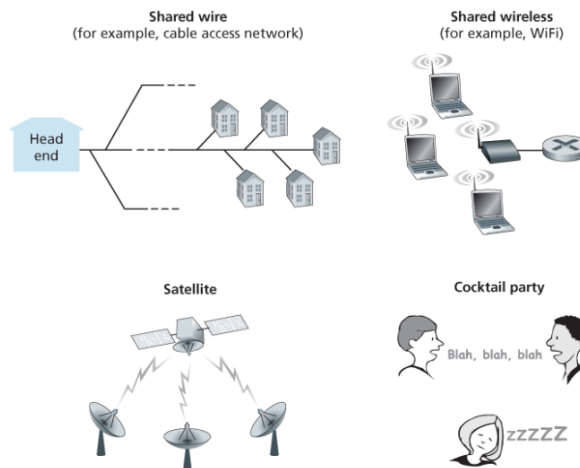


## 6.3 다중 접속 링크와 프로토콜

### 두 종류의 네트워크 링크

- **점대점 링크(point-to-point link)**
  - 링크의 한쪽 끝에 한 송신자와 링크의 다른 쪽 끝에 한 수신자가 있다.
  - PPP(point-to-point protocol)과 HDLC(high-level data link control)이 여기에 속한다. (뒤에 다룬다.)
- **브로드캐스트 링크(broadcast link)**
  - 동일한 하나의 공유된 브로드캐스트 채널에 다수의 송신 노드 및 수신 노드가 연결된다.
  - 임의의 한 노드가 프레임을 전송하면 채널이 그 프레임을 브로드캐스트해서 다른 모든 노드가 그 프레임의 복사본을 수신하기 때문에 브로드캐스트 용어가 쓰인다.

### 다양한 다중 접속 채널



### 다중 접속 문제(multiple access problem)

모든 노드가 프레임을 전송할 수 있으므로 2개 이상의 노드가 브로드캐스트 채널에서 직접 통신할 수 있고, 이런 일이 발생하면 모든 노드는 동시에 여러 개의 프레임을 받게 된다.

즉, 전송된 프레임들이 각 수신자에서 충돌하게 되고 어떤 수신 노드도 전송된 프레임의 의미를 파악할 수 없게 된다.

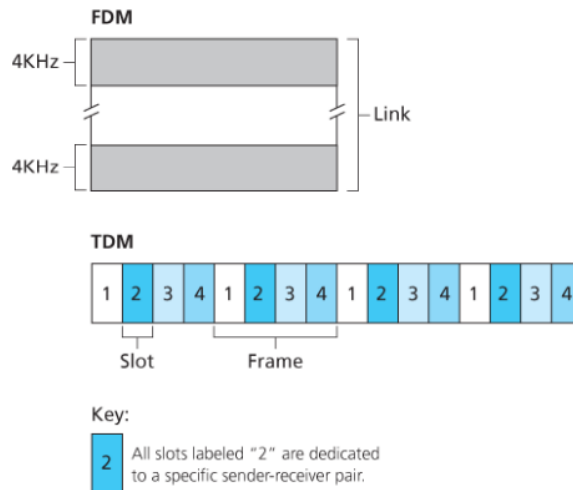
따라서 충돌에 관련된 모든 프레임은 손실되며, 다수의 노드가 빈번히 프레임을 전송하려 한다면 많이 충돌할 것이고 따라서 브로드캐스트 채널의 대역폭이 많이 낭비된다.

### 다중 접속 프로토콜(multiple access protocol)

초당 R 비트의 전송률을 갖는 브로드캐스트 채널에 대한 다중 접속 프로토콜은 다음과 같은 특성을 지니는 것이 바람직하다.

1. 단 하나의 노드가 전송할 데이터가 있을 때는 그 노드가  $R$  bps의 처리율을 갖는다.
2.  $M$ 개의 노드가 전송할 데이터가 있을 때는 각 노드가  $R/M$  bps의 처리율을 갖는다.
  - 항상이 아니며 각 노드가 정의된 시간 동안  $R/M$ 의 평균 처리율을 가짐을 의미한다.
3. 분산되어 있어 고장으로 인해 전체 시스템을 정지시킬 수 있는 마스터 노드가 없다.
4. 단순해서 구현하는 데 비용이 적게 든다.

### 6.3.1 채널 분할 프로토콜



#### 시분할 다중화(time-division multiplexing, TDM)

채널이  $N$ 개 노드를 지원하고 채널 전송률이  $R$  bps라고 하자.

TDM은 시간을 **시간 프레임(time frame)**으로 나누고 또한 각 시간 프레임을  $N$ 개의 **시간 슬롯(time slot)**으로 나눈다.

그 후  $N$ 개의 노드에게 시간 슬롯을 각각 할당한다.

노드는 전송할 패킷이 있을 때마다 **TDM 프레임에서 자신에게 할당된 시간 슬롯 동안 패킷을 전송한다.**

#### 장점

- 충돌을 제거할 수 있다.
- 매우 공정하다.

#### 단점

- 전송할 패킷이 있는 노드가 단 하나인 경우에도 노드 전송률이  $R/N$ 으로 제한된다.
- 노드가 전송 순서상 자신의 차례를 항상 기다려야 한다.

#### 주파수 분할 다중화 (frequency-division multiplexing, FDM)

$R$  bps의 채널을  $R/N$ 의 대역폭을 갖는 다른 주파수로 나눠서 각 주파수를  $N$ 개의 노드 중 하나에게 할당한다.

즉, 하나의 큰  $R$  bps 채널로부터  $N$ 개의  $R/N$  bps의 작은 채널을 만든다.

TDM과 같은 장단점을 갖는다.

## 코드 분할 다중 접속(code division multiple access, CDMA)

CDMA는 다른 코드를 각 노드에게 할당한다.

노드는 전송하는 데이터 비트들을 자신의 유일한 코드로 인코딩한다.

### 장점

- CDMA 네트워크에서 코드들을 신중하게 선택하면 여러 노드들이 동시에 전송할 수 있다.
- 다른 노드들에 의해 전송이 간섭되더라도 각 수신자들이 송신자의 인코딩된 데이터 비트를 정확하게 수신할 수 있다.

## 6.3.2 랜덤 접속 프로토콜

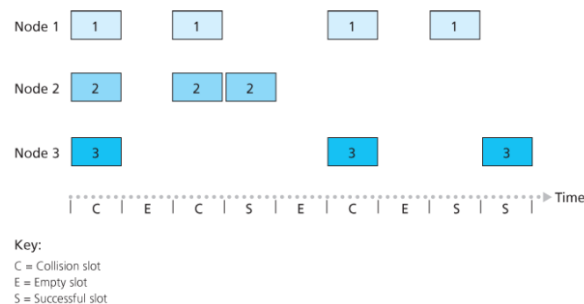
랜덤 접속 프로토콜에서 전송 노드는 항상 채널의 최대 전송률인  $R$  bps로 전송한다.

충돌이 생기면 충돌과 관련된 각 노드는 프레임이 충돌 없이 전송될 때까지 자신의 프레임을 계속해서 재전송한다.

프레임이 충돌했을 때 즉시 재전송하지 않고, 랜덤 지연 시간 동안 기다린 후 재전송 한다.

즉, 출동했던 노드 중 하나는 다른 노드가 선택한 지연 시간보다 충분히 작은 지연시간을 선택함으로써 충돌 없이 자신의 프레임을 채널로 전송할 수 있다.

### 슬롯 알로하(slotted ALOHA)



### 동작 과정

1. 전송할 새 프레임이 있으면 다음 슬롯이 시작할 때까지 기다렸다가 그 슬롯에 전체 프레임을 전송한다.

2. 만약, 충돌하지 않으면 노드는 성공적으로 자신의 프레임을 전송한 것이다. 따라서 그 프레임을 재전송할 필요가 없다.
3. 만약 충돌하면, 노드는 그 슬롯이 끝나기 전에 충돌을 검출한다. 노드는 그 프레임이 충돌 없이 전송될 때까지 확률  $p(0 \sim 1)$  사이로 해당 프레임을 다음 슬롯들에서 재전송한다.
4. 충돌하지 않을 때까지 3번 과정을 반복한다.

## 장점

- 하나의 활성노드로 하여금 채널의 전송력  $R$ 로 계속해서 프레임을 전송할 수 있도록 허용한다.
- 노드가 충돌을 감지하고 언제 재전송할지 각자 결정하므로 분산되어있다.
- 매우 단순하다.

## 단점

- 노드는 슬롯이 언제 시작하는지 동기화되어있어야 한다.
- 활성 노드가 많이 있으면 일부 슬롯이 충돌로 인해 결과적으로 낭비 된다.
- 모든 활성 노드가 확률적인 전송 정책 때문에 전송을 억제하는 경우 일부 슬롯이 비게 된다.

낭비되지 않는 슬롯은 정확히 한 노드만 전송하는 슬롯이고, 이 노드를 **성공한 슬롯(successful slot)**이라 한다.

## 효율성

노드가  $N$ 개가 있을 때 하나의 슬롯이 성공적인 슬롯일 확률은 노드들 중 한 노드만 전송하고 나머지  $N-1$  개의 노드는 전송하지 않는 확률이다.

노드가 전송할 확률이  $p$ 라하면 해당 노드가 성공할 확률은  $p \times (1-p)^{(N-1)}$  이다.

노드가  $N$ 개 있으므로 임의의 한 노드가 성공할 확률은  $N \times p \times (1-p)^{(N-1)}$  이다.

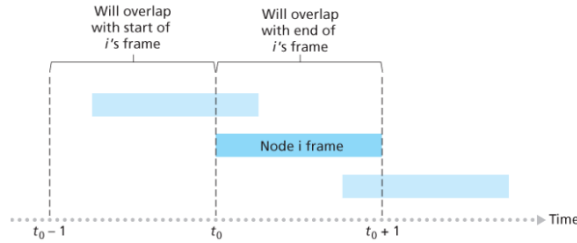
최대의 효율을 구하기 위해서는 이 식을 최대화 하는  $p$ 를 구해야 한다.

활성 노드가 많은 경우의 최대 효율을 구하기 위해  $N$ 이 무한대가 될 때의 극한값을 취한다.

이렇게 계산하면 최대 효율은  $p = 1/e = 0.37$  임을 알 수있다.

즉, 많은 노드가 전송할 프레임이 많을 때 기껏해야 37%의 슬롯만 낭비되지 않는다.

## 알로하(ALOHA)



순수 알로하 프로토콜에서는 슬롯 개념이 없다.

## 동작 과정

1. 프레임이 도착하면 노드는 즉시 프레임 전체를 브로드캐스트 채널로 전송한다.
2. 만약 충돌하면, 노드는 확률  $p$ 로 즉시 재전송 한다.
3. 즉시 재전송하지 않는 경우, 노드는 프레임 전송 시간 동안 기다린다.
4. 기다리고 나서 확률  $p$ 로 전송하거나 아니면  $1-p$  확률로 또 다른 프레임 시간 동안 기다린다.

## 효율성

임의의 시점에 노드가 프레임을 전송할 확률은  $p$ 다.

시간  $t_0$ 에 프레임 전송을 시작한다고 가정하자.

이 프레임이 성공적으로 전송되기 위해서는  $[t_0-1, t_0]$  동안 다른 노드들이 전송을 해서는 안된다. 만일 전송을 하게 되면 노드  $i$ 가 전송 시작 부분과 겹쳐 충돌하게 된다.

이 시간동안 다른 모든 노드가 전송을 시작하지 않을 확률은  $(1-p)^{(N-1)}$  이다.

마찬 가지로 노드  $i$ 가 전송하는 동안에 다른 노드가 전송을 시작해서는 안되고, 이 확률 또한,  $(1-p)^{(N-1)}$  이다.

즉, 성공적으로 전송할 확률은  $i$ 가 전송 할 확률  $p$ 를 포함해서  $p \times (1-p)^{(2(N-1))}$  이다. 슬롯 알로하처럼 극한값을 취하면 최대 효율은  $p = 1/2e$ 로 슬롯 알로하의 절반이다.

즉, 순수 알로하는 완전히 분산되어 동기화하는 것을 안하는 대신, 효율성을 포기한다.

## CSMA

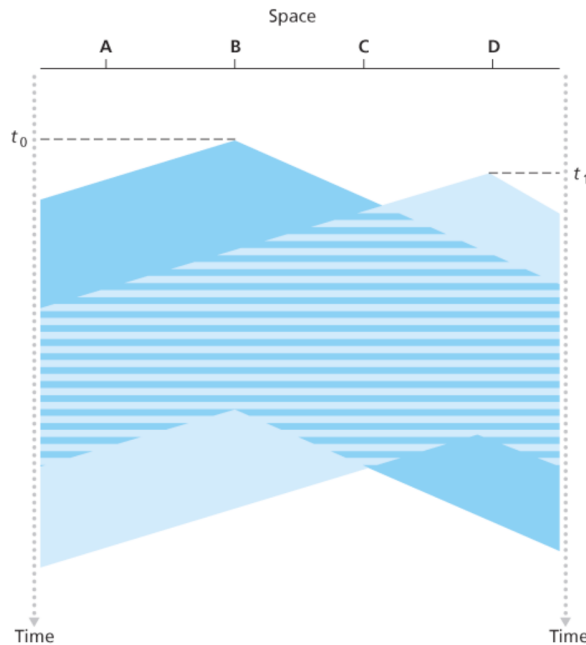
위 두 프로토콜에서는 다른 노드가 전송하고 있건 말건 일단 보낸다.

즉, 충돌이 생기고 결과적으로 효율이 떨어진다.

이러한 충돌을 없애기 위한 규칙을 보자.

- 캐리어 감지(carrier sensing)
  - 만일 다른 노드가 프레임을 채널로 전송하고 있는 경우, 노드는 임의의 짧은 시간 동안 전송 중단을 감지하면 프레임을 전송하기 시작한다.
- 충돌 검출(collision detection)
  - 만일 다른 노드가 방해 프레임을 전송하고 있음을 검출하면, 자신의 전송을 중단하고 랜덤 시간 동안 기다린 후 유희 시 감지 및 전송과정을 반복한다.

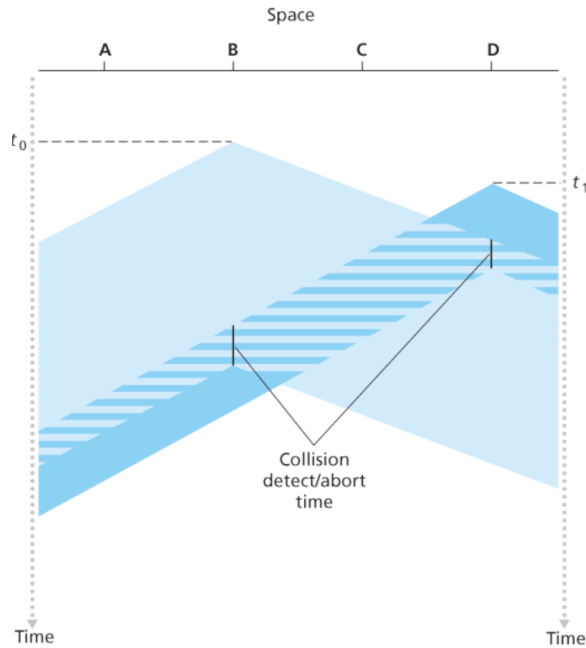
### CSMA에서 충돌이 발생하는 경우



1. 시각  $t_0$ 에 노드 B가 다른 노드가 아무도 전송하고 있지 않으므로 채널이 비어 있는 것으로 감지한다.
2. B는 전송을 시작하고, 전송한 비트들이 브로드캐스트 매체를 따라 양방향으로 전송된다.
  - 시간이 경과함에 따라 아래쪽으로 전파되는 것은 B의 비트들이 실제로 브로드캐스트 매체로 전파할 때 0보다 큰 시간이 필요하다는 것을 의미한다.
3. D가  $t_1$  시점에 전송할 프레임이 생겼고, 노드 B가  $t_1$ 에 전송을 하고 있음에도 불구하고, B에 의해 전송되는 비트들은 D에 도달하지 못했고, 따라서 D는  $t_1$ 일 때 채널이 사용되지 않는 것으로 감지한다.
4. D가 전송을 시작하고, 약간의 시간 후에 B가 전송한 비트와 D의 전송한 비트가 간섭을 일으키기 시작한다.

즉, 브로드캐스트 채널 중단 간의 채널 전파 지연(channel propagation delay)이 길수록 다른 노드에서 이미 시작된 전송을 캐리어 감지 노드가 감지할 수 없는 경우가 증가하기 때문에 채널 중단 간의 채널 전파 지연(channel propagation delay)는 CSMA의 성능을 결정하는데 중요한 역할을 한다.

### CSMA/CD



CSMA는 충돌 검출을 수행하지 않는 반면, CSMA/CD는 충돌 검출을 수행한 후 즉시 전송을 취소한다.

## 동작 과정

1. 어댑터는 네트워크 계층으로부터 데이터그램을 받아서 링크 계층 프레임を 만든 후에 그 프레임을 어댑터 버퍼에 저장한다.
2. 어댑터는 채널이 유헤(idle) 상태임을 감지하면 프레임 전송을 시작한다.
  - 만일 어댑터가 채널이 바쁜(busy) 상태임을 감지하면, **어떤 신호 에너지도 감지되지 않을 때까지 더 기다렸다가 프레임을 전송하기 시작한다.**
3. 전송하는 동안 어댑터는 브로드캐스트 채널을 사용하는 **다른 어댑터로부터의 신호 에너지가 있는지 감시한다.**
4. 프레임 전체를 전송하는 동안 다른 어댑터로부터의 신호 에너지가 감지되지 않으면, 프레임 전송을 완료한다.
  - 감지되면 **자신의 프레임 전송을 취소한다.**
5. 전송 취소 후 **임의의 랜덤 시간**만큼 기다린 후 2단계로 돌아간다.
  - 만일 랜덤 시간이 아니라 고정 시간이라면 동시에 프레임을 전송했을 때 똑같은 시간을 기다린 후 전송을 하므로 계속해서 충돌하게 된다.

## 랜덤 시간을 결정하는 알고리즘: 이진 지수적 백오프(binary exponential backoff)

충돌을  $n$ 번 경험한 프레임을 전송할 때 노드는  $\{0, 1, 2, \dots, 2^n - 1\}$  중에서 랜덤하게  $K$  값을 선택한 후  $K \times \text{비트 시간}$  만큼 기다린다.

이더넷의 경우  $K \times 512$  비트 시간(이더넷으로 512 비트를 전송하는데 걸리는 시간  $\times K$ )가 되며  $n$ 의 최댓값을 10으로 제한한다.

즉, 충돌을 많이 경험할수록 K의 범위가 지수적으로 커지게 된다.

새 프레임을 준비할 때는 최근 발생한 충돌을 고려하지 않고 CSMA/CD를 수행하여 충돌을 경험한 노드보다 먼저 전송될 수도 있다.

## 효율성

$d(prop)$  = 신호 에너지가 임의의 두 어댑터 사이에서 전파되는 데 걸리는 최대 시간  
 $d(trans)$  = 최대 크기의 이더넷 프레임을 전송하는데 걸리는 시간

효율 =  $1/(1+5d(prop)/d(trans))$

전파 지연이 0이 되면 충돌한 노드는 채널을 장비하지 않고 즉시 취소하기 때문에  $d(prop)$ 이 0이 되면 효율은 1에 근접한다.

$d(trans)$ 가 아주 크면 프레임이 채널을 한번 차지하면 아주 오랫동안 채널을 사용하기 때문에 효율은 1에 근접한다.

따라서 채널은 거의 모든 시간 동안 유용하게 쓰인다.

## 6.3.3 순번 프로토콜

다중 접속 프로토콜에서 요구되는 두가지 특성은 다음과 같다.

1. 단 하나의 노드만이 활성이면 Rbps의 처리율을 갖는다.
2. M개의 노드가 활성이면 각 노드가 거의  $R/M$  bps의 처리율을 갖는다.

알로하와 CSMA 프로토콜은 첫번째 특성은 지니고 있으나 두 번째 특성은 없다.

이것이 **순번 프로토콜(taking-turns protocol)** 을 개발하게 된 동기다.

### 폴링 프로토콜(polling protocol)

노드 중 하나를 마스터 노드로 지정한다.

**마스터 노드는 각 노드를 라운드 로빈 방식으로 폴링한다.**

특히, 마스터 노드는 먼저 노드 1에게 노드 1이 최대로 보낼 수 있는 프레임 수에 대한 메시지를 전송하고, 노드 1이 프레임을 전부 보낸 다음 다음 노드도 똑같이 수행하여 순환적으로 각 노드를 폴링하는 방식으로 이 과정을 계속한다.



## 장점

- 빈슬롯을 제거할 수 있다.

## 단점

- 폴링 지연이 있다.
  - 한 노드만 활성이면 활성 노드가 프레임을 최대 개수만큼 보낼 때마다 마스터 노드는 비활성 노드들을 차례로 폴링해야만한다.
- 마스터 노드가 고장나면 전체 채널이 동작하지 못한다.

## 토큰 전달 프로토콜(token-passing protocol)

토큰(token) 이라고 알려진 작은 특수 목적 프레임이 정해진 순서대로 노드 간에 전달된다.

예를 들어, 노드 1은 항상 노드 2에 노드 2는 노드 3에 노드 N은 노드 1에 토큰을 전송한다.

노드가 토큰을 수신하면, 전송할 프레임이 있을 때만 토큰을 붙잡고, 그렇지 않으면 토큰을 전달한다.

프레임을 최대 개수까지 전송한 뒤 토큰을 다음 노드로 전달한다.

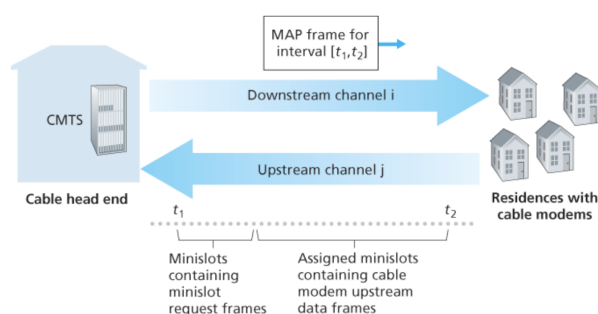
## 장점

- 분산 방식으로 효율이 매우 높다

## 단점

- 노드 하나가 실패하면 채널이 동작하지 않는다.
- 노드가 토큰을 놓아주지 않으면, 토큰이 다시 돌 수 있도록 하는 회복 절차가 수행되어야 한다.

## 6.3.4 DOCSIS: 케이블 인터넷 접속을 위한 링크 계층 프로토콜



DOCSIS(Data-over-Cable Service Interface Specifications)는 케이블 데이터 네트워크의 구조와 프로토콜들을 정의한다.

DOCSIS는 하향 및 상향 네트워크 세그먼트들을 다수의 주파수 채널로 나누기 위해 FDM을 사용한다.

각 상향 및 하향 채널은 브로드캐스트 채널이다.

## 하향 채널

각 하향 채널은 24~192 MHz 대역에 약 1.6 Gbps의 최대 처리율을 제공한다.

CMTS에 의해 하향 채널로 전송된 프레임은 그 채널을 통해 수신하는 모든 케이블 모뎀에 의해 수신된다. (하향 채널로 전송하는 CMTS가 하나이기 때문에 다중 접속 문제는 발생하지 않는다.)

CMTS는 하향 채널상으로 MAP 메시지로 알려진 제어 메시지를 보냄으로써 어떤 케이블 모뎀이 MAP 메시지에서 명시한 시간 간격 동안 어떤 미니슬롯(mini slot)으로 전송할 수 있는지 알려준다.

## 상향 채널

상향 채널은 6.4~96 MHz 대역에 약 1 Gps의 최대 상향 처리율을 제공한다.

다수의 케이블 모뎀이 CMTS로의 동일한 상향 채널(주파수)을 공유하여 충돌이 발생할 수 있다.

상향 채널은 TDM처럼 시간 간격으로 나뉘어 있고, 각 시간 간격은 케이블 모뎀이 CMTS로 전송할 수 있는 일련의 미니슬롯(mini slot)들로 구성되어 있다.

미니슬롯(mini slot) 이 케이블 모뎀마다 명시적으로 할당되어 있기 때문에 CMTS는 미니슬롯(mini slot) 동안은 충돌이 발생하지 않는 것을 확신할 수 있다.

미니슬롯(mini slot)에는 미니슬롯 요청(mini-slot-request) 프레임을 CMTS에게 전송하기 위한 특정 미니슬롯(mini slot) 들이 있다.

각 케이블 모뎀은 미니슬롯 요청(mini-slot-request) 프레임을 CMTS에게 전송하여 어떤 케이블 모뎀이 전송할 데이터가 있는지 알 수 있다.

미니슬롯 요청(mini-slot-request) 프레임 은 랜덤 접속 방식으로 전송되기 때문에 충돌이 발생할 수 있다.

케이블 모뎀은 충돌 검출을 수행하지 않고, 요청된 할당에 대한 응답을 다음 하향 제어 메시지에서 수신하지 못한다면 미니슬롯 요청 프레임이 충돌됐다고 추정한다.

이렇게 추정된 케이블 모뎀은 재전송을 지연시키기 위해 이진 지수적 백오프를 사용한다.