

무선 애드혹 랜 환경에서의 채널 분리에 의한 채널 활용도 향상

(An Improvement of Channel Utilization by Channel Splitting
in Wireless Adhoc LAN Environment)

전 철 호 * 서 영 주 ** 안 성 옥 ***

(Chul-Ho Jun)(Young-Joo Suh)(Syungog An)

요 약 무선 랜용 MAC 프로토콜을 설계함에 있어서, 몇 가지 고려되어야 할 사항들이 있다. 이들 중에서 hidden-terminal 문제와 exposed-terminal 문제는 반드시 고려되어야 할 문제들이다. Hidden-terminal 문제는 많은 사람들에 의해 연구되어 왔으며, 상당한 개선을 이루었으나, exposed-terminal 문제는 그리 중요하게 고려되지 않았다. 이 논문은 두 개의 프로토콜을 제안하였다. 첫 번째 프로토콜은 가용 채널을 제어 채널과 데이터 채널로 나누어 이를 각각 제어 프레임과 데이터 프레임을 위해 사용하는 것이며, 두 번째 프로토콜은 채널을 분리할 경우 제어 프레임이 데이터 프레임과 완전히 독립적으로 전송될 수 있다는 것을 이용하는 프로토콜이다. 두 번째 프로토콜은 한 번의 프레임 전송을 예약하는 첫 번째 프로토콜과 달리 복수의 프레임 전송을 예약한다. 제안된 두 프로토콜은 시뮬레이션을 통해 검증되었으며, 기존의 프로토콜에 비해 향상된 성능을 보였다.

Abstract In designing WLAN MAC protocols, several issues must be considered. Among them, the hidden-terminal and exposed-terminal problems are two of the most important issues. The hidden-terminal problem has been analyzed by many researchers and significant improvements have been made. The exposed-terminal problem, however, has not been fully considered in the design of WMAC protocols. In this paper, we propose two protocols. The first one splits an available channel into a control channel and a data channel and then uses control frames over the control channel. In the second protocol, control frames can be transmitted independently of data frames. The latter protocol reserves several transmissions while the first protocol reserves only one transmission. The proposed protocols are compared with existing protocols and show an improved performance.

1. 서론

최근에 들어, 이동컴퓨팅 분야는 폭발적인 관심을 얻고 있으며, 이에 대한 많은 연구가 진행되고 있다. 이러한 성장은 노트북 컴퓨터나 팜탑 (palm-top) 컴퓨터와 같은 휴대용 컴퓨터들이 상당히 강력한 기능을 가지게

됨으로써 사용자들이 이들 휴대용 컴퓨터를 이용하여 이동 중에도 다양한 종류의 네트워크 서비스를 받을 수 있게 되었기 때문이다. 휴대용 컴퓨터에 네트워크에 대한 연결을 지원하기 위해서 무선 네트워크 기술이 요구되어 지는데, 이 중에서도 무선 랜은 상대적으로 큰 대역폭을 제공하므로 그 중요도가 더욱 높으며, 빠른 기술적 발전을 이루고 있다.

무선 랜 용 MAC (Medium Access Control) 프로토콜을 설계하는데 있어 고려해야 될 몇 가지 문제들이 있다. 이들 중에서 가장 중요한 문제로 들 수 있는 것이 *hidden-terminal* 문제이다 [2]. 무선 랜 환경에서, 한 호스트가 전송한 프레임을 전달받지 못하는 호스트들을 그 호스트의 *hidden-terminal*이라 부르는데, 네트워크 상에 이런 *hidden-terminal*들이 존재할 경우 네트워크

* 학생회원 : 포항공과대학교 컴퓨터공학과
lupus@postech.ac.kr

** 중신회원 : 포항공과대학교 컴퓨터공학과 교수
yjsuh@postech.ac.kr

*** 중신회원 : 배재대학교 컴퓨터전자정보공학부 교수
sungohk@mail.paichai.ac.kr

논문접수 : 1999년 11월 26일

심사완료 : 2001년 2월 2일

채널의 상태를 잘 못 판단함으로써 문제를 야기할 수 있다. 한 이동 호스트가 채널을 사용하고 있는 도중에 이를 알지 못하는 hidden-terminal이 전송을 시작할 경우 충돌을 발생시키게 되며, 이렇게 하여 hidden-terminal 문제는 채널 효율에 상당히 큰 악영향을 미치게 된다.

Exposed-terminal 문제는 무선 랜 환경에서 발생할 수 있는 또 다른 중요한 문제로서, hidden-terminal 문제에 반대되는 경우이다. *Exposed terminal* 문제는, 데이터의 전송을 시작하더라도 프레임 충돌을 발생시키지 않을 이동 호스트가, 채널이 이미 사용 중이라고 판단하고 전송을 지연함으로써 채널 효율을 저하시키게 되는 문제이다. *Exposed terminal* 문제 역시 네트워크의 효율을 저하시키므로 MAC 프로토콜 설계 시에 고려되어야 한다.

Hidden Terminal 문제를 해결하기 위해 지금까지 다양한 방법이 제시되었다. 가장 처음으로 제시된 방법인 BTMA (Busy Tone Multiple Access) [2]는 가용 채널을 두 개의 보조 채널로 나누어 각각 데이터의 전송과 데이터의 전송상태 통지를 위해 사용하였다. 이 두 개의 보조 채널을 각각 메시지 채널과 BT(busy-tone) 채널이라 부른다. 이 프로토콜에서 base station은 전체 채널을 통제한다. Base station은 메시지 채널을 계속해서 감시하여, 만약 어떤 이동호스트가 전송을 시작할 경우 BT 채널을 통해 BT를 전송함으로써 다른 호스트들이 이미 사용중인 채널을 사용하려 하지 않도록 한다. BTMA는 단지 CSMA (Carrier Sense Multiple Access) [1]를 무선 상황에 맞게 약간 수정한 것에 불과하므로, 이를 무선 랜 환경에 적용시킬 경우 여러 가지 문제점을 야기할 수 있다. 예를 들어, BTMA를 사용하는 호스트들은 전송할 데이터가 준비되는 데로 전송을 시도한다. 그러므로, 데이터를 전송하려는 이동 호스트의 수가 많을 경우 BTMA는 CSMA와 같이 데이터 프레임의 충돌로 인한 심각한 성능저하를 경험할 수 있다. 이동 호스트들은 유선 네트워크에서와 같이 충돌을 즉시 감지할 수 없으므로 상황은 CSMA보다 더욱 나빠지게 된다. 더구나, BTMA는 모든 호스트들이 base station을 통해서 간접적으로 통신할 수밖에 없는 infra-structured 네트워크 환경을 위해 설계된 것이기 때문에 ad-hoc LAN과 같이 모든 호스트들이 직접적으로 통신할 수 있는 환경에는 적용될 수 없다. BTMA를 다소 수정한 receiver initiated BTMA가 C. Wu에 의해 제안되었다 [7]. 이 프로토콜은 BTMA처럼 BT를 사용하며, 그 이외에 RTS (Request-to-Send)라 불리

는 제어 프레임을 사용한다. 데이터를 전송하기 전에, 송신자는 RTS 제어 프레임을 수신자에게 송신한다. 수신자가 이 프레임을 문제없이 수신하였다면, BT를 송신함으로써 RTS 제어 프레임을 송신한 호스트 이외의 호스트들이 전송을 시작하지 못하도록 한다. Karn은 RTS 제어 프레임과 CTS (Clear-To-Send) 제어 프레임을 사용하는 MACA (Multiple Access with Collision Avoidance) 프로토콜을 제안하였으며 [3], 이후에 Bharghavan에 의해 MACAW로 개선되었다. 이 프로토콜들에서 각 호스트는 RTS와 CTS 제어 프레임을 교환함으로써, 데이터를 전송하기 위하여 필요한 시간만큼 채널을 예약할 수 있다. 채널을 예약함으로써, 전송되는 프레임의 충돌 확률을 줄이게 되며, 충돌이 발생하더라도 상대적으로 데이터 프레임보다 작은 제어 프레임 사이에 발생하게 되므로, 채널 효율은 상당히 개선된다. MACA를 전후하여 RTS/CTS 제어 프레임을 사용하는 유사한 MAC 프로토콜들이 제안되었으며, 무선 랜을 위한 MAC 프로토콜의 표준을 제정하고 있는 IEEE 802.11 위원회 역시 hidden terminal 문제를 해결하기 위해 이 방법을 적용하였고, 실제로 많은 이점이 있는 것으로 분석되고 있다. [5, 6, 8]. 그러나, 이러한 방법은 필요 이상으로 넓은 통신 지역을 예약함으로써 결과적으로 *exposed-terminal*의 통신을 방해하게 된다. 실제로 통신이 예약된 지역의 *exposed-terminal*이 전송을 시도하게 되면 전송되고 있던 데이터 프레임과 충돌을 일으키게 된다.

위에서처럼 hidden-terminal 문제는 많은 곳에서 연구되어져 왔으며, 상당한 개선이 이루어졌다 [6, 8, 9]. 그러나, *exposed-terminal* 문제는 무선 MAC 프로토콜에 충분히 고려되어지지 않았다. 이동호스트들이 유선 네트워크에 연결되어 있는 base station을 통하여 통신할 수 밖에 없는 infra-structured 네트워크 환경 하에서 *exposed-terminal* 문제는 크게 중요하지 않다. 그러한 환경에선 이미 전송 중인 다른 이동 호스트의 데이터 프레임과 충돌을 일으키지 않으면서 새로운 전송을 시작할 수 있는 *exposed-terminal*이 존재하지 않는다. 그러나, 이동 호스트들이 상대적으로 큰 지역아래서 같은 채널을 공유하며 직접적인 통신을 할 수 있는 ad-hoc 네트워크 환경이나 인접한 셀에서 같은 채널을 사용하기도 하는 특별한 infra-structured 네트워크 환경에선 많은 수의 *exposed-terminal*이 존재할 수 있다. 그러므로, 그러한 네트워크 환경에서 동작할 MAC 프로토콜을 설계할 경우 *exposed-terminal* 문제는 심각하게 고려되어야 한다.

제어 프레임을 사용하여 채널을 예약하는 프로토콜들은 데이터 프레임이 전송 도중에 손실되는 것을 막음으로써 ALOHA나 CSMA와 같은 프로토콜에 비해 많은 성능 향상을 얻을 수 있다. 그러나, 이러한 제어 프레임이 충돌에 의해 손실될 경우나 제어 프레임이 도달하지 않는 곳에 위치했었다가 도달 범위로 이동한 이동 호스트의 경우 채널 예약 상황을 통지 받지 못했기 때문에 전송을 시작할 수 있으며, 이에 따라 이미 전송 중인 데이터 프레임과 충돌을 일으킬 수 있다 [11]. Infra-structured 네트워크 환경 하에선 제어 프레임이 손실되더라도 그에 따른 영향이 그리 크지 않으나 하나의 채널을 많은 호스트들이 공유하는 ad-hoc 네트워크 환경에선 큰 문제가 될 수 있다.

이 논문에서는, 두 개의 프로토콜을 제안하였다. 첫 번째 프로토콜은 가용 채널을 제어 채널과 데이터 채널로 나누어 제어 채널을 RTS/CTS/ACK 등의 제어 프레임의 전송을 위해 사용함으로써 hidden-terminal problem과 exposed terminal problem을 해결하였으며, 제어 프레임의 전송 실패에 의한 손실 문제를 해결하였다. MACA와 같은 프로토콜이 exposed-terminal 문제를 해결하지 못 하는 것은 제어 프레임과 데이터 프레임의 충돌에 기인한다. 제안된 프로토콜은 채널 분리를 통해 이러한 충돌의 가능성을 제거함으로써 exposed-terminal 문제를 해결하였다. 두 번째 프로토콜은 제어 프레임들이 데이터 프레임과 완전히 독립적으로 전송될 수 있는 이점을 보다 적극적으로 사용한다. 이 프로토콜은 하나의 데이터 프레임 전송만을 예약하는 첫 번째 프로토콜과는 달리 다수의 전송을 예약할 수 있다. 제안된 두 개의 프로토콜은 시뮬레이션을 통해 기존의 프로토콜과 함께 비교되었다.

이 논문의 나머지 부분은 다음과 같다. 다음 장에선 기존의 프로토콜에 내재한 문제점들에 대해서 분석하였다. 3장 및 4장에선 제안하는 두개의 프로토콜을 기술하였다. 제안된 프로토콜의 성능은 5장에서 검증되었으며, 검증 결과는 6장에서 요약되었다.

2. 기존 프로토콜의 문제점

이전 장에서 언급되었던 것처럼 exposed-terminal 문제는 MAC 프로토콜을 설계하는데 있어 중요한 문제이며, 반드시 해결되어야 한다. 이번 장에선 기존 프로토콜에 내재한 문제들을 exposed terminal 문제를 포함하여 설명한다.

RTS/CTS 제어 프레임을 사용하는 기존 프로토콜들은 hidden-terminal 문제를 해결하기 위하여 제안되

었다. 이러한 프로토콜에서, 데이터의 송수신자는 제어 프레임을 교환함으로써 hidden-terminal 문제를 해결한다. 호스트들은 데이터의 송수신자가 교환하는 제어 프레임들을 감시하여 채널예약이 발생할 경우 예약된 전송이 끝나는 시점까지 자신의 데이터 전송을 연기한다. 자료를 전송하려는 호스트는 먼저 수신자에게 RTS 제어 프레임을 전송한다. RTS를 수신한 수신자는 CTS 제어 프레임으로 응답한다. 송신자는 CTS를 수신한 후 데이터 전송을 개시한다. 각 RTS/CTS 제어 프레임은 데이터 전송이 완료될 때까지 걸리는 시간에 대한 정보를 담고 있어서 데이터 전송이 완료되는 시점까지 기다려야 하는 다른 호스트들에게 기다려야 하는 시간의 양을 알려주게 된다. 이런 방법을 통해서 데이터 송신자의 hidden-terminal이 채널의 상태를 알지 못하고 전송을 시작함으로써 발생하는 hidden-terminal 문제는 자연스럽게 해결되게 된다.

RTS/CTS 제어 프레임을 사용하는 기존의 MAC 프로토콜들 (예를 들어 MACA)은 exposed-terminal 문제를 해결하지 못한다. 그림 1은 B가 A에게 전송 중인 상황을 나타내고 있다. 이 때, C는 B의 전송지역 내에 위치하므로 채널이 사용 중인 것을 알 수 있으며 (이 상황을 'C가 B에게 exposed 되어 있다'고 표현한다.), 전송할 자료가 있더라도 이의 전송을 연기하게 된다. 그러나, C의 수신자가 B가 아닌 경우에는 (예를 들어, D인 경우) C는 전송을 연기할 필요가 없다. 이 경우에 A는 C의 전송범위 밖에 위치하므로 C의 전송은 B의 전송을 방해하지 않는다. RTS/CTS 제어 프레임을 사용하는 기존의 MAC 프로토콜에서, C가 B의 전송범위 밖에 위치한 D로의 전송을 시도하려 할 때, D가 응답한 CTS 제어 프레임은 B의 데이터 프레임과 충돌하여 C에게 전달될 수 없으며, 결국 C는 데이터를 전송할 수 없다. 심지어 C가 전송을 시작한다 하더라도 B와 C는 서로 각자에게 전송되어 오는 ACK 제어 프레임의 수신을 방해하게 된다. 전송에 실패한 C는 계속해서 전송을 시도하게 되며, 계속되는 실패를 경험한다. 이렇게 하여, 기존의 프로토콜들은 계속되는 제어 프레임 충돌을 일으키게 된다 [4]. IEEE 802.11 표준에서는 [10], 다른 호스트의 RTS 제어 프레임을 감지한 이동 호스트가 그 RTS 제어 프레임에 의해 시작된 데이터 전송과정이 완료 될 때까지 전송 시도를 지연하도록 하였다. 이를 통하여 exposed-terminal의 계속되는 전송 시도와 실패는 피할 수 있다. 위의 예에서 B가 전송한 RTS를 감지한 C는 B의 데이터 전송이 완료될 때까지 전송 시도를 하지 않는다. 그러므로, 이 프로토콜에서 exposed-termi-

nal은 전송이 종료될 때까지 어떠한 전송도 개시하지 못한 채 기다리게 된다.

위에서처럼, RTS/CTS 제어 프레임을 사용하는 기존의 MAC 프로토콜은 exposed-terminal 문제에 상당히 취약한 특성을 가지고 있는데, 이는 exposed-terminal 이 전송을 시도할 경우 제어 프레임과 데이터 프레임이 충돌하는 데에 원인이 있다. 그러므로, 이들 제어 프레임과 데이터 프레임이 충돌 없이 전송될 수 있는 프로토콜이 제안된다면, 이 문제는 해결될 수 있다. 그림 1의 예에서, C가 B의 데이터 프레임과 충돌 없이 D에 의해 전송된 CTS 프레임에 전송 받을 수 있다면, C는 데이터를 D로 전송할 수 있고, B의 데이터 프레임 역시 문제없이 A에 전달될 수 있다.

다음으로 제어 프레임의 전달이 실패하여 생기는 문제에 대해 생각해 보도록 한다. 그림 1의 환경에서 B의 전송이 종료한 후 A가 B로 전송하려 한다고 가정하자. A는 B에게 RTS를 전송할 것이며, B는 CTS로 응답할 것이다. C는 B의 CTS 응답을 감지할 수 있으며, A의 데이터 전송이 끝날 때까지 침묵함으로써 전송을 방해하려 하지 않을 것이다. 그러나 B가 CTS 응답을 전송하는 시점에 D가 제어 프레임이나 데이터를 전송 중이었다면, C는 B의 CTS 응답을 감지할 수 없다. B의 CTS 응답과 D의 전송이 C가 위치한 지점에서 충돌을 일으키기 때문이다. A와 B의 채널 예약을 감지하지 못한 C가 전송을 시도하기 위해 RTS를 전송하게 되면, 이 RTS는 B가 위치한 지점에서 A의 데이터 전송과 충돌하게 되며, A의 데이터 전송은 손실되게 된다. 비슷한 상황이 이동호스트의 이동성 때문에 발생할 수 있다. 먼 곳에 위치하였기 때문에 A와 B의 RTS/CTS 교환을 감지하지 못한 호스트가 (예를 들어 D) B 근처로 이동할 경우 D는 역시 A와 B의 채널 예약을 감지하지 못하였기 때문에 전송을 시도할 수 있으며, 마찬가지로 충돌을 발생시킬 수 있다.

계속되는 3장과 4장에서 이러한 문제들을 해결하는 프로토콜들이 제안되고 있다.

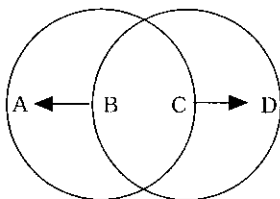


그림 1 Exposed Terminal 문제

3. 채널 분리

Exposed terminal 문제는 다음과 같이 채널을 두 개의 보조 채널로 나누어 사용하는 것에 의해 해결될 수 있다: 먼저 각 채널을 두 개의 보조 채널로 나눈다. 한 보조 채널은 제어 프레임의 전송을 위해 사용되며, 나머지 한 보조 채널은 데이터 프레임의 전송을 위해 사용된다. 제어 프레임의 크기가 상대적으로 작으므로, 제어 프레임에 요구되는 채널 대역폭 또한 작게 설정한다. 각 이동 호스트는 제어 채널을 통하여 데이터 채널을 예약한 후, 데이터 채널을 통하여 데이터를 전송하게 된다.

채널을 분리하여 사용하는 프로토콜은 채널을 분리하였다는 점을 제외하면, RTS/CTS 제어 프레임을 사용하는 기존의 프로토콜에 상당히 유사하다. 한 호스트가 데이터를 전송하려 할 경우, 호스트는 제어 채널을 통해 RTS 프레임을 수신자에게 전송한다. 수신자는 역시 제어 채널을 통해 CTS 프레임을 전송함으로써 이에 응답한다. 이 응답을 받은 송신자는 데이터 채널을 통해 전송하고자 하는 데이터를 전송하게 된다. MACA에서처럼 각 RTS/CTS 제어 프레임은 채널을 예약하고자 하는 시간에 대한 정보를 담고 있다. 데이터를 전송 받은 이후에 수신자는 ACK 프레임을 전송하여 데이터가 문제없이 전송되었음을 알리게 된다.

그림 1의 예에서 채널 분리가 사용될 경우, D의 CTS 응답은 B에 의해 전송중인 데이터 프레임과 충돌 없이 C에게 전송될 수 있다. 그러므로, C는 D로의 전송을 개시할 수 있다. 자료를 전송 받은 이후에 A와 D중 자료 전송을 일찍 끝낸 호스트는 ACK 제어 프레임을 각자의 송신자에게 전송한다. 채널 분리가 사용될 때, A나 D가 전송한 ACK 제어 프레임은 C나 D의 데이터 프레임과 충돌하지 않는다. 결과적으로 exposed-terminal 문제는 발생하지 않는다.

채널 분리는 제어 프레임의 충돌이나 이동호스트의 이동성에 기인한 문제 역시 해결한다. 다른 호스트들이 교환한 제어 프레임을 감지하지 못한 이동 호스트들은 채널이 예약된 상황을 알지 못할 수 있다. 기존의 프로토콜들에서, 이러한 호스트들은 전송을 시도하기 위해 RTS 제어 프레임을 전송함으로써 이미 진행중인 데이터 전송을 방해할 수 있다. 앞장에서 언급한 것처럼 이런 상황은 제어 프레임의 충돌과 이동 호스트의 이동성 때문에 발생하게 되며, 네트워크 성능에 심각한 저하를 가져오게 된다. 채널 분리를 사용할 경우, 다른 호스트들이 교환한 RTS/CTS 제어 프레임을 감지하지 못한 이동 호스트들이 전송을 시도하기 위해 RTS 프레임을

전송하더라도, 이 RTS 제어 프레임은 진행중인 데이터 프레임의 전송을 방해하지 않는다. 그러므로, 제어 프레임의 빈번한 충돌이 발생할 경우, 제안된 프로토콜은 기존의 프로토콜에 비해 상대적으로 안정적이다. 이러한 현상은 각 호스트들의 전송 요구가 많을수록, 이동 호스트들이 밀도 있게 위치할수록 더욱 분명해지며, Ad hoc 네트워크 환경에서의 경우 많은 수의 이동 호스트가 같은 채널을 공유하게 되므로 이런 상황은 더욱 두드러지게 된다.

그림 2는 채널 분리를 사용하는 프로토콜의 예이다. 제어 채널을 통한 한 번의 RTS/CTS 제어 프레임 교환이 데이터를 전송하는데 필요한 시간만큼 데이터 채널을 예약한다. CTS 프레임은 전송 받은 이후에 송신자는 데이터 채널을 통해 데이터를 전송하며, 수신자는 데이터의 전송이 끝나자마자 제어 채널을 통해 ACK 제어 프레임으로 응답한다. 이런 방법으로 데이터 채널은 예약되고 사용된다.

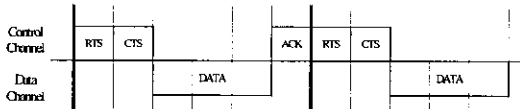


그림 2 첫 번째 프로토콜의 실행 예

채널을 분리하는 것이 장점만을 가진 것은 아니다. 우선 제어 프레임에 일정 대역을 정적으로 할당하므로, 데이터의 전송에 사용될 수 있는 대역폭이 감소하게 된다. 또한 제어 프레임이 작은 대역폭을 가진 제어 채널을 통해 전송되므로 제어 프레임의 전송시간은 기존의 프로토콜에 비해 길다. 단위 시간당 전송 시도의 수가 같을 때, 제어 프레임의 증가된 전송시간은 제어 프레임의 충돌 가능성을 높일 수 있으며, 이는 곧 채널의 효율의 감소로 귀결될 수 있다.

4. 전송 예약

앞장에선, 채널을 분리한 후 RTS/CTS 제어 프레임을 사용하는 프로토콜을 제안하였다. 이번 장에선 채널을 분리한 후 RTR (Request to Reserve) 제어 프레임과 ATR (Acknowledgement to Reserve) 제어 프레임 사용하는 다른 프로토콜을 제안한다.

MACA에서 RTS/CTS 제어 프레임의 교환은 앞으로 진행하려는 데이터의 전송을 방해할 수 있는 새로운 RTS 제어 프레임의 전송을 금지한다. 그러나, 만약 데이터 채널과 제어 채널이 분리될 경우 제어 프레임은

데이터 프레임의 전송여부와 관계없이 전송될 수 있다. 이 특성을 적극적으로 활용할 경우 네트워크의 성능은 개선될 수 있다.

두 번째 프로토콜은 RTS/CTS 제어 프레임 대신에 RTR/ATR 제어 프레임을 사용한다. 이 제어 프레임들은 데이터 프레임의 전송여부에 관계없이 자유롭게 전송될 수 있다. 먼저 전체 시간을 데이터 프레임을 전송하는데 필요한 시간단위인 데이터 슬롯(data-slot)으로 나눈다. 또, 하나의 데이터 슬롯은 하나의 제어 프레임을 전송하는데 필요한 시간인 제어 슬롯으로 나누어 진다. 호스트는 성공적인 RTR/ATR 제어 프레임의 교환을 통해서 예약 가능한 N개의 데이터 슬롯 중 하나를 예약할 수 있다. 각 이동 호스트는 각 데이터 슬롯에 해당하는 플래그(flag)들을 가지고 있어서 채널 예약 상황에 대한 정보를 유지한다.

전송할 데이터를 가지고 있는 이동 호스트는 RTR 제어 프레임을 브로드캐스트(broadcast)함으로써 전송을 시도한다. RTS 제어 프레임처럼, RTR 제어 프레임은 이에 대한 응답인 ATR 제어 프레임이 전송될 때까지 제어 채널을 예약한다. RTR 제어 프레임을 전송 받은 이동 호스트는 데이터 채널을 예약하는 ATR 제어 프레임으로 응답한다. 기본적으로 RTR 제어 프레임의 기능은 RTS 제어 프레임과 같으며, ATR 제어 프레임의 기능 역시 CTS 제어 프레임과 비슷하다. 그러나, 채널을 예약하는 시간을 담고 있는 RTS/CTS 제어 프레임과는 달리 RTR 제어 프레임은 그러한 정보를 가지고 있지 않으며, ATR 제어 프레임은 예약 시간을 가지고 있지 않은 대신에 예약한 데이터 슬롯의 번호를 가지고 있다. ATR 제어 프레임을 전송 받은 이동 호스트는 예약된 데이터 슬롯 동안에 데이터를 전송할 수 있다. ATR 제어 프레임을 전송한 호스트와 전송 받은 호스트는 채널 예약 상황을 나타내기 위해 해당 데이터 슬롯의 플래그를 갱신한다. 각 이동호스트는 전송을 시도하기 전에 또 자신에게 보내진 RTR 제어 프레임에 응답하기 전에 이 플래그의 값을 참조한다. 모든 채널이 예약되었다면, 이동호스트는 전송을 시도하지 않으며, RTR 제어 프레임에 응답하지도 않는다. 전송을 예약한 이동 호스트는 예약한 데이터 슬롯 번호와 수신 받을 이동 호스트에 대한 정보를 관리하여, 예약된 데이터 슬롯이 되면 해당하는 이동 호스트로의 전송을 시작한다.

제어 채널을 통해 전송중인 ATR 프레임을 감지한 이동 호스트들은 데이터 채널의 예약 상황을 알 수 있다. 전송 받은 ATR 제어 프레임의 수신자가 아닐 경우 이동 호스트는 예약된 데이터 슬롯의 끝까지 자료 전송

을 시도하지 않는다. 그 이유는 다음과 같다. 그림 1에서 A가 B로 데이터를 전송하기 위해 데이터 슬롯 1을 예약하게 되었다면, B가 송신한 ATR 1은 A와 C에 의해 수신되게 된다. 이 상황에서 C가 D에게 RTR을 송신함으로써 D로의 전송을 시도한다면, D는 B가 송신한 ATR 1을 수신하지 못했기 때문에 데이터 슬롯 1의 예약 상황을 알지 못하므로 같은 ATR 1으로 응답할 수 있다. 이 ATR 응답은 데이터 슬롯 1에 대한 중복되는 예약을 초래하게 된다. 결국 데이터 슬롯 1에서 A와 C는 동시에 데이터를 송신하게 되고, B에서 데이터 프레임은 충돌하게 된다. ATR 1을 수신한 C가 데이터 슬롯 1의 끝까지 전송을 시도하지 않는다면, 이 상황은 피할 수 있다.

한 데이터 슬롯동안 한 호스트는 두 개 이상의 데이터 슬롯을 예약할 수 없다. 두 개 이상의 예약이 허용된다면, 한 이동 호스트에 의한 채널의 독점이 발생할 수 있다. ATR을 응답 받은 이동 호스트가 예약된 데이터 슬롯까지 전송을 시도하지 않는다면 이러한 독점은 피할 수 있다.

앞장에서 제안된 프로토콜의 경우, 제어 프레임의 충돌 가능성은 기존의 RTS/CTS 제어 프레임을 사용하는 프로토콜보다 높다. 제어 프레임의 교환이 완료되기 전까지는 데이터를 전송할 수 없으므로, 제어 프레임의 충돌이 빈번할 경우 채널의 효율은 상당히 낮아질 수 밖에 없다. 이번 장에서 제안된 프로토콜은 현 데이터 슬롯 동안에 발생한 RTR/ATR 교환으로 다음 N개의 데이터 슬롯 중 하나를 예약한다. 그러므로, 제어 프레임의 충돌이 빈번하더라도 N개의 데이터 슬롯 동안에 단지 N 개의 RTR/ATR 교환이 성공적으로 완료될 수 있다면, 채널의 효율의 감소는 발생하지 않는다.

ACK 제어 프레임을 보다 안전하게 전송하는 방법에 대해 고려해 볼 수 있다. ACK 제어 프레임은 데이터 프레임의 전송이 완료되자마자 전송되게 된다. 다시 말해서, 제안된 프로토콜의 경우 ACK 제어 프레임은 각 데이터 슬롯의 시작 부분에 전송되게 된다. ACK 프레임이 전송되는 시점이 고정되어 있으므로, ACK 제어 프레임이 전송되는 시점, 즉 각 데이터 슬롯의 시작부분에서 RTR/ATR 제어 프레임이 전송되지 않도록 한다면, ACK 제어 프레임의 충돌 가능성은 상당히 줄어들게 된다. 데이터 슬롯 시간의 길이가 T이고 시간 T동안 전송될 수 있는 제어 프레임의 수가 N이라고 할 때, 전송 시도를 위하여 RTR 제어 프레임을 전송할 수 있는 시점은 각 데이터 슬롯의 T/N부터 T - 2T/N까지이다. ACK 제어 프레임은 긴 시간동안의 데이터 전송

이 성공적으로 완료된 후에 전송되는 것이므로, 이 프레임이 빈번하게 손실될 경우 네트워크의 성능에 큰 악영향을 미치게 된다. 그러므로, ACK 제어 프레임의 안전한 전송이 보장될 경우 네트워크의 성능은 크게 향상되게 된다.

그림 3은 제안된 프로토콜의 수행과정을 보여주는 한 예이다. 데이터 슬롯 0에서 RTR/ATR 제어 프레임의 교환을 통해 데이터 슬롯 1이 예약되었으며, 예약된 데이터는 데이터 슬롯 1동안에 데이터 채널을 통해 전송되었다. 역시 데이터 슬롯 1동안에 데이터 슬롯 2와 3에 대한 예약이 제어 채널을 통해 이루어졌으며, 그림에서와 같이 예약된 데이터 프레임은 데이터 슬롯 2와 3동안에 전송되었다. 이 그림에서 흐리게 칠해진 부분은 ACK 제어 프레임의 전송을 위해 남겨진 시간을 의미하며, 이 시간동안 RTR 제어 프레임은 전송되지 않아야 한다.

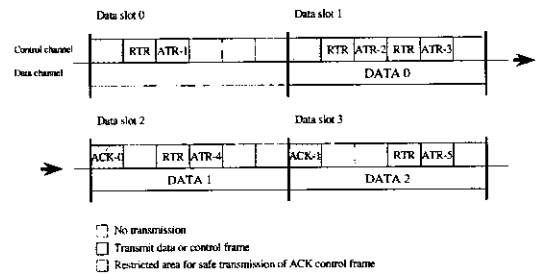


그림 3 두 번째 프로토콜의 실행 예

제안된 프로토콜에서, 제어 채널에 할당하는 대역폭의 크기는 네트워크의 성능에 상당한 영향을 끼치게 된다. 제어 채널에 너무 큰 대역폭을 할당하였을 경우 데이터 채널에 할당해야 할 대역폭이 작아지게 되므로 채널 효율에 문제를 가져오게 될 것이며, 반대로 너무 작게 할당하였을 경우 제어 프레임의 전송 시간이 증가하게 되어 역시 문제를 야기하게 된다. 제어 프레임의 증가된 전송시간은 제어 프레임의 충돌 가능성을 높일 수 있으며, 이는 곧 채널의 효율의 감소로 이어질 수 있다. 그러므로, 제어 채널에 할당할 대역폭의 최적화가 요구되어 진다.

5. 성능 평가

이번 장에서는 시뮬레이션을 통해 제안된 프로토콜들의 성능을 검증하고 기존의 프로토콜과 비교한다. 시뮬레이션에 사용된 파라미터들은 다음과 같다.

네트워크 형태: Ad-hoc

전체 네트워크 범위: 400m × 400m

이동 호스트의 전송범위(반경): 100m

데이터 프레임의 크기: 2000bytes

제어 프레임의 크기: 20bytes

전체 채널 대역폭: 2Mbps

제어 채널 대역폭 : 데이터 채널 대역폭 = 1:5

예약 될 수 있는 데이터 슬롯의 수 3

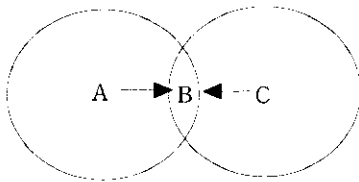


그림 4 Hidden terminal 상황 (A와 C는 서로 hidden terminal 이다.)

먼저 그림 4에서와 같은 hidden-terminal 상황에서의 성능을 살펴본다. 그림 4에서 이동 호스트 A와 C가 이동 호스트 B로 데이터를 전송하려 한다. 여기서 A와 C는 서로 hidden-terminal이므로, 서로의 전송을 방해한다. 그림 5는 그림 4의 hidden-terminal 상황에서의 시뮬레이션 결과이다. 그림 5는 가변의 트래픽 로드에서 CSMA, MACA와 본 논문에서 제안된 두 가지 프로토콜에 대한 전체 네트워크의 전송량을 비교하였다. 여기서 전송량은 초당 전체 네트워크에서 성공적으로 전송된 데이터 프레임 수를 의미하며, 트래픽 로드는 전체 네트워크에서의 총 트래픽 로드를 의미한다. 트래픽 로드가 1 일 경우, 네트워크는 전체 대역폭인 2Mbps를 사용한다. 그림 5의 경우 데이터 프레임 크기가 2000bytes(=16000bits)이므로 전체 네트워크에서 초당 최대 프레임 전송 횟수는 125 (2Mbps/16000bits=125)가 된다. 그러므로, 트래픽 로드가 1이고 두 호스트 A, C가 B로 전송을 시도할 경우, 각 호스트 별로 최고 초당 62.5의 프레임이 전송되어질 수 있다. 그림 5에서 CSMA는 A와 C가 서로 hidden-terminal이므로 서로의 전송을 방해하게 되어 전송 충돌에 의한 상당히 낮은 전송량을 보이고 있으나, MACA의 경우 hidden-terminal 문제를 해결하므로 가장 좋은 성능을 보이고 있다. 제안된 프로토콜들 역시 hidden-terminal 문제를 해결하고 있으나 MACA에 비해 비교적 낮은 성능을 보이고 있다. 이는 제어 채널의 분리에 따른 줄어든 데이터 프레임 대역폭과 제어 프레임의 교환에 따

른 증가된 지연 시간에 기인한다. 2000bytes 크기의 데이터 프레임 하나를 전송하는데, MACA는 전체 대역폭 2Mbps를 사용하여 $(2000 \times 8\text{bits}) / 2\text{Mbps} = 8\text{msec}$ 의 전송 시간이 필요하나, 제안된 프로토콜의 경우 전체 대역폭 2Mbps 중 5/6을 데이터 프레임 전송에 사용하므로 $(2000 \times 8\text{bits}) / (5/6 \times 2\text{Mbps}) = 9.6\text{msec}$ 의 전송 시간이 필요하다. 이와 같이 비교적 긴 전송시간이 필요함에도 RTR/ATR 예약 프로토콜의 경우 MACA에 근접하는 성능을 보이고 있다. RTR/ATR 예약 프로토콜에서 호스트는, 다른 호스트들이 이전에 몇 개의 데이터 슬롯을 예약을 하였을 경우, 자신이 예약한 데이터 슬롯이 시작되기까지 상당한 시간을 기다리게 됨을 예측할 수 있다. 그러나, 이러한 전송 지연은 데이터 프레임들 간의 전송 충돌을 피하기 위한 지연이라 할 수 있다. 그림 6은 그림 5와 같은 hidden-terminal 상황에서 트래픽 로드에서 다른 프레임간의 전송 충돌 횟수를 보이고 있다. 그림 6에서 보는 바와 같이 CSMA는 네트워크 성능을 저하시키는 많은 전송 충돌 횟수를 보이고 있다. MACA와 제안된 두 프로토콜은 전송 충돌 횟수가 작으나, 이들 중에서도 제안된 프로토콜이 MACA보다 더욱더 작은 횟

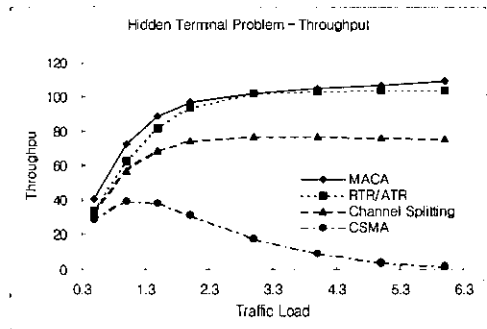


그림 5 Hidden terminal 상황에서의 전송량

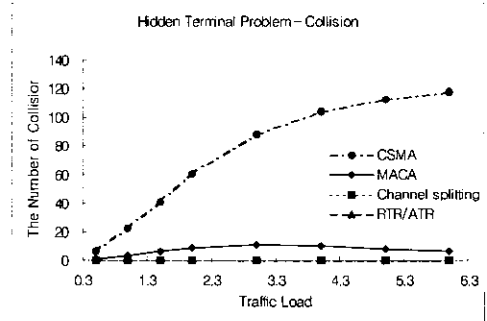


그림 6 Hidden terminal 상황에서의 데이터 손실 수

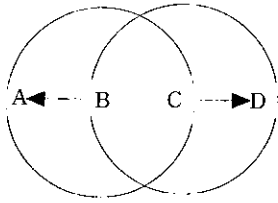


그림 7 Exposed terminal 상황

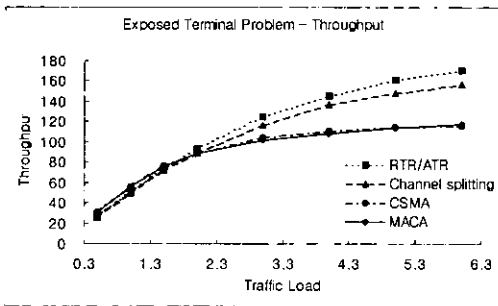


그림 8 Exposed terminal 상황에서의 전송량

수의 전송 충돌을 보이고 있음을 알 수 있다.

그림 7은 B와 C가 각각 A와 D로 전송을 원하는 exposed-terminal 환경의 네트워크를 보이고 있다. 그림 7에서 보는 바와 같이 B와 C가 가 동시에 전송을 하더라도 수신자가 멀리 떨어져 있으므로 문제가 되지 않는다. 그림 8은 각 프로토콜의 전송량을 보이고 있다. 이 그림에서 MACA는 CSMA와 같이 exposed-terminal 문제를 해결하지 못하므로, CSMA와 유사한 성능을 보임을 알 수 있다. 트래픽 로드가 낮을 경우 (2 이하일 경우), 제안된 프로토콜들은 CSMA와 MACA와 유사한(약간 낮은) 성능을 보이나, 트래픽 로드가 증가함에 따라 exposed-terminal이 되는 횟수가 증가하므로 이를 해결하는 제안된 프로토콜들은 CSMA와 MACA보다 향상된 성능을 보임을 알 수 있다.

다음의 그림들은 MACA와 제안된 프로토콜을 일반적인 환경에서 시뮬레이션을 수행하여 성능 평가를 한 것이다. 네트워크의 성능은 이동 호스트의 밀집도와 관계가 있기 때문에 20대의 호스트가 통신하는 저 밀집도 환경, 60대의 호스트가 통신하는 중 밀집도 환경, 120대가 통신하는 고 밀집도 환경 등 세 가지의 환경 하에서 실험하였다. 시뮬레이션에서 데이터 채널은 제어 채널의 5배에 해당하는 대역폭을 가지도록 설정하였으며, RTR/ATR 제어 프레임을 사용하는 프로토콜에서 예약될 수 있는 데이터 슬롯의 개수는 3으로 설정하였다.

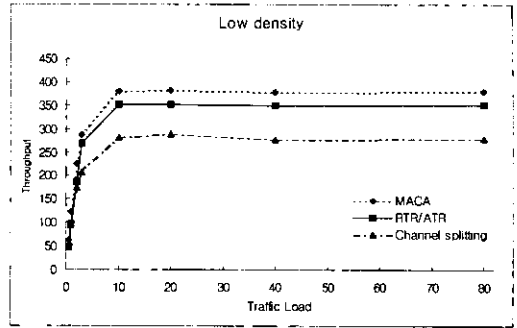


그림 9 저 밀집도 환경에서의 네트워크 성능

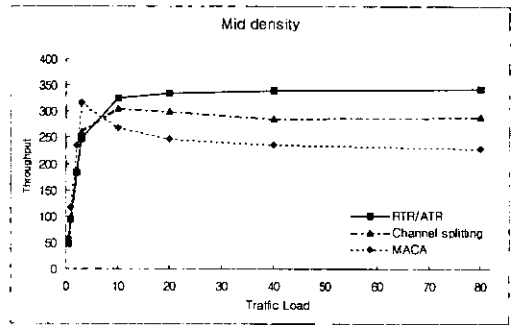


그림 10 중 밀집도 환경에서의 네트워크 성능

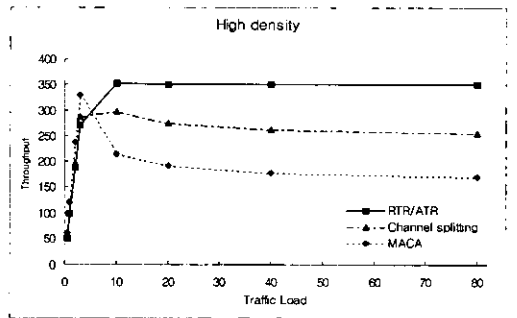


그림 11 고 밀집도 환경에서의 네트워크 성능

그림 9, 10, 11은 저 밀집도, 중 밀집도, 고 밀집도 환경 각각에 대하여 전송량을 트래픽 로드와 대한 함수로 보이고 있다. 그림 9에서와 같이 저 밀집도 환경에서는 MACA는 가장 나은 성능을 보이는 반면, RTR/ATR 예약 프로토콜은 MACA에 비해 약간 낮은 성능을 보인다. 이는 저 밀집도 환경에서는 exposed terminal의 수가 적으며, 인접 이동 호스트의 간섭 역시 상대적으로

적기 때문에 제안된 프로토콜들이 적용되었을 때의 장점이 두드러지지 않기 때문이다. 그림 10은 중 밀집도 환경에서의 전송량을 보이고 있다. 트래픽 로드가 작을 경우 MACA와 제안된 프로토콜들은 비슷한 성능을 보이나 트래픽 로드가 증가할수록 제안된 프로토콜들은 MACA에 비해 나은 성능을 보인다. 이는 밀집도가 증가함에 따라 이웃한 호스트간의 전송 충돌 횟수가 증가함에 기인한다. 그림 11에 보인 바와 같이 고 밀집도 환경 또한 중 밀집도 환경과 비슷한 결과를 보이고 있다. 이와 같이 제안된 프로토콜들은 트래픽 로드가 많을 수록, 단위 면적 당 이동 호스트의 수가 클수록 좋은 성능을 보인다는 것을 알 수 있다.

그림 12, 13, 14는 각각 저 밀집도, 중 밀집도, 고 밀집도 환경에서의 전송 실패의 수를 트래픽 로드(traffic load)에 대한 함수로써 보이고 있다. 여기에서 전송 실패의 수는 호스트가 데이터를 실제로 전송하였지만 충돌이 발생하여 손실된 경우의 수를 의미한다. 그림을 보면, 제안된 프로토콜들은 모든 환경에서 가장 적은 전송 실패의 수를 보임을 알 수 있다. 저 밀집도 환경에서의 그림 9와 12를 보면, MACA는 제안된 프로토콜 보다 많은 전송 실패를 경험하였지만, 최고의 성능을 보이고 있다. 이것은 다음과 같이 설명될 수 있다. 저 밀집도 환경에서 MACA가 유발한 전송 실패의 수(약 150)는 중 밀집도 환경이나 고 밀집도 환경에서의 전송 실패의 수(각각 400, 520 정도)에 비해 상당히 작다. 이에 따라, 저 밀집도 환경에서의 MACA와 제안된 프로토콜들의 전송 실패 차(채널 분리: 약 90, RTR/ATR: 약 110)는 중 밀집도 환경(채널 분리: 약 150, ATR/RTR: 약 310)이나 고 밀집도 환경(채널 분리: 약 200, ATR/RTR: 약 420)에서와 비교했을 때 매우 작다. 이렇게 작은 수의 전송 실패는 MACA의 성능을 그리 크게 저하시키지 않으며, 채널을 분리함에 따르는 효율 저하에 비해 큰 부담이 되지 않으므로 가장 높은 성능을 보이게 된다. 그러나, 밀집도가 증가함에 따라 전송 실패의 수는 증가하게 되며, 이러한 전송 실패는 MACA의 성능을 심각하게 저하시키게 된다. 주목할 사실은 제안된 프로토콜들에서 채널 분리 방식은 중 밀집도, 고 밀집도에서 각각 약 250 과 320의 충돌 횟수를 가지며, RTR/ATR 예약 방식은 중 밀집도, 고 밀집도에서 각각 약 90 과 100의 충돌 횟수를 가진다는 것이다(그림 13, 14). 이와 같이 제안된 프로토콜들이 각 밀집도에 대하여 낮은 충돌 횟수 편차를 가지므로, 제안된 프로토콜들의 각 밀집도에 대한 전송량은 거의 일정한 상수 값을 유지한다. (예를 들어 RTR/ATR 예약 프로토콜의 경우 각 밀집

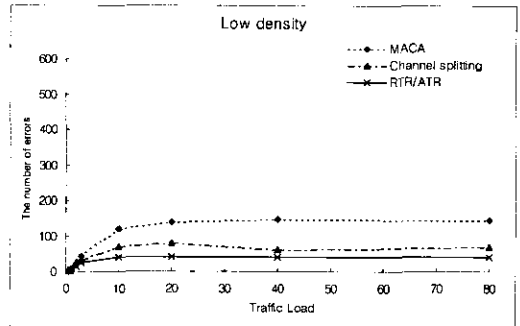


그림 12 저 밀집도 환경에서의 에러 수

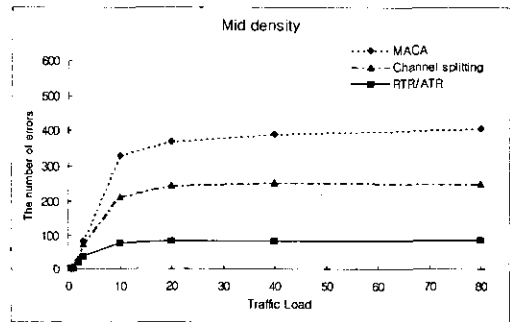


그림 13 중 밀집도 환경에서의 에러 수

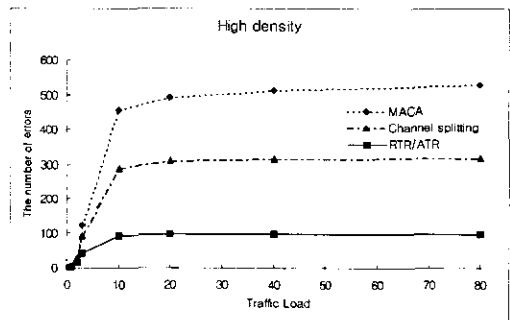


그림 14 고 밀집도 환경에서의 에러 수

도에 대해서 350 정도의 일정한 전송량을 유지한다.) 이와 같은 시뮬레이션 결과를 바탕으로 제안된 프로토콜들은 충돌이 일어날 환경에 강하며, 트래픽 로드가 높거나 단위 면적 당 이동 호스트의 수가 많을 경우 MACA에 비하여 향상된 성능을 보임을 알 수 있다.

6. 결론

무선 랜용 MAC 프로토콜을 설계함에 있어서 hid-

den-terminal 문제와 exposed-terminal 문제를 해결하는 것은 매우 중요한 과제이다. 이를 해결하기 위하여, 이 논문은 두 개의 새로운 프로토콜을 제안하였다. 첫 번째 프로토콜은 주어진 채널을 제어 채널과 데이터 채널로 분리한 후 기존의 프로토콜에 사용한 RTS/CTS 제어 프레임 대신에 새로운 RTR/ATR 제어 프레임을 사용하는 프로토콜이다. 제안된 프로토콜은 hidden-terminal 문제와 exposed-terminal 문제를 해결하고 있으며, 이 외에 제어 프레임의 충돌이나 이동 호스트의 이동성에 의해 발생하는 문제 역시 완화한다. 제안된 프로토콜은 기존의 프로토콜에 비해 향상된 성능을 보였으며, 이동 호스트의 밀집도가 높거나 전송 요구가 높은 환경에서도 안정적인 성능을 보였다.

참 고 문 헌

- [1] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I Carrier Sense Multiple Access Modes and Their Throughput Delay Characteristics," *IEEE Trans. on Communication*, vol. COM-23, no. 12, pp. 1400-1416, 1975.
- [2] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II The Hidden Terminal Problem in Carrier Sense Multiple-Access Modes and the Busy-Tone Solution," *IEEE Trans. on Communication*, vol. COM-23, no. 12, pp. 1417-1433, 1975.
- [3] P. Karn, "MACA - A New Channel Access Method for Packet Radio," *Proceedings of ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134-140, 1990.
- [4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *Proceedings of ACM SIGCOMM 94*, pp. 212-225, 1994.
- [5] J. Weinmiller, M. Schlager, A. Festag, and A. Wolisz, "Performance Study of Access Control in Wireless LANs IEEE 802.11 DFWMAC and ETSI RES 10 Hiperlan," *ACM MONET*, pp. 55-67, 1997.
- [6] H. S. Chhaya and S. Gupta, "Performance Modeling of Asynchronous Data Transfer Methods of IEEE 802.11 MAC Protocol," *ACM Wireless Networks*, pp. 217-234, 1997.
- [7] C. Wu and V.O.K. Li, "Receiver-initiated Busy-Tone Multiple Access in Packet Radio Networks," *Proceedings of ACM SIGCOMM 87*, 336-342, 1987.
- [8] S. Khurana, A. Kahon, and A. P. Jayasumana, "Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC Protocol," *Proceedings of the 23rd. Annual Conference on IEEE Local Computer Networks*, 1998.
- [9] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proceedings of ACM SIGCOMM 97*, pp.39-49, 1997.
- [10] IEEE Standards Department, *Draft Standard IEEE 802.11, Wireless LANs (P802.11D3)*, 1996
- [11] Mario Joa-Ng and I-Tai Lu, "Spread Spectrum Medium Access Protocol with Collision Avoidance in Mobile Ad-hoc Wireless Network," *Proceedings of IEEE INFOCOM 99*, pp. 776-783, 1999
- [12] Zhenyu Tang and J. J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks," *Proceedings of IEEE INFOCOM 99*, pp. 194-201, 1999
- [13] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," *Proceedings of ACM SIGCOMM 95* pp. 262-273, 1995
- [14] Z. Cai and M. Lu, "SNDR: A New Medium Access Control for Multi-channel Ad-Hoc Networks," *Proceedings of the 51st vehicular technology conference (VTC)*, pp. 966-971, 2000
- [15] S. Wu, Y. Tseng, and J. Sheu, "Intelligent Medium Access for Mobile Ad-Hoc Networks with Busy Tones and Power Control," *Proceedings of the 8th international conference on computer communications and networks*, pp. 71-76, 1999
- [16] J. Deng and Z. J. Hass, "Dual Busy Tone Multiple Access: A New Medium Access Control for Packet Radio Networks," *Proceedings of the IEEE 1998 International Conference on Universal Personal Communications*, pp. 973-977, 1998.



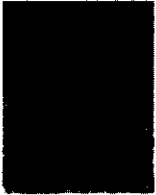
전 철 호

1998년 동국대학교 컴퓨터공학과 학사 졸업. 2000년 포항공과대학교 컴퓨터공학과 석사졸. 2000년 ~ 현재 포항공대 ANMC lab 연구원. 관심분야는 Wireless MAC Protocol, Mobile Networking Protocols.



서 영 주

1985년 한양대 전자공학 학사. 1987년 한양대 전자공학 석사. 1996년 미국 Georgia Tech 컴퓨터공학 박사. 1988년 ~ 1990년 LG전자 연구원. 1990년 ~ 1993년 충청대학 조교수. 1996년 ~ 1997년 미국 Georgia Tech Postdoc Researcher. 1997년 ~ 1998년 미국 University of Michigan Research Fellow. 1998년 ~ 현재 포항공대 컴퓨터공학과 조교수. 관심분야는 Network protocols, Wireless Mobile Networks, Interconnection Networks.



안 성 욱

1983년 2월 고려대학교 수학교육과 졸업(학사). 1985년 2월 고려대학교 대학원 수학과 졸업(전산학전공 석사). 1989년 8월 고려대학교 대학원 수학과 졸업(전산학전공 박사). 1991년 ~ 현재 배재대학교 정보통신공학부 부교수. 1993년 ~ 1994년 펜실베이니아주립대학교 컴퓨터학과 교환교수. 1994년 ~ 1995년 뉴욕주립대학교 컴퓨터학과 교환교수. 관심분야는 데이터베이스, 컴퓨터네트워크, XML, 저장관리기, 컴퓨터 교육