Revised MAC Frame Formats

B2 Overall Operation

- There are two binary flags controlling the operation of a STA
  - Infrastructure
  - Acting_as_AP
- The combo of these bits defines a station’s role in the network

<table>
<thead>
<tr>
<th>Inf</th>
<th>AP</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>STA in infrastructure</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>AP in infrastructure</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>STA in ad-hoc</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Nonsensical</td>
</tr>
</tbody>
</table>
How the B2 draft works

**STA 1**

- **RTS:** $\text{MPDU} = x$, $\text{NID} = y$, $\text{DA} = \text{STA 2}$

**STA 2**

- **CTS:** $\text{MPDU} = x$

Data:

- **Data:** $\text{MPDU} = x$, $\text{NID} = y$, $\text{DA} = \text{STA 2}$, $\text{SA} = \text{STA 1}$

Ack:

- **Ack:** $\text{MPDU} = x$

---

**B2 Processing**

**Transmit RTS**

- **Compute MPDU**
- **Form RTS from MPDU, NID, final DA, Duration, & misc. bits**
- **Transmit RTS**
B2 Processing
Receive RTS

if (Infrastructure && Acting_as_AP && ToAP && NID == MyNID) { // AP
    receive RTS
    set NAV
    transmit CTS
}
else if (!Acting_as_AP && !ToAP && DA == MA) { // STA of some sort
    receive RTS
    ...
}
else {
    NAV = Duration
    ignore RTS
}

B2 Processing
Transmit CTS

Copy MPDUID and Duration from RTS into CTS
Transmit CTS
B2 Processing
Receive CTS

if (MPDUID == RTSMPDUID) {
    transmit data
}  
else {
    NAV = Duration
    ignore CTS
}

B2 Processing
Transmit Data

if (Infrastructure && Acting_as_AP) { // AP
    NID = MyNID, DA = True DA, SA = True SA
    ToAP = 0
}
else if (Infrastructure && !Acting_as_AP) { // STA in Infra
    NID = MyNID, DA = True DA, SA = MA
    ToAP = 1
}
else { // STA in ad-hoc
    NID = MyNID, DA = True DA, SA = MA
    ToAP = 0
}

Transmit Data Frame
B2 Processing
Receive Data

if (Infrastructure && Acting_as_AP && ToAP && NID == MyNID) { //AP
receive frame
NAV = Duration
transmit Ack
}
else if (!Acting_as_AP && !ToAP && DA == MA) { // STA
receive frame
...
}
else {
    NAV = duration
    ignore data frame
}

B2 Processing
Transmit Ack

Copy Data frame MPDU ID, Duration, and Frag# to the Ack
Transmit the Ack
B2 Processing
Receive Ack

if (MPDUID == DataMPDUID) {
    transmission complete
    next fragment
}
else {
    NAV = Duration
    ignore Ack
}
MPDUID Problems

• It's not very unique
  - Only two bytes
• Can cause non-duplicates to be rejected
  - If someone else hash's down to the same MPDU
• Can cause RTS/CTS/Data/Ack to be mismatched
  - Since frames are tied together using just MPDUID rather than the addresses of the source and destination, you could receive a CTS from someone else

MPDUID Fix

• Since you have a Seq # anyway, use it instead
• Perform duplicate detection using (SA, Seq #) pair
• Use directed RTS/CTS/Data/Ack to eliminate possibility of mismatching
MPDUID Fix Example

STA 1

RTS: DA = STA 2, SA = STA 1

CTS: DA = STA 1

Data: SEQ# = x, DA = STA 2, SA = STA 1

Ack: DA = STA 1

STA 2

MPDUID Fix Benefits

- You get better filtering
- You get better matching of RTS/CTS/Data/Ack
- You don't have to compute a hash
NID

- NID = concat(1-bit Inf/AdH, 13-bit ESSID, 10-bit BSSID)
- NID is overloaded
  - Used by APs as an address match field
  - ESSID portion is used by STAs to identify APs as belonging to an administrative domain
  - Used to identify an ad-hoc BSS

NID Problems

- It requires work to keep them unique
- Human must ensure that ESSID is unique for any overlapping ESSs
  - Difficult in multi-office buildings
  - It's only 13 bits, so there isn't much uniqueness
- Human must ensure that BSSID within ESSID is unique for all overlapping BSSs
  - Requires human to keep paperwork of which BSSIDs have been used and where they are
NID Fix

- Make NID = 48-bit address of AP
  - Now NID essentially is a BSSID
  - Ad-Hoc handled specially (later slide)
- Rename NID to "Via"
- Via amounts to the address of the AP a STA is associated with (MyAP)
- Leave ESSID only in Beacon/Probe Response frames
  - Since NID is now unique, ESSID is only needed to distinguish APs belonging to an administrative domain while looking for new APs to associate with
- Make ESSID a variable length string
  - No longer impacts header size

Example Cases
Case 1: From STA 1 to STA 2
enter DS through AP1

Case 2: From STA 1 to STA 2
exit DS through AP2
Case 3: Management frame from STA 3 to AP3

RTS: DA = AP3, SA = STA3
CTS: DA = STA3
Data: SEQ# = x, Via = AP3, DA = AP3, SA = STA3
Ack: DA = STA3

Case 4: Management frame from AP3 to STA3

RTS: DA = AP3, SA = AP3
CTS: DA = AP3
Data: SEQ# = x, Via = AP3, DA = STA3, SA = AP3
Ack: DA = AP3
Via and Ad-Hoc

- In Ad-Hoc BSSs, set the Via field to the originator of the frame

Case 5: Ad-hoc frame from STA 4 to STA 5

STA4

RTS: DA = STA5, SA = STA4

CTS: DA = STA4

Data: SEQ# = x, Via = STA4, DA = STA5, SA = STA4

Ack: DA = STA4
New Processing
Transmit RTS

if (Infrastructure && !Acting_as_AP) { // Infra STA
    RTS DA = MyAP
}
else { // AP or Ad-hoc STA
    RTS DA = Final DA
}
Fill in SA=MA & Duration
Transmit RTS

New Processing
Receive RTS

if (RTS DA == MA) {
    NAV = Duration
    receive RTS
    transmit CTS
}
else {
    NAV = Duration
    ignore RTS
}
New Processing
Transmit CTS

Copy RTS SA to CTS DA
Copy RTS Duration to CTS Duration
Transmit CTS

New Processing
Receive CTS

if (CTS DA == MA) {
    receive CTS
    transmit Data
}
else {
    NAV = Duration
    ignore CTS
}
New Processing
Transmit Data

if (Infrastructure && Acting_as_AP) { // AP
  Via = MA, DA = True DA, SA = True SA
  ToAP = 0
}
else if (Infrastructure && !Acting_as_AP) { // Infra STA
  Via = MyAP, DA = True DA, SA = MA
  ToAP = 1
}
else { // Ad-hoc STA
  Via = MA, DA = True DA, SA = MA
  ToAP = 0

  Fill in Duration, Frag#, etc.
  Transmit Data

New Processing
Receive Data

if (Acting_as_AP && ToAP && Via == MA) { // AP
  receive frame
  NAV = Duration
  transmit Ack
}
else if (!Acting_as_AP && !ToAP && DA == MA) { // STA
  receive frame
  ...
}
else {
  NAV = Duration
  ignore Data
}
New Processing
Transmit Ack

if (Acting_as_AP) { // AP
    Ack DA = Data SA
}
else { // STA
    Ack DA = Via
}
Copy Duration, Frag#, etc.
Transmit Ack

New Processing
Receive Ack

if (DA == MA) {
    transmission complete
    next fragment
}
else {
    NAV = Duration
    ignore Ack
}
### Processing Complexity Summary

<table>
<thead>
<tr>
<th>Action</th>
<th>Change (B2 to New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tx RTS</td>
<td>Much less (no MPDU ID computation)</td>
</tr>
<tr>
<td>Rx RTS</td>
<td>Slightly less</td>
</tr>
<tr>
<td>Tx CTS</td>
<td>Same</td>
</tr>
<tr>
<td>Rx CTS</td>
<td>Same</td>
</tr>
<tr>
<td>Tx Data</td>
<td>Same</td>
</tr>
<tr>
<td>Rx Data</td>
<td>Same</td>
</tr>
<tr>
<td>Tx Ack</td>
<td>Slightly more</td>
</tr>
<tr>
<td>Rx Ack</td>
<td>Same</td>
</tr>
</tbody>
</table>

Overall | Less complex

### Frame Fields Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>FC</th>
<th>Via</th>
<th>DA</th>
<th>SA</th>
<th>Seq#</th>
<th>Frag</th>
<th>Dur</th>
<th>B2 Len</th>
<th>New Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTS</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Data</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>Ack</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td>x**</td>
<td></td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

* Data frame duration is for next fragment

**Ack duration is duration of next fragment and is copied from Data frame duration
Via Fix Benefits

- Take advantage of the fact that all APs already have a unique identifier that is universally administered
- Network manager has virtually no administration to set up an AP
  - No longer needs to choose and administer a unique BSSID
- Better for “shrink-wrap” products

ESSID Details

- One role that NID filled was to identify APs belonging to an ESS via the ESSID portion of NID contained in Beacons and Probe Responses
- Do same thing but with larger, more descriptive field
ESSID Fix Benefits

- Since it's only in Beacon/Probe Responses, it doesn't have to be short
- It can be much more descriptive
  - E.G., "FooBar Corp. WLAN 1"
  - Helps diagnose problems
- Much less chance of collisions
- Better for "shrink-wrap" products

Frame Control Field

<table>
<thead>
<tr>
<th>Ver</th>
<th>Type</th>
<th>Subtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToAP</td>
<td>More</td>
<td>Retry</td>
</tr>
<tr>
<td>Pwr Mgmt</td>
<td>CF Ack</td>
<td>CF Poll</td>
</tr>
<tr>
<td>EP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>