Interleaving Collision Resolution Engines in n-ary Tree Protocols

Wei-Ming Yin and Ying-Dar Lin, *Member, IEEE*

***Abstract*—N-ary tree protocols are used for access control on shared channels to resolve collisions among channel users. For exercising a single collision resolution engine (CRE), throughput and latency of such protocols have been comprehensively studied. This study investigates the same protocols with multiple interleaved CREs in slotted networks. $Power$, defined as the ratio of throughput over latency, is measured as the criteria to launch or terminate CREs. Analysis results indicate that the appropriate number of CREs to trigger depends on the traffic load and the collision resolution protocol. As the load grows to 0.25, 0.35, 0.5, and 0.8, the optimal number of interleaved CREs becomes 2, 3, 5, and 8, respectively. Moreover, the power of n-ary tree protocol with n determined dynamically outperforms the 3-ary tree protocol by 13%.**

***Index Terms*—Collision resolution engine, interleaving, n-ary tree.**

I. Introduction

A CONTENTION resolution process consists of two phases [1]. First, the initial resolution phase in which newly activated users follow a first transmission rule (FTR) to send their packets. Second is the collision resolution phase in which the users retransmit collided packets according to a retransmission rule (RTR). FTR's can be further classified into two types, free-access and blocked-access. The former allows newly activated users to contend immediately for slots while the latter forbids this and is investigated herein. N-ary tree protocols [2] are RTR, which control the slot access of shared channels among retransmission users. A collision resolution engine (CRE) exercises the contention resolution process by allocating a group of slots to resolve collisions.

To reduce the packet access latency while adopting blocked-access protocols, another group of slots may be allocated to process the blocked requests, i.e., triggering another CRE, as shown in Fig. 1. In Fig. 1, user$_A$ has a newly arriving packet at T_1 and will burst the packet at T_3 while only exercising one CRE. If there are two CREs exercised and interleaved, the packet will be transmitted at T_2, which results in shorter packet access latency. However, with multiple CREs, fewer users are involved in a CRE and the system produces lower slot throughput owing to higher estimation error ratio in light traffic load [3].

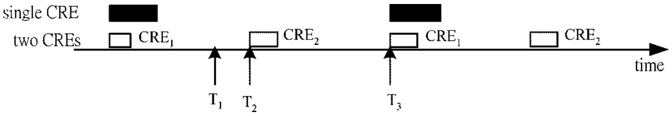

Fig. 1. Interleaving two collision resolution engines.

II. Problem Statement

This work seeks to determine the number of CREs required, given traffic load, to maximize the performance. Since more CREs results in shorter latency but lower slot throughput, and vice versa, neither minimizing latency nor maximizing slot throughput is appropriate. Therefore, the $power$ is introduced and is, given traffic load λ and the number of CREs m, defined as

$$\varepsilon_{\lambda,m} = \frac{\sigma_{\lambda,m}}{\delta_{\lambda,m}} \tag{1}$$

where $\sigma_{\lambda,m}$ and $\delta_{\lambda,m}$ represent the corresponding slot throughput and latency, respectively. The normalized slot throughput is the mean number of user transmissions that a slot successfully resolves, whereas the latency here is the mean number of rounds to resolve a user transmission. Therefore, when slot throughput and latency are derived the number of CREs that maximizes the power could be obtained. Consequently, a policy which dynamically varies the number of CREs based on traffic load could be devised.

III. Analysis Work

In this study, our analysis focuses on n-ary tree protocols. First, the slot throughput and the latency are derived given the number of users and one CRE. The derivation first establishes the recurrence equations for the allocated contention slots and accumulated rounds, given a number of users collided initially. Then, the moment generating function provides a systematic approach to solve these equations [2] and the solutions are used to calculate the slot throughput and the latency, respectively. Thereafter, an equation that binds the traffic load and the number of users is established. Consequently, the power with respect to the traffic load is obtained. The analysis for m interleaving CREs can be obtained by substituting some parameter.

A. Single Collision Resolution Engine

Total Number of Rounds for Resolving R Users, L_R: Denote the random number $\overset{\frown}{X_U}$ as the number of accumulated rounds

---

Manuscript received May 4, 2001. The associate editor coordinating the review of this letter and approving it for publication was Dr. I. S. Venieris. This work was supported in part by the MOE Program of Excellence Research under 89-E-FA04-1-4, Taiwan, R.O.C.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Publisher Item Identifier S 1089-7798(01)11062-8.

to resolve U users that initially contend for the same slot. This produces

$$\widetilde{X_U} = U + \sum_{j=1}^{Q} \widetilde{X_{I_j}},$$

where U users are split into Q subtrees and the random variable I_j denotes the number of users contend for the jth slot, thereby $U = \sum_{i=1}^{Q} I_j$. The moment generating function of $\widetilde{X_U}$ is thus derived as

$$\begin{aligned} G_U(s) &\equiv \sum_{k=0}^{\infty} Pr\{X_U = k\} * s^k = E[s^{X_U}] = E[s^X|U] \\ &= E\left[E\left[s^{1+\sum_{j=1}^{Q} X_{I_j}}\right]\middle| U\right] \\ &= sE[G_{I_1}(s)G_{I_2}(s)\cdots G_{I_Q}(s)] \\ &= s\sum_{i_1\cdots i_Q}^{U} \binom{U}{i_1 \cdots i_Q} \prod_{j=1}^{Q} p_j^{i_j} G_{i_j}(s) \end{aligned}$$

where p_j is the probability of choosing jth subtree. Its mean is calculated by taking the first derivative of $G_U(s)$ with respect to s and evaluating it at $s = 1$, which yields

$$\begin{aligned} X_U &\equiv \left.\frac{dG_U(s)}{ds}\right|_{s=1} \\ &= \left(U + \sum_{u=1}^{\infty} \beta_{U,u}\Psi_{U,u}\right)\left(1 + \sum_{u=1}^{\infty} \beta_{U,u}\left(\frac{1}{u^{U-1}}\right)\right)^{-1} \end{aligned} \tag{2}$$

where $\beta_{U,u}$ indicates the probability of the estimated number of users is u given the actual number of users is U in the n-ary tree protocol and $\Psi_{U,u}$ is given as

$$\Psi_{U,u} = \sum_{j=1}^{u}\sum_{i_j=0}^{U-1} \binom{U}{i_j} \left(\frac{1}{u}\right)^{i_j} \left(1 - \frac{1}{u}\right)^{U-i_j} X_{i_j}.$$

The initial conditions are $X_0 = 0$ and $X_1 = 1$. Therefore, if R users participate in a contention resolution cycle, the mean number of total rounds summed for R users, L_R, is given as

$$L_R = \sum_{r=1}^{\infty} \alpha_{R,r} \left(\sum_{j=1}^{r}\sum_{j=0}^{R-1} \binom{R}{j} \left(\frac{1}{r}\right)^{j} \left(1 - \frac{1}{r}\right)^{R-j} X_j\right) \tag{3}$$

where $\alpha_{R,r}$ indicates the probability of the estimated number of users is r given the actual number of users is R in the initial contention phase. Note that the initial condition is $L_0 = 0$.

Total Number of Slots for Resolving R Users, A_R: Similarly, the random number $\widetilde{Y_U}$ is denoted as the number of slots allocated to resolve U users collided initially in a slot. This produces

$$\widetilde{Y_U} = 1 + \sum_{j=1}^{Q} \widetilde{Y_{I_j}},$$

where U users are split into Q subtrees and the random variable I_j denotes the number of users contend for the jth slot, thereby $U = \sum_{i=1}^{Q} I_j$. According to the above, its mean is

$$Y_U = \left(1 + \sum_{u=1}^{\infty} \beta_{U,u}\Omega_{U,u}\right)\left(1 + \sum_{u=1}^{\infty} \beta_{U,u}\left(\frac{1}{u^{U-1}}\right)\right)^{-1} \tag{4}$$

where $\Omega_{U,u}$ is given as

$$\Omega_{U,u} = \sum_{j=1}^{u}\sum_{i_j=0}^{U-1} \binom{U}{i_j} \left(\frac{1}{u}\right)^{i_j} \left(1 - \frac{1}{u}\right)^{U-i_j} Y_{i_j}.$$

Therefore, the total number of slots allocated to resolve R users participating in a contention resolution cycle is

$$A_R = \sum_{r=1}^{\infty} \alpha_{R,r} \left(\sum_{j=1}^{r}\sum_{j=0}^{R-1} \binom{R}{j} \left(\frac{1}{r}\right)^{j} \left(1 - \frac{1}{r}\right)^{R-j} Y_j\right). \tag{5}$$

B. Multiple Collision Resolution Engines

Regarding interleaving CREs, without loss of generality, the population of channel users is assumed to be distributed uniformly among all CREs. Therefore, the latency and the slot throughput given R users and m CREs can be calculated as $\delta_m(R) = L_{\lceil R/m \rceil}/\lceil R/m \rceil$ and $\sigma_m(R) = \lceil R/m \rceil / A_{\lceil R/m \rceil}$, respectively. Since the number of users R could be obtained from

$$R = \frac{C}{\Lambda} * \lambda * \Gamma_\lambda$$

where C, Λ, λ, and Γ_λ represent *channel capacity, mean packet size, traffic load,* and *mean cycle length,* respectively, the power is then calculated as

$$\varepsilon_{\lambda,m} = \frac{\sigma_{\lambda,m}}{\delta_{\lambda,m}} = \frac{\sigma_m\left(\frac{C}{\Lambda} * \lambda * \Gamma_\lambda\right)}{\delta_m\left(\frac{C}{\Lambda} * \lambda * \Gamma_\lambda\right)}. \tag{6}$$

IV. NUMERICAL RESULTS

This work investigates the power of n-ary tree protocol with n fixed to 3 or computed according to statistically optimized minislot allocation (SOMA) [3] which is an n-ary tree protocol with n dynamically determined based on the estimated number of collided users. Consistent with $\alpha_{R,r}$, $\beta_{U,u}$, and Γ_λ obtained from [3], C and Λ are hereby given as 6 Mb/s and 368.1 bytes.

Since exercising more CREs during high traffic load leads to fewer users that are resolved in a CRE, the latency is probably shortened and higher power is thereby produced. Figs. 2 and 3 demonstrate that additional CREs should be triggered as the load increases. However, triggering more CREs does not necessarily imply improved power. In fact, given traffic load, the number of CREs required to achieve maximal power can be determined from our analysis, and varies among collision resolution protocols. In general, when the load is below 0.2, one CRE would be the best. As the load grows to 0.25, 0.35, 0.5, and 0.8, it would be better to have 2, 3, 5, and 10 CREs, respectively. This observation could be applied to any collision resolution protocols whose slot throughput and latency compromise each other. Moreover, the power of the n-ary tree protocol with n determined dynamically averagely outperforms, thanks to its flexibility, the 3-ary tree protocol by 13%.

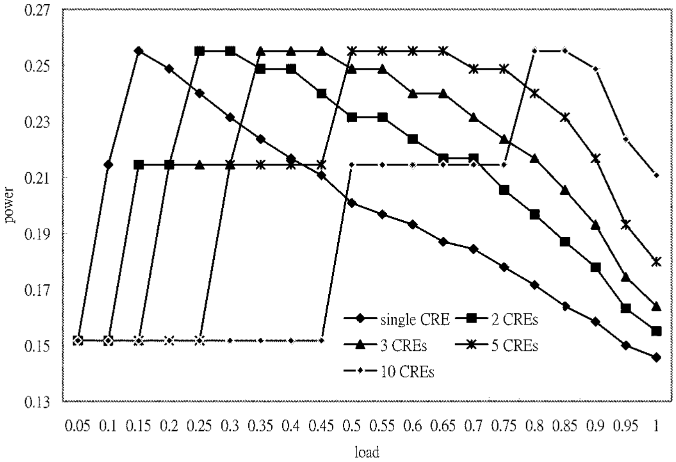

Fig. 2. Power of 3-ary tree protocol.

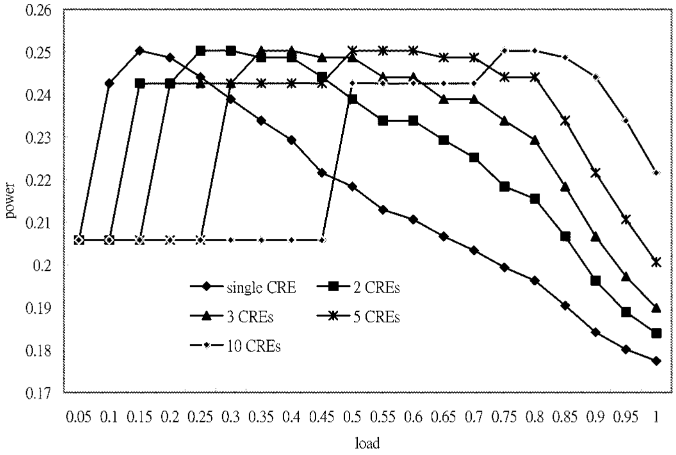

Fig. 3. Power of n-ary tree protocol with n dynamically determined.

V. CONCLUSION

The performance of n-ary tree protocols interleaving multiple CREs was investigated herein. Performance is measured with the power that considers both slot throughput and latency. The analysis based on a systematic approach indicates that the appropriate number of CREs to be trigger depends on the traffic load and varies with collision resolution protocols. This observation could be applied to any collision resolution protocols whose slot throughput and latency compromise each other. Moreover, the n-ary tree protocol with n dynamically determined outperforms the 3-ary tree protocol by 13% owing to its flexibility.

REFERENCES

[1] N. Golmie, Y. Saintillan, and D. Su. (1999) A review of contention ‘resolution algorithms for IEEE 802.14 networks. *IEEE Commun. Surveys* [Online]. Available: http://www.comsoc.org/pubs/surveys

[2] P. Mathys and P. Flajolet, “Q-ary collision resolution algorithms in random-access systems with free and blocked channel access,” *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 217–243, Mar. 1985.

[3] W. M. Yin and Y. D. Lin, “Statistically optimized minislot allocation for initial and collision resolution in hybrid fiber coaxial networks,” *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1764–1773, Sept. 2000.