This document will continuously be updated.

1. Overview

The A-DCF simulator is based on NS-3. We extend the wifi module to implement A-DCF and O-DCF. For details on NS-3, refer to http://www.nsnam.org.

2. Building the simulator

Decompress adcf-simulator-0.1.tar.bz2 by typing the following:

```
tar xvjf adcf-simulator-0.1tar.bz2
```

Change into the directory by typing the following:

```
cd ns-allinone-3.14.1-adcf-0.1
```

Build by typing the following:

```
./build.py
```

3. Running a scenario

Change into the directory by typing the following:

```
cd ns-3.14.1
```

Run the FIM scenario by typing the following:

```
./waf --run tcp-fim
```

4. Script details

You will find sample scenario scripts in directory scratch/, where tcp-fim.cc and tcp-fc.cc
correspond to flow-in-the-middle and fully-connected topologies, respectively. You can change the simulation configuration by changing values for appropriate variables. To use 802.11 DCF, set \texttt{macType} to “\texttt{ns3::AdhocWifiMac}”. To use O-DCF, set \texttt{macType} to “\texttt{ns3::ODcfAdhocWifiMac}” and set \texttt{ta} to “queue”. To use A-DCF, set \texttt{macType} to “\texttt{ns3::ODcfAdhocWifiMac}” and set \texttt{ta} to “delay”.

5. Implementation details

We implement A-DCF and O-DCF by extending the \texttt{wifi} module. You will find source codes for A-DCF and O-DCF in \texttt{src/wifi/model/} directory. Main classes are \texttt{ODcf}, \texttt{ODcfQueue}, and \texttt{ODcfDcaTxop} which are in \texttt{odcf.h}, \texttt{odcf-queue.h}, and \texttt{odcf-dca-txop.h}. 