Implementation of Time Division Multiply Access for mac80211 Wireless Stack
Implementation “A”. Release 3
Implementation “B”. Release 3
Brief overview
Content

Abstract..................................................................................................................................................3
Disclaimer..................................................................................................................................................3
Base Features...........................................................................................................................................3
Implementation details.............................................................................................................................5
Implementation Limitations......................................................................................................................14
External burst synchronization..............................................................................................................15
Conditions...............................................................................................................................................15
Brief Comparison with some TDMA-like Solutions............................................................................16
Tips and Tricks.........................................................................................................................................16
Spectral Efficiency and Hardware Recommendations...........................................................................18
How to configure.......................................................................................................................................18
Abstract

The document contains implementation details of Time Division Multiply Access for standard Linux wireless stack and ordinary 802.11abgn compatible hardware.

Disclaimer

NETSHe Lab Ltd. can modify this document as well as the described product at any time without any notifications.

Base Features

TDMA is implemented like collision-less multiply media access for ordinary IEEE 802.11 compatible hardware.

Every device in the wireless network (We call it the 'node') has own timeslot and send data to the network within assigned timeslot only.

TDMA is implemented as an additional interface mode of standard Linux wireless stack. The wireless stack with TDMA is absolutely backward compatible with unmodified and has some unique features.

The new functionality is accessible through standard Linux netlink NL80211 family API and through standard Linux user-space iw utility.

For example, iw utility has additional command:

- to create TDMA interface - iw phy0 interface add wlan0 type tdma;
- to join fair TDMA network with 12 nodes, bitrate 54Mb/s and frequency 2417MHz - iw wlan0 tdma join 2417 ts 12 ver 1 legacy 54;
- to join TDMA like network with 8 nodes, all possible bitrates and frequency 2417MHz - iw wlan0 tdma join 2417 HT20 ts 8 ver 1 mcs 0 1 2 3 …. 15;
- to leave the network - iw wlan0 tdma leave.

At the same time, iw keeps legacy command to perform additional tasks, like:

- to see stations in an associated and associating state - iw wlan0 station dump;
- to see interface information – iw wlan0 info.

TDMA implementation has forth different TDMA modes in one stack:

- «Fair» or v0;
- «TDMA mesh» or v1;
- TDMA Base-station (BS) or v2;
- and TDMA CPE or v3.
TDMA BS and CPE modes have configurable uplink/downlink ratio. We recognize TDMA BS and CPE modes as 'Point-to-Multipoint' (P2MP) modes, if the number of nodes in the network is greater than 2. We recognize these modes as 'P2P', if the number of nodes is two.

TDMA implementation provides symmetrical network by design. It means that all timeslots have single size. Timeslots in BS and CPE modes can have different slot sizes for transmit and receive. The transmit slot size for BS is equal to the receive slot size for CPE and the transmit slot size for CPE is equal to the receive slot size for BS. Polling extension makes different slot sizes possible for every mode. Real slot size depends on load and some other aspects and is limited up with configured value.

The number of nodes is configurable value. P2P mode operates with two nodes only. Current implementations require network restart to change the number of nodes.

TDMA implementation allocates timeslot for node in an automatic manner. Timeslot allocation procedure prevents 'hidden nodes issue' in the network.

At least, one data frame can be transmitted over the network per one timeslot. Next types of wireless frames exist in wireless TDMA network:

- beacons;
- data frames;
- ACK frames for modes without group acknowledgements and version 1 and above;
- polling extension produces action frames in some cases to indicate 'TX finish' state.

TDMA implementation recognizes three state for nodes in the network:

- 'Unknown' state;
- 'Associating' state;
- 'Associated' state.

Only nodes in 'Associated' state can send all types of wireless frames to the network.Nodes in 'Associating' state can send beacon frames to the network to allocate the
Nodes in 'Unknown' state can listen the network to synchronize own state.

TDMA implementation uses relative and multi-source synchronization model:

- The BS is the synchronization point for BS/CPE modes;
- For other modes, every node in 'Associated' state is the source to synchronize clock and events for the others.

Synchronization process is permanent.

The implementation uses Individual frame acknowledgement for implementation “A” (as described in 802.11-family standards) and unique bulk (or group) acknowledgement for implementation “B”.

TDMA uses '4 addresses' wireless frame format for data frames. Thus, wireless interfaces in TDMA modes can be bridged.

TDMA implementation can support any wireless device that:

- has driver in mac80211 stack;
- supports AdHoc mode;
- uses Minstrel rate control algorithm.

The implementation has minimal hardware driver-specific part.

TDMA has internal mesh-similar protocol to exchange network information and allocate timeslots. Beacon frames are used to propagate service information over to the network. Protocol avoids the 'hidden nodes issue' and increases network capacity.

The nodes can use the same timeslots, if they do not have direct connection and general neighbours.

The implementation has initial support for mobile nodes.

**Implementation details.**

TDMA adds capability to create new wireless interface with type TDMA. It can be used through iw utility and/or through netlink calls.
TDMA interface does not move frames from interface queues to driver queues directly, but moves frames to own internal queue. TDMA interface releases frames from own queue to the hardware within the timeslot only. Thus, node transmits frames within own timeslot.

TDMA interface uses high resolution timer for synchronization.

To limit CSMA/CA contention challenge and have predicable transmission time, implementation minimizes contention related values. The implementation does not disable 'carrier ready check' procedure.

The implementation uses only one internal queue to manage the transmission.

Every node in 'ASSOCIATED' state has own assigned timeslot. It uses own timeslot to transmit the data over to the network. The node transmits at least two frames per timeslot. The first transmitted frame is the beacon frame. It is a mandatory frame. When the beacon is transmitted, the node transmits one or more data frame from the internal queue.

TDMA uses interrupts to find when the timeslot begin and to start data frame transmission from the internal queue.

Below we will consider frame transmit process for implementations “A” and “B” separately.

**Implementation “A”**

The wireless stack transmits all data frames with single data-rate and without acknowledgements in case of «Fair TDMA» (v0) mode. Thus, We do not have any form of retransmissions in this case and transmission time is predicable.

The stack transmits only one data frame and uses transmission status callback to try to transmit next frame in case if other TDMA modes. The stack finishes transmission, if predicated frame transmission time exceeds remained slot time.

The described procedure can produce collisions or inefficient using of slot time, because transmission time can not be predicted exactly.

To make transmission time much predicable, We use '3' as maximal number of
transmission retries.

**Implementation “B”**

The node transmits all data frames without individual acknowledgements. Acknowledgements data is placed into beacon frames and used for retransmission and rate control.

**Timeslot Size**

The developers call as 'Round' the set of timeslots for configured number of nodes. The timeslot size is calculated during the TDMA interface initialization process. It is based on the specified data-rate, current MTU size for the interface, tx/rx ratio for BS (for P2P and P2MP modes) and number of nodes in the network.

Below is showed the simplified timeslot length calculation:

- It will be $T_{\text{unaligned timeslot}} = T_{\text{beacon}} + T_{\text{sifs}} + T_{\text{mtu+llcheader}}$ for «Fair TDMA» and implementation “B”;
- and will be $T_{\text{unaligned timeslot}} = T_{\text{beacon}} + T_{\text{sifs}} + 3 \times T_{\text{mtu+llcheader}}$ for other modes.

Beacon frame has fixed maximal length, thus $T_{\text{beacon}}$ is the fixed value for the specified data-rate and mode. $T_{\text{sifs}}$ is the fixed value in 10 microseconds.

To calculate transmission time for data frames, We use preconfigured speed for «Fair TDMA» or minimal available speed (e.g. 1Mb/s for 20MHz width and 802.11g) for others. To have compatibility with some (usually old) wireless devices, We align timeslot length to the TU value (time unit — 1024 microseconds).

$$T_{\text{TUaligned}} = \left\lfloor \frac{T_{\text{unaligned timeslot}}}{1024} \right\rfloor + 1 \times 1024$$

At the next step, We align timeslot length to match $T_{\text{round}}$ to the minimal round size in $12\text{TU}$.

$$T_{\text{round}} = T_{\text{TUaligned}} \times N_{\text{nodes}}$$

$$T_{\text{round}} = \left\lfloor \frac{T_{\text{round}}}{1024} \right\rfloor \times 12 \times 1024; T_{\text{round}}$$
\[ T_{\text{aligned}} = \left\lceil \left( T_{\text{round}} / N_{\text{nodes}} \right) / 1024 \right\rfloor + 1 \times 1024 \]

After all transformations, we have some value for the timeslot size

\[ T_{\text{aligned}} - T_{\text{beacon}} - T_{\text{sifs}} \geq T_{\text{mtu+llheader}} \]

Thus, we provide capability to transmit at least one data frame within the timeslot and more frames. When the actual data-rate is bigger than preconfigured.

TDMA has hard-coded minimal timeslot sizes for every mode. TDMA has hard-coded minimal round size for every mode.

The developers added capability to have various preconfigured timeslot sizes for all modes except “Fair TDMA”. Every preconfigured timeslot sizes is bigger or equal than minimal and can be used in an appropriate case. We call preconfigured timeslot sizes as “Large', "Medium', 'Small' and 'Extra small'.

So, the default timeslot size is around 41 milliseconds for implementation “A” and 802.11g. It is suitable for long distance network and high noised environments. We call it as 'Large' timeslot. Also, timeslots with 'Medium' size (around 20 milliseconds), 'Small' size (around 10 ms) and 'Extra small' (around 9ms) size are available.

'Extra small' time slot size is 6ms and all others are 8ms for implementation “B”.

All typical timeslot sizes are showed below in a table form for both implementations.

**Implementation “A”. Time slot sizes**

<table>
<thead>
<tr>
<th>#</th>
<th>Channel width</th>
<th>Extra small</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5MHz</td>
<td>31 ms</td>
<td>34 ms</td>
<td>38 ms</td>
<td>45 ms</td>
</tr>
<tr>
<td>2</td>
<td>10MHz</td>
<td>18 ms</td>
<td>22 ms</td>
<td>30 ms</td>
<td>45 ms</td>
</tr>
<tr>
<td>3</td>
<td>20Mhz</td>
<td>10 ms</td>
<td>12 ms</td>
<td>26 ms</td>
<td>45 ms</td>
</tr>
<tr>
<td>4</td>
<td>40Mhz</td>
<td>10 ms</td>
<td>12 ms</td>
<td>26 ms</td>
<td>45 ms</td>
</tr>
</tbody>
</table>

**Implementation “B”. Time slot sizes**

<table>
<thead>
<tr>
<th>#</th>
<th>Channel width</th>
<th>Extra small</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5MHz</td>
<td>16 ms</td>
<td>20 ms</td>
<td>20 ms</td>
<td>20 ms</td>
</tr>
<tr>
<td>2</td>
<td>10MHz</td>
<td>10 ms</td>
<td>14 ms</td>
<td>14 ms</td>
<td>14 ms</td>
</tr>
<tr>
<td>3</td>
<td>20Mhz</td>
<td>6 ms</td>
<td>8 ms</td>
<td>8 ms</td>
<td>8 ms</td>
</tr>
</tbody>
</table>
The available time slot sizes are limited by the wireless hardware. Some old radios do not support time slot sizes less than 8ms.

Timeslot size defines latency in the wireless network. Latency will be described below.

It should be noted, that 'Small' and 'Extra small' timeslot sizes provide low latency and lesser throughput due to higher overhead. Also, network with small timeslot size is less suitable for noised environment.

**Polling**

Polling extension allows the current node in wireless network to inform next node that TX is finished (no more frames in the queue).

The node can use two ways to inform next node:
- Set special flag in the last frame;
- or send special action frame.

Current node finishes transmission immediately when information is sent.

Polling must be setuped manually through iw utility (when interface is setuped) or through WebUI.

Polling does not work when next timeslot is not allocated.

Polling makes using of timeslot much efficiency (especially in cases of under-load), reduces latency and can show better performance in some cases. But it may provide throughput degradation in noisy environment.

Polling extension is available for release 3 and later.

### Frame Transmission chart

Figure 1 shows transmission chart for «Fair TDMA» and implementation “B”.

<table>
<thead>
<tr>
<th>#</th>
<th>Channel width</th>
<th>Extra small</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>40Mhz</td>
<td>6 ms</td>
<td>6 ms</td>
<td>6 ms</td>
<td>6 ms</td>
</tr>
</tbody>
</table>
**Node Living Sequence in TDMA Network**

The node starts in the 'Unknown' state and chooses number of rounds to be in 'Unknown' state randomly to prevent races between two or more nodes in this state.

\[ N(r) = \text{Random}(\text{UNKNOWCOUNTMIN, UNKNOWCOUNTMAX}) \]

The node chooses own timeslot randomly and moves itself in the 'ASSOCIATED' state, when it is in 'UNKNOWN' state and does not receive beacons from nodes in 'ASSOCIATED' state within \( N(r) \) rounds.

The node listens the network, collects information about neighbours, assigned timeslots and performs internal time and event synchronization, when it is in 'UNKNOWN' state and receives beacons from nodes in 'ASSOCIATED' state within \( N(r) \) rounds.

When \( N(r) \) time is gone, the node has synchronized internal clock and network map. The node chooses unassigned timeslot randomly and moves in the 'ASSOCIATING' state.

Nodes in 'ASSOCIATED' and 'ASSOCIATING' state transmit beacons frame with some information. They include in the beacon information about all visible and right configured neighbours.
If the node in 'ASSOCIATING' state receives own MAC-address/node ID in beacons from every valid neighbour within predefined time, the node moves nit the 'ASSOCIATED' state. The node moves reset in the 'UNKNOWN' state in other case.

Every node perform internal synchronization permanently (when valid beacon is received). They use nodes in 'ASSOCIATED' state as a synchronization sources. The software calculates start of zero timeslot on the base of beacons from nodes in 'ASSOCIATED' state and uses high resolution timer to adjust start of self beaconing to appropriate timeslot.

Additional Features

The software supports WEP-40, WEP-104 and AES128 (CCMP) data encryption for every node.

TDMA supports frame retransmission through standard 'ACK' wireless frames in "TDMA mesh", "P2P TDMA" and "P2MP" modes and implementation "A".

TDMA supports frame retransmission through unique 'Bulk ACK' method in implementation "B".

The software supports variable data-rates and rate control in "TDMA mesh", "P2P" and "P2MP" modes.

TDMA supports NETSHe software frame compression method. LZO compression is used for data frames. Software transmits uncompressed frame, if compressed frame is bigger than original.

TDMA has built-in software A-MSDU aggregation and provides it everywhere. The aggregation process is tune-able by user.

The software can do frame reordering within internal queue to optimize throughput. Frame reordering is activated by default and can be disabled by user request. The reordering process is showed in the figure below.
As showed in the figure above, we did not reach the timeslot end and have some time for transmission. The first frame in the queue is too large to be transmitted within remained timeslot. When the reordering is activated, the second frame will be transmitted within this timeslot because its size matches remained timeslot size.

The frame reordering increases throughput in most cases, especially in case of very non-uniform traffic with small- and medium-sized frames. But it may have negative effect in some cases with TCP (in cases of high latency) or UDP (voice and video) applications.

The latency for 'request-reply' typed traffic (TCP/ICMP) depends on distance in timeslots between nodes and timeslot size.

'Group acknowledgement' for all TDMA modes is available since release 2. It avoids individual frame acknowledgements and uses "Group" acknowledgement for "TDMA mesh", "BS" and "CPE" modes. 'Group acknowledgements' uses beacon frames to propagate information about received data frames within prior round.

'Group acknowledgements' can be activated within configuration process and uses variable length beacon frames.

**Frame Transmission Sequence**

The frame transmission sequence for Implementation "A" (except 'TDMA fair mode') is showed below.
The frame transmission sequence for implementation “B” is showed below.

Timeslot charts.

Timeslot chart for 'Fair TDMA ' (v0) and 'TDMA like' (v1) modes with N nodes is showed below.

Timeslot chart for 'TDMA P2P' mode is showed below.
Timeslot chart for 'TDMA P2MP' mode with N nodes is showed below.

**Implementation Limitations**

TDMA does not support Block acknowledgements in all modes and any form of retransmissions in «Fair TDMA» mode without 'Group acknowledgement'.

TDMA supports only one preconfigured data-rate in “Fair TDMA” mode. TDMA does not provide data-rate adaptation and acknowledgements in «Fair TDMA» mode (without 'Group acknowledgement').

“Fair TDMA” mode supports maximal symbol rate in 54Mb/s.

TDMA supports configurable upload (UL)/download (DL) ratio in «"P2P" and “P2MP” modes.

All nodes must have the same MTU.

Nodes in different TDMA modes can not be mixed in one network.
We offer to use batman_adv to create full operate mesh network for “Fair TDMA” and “TDMA mesh” modes.
External burst synchronization.

It should be noted, that external burst synchronization is excluded from the release 2.

External burst synchronization provides time synchronization of zero timeslot for every device over the wired network.

To use external burst synchronization, the devices should be synchronized via GPS/GLONASS and/or NTP.

To do burst synchronization, the software uses multicast addresses from 233.0.255.1 and ports from 7000.

The device must have own unique identifier to be involved in the burst synchronization as a source. The identifier must be configured manually during the set-up process. The software will send synchronization packets to the address/port pairs according to the specified identification.

Every device must use identifiers of the sources to be involved in the burst synchronization as a client.

The identifiers must be configured manually during the set-up process. The software will read synchronization packets from the address/port pairs according to the specified identifications.

Conditions

«Fair TDMA» mode is assigned to version 0 of protocol (implementation).

«TDMA mesh» mode is assigned to version 1 of protocol.

«TDMA Base-station» mode is assigned to version 2 of protocol.

«TDMA CPE» mode is assigned to version 3 of protocol.

Minimal round size is around 12 milliseconds. Thus, for P2P mode with symmetrical UL/DL ratio the minimal timeslot size is around 6ms. We do not support lesser round size due to
synchronization troubles. For 5GHz band, the minimal timeslot size is around 6 (Implementation “B”)/10 (implementation “A”) ms (equal to the 'Extra small' timeslot size).

**Brief Comparison with some TDMA-like Solutions**

Current TDMA implementation is closer to fair TDMA. It is not polling-based solution and it is not similar to WiMAX. Of course, it has own pros and cons.

**Cons:**
- TDMA has bigger latency than in typical polling based solution. *
- TDMA has lesser throughput that in typical polling based solution. *

**Pros:**
TDMA has better throughput in nosed environment and for long distance links. TDMA provides unique flexibility to adopt solution to any form of links and appliances. TDMA provides mesh-similar modes with decentralized coordinate function. TDMA supports 802.11, 802.11a, 802.11b, 802.11g, 802.11n modes. TDMA supports HT data-rates for 2.5, 5 and 10MHz channel widths.

* TDMA implementation “B” (group/bulk acknowledgements) has throughput and latency which is very close to the best CSMA/CA and polling based solutions.

For the release 3, polling makes the solution very close to typical polling-based solutions.

**Tips and Tricks**

Our solution is very suitable for high noised environment and long distance links. We share some hints below to have efficiency solution with our software.
Hints to have Maximal Throughput

Do not use “Fair TDMA" mode.
Enable A-MSDU aggregation.
Enable Frame reordering.
Find optimal timeslot size in an experiment.
Use compression if your hardware allows it.
Use bulk acknowledgements if noises is low and environment is clear.

Hints to have Minimal Latency

Enable A-MSDU aggregation.
Disable Frame reordering.
Use minimal possible timeslot size.
Do not use encryption and compression.
Use polling.
Use bulk acknowledgements if noises is low and environment is clear.

Hints for noisy Environments to have Maximal Throughput and Stability

Do not use “Fair TDMA” mode and sub-modes with “Group/bulk acknowledgements” for such environments.
Do not use polling.
Disable A-MSDU aggregation.
Disable Frame reordering.
Find optimal timeslot size in an experiments.

Hints for Long Distance Links

Enable A-MSDU aggregation.
Enable Frame reordering.
Configure link distance properly.
Use mid-sized timeslots.

Use polling.

Select “TDMA fair” mode and (or) modes with “Group acknowledgements” if the environment is clear.

**Spectral Efficiency and Hardware Recommendations**

Our solution has maximal spectral efficiency more than 2 bits per Hz for 1-MIMO (See last test results). It means that the aggregated network throughput will be around 20Mbps for 802.11n 2-MIMO radios and 5MHz channel width. The aggregated throughput will be more than 70Mbps for 2-MIMO hardware and 20MHz channel width.

It should be noted, that the spectral efficiency is better for the narrow channels.

The implementation has best results for radios with support of short guard intervals for 20MHz and 40MHz channel widths, RX STBC, maximal A-MSDU size up to 8KB and LDPC.

**How to configure**

**Configuration NETSHe-based Firmware**

To configure TDMA, choose mode «TDMA» in the list in wireless device configuration page, frequency and right modulation.

It should be noted, that you must not choose 'Auto' in the Frequency list and must choose the same frequency for all modes in the network.

It should be noted, that you can choose different modulations for TDMA version 1, 2 and 3.

**Encryption Settings**

You can use one of three types of Encryption:

- No encryption as showed in a Figure below.
- WEP encryption as showed in a Figure below. To use WEP, you must set-up WEP
key.

- WPA2 (Personal) encryption (CCMP with AES-128). To use WPA2 encryption, you must set up AES-128 key.

It should be noted, that the key for WEP-40 encryption must have 5 symbols. The key for WEP-104 must have 13 symbols. The key for AES-128 encryption must have 16 symbols.

No encryption.
WEP-40 encryption.

WEP encryption

WEP-104 encryption
AES-128 encryption.

To configure TDMA version, please, choose right version in the list as showed below

TDMA version 0 («Fair TDMA»).

It should be noted, that you must choose exact and the same data-rate for every node in
TDMA version 1 («TDMA mesh»).

Operational channel width and 802.11n modes should be configured as for other types of wireless interfaces and should be the same for all nodes in the network.
TDMA version 2 ("TDMA Base-station")

The base-station in the TDMA network can be identified by SSID or MAC-address. To provide MAC-address-based identification, please leave SSID field empty. To provide SSID-based identification, please specify valid SSID. BS and every CPE in the network must be configured with the same frequency and channel width. CPE obtains settings like number of nodes in network, tx/rx ratio, timeslot size, frame reordering and bulk acknowledgement from the base-station.

TDMA version 3 ("TDMA CPE").

To configure CPE, you must specify right mode, frequency, channel width, encryption and network identification (via SSID or BS MAC-address). CPE obtains other settings from the Base-station. Settings for SSID identified network are showed below.
CPE settings with MAC-based identification are showed below.

Fine tuning of TDMA Network

Fine tuning of settings must be done for every node in the network manually. Frame
reordering and bulk acknowledgement settings for CPE mode can not be tuned manually (CPE obtains it from the Base-station).

All tunable settings are placed to the tab 'Advanced' in wireless configuration page. The settings are:

- frame TX optimization (reordering)
- bulk acknowledgements
- disabling of A-MSDU