* This mechanism uses checksums to detect corrupted messages. Every message with a wrong checksum is discarded.
4) *Truncated packet filtering.* This mechanism lets us reduce the load on the system if the message is already corrupted (detection during round trip time), and filter them out on the hardware level.
5) *Collision consensus enforcement.* When a station sees that its message is corrupted, it jams the whole network by sending special "jammed" information. This mechanism ensures that no station will consider the corrupted message in the network to be a good one.

Due to these mechanisms, a message is not sent when it is known that there is information in the medium, and if a collision happens, it can be clearly sensed. But there is still a possibility that one station decides that the medium is empty, while another has already started a transmission. There will be interference at some point of the network. A probabilistic protocol will help us to avoid this problem. Our work examines a general type of the probabilistic protocol.

A. Protocol

Let there be N stations and every station have a queue of messages to send. These stations are connected to a shared medium, where collisions may happen time to time. To deal with such collisions, the backoff protocol from the Aloha network was adopted (see [2], [3]). If a collision occurs in the backoff protocol, the next retransmission will be done in one of the next W moments, where W is a time window of certain size and the retransmission in the window is chosen uniformly. *A time slot* (or just *a slot*) is a time equal to the round trip time. We can vary this time in our model in order to get model closer to the real protocol, where is no time synchronization. If a collision does not happen during the first time slot (for a message that requires more than one time slot to be transmitted), it will not happen at all, due to the carrier detection mechanism. Therefore, we can consider that the transmission of one message takes only one time slot in our model. The segment of time slots $[1 \dots W]$ is called *a contention window*. The idea behind the contention window is that we select a time slot for transmission uniformly in the contention window. The main goal of this principle is to reduce the load on the network, and hence to increase the probability of successful resending of a message during one of the next time slots, within the contention window.

Manuscript received December, 2007 and revised February, 2008.

This Paper was presented as part at the Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN) 2007.

The paper was written in part while A. Lukyanenko was with University of Kuopio, P.O.Box 1627, FI-70211 Kuopio FINLAND and supported in part by CIMO Fellowship, 2006, Finland.

A. Lukyanenko and A. Gurtov are with Helsinki Institute for Information Technology, Helsinki, P.O. Box 9800, FIN-02015 TKK, Finland e-mail: (see http://hiit.fi/people).

Backoff protocols are acknowledgment-based protocols. This means that they store information about their own transmissions. This information is the number of uninterrupted unsuccessful transmissions up to this moment. It is called *a backoff counter,* and denoted by b_i for station i. At first, the counter is equal to 0, and it increases after every unsuccessful attempt. The counter returns back to 0 after a successful transmission, and the message is removed from the top of the corresponding queue. The counter is not increased endlessly; at some moment of time it is stopped, and we decide that we cannot successfully send the current message, and discard it. In Ethernet the upper bound for the backoff counter is 16.

In general, in any backoff protocol, the contention window changes with the change of the backoff counter. The probability of sending at a time slot of the contention window (W_{b_i} for station i) is a function of the backoff counter (b_i for station i), and we call this probability *the backoff function*. We consider $f(b_i)$ as a probability, but not necessarily $\sum_{b_i} f(b_i) = 1$. At any moment of time the probability $f(b_i)$ defines uniform distribution for the next attempt to transmit when the backoff counter b_i is known. We can set the contention window size via $f(b_i) \leq 1$, as $W_i = f^{-1}(x)W_0 \geq 1$, where W_0 is the minimal contention window size and $f^{-1}(x) \stackrel{\text{def}}{=} \frac{1}{f(i)}$. For W_0 we use value 1 by default, when the opposite is not mentioned. Note, function $f^{-1}(x)$ not necessary gives integer numbers, in that case, we will define below somewhat modified uniform distribution more precisely.

We need to retransmit a message after every collision, or discard it. First of all, we increase the backoff counter, which represents the load of the system. If we know the backoff function for this counter, we can determine the contention window size. Then we take a random value from the contention window representing some time delay in the slots. This is the time that must elapse before the next transmission attempt. This random value we call *a backoff time*, and it is uniformly distributed on the contention window.

As an example, in Ethernet the backoff protocol is called the BEB, and $f(b_i) = 2^{-b_i}$, for $b_i \leq 10$, and $f^{-1}(b_i) = 1024$, for $b_i > 10$. As we mentioned before, after $M = 16$ we discard the packet.

B. Related work

BEB protocol has over 30 years of research history. Results that have been received, appear to contradict each other; some authors say that the protocol is stable, some say it is not. The result of analyses greatly depends on the used mathematical model, i.e. how they mathematically approximate the Ethernet protocol. Here, we are going to mention some of the most interesting research outcomes.

Kelly [8] (later with MacPhee [9]) showed that BEB protocol is unstable for $\lambda > 0.693$, and strongly stable for $\lambda < 0.567$ for infinite model ($N = \infty$). Furthermore, this author says that "the expected number of successful transmissions is finite for any acknowledgment based scheme with slower than exponential backoff". Then Aldous [1] with almost the same model found that all acknowledgment-based protocols are unstable. They used a queue-free infinite model. However, later Håstad et al. [6] discovered that a finite model (i.e. model with a finite number of stations) with queues is stable for polynomial backoff protocol and BEB protocol is unstable for

$$\lambda \geq \lambda_0 + \frac{1}{4N-2} \text{ with } \lambda_0 \approx .567.$$

Additionally, the definition of stability in [6] differs from the first authors. They define stability as the finiteness of expected time to return to zero (which is also called positive recurrence [6]) and finiteness of expected average total queue-size. The first two authors talk about stability in terms of throughput rate; several other results about the stability of Ethernet can be found in literature, see [10]–[14]. However, we are mostly interested in works of Bianchi [5] and Kwak et al. [4]. The models proposed by the latter authors seem to be the most reasonable. In [5] the throughput rate of wireless local area network is analyzed for a protocol which is close to Ethernet protocol. A similar model was considered in [4], where some results on the expected delay time and throughput for exponential backoff protocol from Ethernet network with a general factor r (in BEB protocol the factor r equals to 2) are obtained.

II. Analysis

Our analysis is based on the work of Kwak et al. [4] and Bianchi [5]. We use their model and now present some assumptions adopted from [4], [5].

A. Model with unbounded backoff counter

We have the following assumptions

- Our first assumption is that our system is in *a steady state.* It is a reasonable assumption for a big enough number of stations N. We assume that a large number of stations makes the network work uniformly. In other words, if the number of stations is large, a single station does not have a great effect on the performance of the system. On the other hand, for a small number of stations this assumption may be far from reality (it is possible, for example, that one station may capture the channel, what is called *a capture effect*). By this assumption at any moment of time we have the same probability p_c for the collision of a message sent to the medium.
- The second assumption is that all *stations are identical*, so the performance of every station is the same.
- The third assumption is that our model is under the *saturation condition*. Hence, there are always messages waiting in the input. Without saturation assumption the system might show "better" results, but this assumption lets us understand the worse case.
- The last assumption is that the *time is divided into time slots* of equal length. During every time slot we can send a message, the propagation time of the message is assumed to be equal to the time slot. Every message is synchronized to the time slot bounds. We know that if a collision has not happened during the first time slot for some large message (large means that the message transmission duration is longer then time slot duration),

then most likely, it will not happen in the remaining time slots of this message with high probability.

When a new packet is sent for the first time, the (initial) contention window size is W_0. After the first attempt, the time within which the transmission will be tried is delayed by an amount of time which is uniformly distributed over the set $\{1,\ldots,W_0\}$. Every time we have a collision we increase this delay set according to a backoff function $f(i)$, where $0 < f(i) < 1$ for $i > 0$ and we assume that $f(0) = 1$. After the i^{th} collision, the delay is distributed over $\{1,\ldots,\lceil f^{-1}(i)W_0\rceil\}$. The initial value W_0 can be interpreted as the multiplier value for function $f^{-1}(i)$ (we always see it as a multiplier). After a successful transmission the delay again is distributed in $\{1,\ldots W_0\}$.

In our model, the backoff counter specifies the state of each station. Let D_i be the time of staying in state i, called delay, thus we have the following formula for D_i:

$$\begin{aligned} Pr\{D_i = k\} &= \frac{1}{X_i} - \frac{Y_i}{X_i(X_i+1)} \\ &= \frac{X_i+1-Y_i}{X_i(X_i+1)}, k = 1,\ldots,X_i \\ Pr\{D_i = X_i+1\} &= \frac{Y_i}{X_i+1}, \end{aligned} \tag{1}$$

where $X_i = \lfloor f^{-1}(i)W_0\rfloor$ and $Y_i = f^{-1}(i)W_0 - X_i$. The construction above helps to deal with continuous backoff function, and it is applicable if

$$f^{-1}(i)W_0 \geq 1, \qquad \text{for all } i. \tag{2}$$

If $f^{-1}(i)W_0$ is an integer then equation (1) has the following (uniform) distribution,

$$Pr\{D_i = k\} = \frac{1}{f^{-1}(i)W_0}, k = 1,\ldots,f^{-1}(i)W_0.$$

Definition (1) is almost the same as in [4]; now X_i and Y_i are the integer and fractional parts not only for r^iW_0, but for $f^{-1}(i)W_0$ in general (in [4] $f(i) = \frac{1}{r^i}$).

Now we know how long we are going to stay in the state i. Next what we should do, is to find the probability P_i to succeed state i. The state model remains the same as in [4] (see Figure 1), hence the probability P_i is

$$P_i = (1-p_c)p_c^i, \tag{3}$$

where the collision probability p_c to be determined below.

B. System load

Let ED_i be the expected delay for state i. It then follows from (1) that

$$ED_i = \frac{W_i+1}{2}, \tag{4}$$

where $W_i = f^{-1}(i)W_0$. We know that we enter state i with probability P_i and stay in i for ED_i time in average. Thus, we can find the probability γ_i to be in state i at any instant. It corresponds to the fraction of time that system spends in this state in steady-state model:

$$\begin{aligned} \gamma_i &= \frac{ED_iP_i}{\sum_{j=0}^{\infty} ED_jP_j} = \frac{(W_i+1)(1-p_c)p_c^i}{\sum_{j=0}^{\infty}(W_j+1)(1-p_c)p_c^j} \\ &= \frac{(W_i+1)(1-p_c)p_c^i}{W_0(1-p_c)\sum_{j=0}^{\infty} f^{-1}(j)p_c^j + 1}. \end{aligned} \tag{5}$$

In general we cannot find the exact value of $\sum_{j=0}^{\infty} f^{-1}(j)p_c^j$, and furthermore we cannot expect that the sum of the series even converges on $0 < p_c < 1$. Let us define a new function

$$F(z) \stackrel{\text{def}}{=} \sum_{j=0}^{\infty} f^{-1}(j)z^j. \tag{6}$$

Denote by $\xi = \xi(p_c)$ the (random) number of successive collisions before the successful transfer, then

$$E\left[f^{-1}(\xi)\right] = \sum_{i=0}^{\infty} f^{-1}(i)P\{\xi = i\} = (1-p_c)F(p_c).$$

Note that we cannot consider $F(p_c)$ as a generating function, because of a dependence between p_c and the set of the backoff functions $\{f(i), i \geq 0\}$.

Substituting (6) into (5), we obtain a compact form of the equation for γ_i:

$$\gamma_i = \frac{(W_i+1)(1-p_c)p_c^i}{W_0(1-p_c)F(p_c)+1}. \tag{7}$$

It follows from [4] that the probability to be in state i with backoff timer equal to zero (the station is transmitting in state i) is exactly $\frac{\gamma_i}{ED_i}$, hence the transmission probability p_t at any instant is (see (4))

$$p_t = \sum_{i=0}^{\infty} \frac{\gamma_i}{ED_i} = \sum_{i=0}^{\infty} \frac{2(1-p_c)p_c^i}{W_0(1-p_c)F(p_c)+1}.$$

This immediately implies

$$p_t = \frac{2}{W_0(1-p_c)F(p_c)+1}. \tag{8}$$

Another dependence between p_t and p_c can be taken from [5]:

$$\begin{aligned} p_c &= P\{\text{collision}\} \\ &= 1 - P\{\text{no transmissions from other } N-1 \text{ stations}\} \\ &= 1 - (1-p_t)^{N-1}. \end{aligned} \tag{9}$$

So we obtain another equation connecting p_t and p_c:

$$p_t = 1 - (1-p_c)^{\frac{1}{N-1}}. \tag{10}$$

Combining (8) and (10) implies

$$\frac{2}{W_0(1-p_c)F(p_c)+1} = 1 - (1-p_c)^{\frac{1}{N-1}}. \tag{11}$$

Note that the right-hand side of (11) is 0, when $p_c = 0$, it is 1, when $p_c = 1$ and it monotonically increases with p_c. Let

$$G(p_c) \stackrel{\text{def}}{=} \frac{2}{W_0(1-p_c)F(p_c)+1}. \tag{12}$$

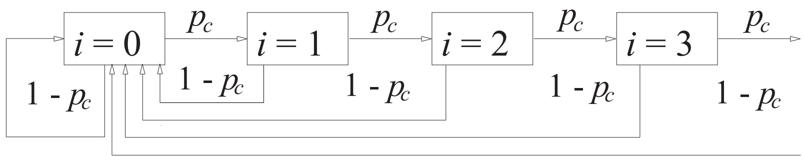

Fig. 1. State model.

Putting $p_c = 0$ in (12) and taking into account (2) we obtain

$$0 < G(0) = \frac{2}{W_0 f^{-1}(0) + 1} = \frac{2}{W_0 + 1} \leq 1.$$

To have a unique solution p_c of (11) the monotone decrease of function $G(p_c)$ (for $0 < p_c < 1$) is sufficient. To check this, we calculate derivative

$$\begin{aligned} G'(p_c) &= \left(\frac{2}{W_0(1-p_c)F(p_c)+1} \right)'_{p_c} \\ &= -\frac{2}{\left(W_0(1-p_c)F(p_c)+1\right)^2} \times \\ & \quad \left(-W_0 F\left(p_c\right) + W_0\left(1-p_c\right) F'_{p_c}\left(p_c\right) \right). \end{aligned}$$

As we can see, only the rightmost parentheses part in the equation above determines the sign of the derivative $G(p_c)'$. Recall that $F(p_c) \overset{\text{def}}{=} \sum_{j=0}^{\infty} f^{-1}(j) p_c^j$. Thus we have

$$\begin{aligned} W_0 & \left[-F\left(p_c\right) + \left(1-p_c\right) F'(p_c) \right] = \\ W_0 & \left[-\sum_{j=0}^{\infty} f^{-1}(j) p_c^j \right. \\ & \left. + \left(1-p_c\right) \sum_{j=0}^{\infty} (j+1) f^{-1}(j+1) p_c^j \right] = \\ W_0 & \left[-\sum_{j=0}^{\infty} f^{-1}(j) p_c^j + \sum_{j=0}^{\infty} (j+1) f^{-1}(j+1) p_c^j = \right. \\ & \left. -\sum_{j=0}^{\infty} (j+1) f^{-1}(j+1) p_c^{j+1} \right] = \\ W_0 & \left[-\sum_{j=0}^{\infty} f^{-1}(j) p_c^j + \sum_{j=0}^{\infty} (j+1) f^{-1}(j+1) p_c^j = \right. \\ & \left. -\sum_{j=0}^{\infty} j f^{-1}(j) p_c^j \right] = \\ W_0 & \left[\sum_{j=0}^{\infty} (j+1) \left(f^{-1}(j+1) - f^{-1}(j) \right) p_c^j \right]. \end{aligned}$$

From the last equations we have that condition $f^{-1}(i+1) \geq f^{-1}(i)$ for every i is enough to have non-increasing function G. Hence, if we have for at least one k that $f^{-1}(k+1) > f^{-1}(k)$ (in addition to condition $f^{-1}(i+1) \geq f^{-1}(i)$ for every i), then we have only one intersection. Note, that if $W_0 > 1$ then we need only condition $f^{-1}(i+1) \geq f^{-1}(i)$. This is a sufficient condition to have unique solution p_c satisfying (11).

Note that if we have $f(i) = d$ for all i (Aloha protocol), then the function G will be a horizontal line.

Now we resolve equation (11) in such a way to obtain $F(p_c)$:

$$F(p_c) = \frac{1 + (1-p_c)^{\frac{1}{N-1}}}{W_0(1-p_c)\left(1-(1-p_c)^{\frac{1}{N-1}}\right)}. \tag{13}$$

For $z \in (0,1)$, we introduce the function

$$L(z) \overset{\text{def}}{=} \frac{1 + (1-z)^{\frac{1}{N-1}}}{W_0(1-z)\left(1-(1-z)^{\frac{1}{N-1}}\right)}. \tag{14}$$

Thus, the solution of the equation $F(p_c) = L(p_c)$ gives us the value of p_c (See Figure 2). We can see that the faster we increase the resolution window, the smaller is the probability of collision. So by this graphics, BEB protocol seems to be better than, for example, polynomial as the contention window of exponential backoff increases faster than the contention window for polynomial. Later we will show that with small number of collisions (big contention window) the channel becomes more and more loaded. (More packets wait in the queue, what is called instability in [6]).

C. Expected transmission time

Next, we should find the average time for a message to leave the system successfully. In this model we do not have another choice, as we do not discard messages. For this reason we should introduce a new random variable. Let N_R be the random variable of state at which a node transmits successfully. In other words, it is the number of collisions before a successful transmission.

Obviously, the probability to transmit successfully exactly at the i^{th} state is

$$P_i = P(N_R = i) = (1-p_c)p_c^i \qquad i \geq 0. \tag{15}$$

Hence, the average number of collisions before transmission is

$$E[N_R] = \sum_{i=0}^{\infty} i P_i = \frac{p_c}{1-p_c}.$$

Let S be the service time of a message. That is the time since the first attempt to transmit up to the instant when the message leaves the system. (In other words, that is complete time of transmission of a message). Now we can compute the average service time ES of a message being in the top of the queue. Because variables N_R and D_i are independent, we can use the known property of conditional expectation to obtain

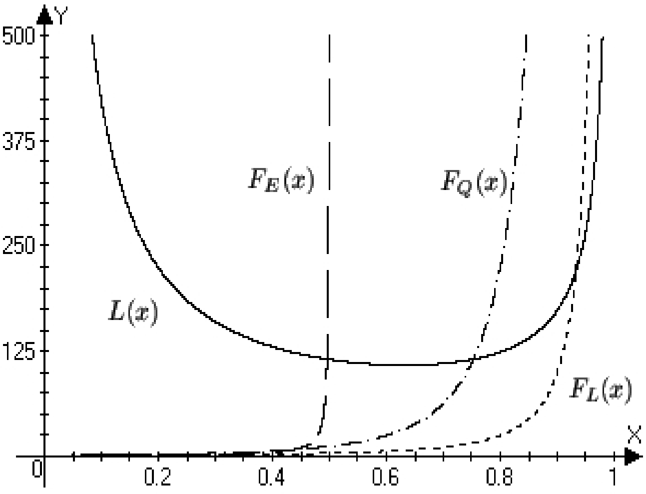

Fig. 2. Intersection points for equation $F(x) = L(x)$, where $F(x)$ is observed in particular cases. $F_Q(x) = \frac{1+x}{(1-x)^3}$ for quadratic polynomial function, $F_L(x) = \frac{1}{(1-x)^2}$ for linear function, and $F_E(x) = \frac{1}{1-2x}$ for BEB protocol.

$$\begin{aligned} ES &= E\left[\sum_{i=0}^{N_R} D_i\right] = E_{N_R}\left[E\left[\sum_{i=0}^{N_R} D_i \middle| N_R\right]\right] \\ &= E\left[\sum_{i=0}^{N_R} \frac{W_i+1}{2}\right] \\ &= \frac{W_0}{2} E\left[\sum_{i=0}^{N_R} f^{-1}(i)\right] + \frac{E[N_R]+1}{2}. \end{aligned} \tag{16}$$

Furthermore, let the indicator function for a boolean variable A be $\mathbb{1}(A)$

$$\mathbb{1}(A) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } A, \\ 0, & \text{otherwise.} \end{cases}$$

Using function $\mathbb{1}(A)$ and its property we find the value of the first term in the sum

$$\begin{aligned} E\left[\sum_{i=0}^{N_R} f^{-1}(i)\right] &= E\left[\sum_{i=0}^{\infty} f^{-1}(i)\mathbb{1}\{N_R \geq i\}\right] \\ &= \sum_{i=0}^{\infty} f^{-1}(i) P(N_R \geq i), \end{aligned}$$

where $\mathbb{1}(\cdot)$ is the indicator function. Recall that $P(N_R \geq i) = p_c^i$. Then (16) becomes

$$ES = \frac{1}{2}\left(W_0 \sum_{i=0}^{\infty} f^{-1}(i) p_c^i + \frac{1}{1-p_c}\right).$$

Thus, we have finally

$$ES = \frac{1}{2}\left(W_0 F(p_c) + \frac{1}{1-p_c}\right). \tag{17}$$

Now we insert (13) into (17) and obtain

$$ES = \frac{1}{(1-p_c)\left(1-(1-p_c)^{\frac{1}{N-1}}\right)}. \tag{18}$$

By easy algebra, (18) gets minimum at

$$p_c^* = 1 - \left(1 - \frac{1}{N}\right)^{N-1} \tag{19}$$

Recall the well-known limit

$$\left(1 - \frac{1}{N}\right)^N \xrightarrow{N\to\infty} e^{-1}. \tag{20}$$

Hence,

$$p_c^* = 1 - \left(1 - \frac{1}{N}\right)^{N-1} \xrightarrow{N\to\infty} 1 - e^{-1}, \tag{21}$$

and this simple expression gives us the optimal collision probability p_c^* as the number of stations N tends to infinity.

D. Stability condition

Let λ be the incoming rate for the system. We assume that the incoming message uniformly chooses the station in the network. Hence, the incoming rate for a single station is $\frac{\lambda}{N}$. The condition that the queue of the station decreases over time is $ES < \frac{N}{\lambda}$. This gives the following stability condition of the protocol:

$$\lambda < N(1-p_c)\left(1-(1-p_c)^{\frac{1}{N-1}}\right). \tag{22}$$

E. Optimality condition

Now we could clearly say what conditions should be set to have the best possible protocol (over the class of the backoff protocols). Optimal value (19) of collision probability $p_c^* = 1 - \left(1 - \frac{1}{N}\right)^{N-1}$ tends to $1 - e^{-1}$ as the number of stations N tends to infinity. Also, for this (optimal) value the maximal attainable throughput of the system is

$$\lambda^* = \sup\left\{\lambda : \lambda < \left(1 - \frac{1}{N}\right)^{N-1}\right\}, \tag{23}$$

which tends to e^{-1} as N tends to infinity. It then follows from (18) and (19) that optimal point $p_c = p_c^*$ gives the following minimal average service time

$$ES = \frac{N}{\left(1 - \frac{1}{N}\right)^{N-1}}. \quad (24)$$

This expectation tends to infinity as N tends to infinity.

Note that for an individual station this tendency means instability, if we have infinite number of stations. These results agree with previous results of other authors on stability of infinite model. In spite of tendency to infinity for individual stations the whole network has finite expected service time when $N \to \infty$. For us, it is most significant to know the service time of every station for some fixed finite parameter N (the number of stations), by which we can tune real network for the best performance.

Now we see from (13) that to achieve the optimal point, the backoff parameters shall satisfy the following condition

$$F(p_c^*) = \frac{2N-1}{W_0\left(1 - \frac{1}{N}\right)^{N-1}}. \quad (25)$$

It means that if we can find such a protocol (i.e. the set of backoff functions $\{f^{-1}(i)\}$) that (25) holds, then the protocol is the best over the class of backoff protocols in the sense of minimization of the average transmission time of a message.

F. Elimination the saturation conditions

We extend our analysis by omitting the saturation condition. To get it we change the model on the following, see Figure 3. In the new model almost everything remains the same except we add a new state -1 representing incoming messages. We assume that there is always some incoming rate ($\lambda > 0$). Hence there are always messages in the incoming queue, the only difference is that we need to wait for them for a certain delay. Let this time be the following random variable

$$D_{-1} \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } q > 0, \\ \tau, & \text{if } q = 0, \end{cases}$$

where τ is a random variable representing the time between incoming messages ($E[\tau] = \frac{1}{\lambda}$) and q is the number of waiting messages in the queue. We can write the time delay for incoming queue in the following way, $D_{-1} = \tau \cdot \mathbb{1}\{q = 0\}$. As we can see there is a tight dependence between τ and queue size q.

A technique of negative drift could help us to analyze the stability of this model. This technique states that if we outside some finite set have negative expected tendency for the change in the queue size then the system is positive recurrent (see [15]). (Also another condition is required to exclude an *infinite jump*.) To use this technique we define a random process $X(t) = \{q(t), b(t), l(t)\}$ for some station, where $q(t)$ is the number of waiting massages in the queue, $b(t)$ is the size of backoff counter, and $l(t)$ is the number of time slots remained till the end of the current contention window at some moment of time t (Note that $b(t)$ is 0 or $l(t)$ is 0 if $q(t)$ is 0 at some moment of time t). For us the recurrence is enough for the system to be stable. Let us define the finite set outside which we need the negative drift, as the set representing station with at least one waiting message in it. Due to this condition the station becomes saturated, and our model becomes identical to the already studied saturation model. The condition of the negative drift hence is identical to the inequality (22).

G. Model with bounded backoff counter

We can easily extend previous results on the model with an upper bound on the backoff counter (see Figure 4). In this model, if the backoff counter exceeds some value M then the message becomes discarded and we take a new one from the queue. Probability to discard a message is $P\{\text{discard}\} = p_c^{M+1}$. Now we shall recalculate the values for the new model, but some of them do not depend on the number of states and hence they remain the same. One of the unchanged values is the delay time D_i.

The probability to enter state i is to be modified, now we can find it as

$$P_i = \frac{(1-p_c)p_c^i}{1-p_c^{M+1}}. \quad (26)$$

Hence the probability to be in state i at any moment of time is

$$\begin{aligned} \gamma_i &= \frac{P_i ED_i}{\sum_{j=0}^M P_j ED_j} = \frac{(W_i+1)(1-p_c)p_c^i}{\sum_{j=0}^M (W_j+1)(1-p_c)p_c^j} \\ &= \frac{(W_i+1)(1-p_c)p_c^i}{W_0(1-p_c)\sum_{j=0}^M f^{-1}(j)p_c^j + 1}. \end{aligned} \quad (27)$$

Using the same arguments we find that

$$\begin{aligned} p_t &= \sum_{i=0}^M \frac{\gamma_i}{ED_i} = \sum_{i=0}^M \frac{2(1-p_c)p_c^i}{W_0(1-p_c)F_M(p_c) + 1 - p_c^{M+1}} \\ &= \frac{2(1-p_c^{M+1})}{W_0(1-p_c)F_M(p_c) + 1 - p_c^{M+1}}, \end{aligned} \quad (28)$$

where $F_M(p_c) = \sum_{i=0}^M f^{-1}(i)p_c^i$. Note that equation (8) is independent of the upper bound for backoff counter. Hence we can use it here. Combining (28) and (8) we have solution for $F_M(p_c)$:

$$F_M(p_c) = \frac{\left(1-p_c^{M+1}\right)\left(1 + (1-p_c)^{\frac{1}{N-1}}\right)}{W_0\left(1-p_c\right)\left(1 - (1-p_c)^{\frac{1}{N-1}}\right)}. \quad (29)$$

Applying almost the same formula for the service time (now we have a finite sum instead of an infinite sum) we have

$$\begin{aligned} ES &= E\left[\sum_{i=0}^{min\{M,N_R\}} D_i\right] \\ &= E_{N_R}\left[E\left[\sum_{i=0}^{min\{M,N_R\}} D_i \,\middle|\, N_R\right]\right] \\ &= E\left[\sum_{i=0}^{min\{M,N_R\}} \frac{W_i+1}{2}\right] \end{aligned}$$

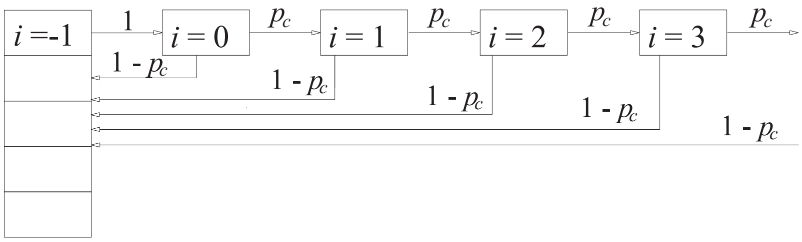

Fig. 3. State model without saturation condition.

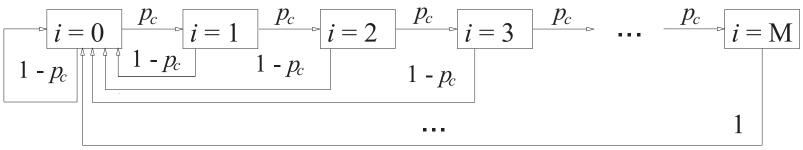

Fig. 4. State model with bounded counter.

$$= \frac{W_0}{2} E\left[\sum_{i=0}^{min\{M,N_R\}} f^{-1}(i)\right] + \frac{E[N_R]+1}{2}, \quad (30)$$

where similar computation for the last component is possible by virtue of the same random variable N_R.

$$\begin{aligned} E\left[\sum_{i=0}^{min\{M,N_R\}} f^{-1}(i)\right] &= E\left[\sum_{i=0}^{M} f^{-1}(i)\mathbb{1}\{N_R \geq i\}\right] = \\ &= \sum_{i=0}^{M} f^{-1}(i)P(N_R \geq i) \\ &= \sum_{i=0}^{M} f^{-1}(i)p_c^i = F_M(p_c). \end{aligned}$$

Combining again the last equations we have

$$ES = \frac{1}{2}\left(W_0 F_M(p_c) + \frac{1}{1-p_c}\right). \quad (31)$$

But we already found $F_M(p_c)$ for (31), hence

$$ES = \frac{\left(1-p_c^{M+1}\right)}{(1-p_c)\left(1-(1-p_c)^{\frac{1}{N-1}}\right)}, \quad (32)$$

which is equal to (17) when $M = \infty$. The negative drift condition for the bounded model will be

$$\lambda < \frac{N(1-p_c)\left(1-(1-p_c)^{\frac{1}{N-1}}\right)}{\left(1-p_c^{M+1}\right)}. \quad (33)$$

When $M = \infty$ (33) gives (22).

III. Application to the Ethernet case

Now we try to apply these results to the real Ethernet protocol (See Section I-A). In addition, we present two exponential protocols that seem to show better performance in mathematical models. We will probe these protocols in cases, when the number of stations is 11, 51, 101, 501 and 1001. In the end we will give an outline of the estimated performance for these cases.

In the introduction we said that Ethernet is a bounded BEB protocol with $M = 16$. Hence, for the Ethernet policy we have

$$\begin{aligned} F_E^{16}(p_c) &= \sum_{i=0}^{10} 2^i p_c^i + \sum_{i=11}^{16} 2^{10} p_c^i \\ &= \frac{1-(2p_c)^{11}}{1-2p_c} + 2^{10} p_c^{11} \frac{1-p_c^6}{1-p_c}. \end{aligned}$$

Additionally, we consider another (exponential) set of backoff functions

$$F_a^M(x) = \sum_{i=0}^{M} a^i x^i = \frac{1-(ax)^{M+1}}{1-ax}.$$

Especially, we are interested in some sets of functions $F_a^{16}(x)$ (particularly $F_{2.1}^{16}(x)$ and $F_{2.4}^{16}(x)$). See Figure 5 to understand the behavior of the functions.

In the table III we give comparative data on the behavior of the network depending on the protocol and the number of stations. In the table, the leftmost column shows the number of the stations, the next row shows the point of intersection p_c, after that column shows the average service time for the network, and the last column shows the probability of discarding for that protocol. Every cell has 3 numbers, separated by '/', these numbers are correspondingly data for $F_{2.4}^{16}(x)$, $F_{2.1}^{16}(x)$, and $F_E^{16}(x)$.

From the table we derive that Ethernet protocol is better to use for a small number of stations (something like 10 stations), on the other hand the protocol $F_{2.1}^{16}(x)$ is better to use for 51, 101 stations, and the protocol $F_{2.4}^{16}(x)$ is a good choice for 501, 1001 active stations. Also the existing Ethernet protocol is better not to use if the number of stations is greater than 500, because the probability that your message will be discarded is high. In real network, where some stations may be almost "silent", the number of active stations much lower than the actual number of the stations in the network.

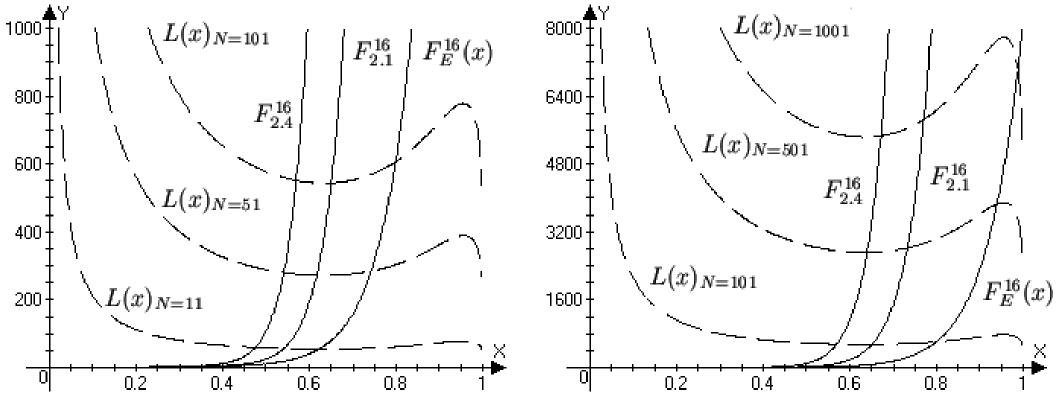

Fig. 5. Collision probability for Ethernet, where $L(x)$ plots for 11, 51, 101, 501 and 1001 stations, $F_E^{16}(x)$ is function for Ethernet, $F_{2.1}^{16}$, $F_{2.4}^{16}$ are exponential backoff functions (with parameter $a = 2.1$ and $a = 2.4$, respectively) with at most 16 attempts to transmit (as for Ethernet).

Number of stations	Backoff function	p_c	$\frac{ES}{N}$	$P\{\text{discard}\}$
11	$F_{2.4}^{16}(x)$	0.48	2.77	$3 * 10^{-6}$
11	$F_{2.1}^{16}(x)$	0.54	2.64	$3 * 10^{-5}$
11	$F_{E}^{16}(x)$	0.62	2.59	$3 * 10^{-4}$
51	$F_{2.4}^{16}(x)$	0.54	2.76	$3 * 10^{-5}$
51	$F_{2.1}^{16}(x)$	0.62	2.69	$3 * 10^{-4}$
51	$F_{E}^{16}(x)$	0.74	2.83	$6 * 10^{-3}$
101	$F_{2.4}^{16}(x)$	0.57	2.74	$6 * 10^{-5}$
101	$F_{2.1}^{16}(x)$	0.65	2.71	$7 * 10^{-4}$
101	$F_{E}^{16}(x)$	0.80	3.02	0.022
501	$F_{2.4}^{16}(x)$	0.64	2.72	$5 * 10^{-4}$
501	$F_{2.1}^{16}(x)$	0.73	2.82	$5 * 10^{-3}$
501	$F_{E}^{16}(x)$	0.94	3.86	0.349
1001	$F_{2.4}^{16}(x)$	0.67	2.73	$1.2 * 10^{-3}$
1001	$F_{2.1}^{16}(x)$	0.77	2.93	0.012
1001	$F_{E}^{16}(x)$	0.99	3.52	0.809

Fig. 6. Numeric results for bounded backoff protocols $F_{2.4}^{16}(x)$, $F_{2.1}^{16}(x)$, and $F_E^{16}(x)$.

IV. Discussion and Future Work

So far we have focused on the binary backoff protocol in the traditional Ethernet network where medium is shared through collision resolution. The modern Ethernet standards are based on switching frames rather than allowing frames to collide and launching the backoff protocol on stations. Only stations residing in a single collision domain in Ethernet network, that is connected to a hub, have collisions.

Two important areas where the backoff protocol is still used is congestion control in transport protocols and medium access control in wireless networks. The stability of Internet is largely due to the voluntarily use of additive increase – multiplicative decrease TCP congestion window combined with packet retransmissions following the backoff protocol [16]. This approach is copied to several other transport protocols, including Stream Control Transport Protocol (SCTP) and TCP Friendly Rate Control (TFRC).

In wireless networks, most importantly IEEE 802.11 WLAN, stations still compete for medium access through collisions. One important difference to wireline Ethernet is that once the wireless station started transmitting a frame, it cannot detect a collision until the transmission has completed. Therefore, the price of collisions is higher than in Ethernet and the MAC protocol performs collision avoidance. This area is of more practical importance today since the use of WLAN is rapidly growing while CSMA/CD Ethernet deployment is shrinking. Our backoff model will be significantly affected by the new assumptions. Its derivation and evaluation is a subject of our current work.

The use of constant two in increasing the interval between TCP retransmissions is mostly motivated by the easiness of implementation. The multiplication by two can be implemented as a simple bit shift operation, there as multiplication by a real number requires many more CPU cycles. However, this argument is largely obsolete now as the computational capabilities of modern process had significantly increased.

An interesting scenario appears when the stations do not follow the voluntary TCP congestion control, but attempt to behave selfishly to maximize own throughput. Recent work evaluated the impact of congestion control parameters on various TCP flavors [17]. However, the paper mostly concentrated on the effect of additive increase and multiplicative decrease coefficients, giving little consideration to the effect of backoff behavior. During the network overload when all stations are transmitting aggressively, the backoff conditions are likely to prevail since the TCP protocol would need to recover many lost packets.

We plan to develop a game-theoretic model of selfish backoff strategy and search for equilibrium conditions. The results will be evaluated with ns-2 simulations with TCP modified to use the optimal backoff constants.

V. Conclusions

We have found stability conditions for the steady state models of backoff protocols. These conditions were obtained both for the bounded and for the unbounded retry limit models. Consequently, we can analytically measure the throughput of the system and the service time. We have also found the optimality conditions. The question of the optimality is still open, but for our model (unbounded retry limit) we prove that exponential function is the best choice for the backoff function. In the paper we present graphics that show the correlation between the level function $L(x)$ and the "extended" backoff function $F(x)$. Moreover, from the graphic in Figure 2 we can see why sub-linear and super-exponential functions are not good choices. Finally, we show the "connection" of the stability for bounded backoff and the successful throughput rate $(1 - P\{\text{discard}\})$.

In this paper, we present analytical solutions, but some questions still remain open. For example, the optimality (for different M) of the general bounded backoff (particularly Ethernet) and the appropriateness of the steady state assumption. A simulation would help to answer the last question, but at the present there are good reasons for supposing that this assumption is appropriate.

A. Acknowledgment

Andrey Lukyanenko would like to thank Prof. Martti Penttonen, who introduced him to the problem arising in Ethernet and helped a lot with editing of this article. Andrey thanks Prof. Evsey Morozov for many valuable comments which have improved presentation of the paper and attracting his attention to the principle of negative drift, and an excellent book of Meyn and Tweedie.

References

[1] D. J. Aldous, "Ultimate Instability of Exponential Back-Off Protocol for Acknowledgement-Based Transmission Control of Random Access Communication Channel", *IEEE Trans. on Information Theory*, IT-33(2), pp. 219-223, 1987.

[2] N. Abramson, "The ALOHA system-another alternative for computer communications", *AFIPS*, vol. 37, pp. 281-285, 1970.

[3] N. Abramson, "Development of the ALOHANE", *IEEE Trans. on Inform. Theory*, vol. 31(2), pp. 119-123, 1985.

[4] B. Kwak, N. Song, and L. E. Miller, "Performance analysis of exponential backoff", *IEEE/ACM Trans. Netw.* vol. 13(2), pp. 343-355, 2005.

[5] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE J. on Sel. Areas in Commun.*, vol. 18(3), pp. 535-547, 2000.

[6] J. Håstad, T. Leighton, and B. Rogoff, "Analysis of backoff protocols for multiple access channel", SIAM J. Comput,. vol 25, no. 4, pp. 740-774, 1996.

[7] R. Metcalfe, D. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks", *Communications of the ACM*, vol. 19(7), pp. 395-404, July 1976.

[8] F. P. Kelly, "Stochastic models of computer communication systems", *J. Roy. Statist. Soc. B*, vol.47., pp. 379-395, 1985.

[9] F.P. Kelly and I.M. MacPhee, "The number of packets transmitted by collision detect random access scheme", *Annals of Prob.*, vol. 15, pp. 1557-1568, 1987.

[10] J. Goodman, A.G. Greenberg, N. Madras and P. March, "Stability of binary exponential backoff", *J. of the ACM*, vol. 35(3), pp. 579-602, 1988.

[11] G. Fayolle, P. Flajolet and M. Hofri, "On a functional equation arising in the analysis of a protocol for a multi-access broadcast channel", *Adv. Appl. Prob.*, vol. 18, pp. 441-472, 1986.

[12] W.A. Rosenkrantz, "Some theorems on the instability of the exponential back-off protocol", in Proc. of *Performance '84*, pp. 199-205, 1985.

[13] L. A. Goldberg and P. MacKenzie, "Analysis of Practical Backoff Protocols for Contention Resolution with Multiple Servers", *J. of Comp. and System Sciences*, vol. 58, pp. 232-258, 1999.

[14] L. A. Goldberg, P. MacKenzie, M. Paterson and A. Srinivasan, "Contention Resolution with Constant Expected Delay", *Journal of the ACM (JACM)*, vol. 47(6), pp. 1048-1096, 2000.

[15] S. P. Meyn and R. L. Tweedie, "Markov Chains and Stochastic Stability", *Springer-Verlag*, London, 1993.

[16] V. Jacobson, M. J. Karels, "Congestion Avoidance and Control", in Proc. of ACM SIGCOMM, 1998.

[17] A. Akella, S. Seshan, R. Karp, S. Shenker, "Selfish Behavior and Stability of the Internet: A Game-Theoretic Analysis of TCP", in Proc. of ACM SIGCOMM, 2002.

Andrey Lukyanenko received his M.Sc. degree in Computer Science from the University of Kuopio, Finland in 2005 and the Petrozavodsk State University, Russia in 2005. For half a year he worked as a Senior Programmer and Team Leader at the PetrSU Metso Automation System Center, Petrozavodsk, Russia. He started his Ph.D. studies at the University of Kuopio in 2006; now he is a researcher at the Helsinki Institute for Information Technology focusing on Queuing and Game Theory for next generation Internet architecture in TrustInet project.

Andrei Gurtov received his M.Sc. and Ph.D. degrees in Computer Science from the University of Helsinki, Finland in 2000 and 2004. At the present, he is Principal Scientist leading the Networking Research group at the Helsinki Institute for Information Technology focusing on the Host Identity Protocol and next generation Internet architecture. He is co-chairing the IRTF research group on HIP and teaches as an adjunct professor at Telecommunications and Multimedia Laboratory of the Helsinki University of Technology. Previously, his research focused on the performance of transport protocols in heterogeneous wireless networks. In 2000-2004, he served as a senior researcher at TeliaSonera Finland contributing to performance optimization of GPRS/UMTS networks, intersystem mobility, and IETF standardization. In 2003, he spent six months as a visiting researcher in the International Computer Science Institute at Berkeley working with Dr. Sally Floyd on simulation models of transport protocols in wireless networks. In 2004, he was a consultant at the Ericsson NomadicLab. Dr. Gurtov is a co-author of over 35 publications including research papers, patents, and IETF RFCs.