CHAPTER 4

Simulation and Evalueation

4.1 Simulation

We use NS-2 (Network Simulator ver. 2.28) with TKN 802.11e extended module for network simulation. The video codec is H.264/AVC JM (Joint Model). We also modified myEvalvid-NT (myEvalvid Network Trace) [19] for emulation.

4.1.1 myEvalvid-NT

The operation flow of myEvalvid-NT for H.264/AVC is shown in Figure 4.1.

1. Encode raw YUV video file into H.264/AVC format using JM.
2. Parse the H.264/AVC video file into traffic trace file. Traffic trace file contains packet id, send time and packet size used for NS2.
3. After simulation, NS2 generates sender trace file and receiver trace. Sender file contains packet id, order and send time. Receiver file contains packet id, order and receive time.
4. Calculate packet loss rate, delay time, jitter and throughput using trace file.
5. Reconstruct erroneous H.264 video file using trace file and Error Inserter.
6. Decode H.264/AVC video file into fixed YUV video using JM.
7. Compare video quality between raw and fixed YUV videos in terms of PSNR.

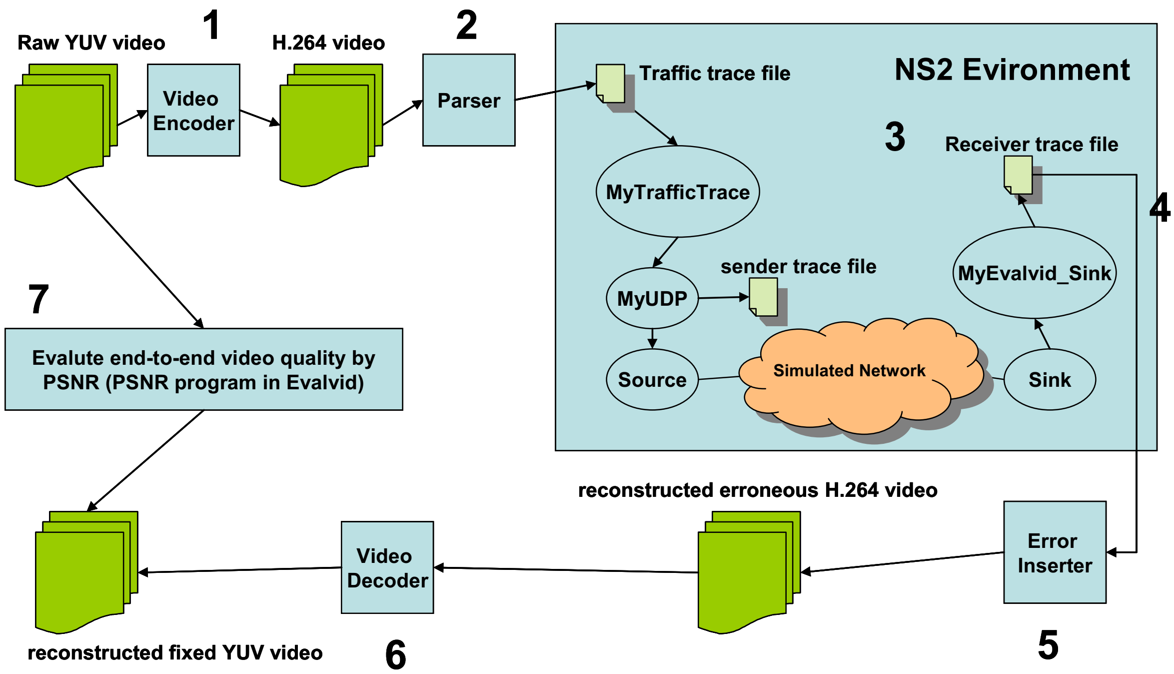

Figure 4.1: myEvalvid-NT for H.264/AVC.

4.1.2 Scenario and parameter

The simulation scenario is as follows. There are nine nodes in the network and these nodes are divided into two overlapped groups of six nodes as shown in Figure 4.2. Each node in the same group is assumed to be within each other's transmission range. Link bandwidth is set 1Mbps. Assume that there are My_video, CBR and FTP three traffics. My_video is the main traffic to observe, whereas FTP and CBR traffics are used to interfere with My_video traffic. We start to transmit CBR traffic at first, and start My_video at random between 1.0 second and 1.1 second. Finally, we transmit FTP traffic after My_video is transmitted in the halfway.

Assume the My_video traffic is film Foreman or film Stefan, film length of Foreman is 10 seconds, film length of Stefan is 3 seconds, resolution is 352×288 CIF, video rate is 30 FPS, compression ratio of JM is 17, image compression sequence is IPBBPBBPBB, GOP (group of picture) is 10. We have three experimental groups and one control group. Three experimental groups, IEEE 802.11e Timebase, IEEE 802.11e MultiQ (single-video multi-level queue) and IEEE 802.11e Hybrid, are our proposed methods. The control group is IEEE 802.11e. Performance metrics are average packet loss rate, invalid packet ratio, lost and invalid packet ratio, delay time, jitter, throughput and PSNR.

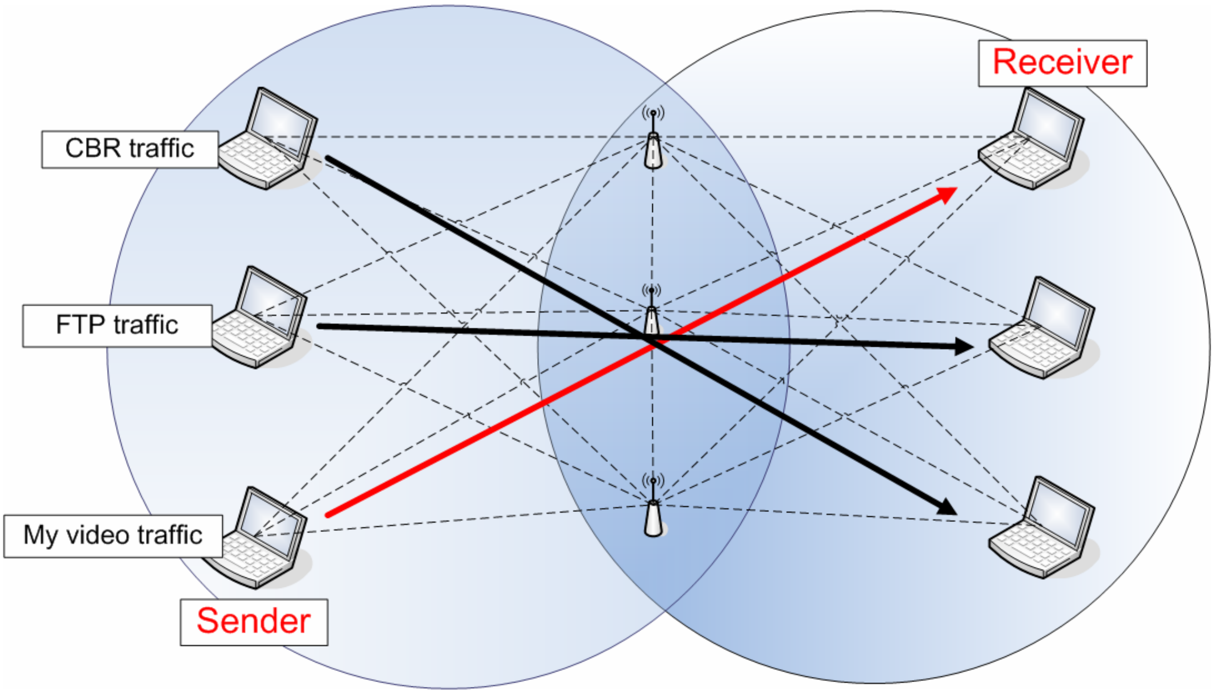

Figure 4.2: Proposed scenario.

4.1.3 Evaluation

We use PSNR (Peak Signal to Noise Ratio) [21] for our evaluation. PSNR is usually expressed in terms of the logarithmic decibel scale.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (4.1)$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\| I(i,j) - K(i,j) \right\|^2 \quad (4.2)$$

The PSNR is most commonly used as a measure of quality of reconstruction in image compression etc. It is most easily defined via the mean squared error (MSE) which for two m×n monochrome images I and K where one of the images is considered a noisy approximation of the other is defined as Formula 4.2 and The PSNR is defined as Formula 4.1. MAX_I is the maximum pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better. [20] maps PSNR to 5 MOS (Mean Opinion Score) level for evaluation (Table 4.1), that we can understand quality of the PSNR for video.

PSNR[dB]	MOS
>37	5(Excellent)
31-37	4(Good)
25-31	3(Fair)
20-25	2(Poor)
<20	1(Bad)

Table 4.1: PSNR and MOS [20].

4.2 Simulation results and analysis

Figure 4.3 to Figure 4.10 show the performances of packet loss rate. IEEE 802.11e Hybrid has close to zero I/P packet loss rates, however, its B packet loss rate is higher than the other groups. As mentioned above I-slice (packet) is the most important slice, P-slice is less important and B-slice is the least important. We trade poor B packet loss rate for better

I/P packet loss rates. Figure 4.11 to Figure 4.18 show the performances of invalid packet ratio. If a system is transmitting an overdue packet then this packet is called invalid. Invalid packet ratio is the ratio of the number of invalid packets to total number of packets. We see that IEEE 802.11e Hybrid has close to zero invalid packet ratio since we strictly limit transmission time within deadline. Figure 4.19 to Figure 4.26 show the performances of lost and invalid packet ratio. This is the ratio of the number of both lost and invalid packets to total number of packets. We see that IEEE 802.11e Hybrid is lower than all the other three. Figure 4.27 to Figure 4.30 show the average delay time and the average jitter, and IEEE 802.11e Hybrid has the best performances. Figure 4.31 and Figure 4.32 show the performance of average throughput. IEEE 802.11e Hybrid still shows good, thought not the best, performance. Finally, IEEE 802.11e Hybrid shows the best performance in PSNR as shown in Figure 4.33 and Figure 4.34. As a whole, simulations show that our IEEE 802.11e Hybrid, IEEE 802.11e Timebase and IEEE 802.11e MultiQ outperforms IEEE 802.11e in packet loss rate, invalid packet ratio, lost and invalid packet ratio, delay time, jitter, and PSNR.

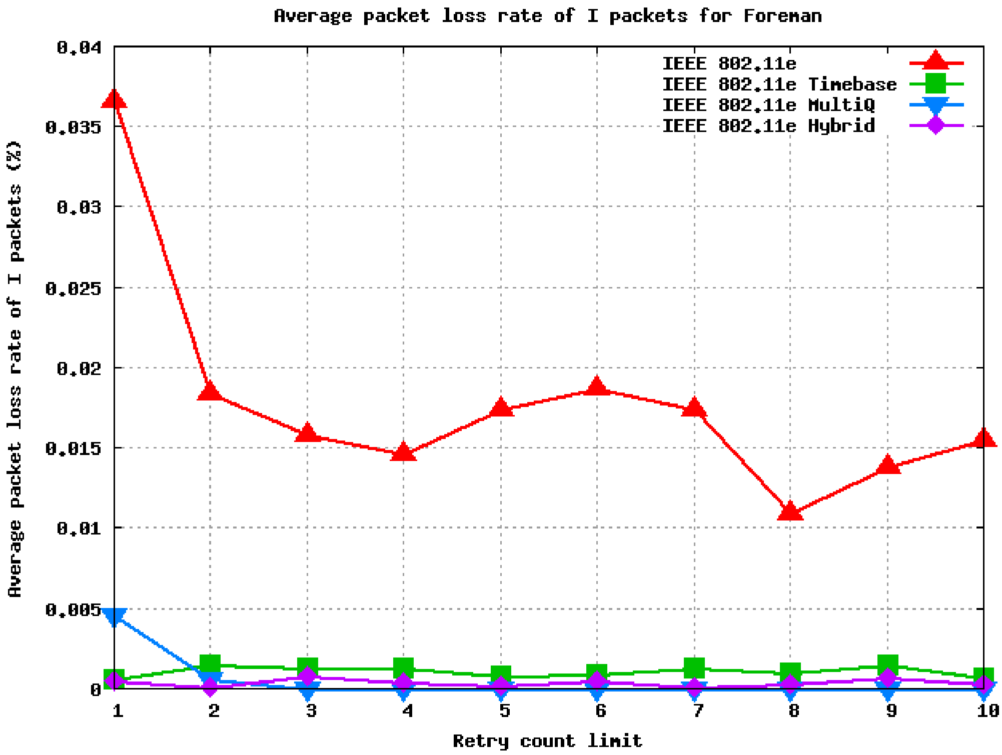

Figure 4.3: Average packet loss rate for I packets against retry limit for Foreman.

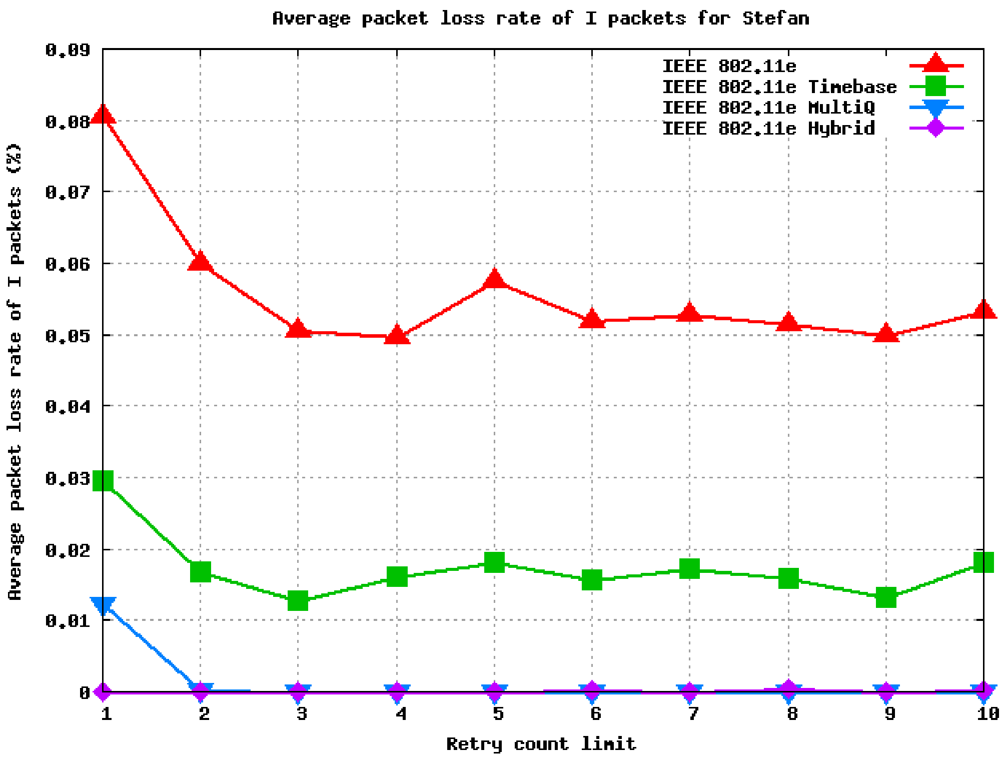

Figure 4.4: Average packet loss rate for I packets against retry limit for Stefan.

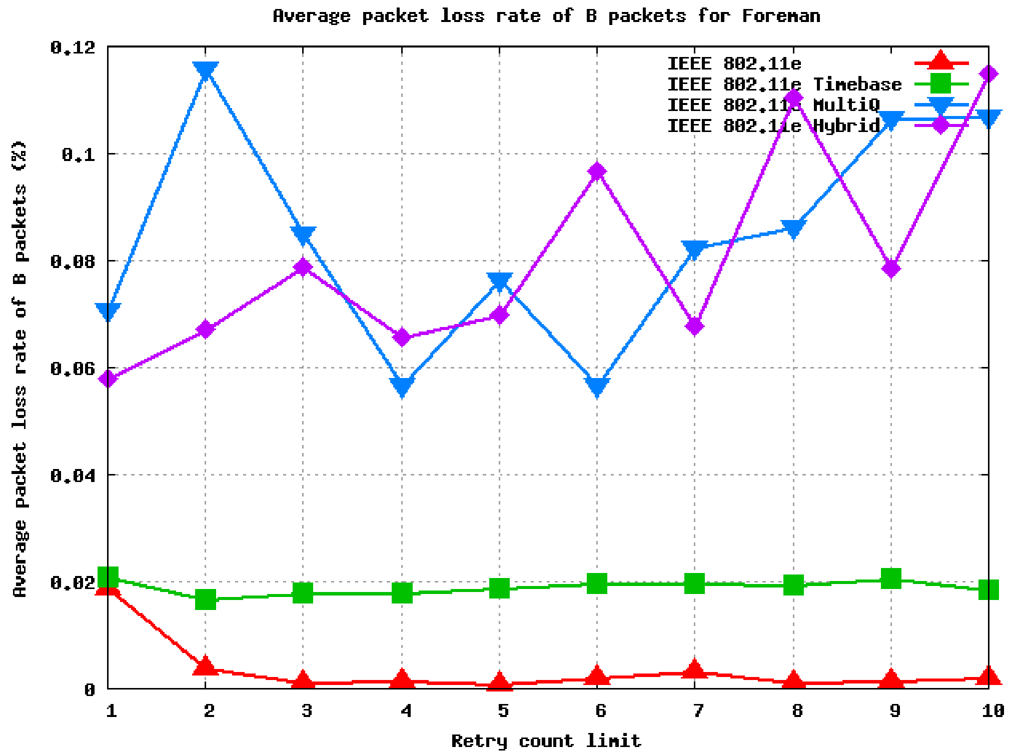

Figure 4.5: Average packet loss rate for B packets against retry limit for Foreman.

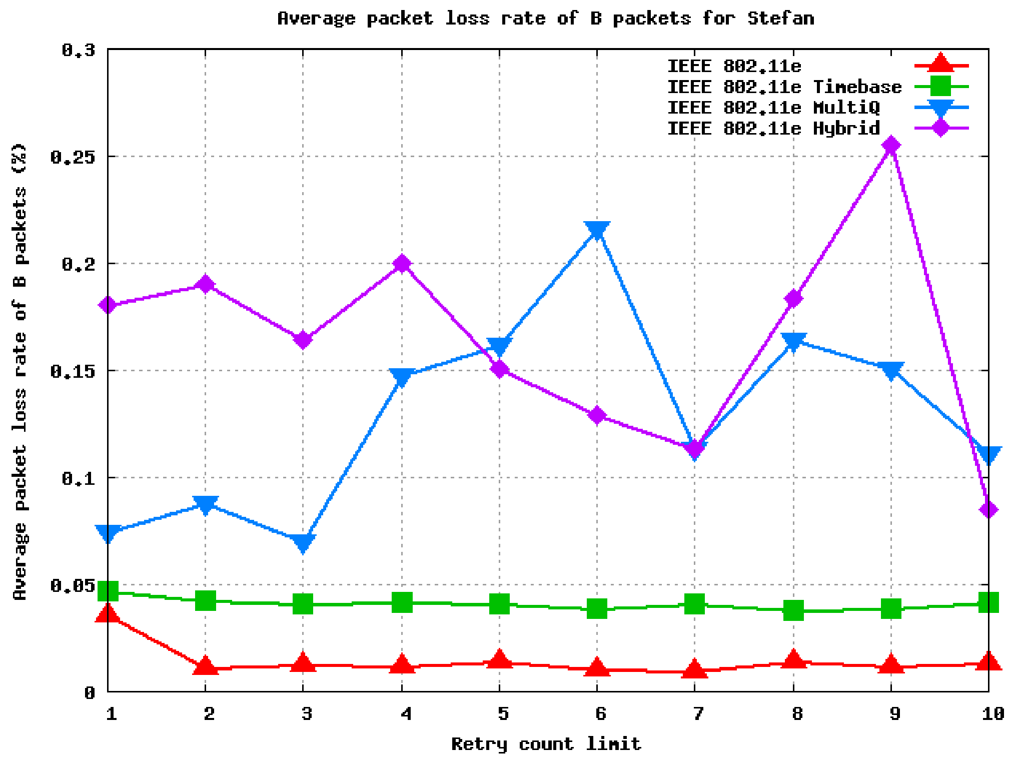

Figure 4.6: Average packet loss rate for B packets against retry limit for Stefan.

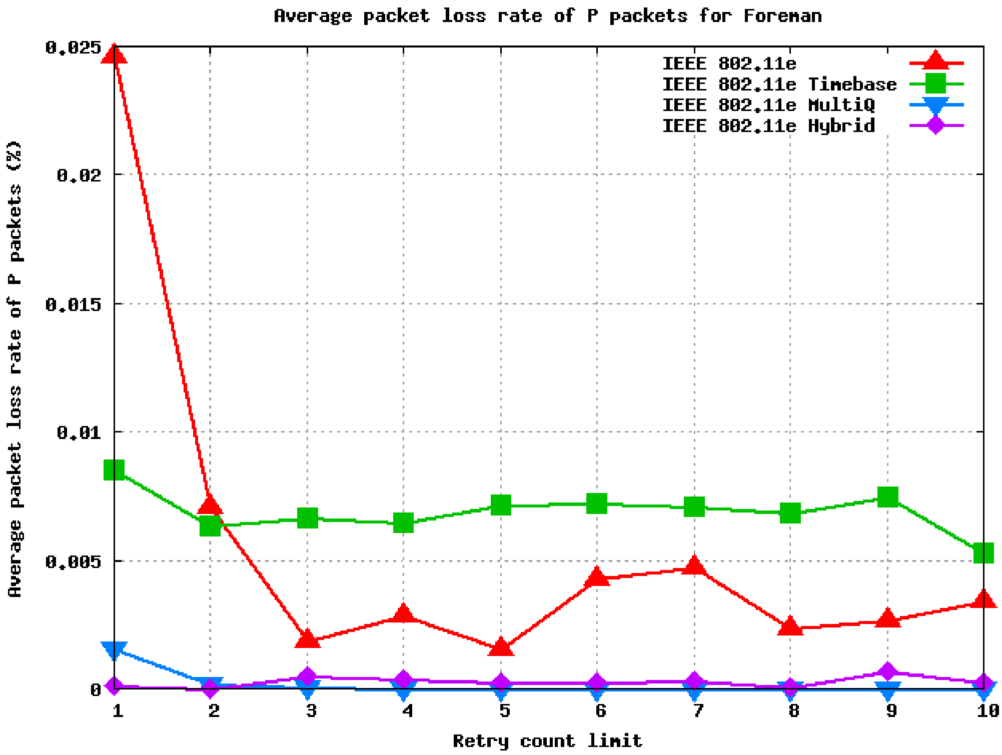

Figure 4.7: Average packet loss rate for P packets against retry limit for Foreman.

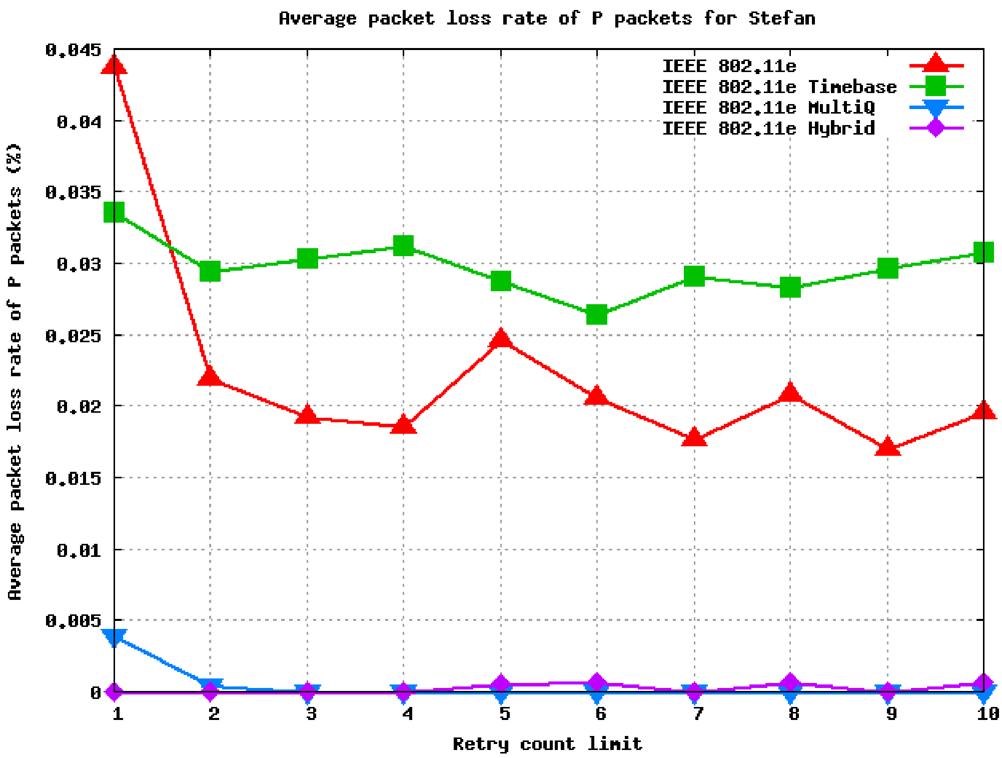

Figure 4.8: Average packet loss rate for P packets against retry limit for Stefan.

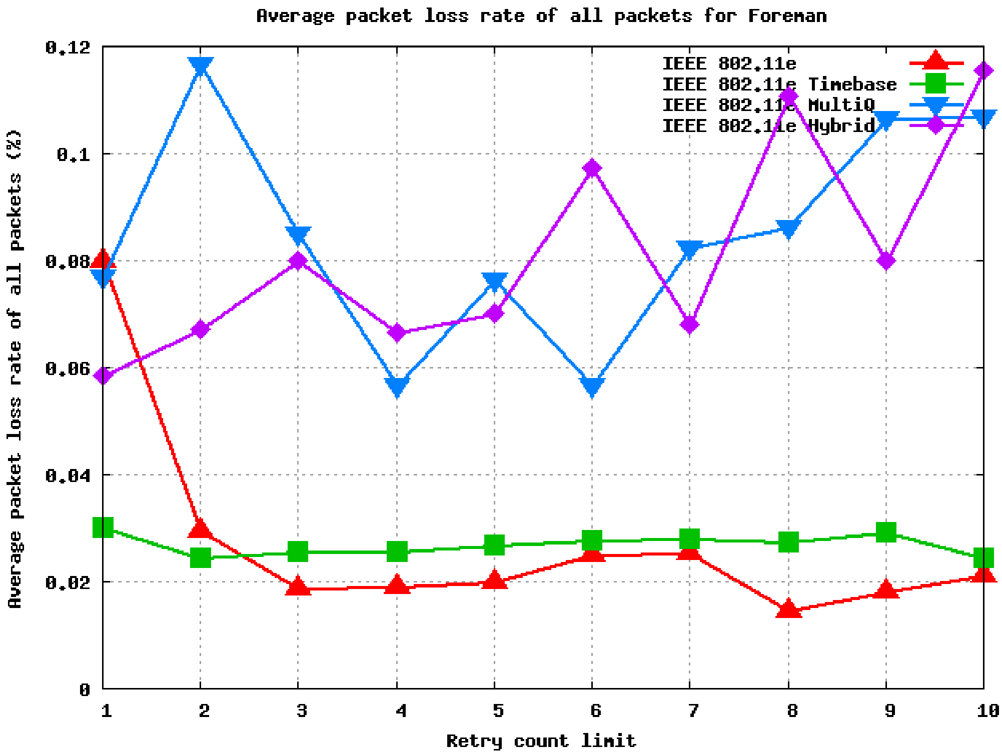

Figure 4.9: Average packet loss rate for all packets against retry limit for Foreman.

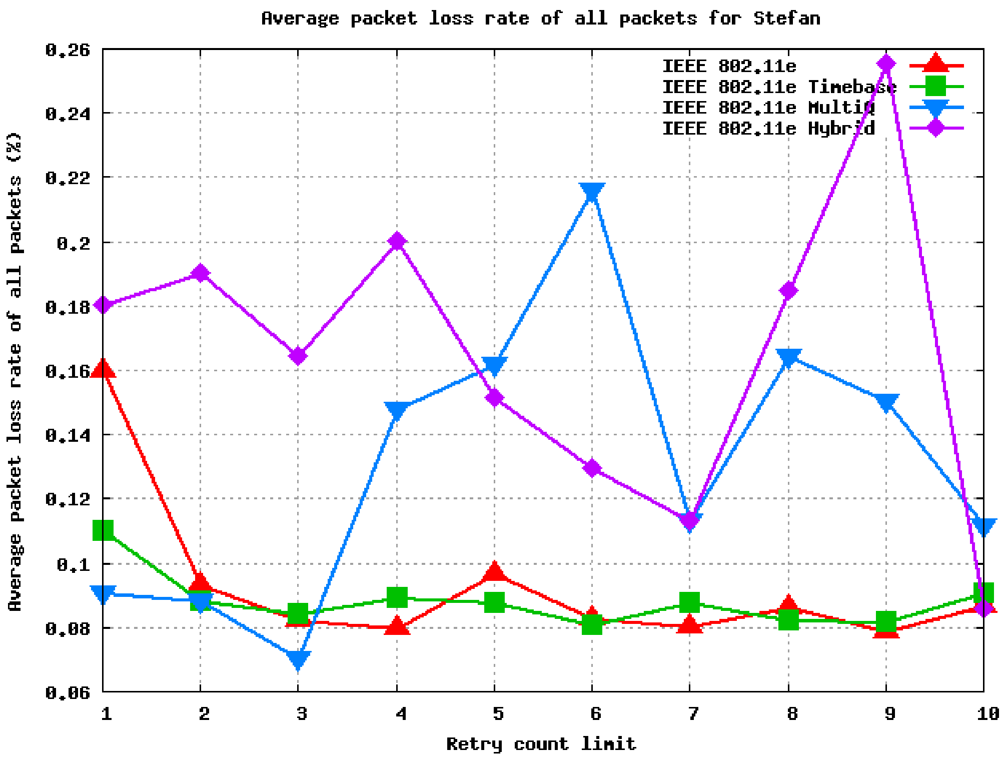

Figure 4.10: Average packet loss rate for all packets against retry limit for Stefan.

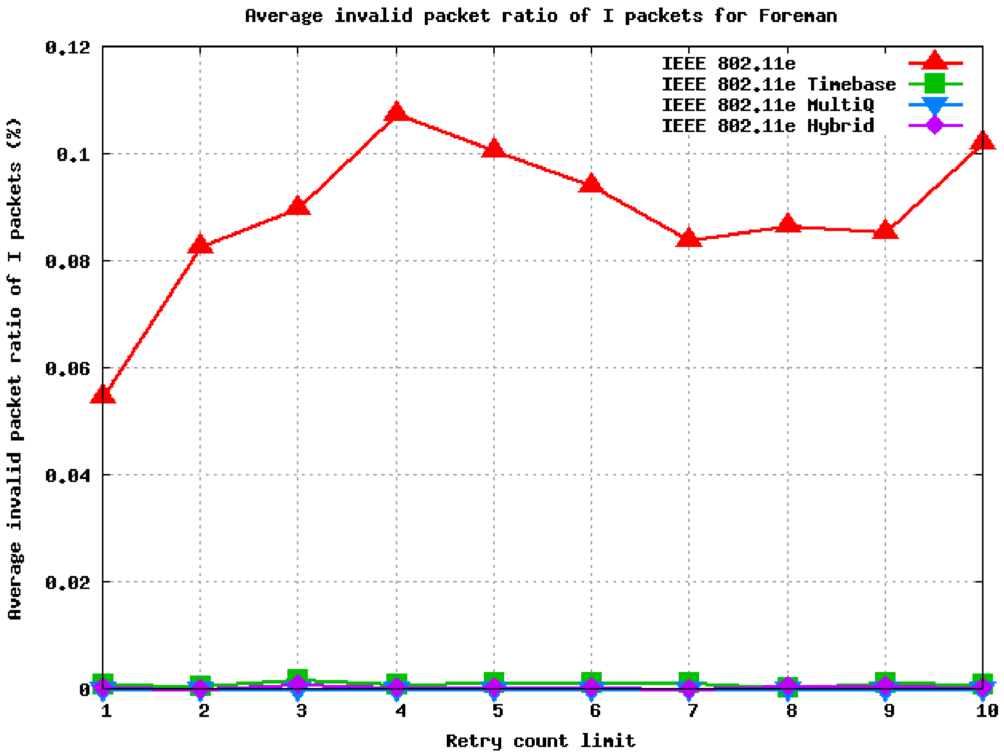

Figure 4.11: Average invalid packet ratio for I packets against retry limit Foreman.

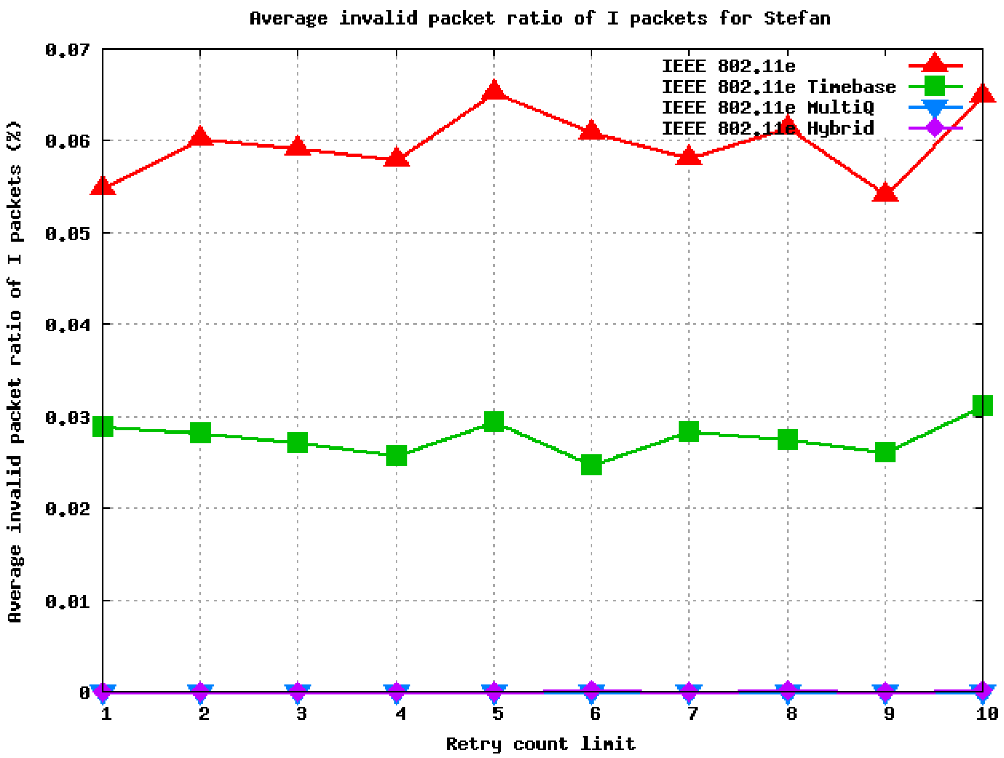

Figure 4.12: Average invalid packet ratio for I packets against retry limit for Stefan.

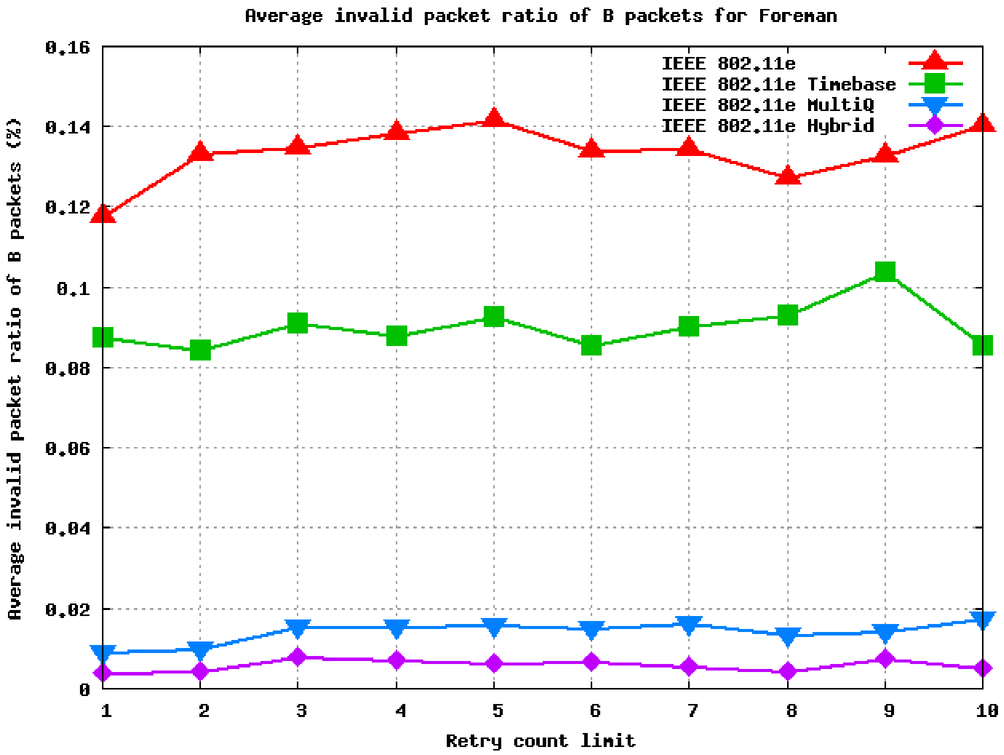

Figure 4.13: Average invalid packet ratio for B packets against retry limit for Foreman.

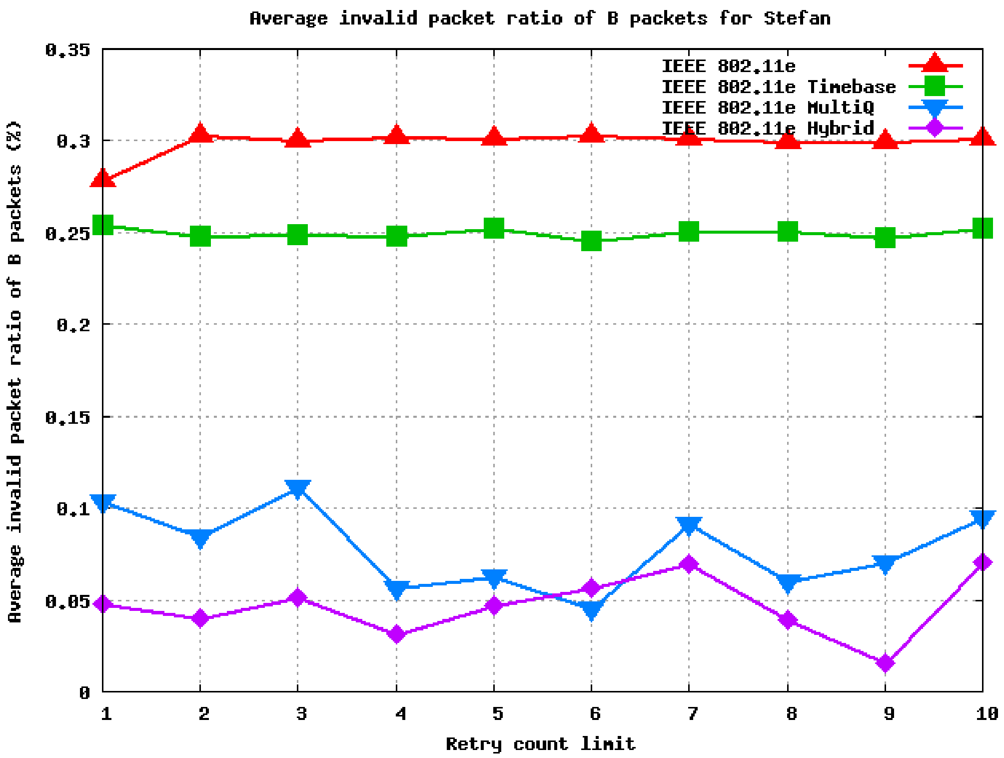

Figure 4.14: Average invalid packet ratio for B packets against retry limit for Stefan.

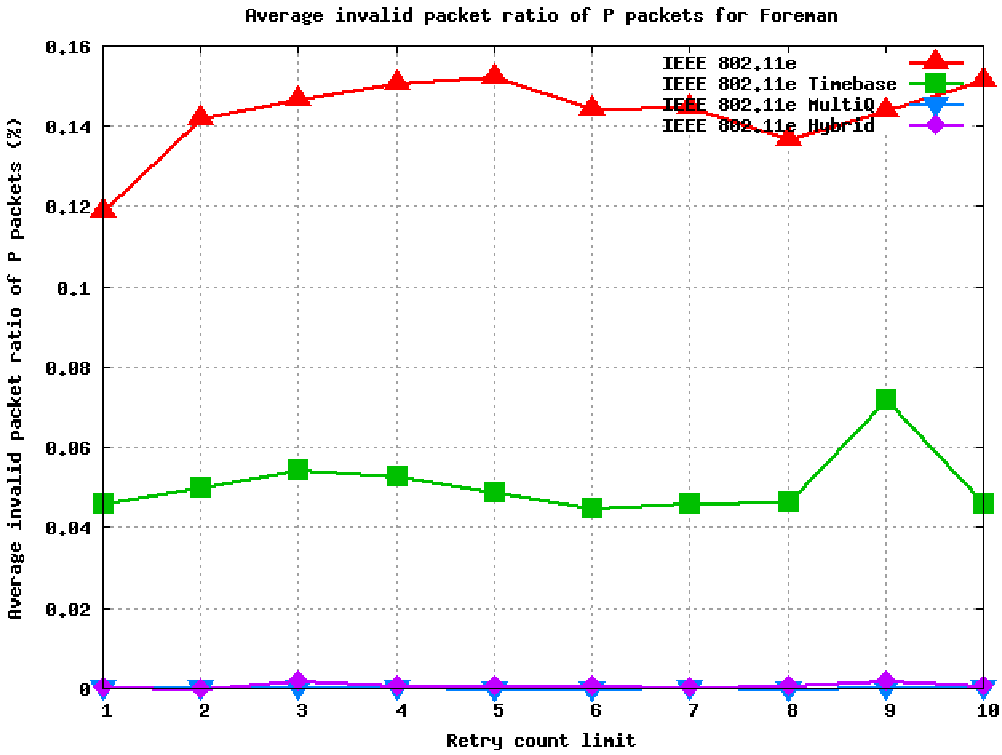

Figure 4.15: Average invalid packet ratio for P packets against retry limit for Foreman.

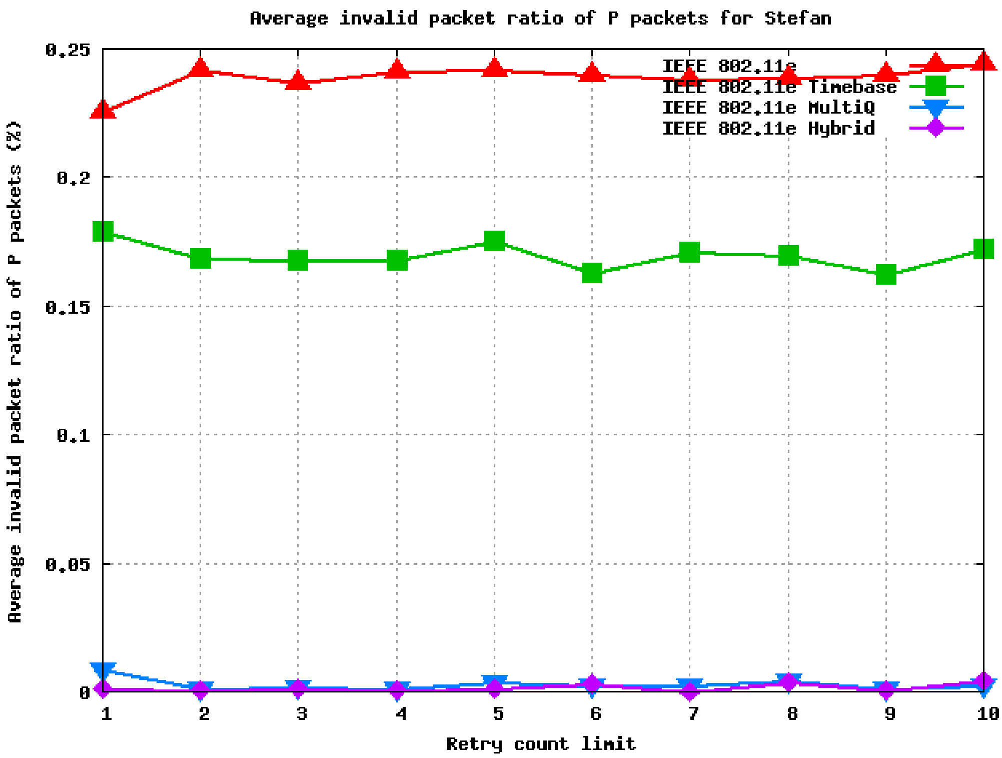

Figure 4.16: Average invalid packet ratio for P packets against retry limit for Stefan.

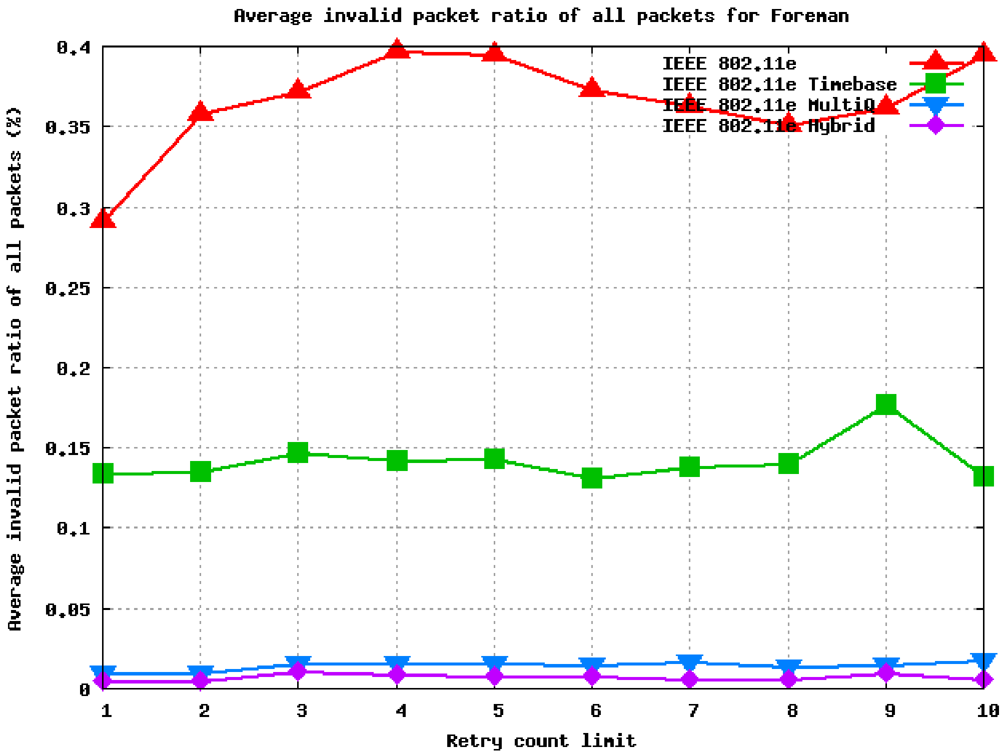

Figure 4.17: Average invalid packet ratio for all packets against retry limit for Foreman.

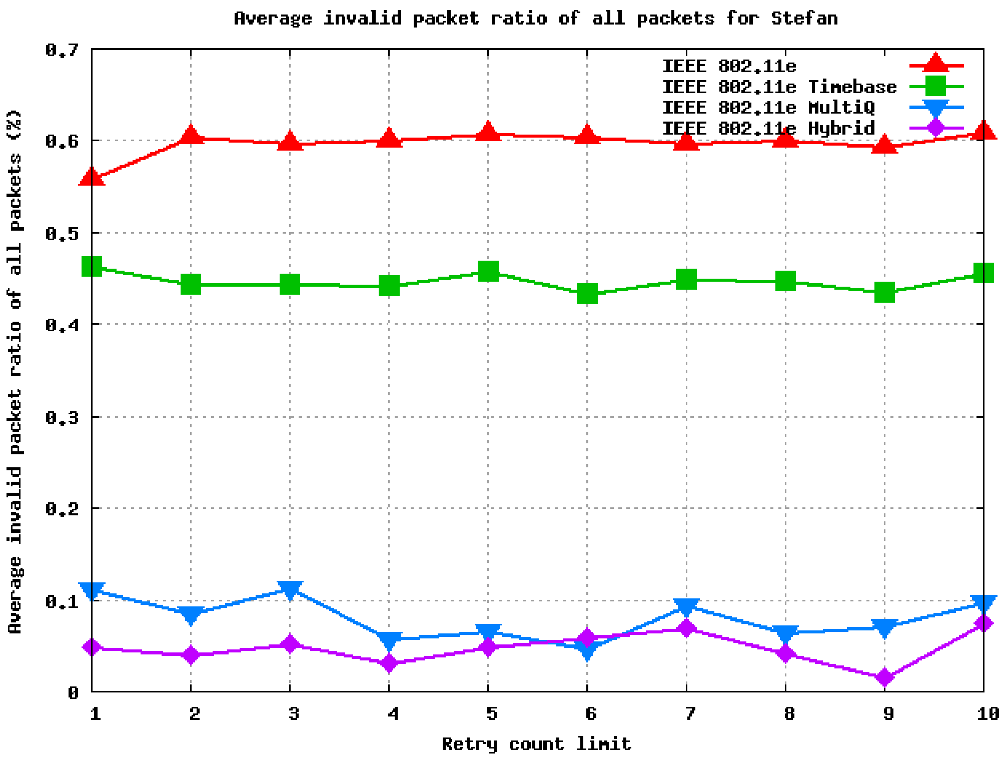

Figure 4.18: Average invalid packet ratio for all packets against retry limit for Stefan.

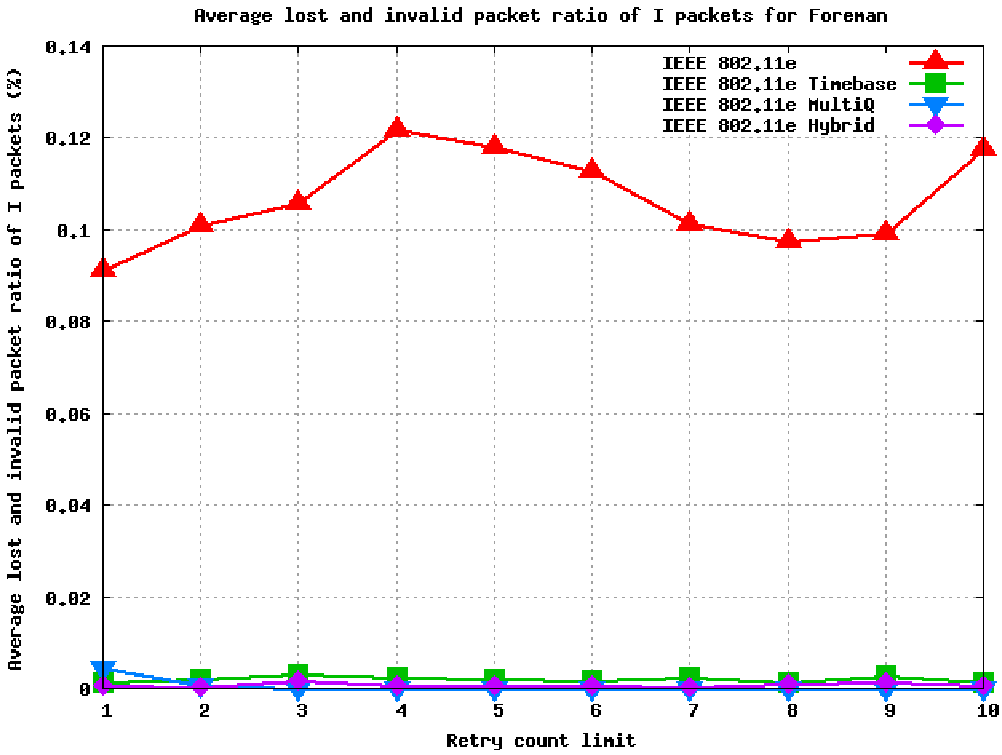

Figure 4.19: Average lost and invalid packet ratio for I packets against retry limit for Foreman.

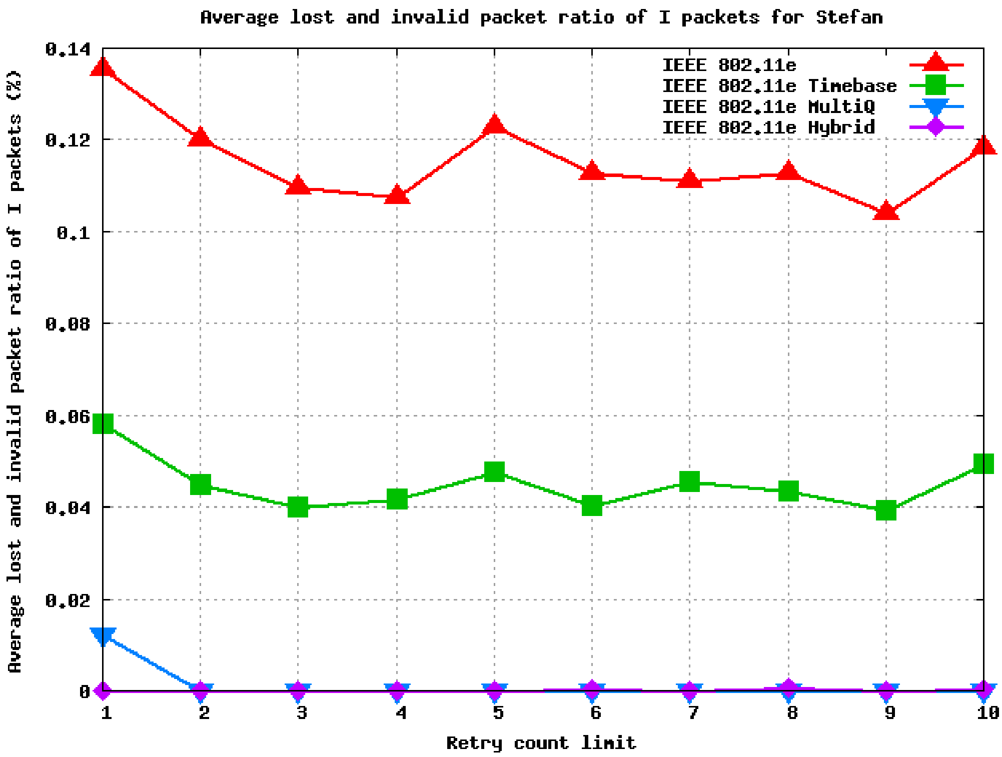

Figure 4.20: Average lost and invalid packet ratio for I packets against retry limit for Stefan.

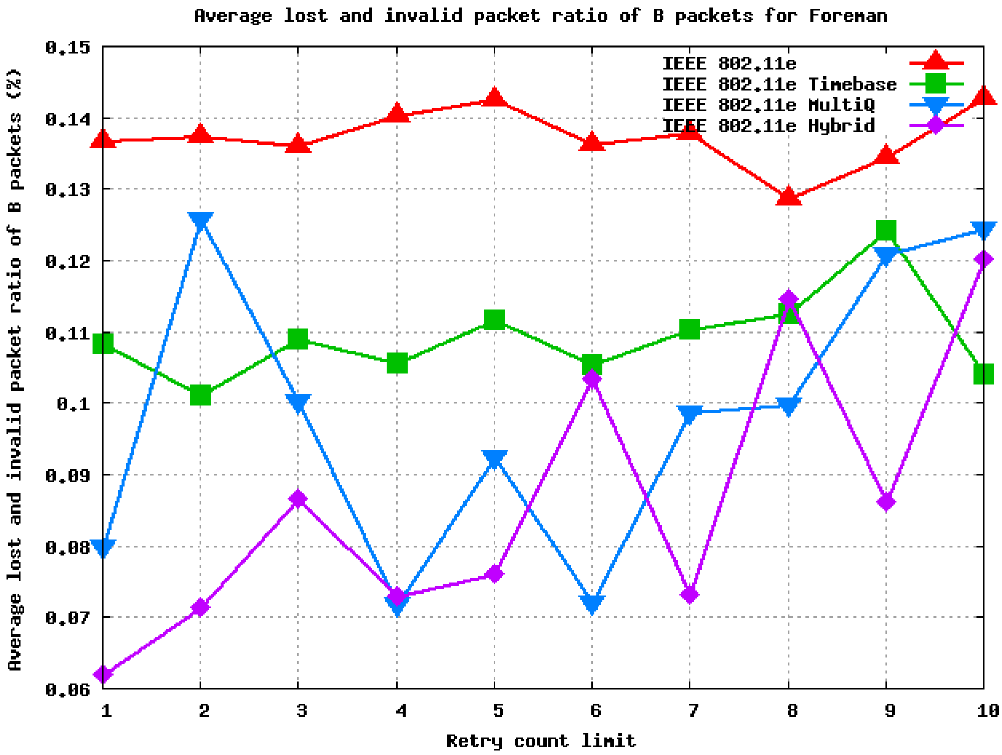

Figure 4.21: Average lost and invalid packet ratio for B packets against retry limit for Foreman.

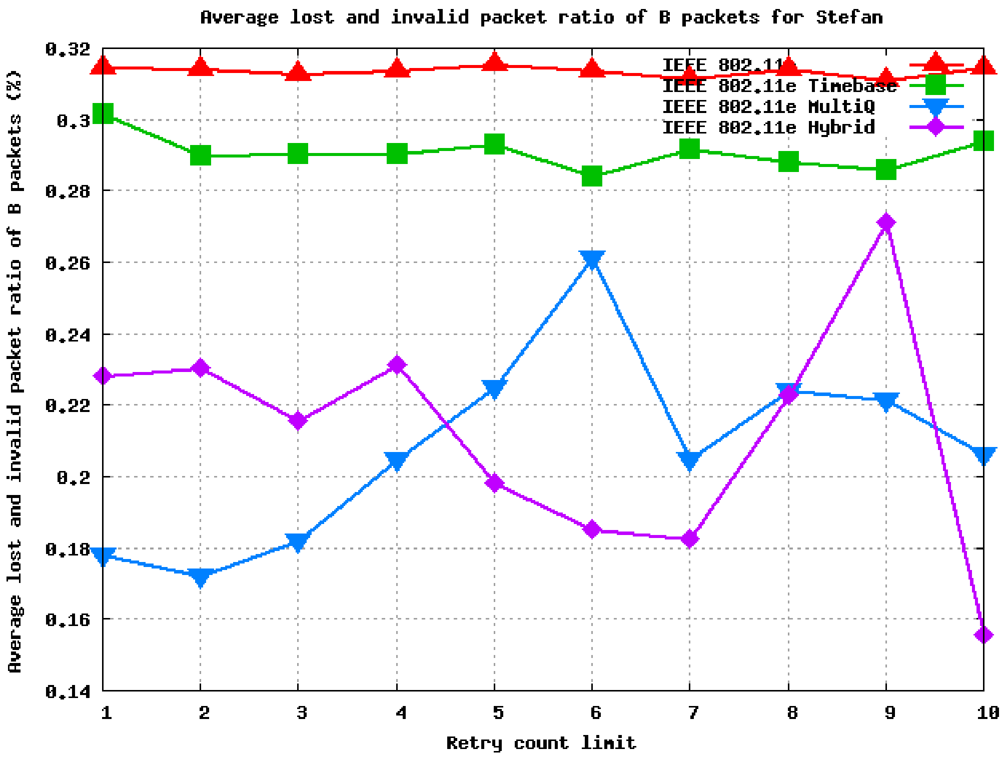

Figure 4.22: Average lost and invalid packet ratio for B packets against retry limit for Stefan.

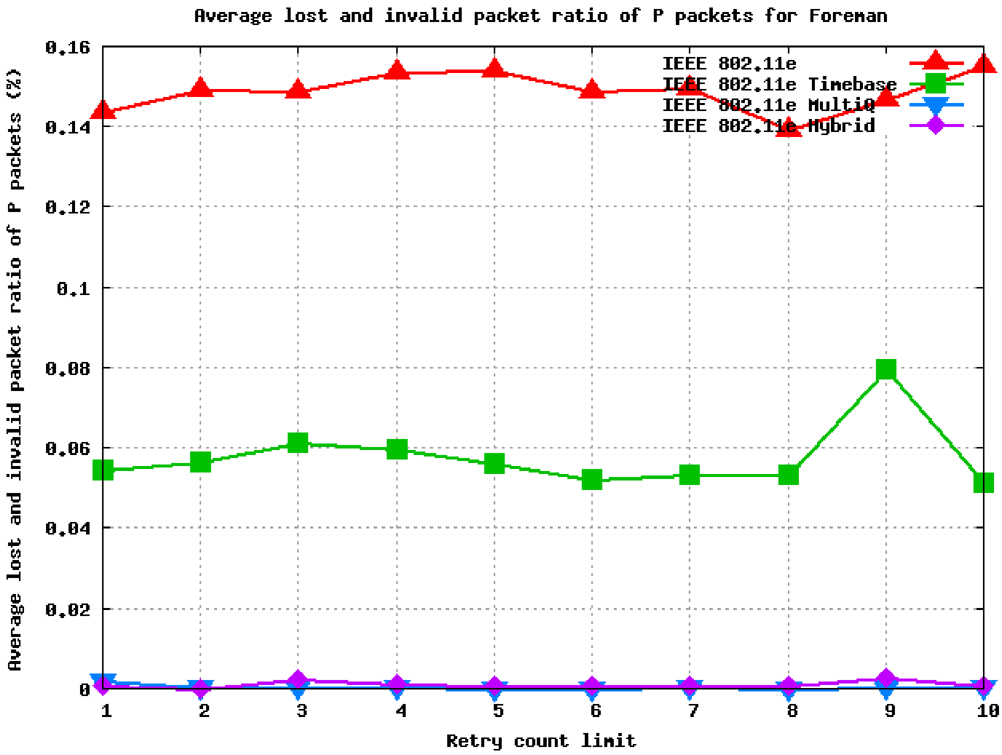

Figure 4.23: Average lost and invalid packet ratio for P packets against retry limit for Foreman.

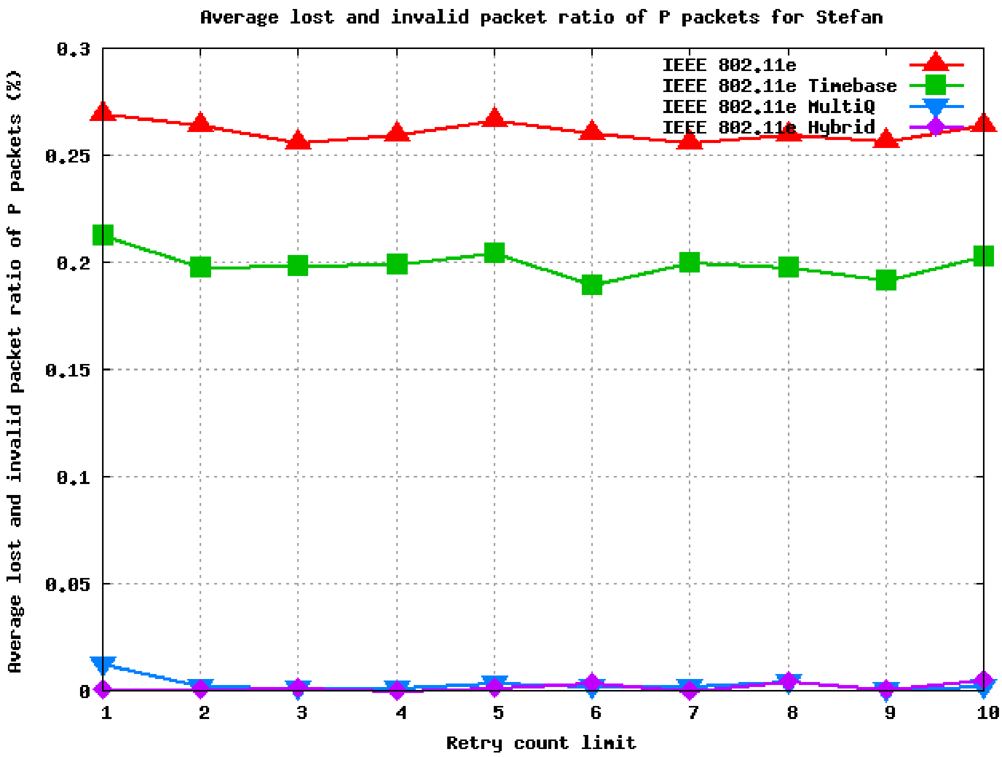

Figure 4.24: Average lost and invalid packet ratio for P packets against retry limit for Stefan.

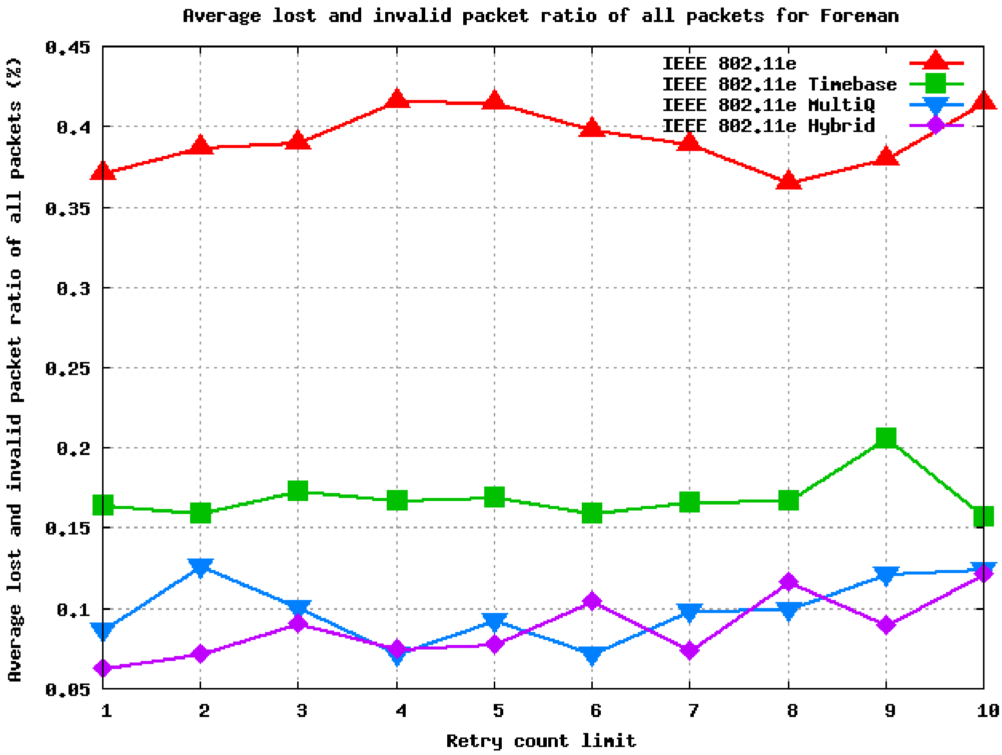

Figure 4.25: Average lost and invalid packet ratio for all packets against retry limit for Foreman.

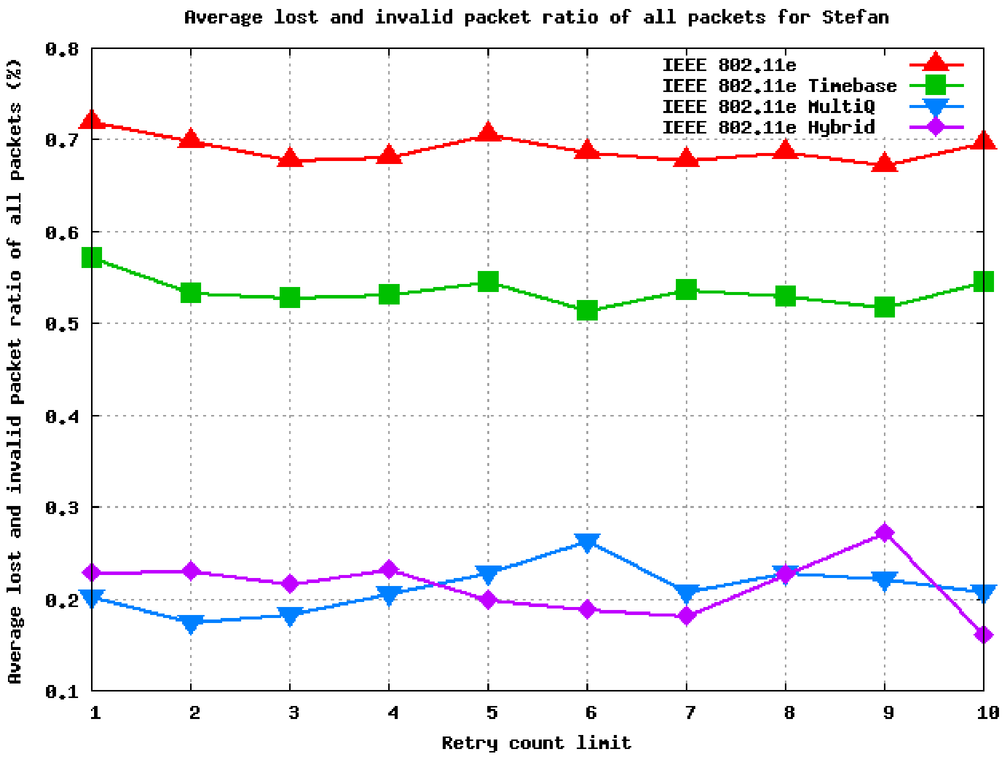

Figure 4.26: Average lost and invalid packet ratio for all packets against retry limit for Stefan.

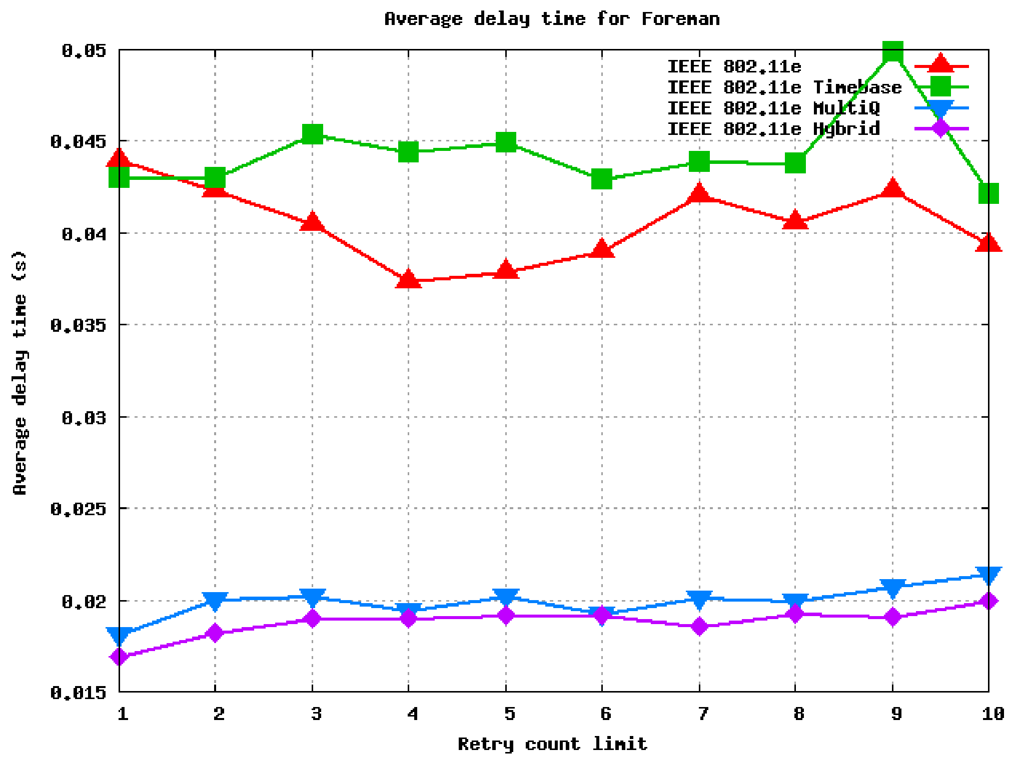

Figure 4.27: Average delay time against retry limit for Foreman.

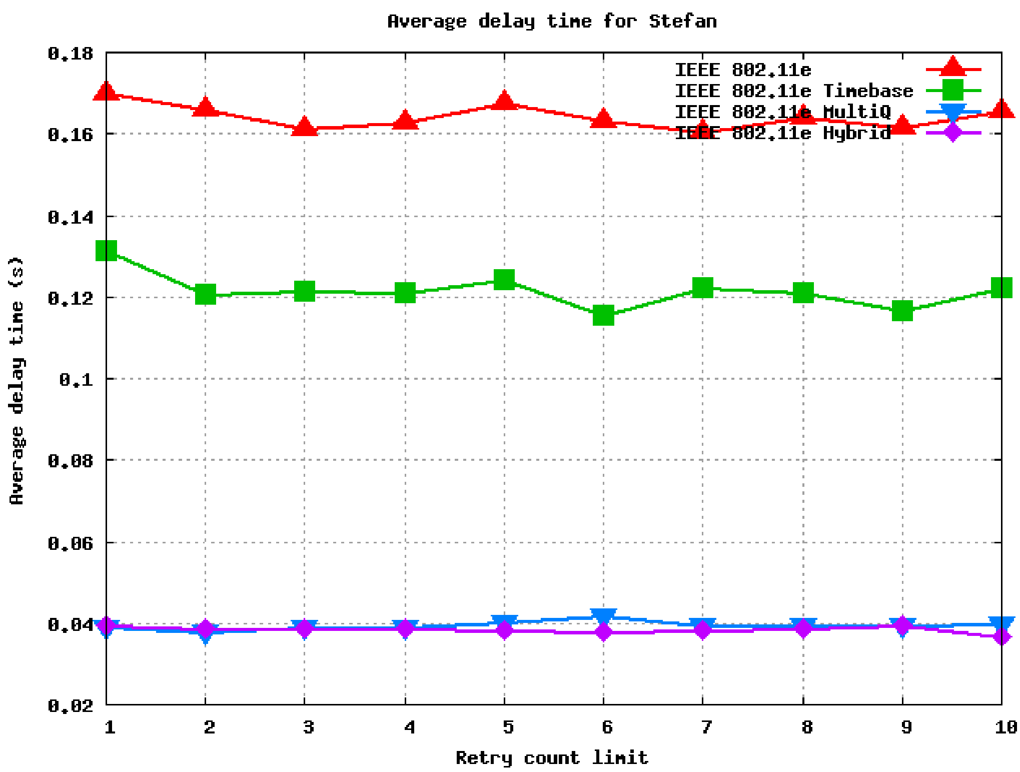

Figure 4.28: Average delay time against retry limit for Stefan.

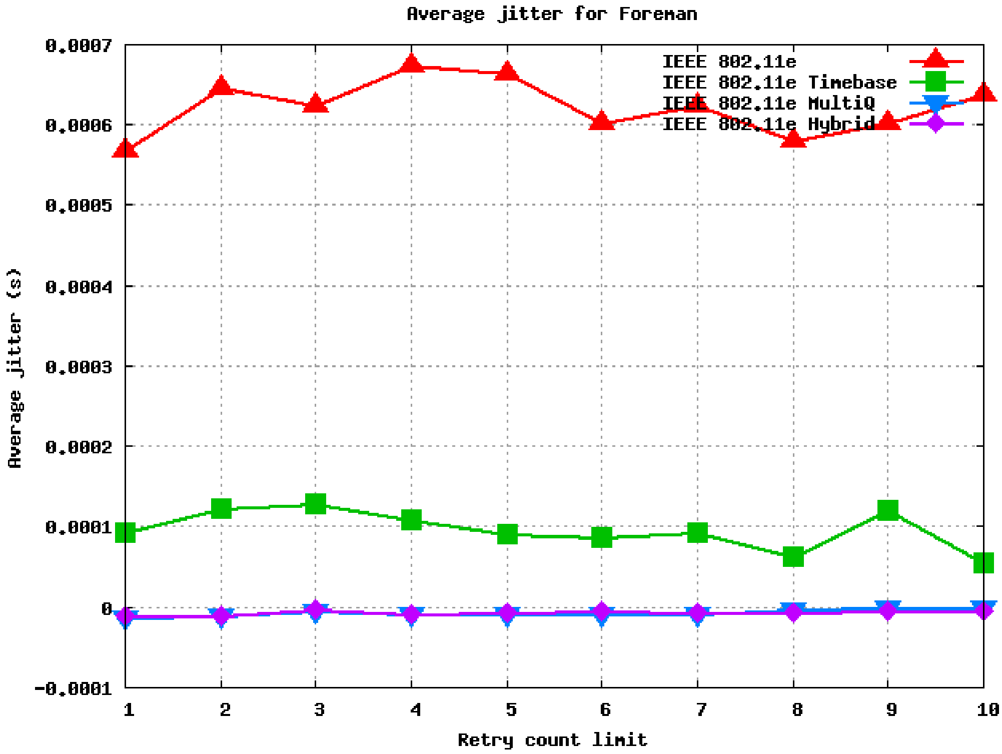

Figure 4.29: Average jitter against retry limit for Foreman.

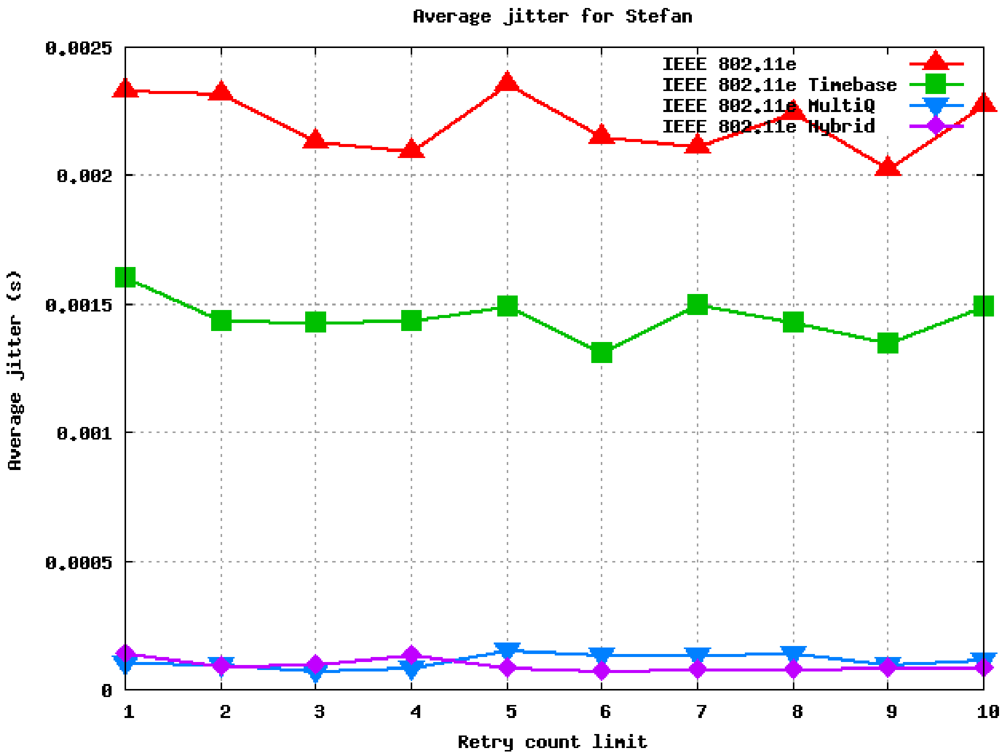

Figure 4.30: Average jitter against retry limit for Stefan.

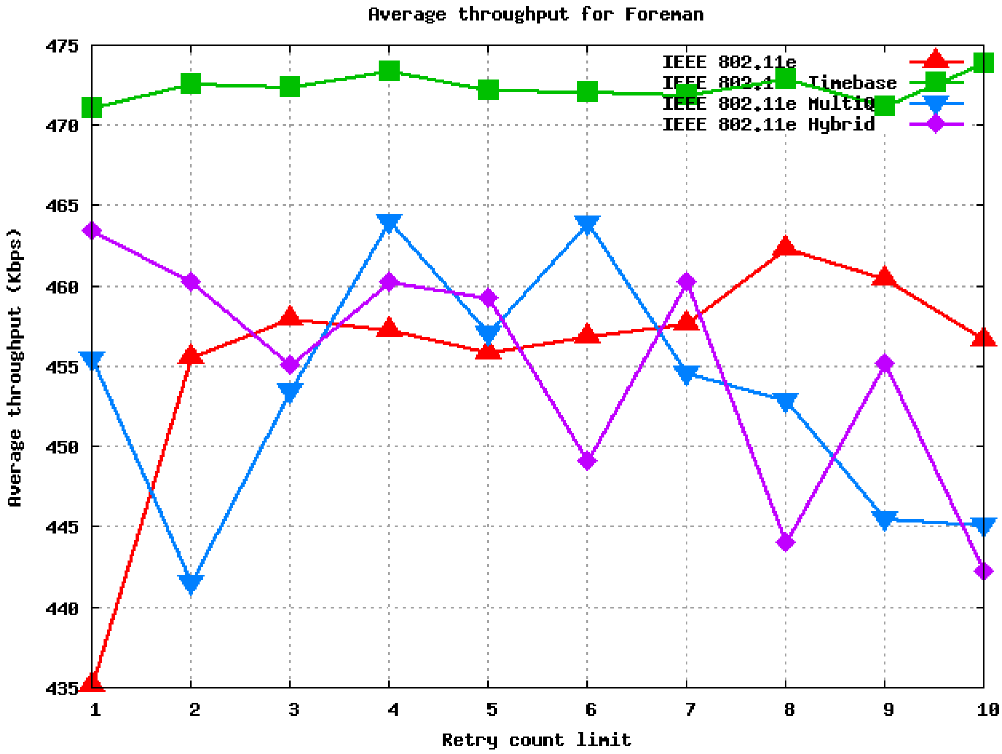

Figure 4.31: Average throughput against retry limit for Foreman.

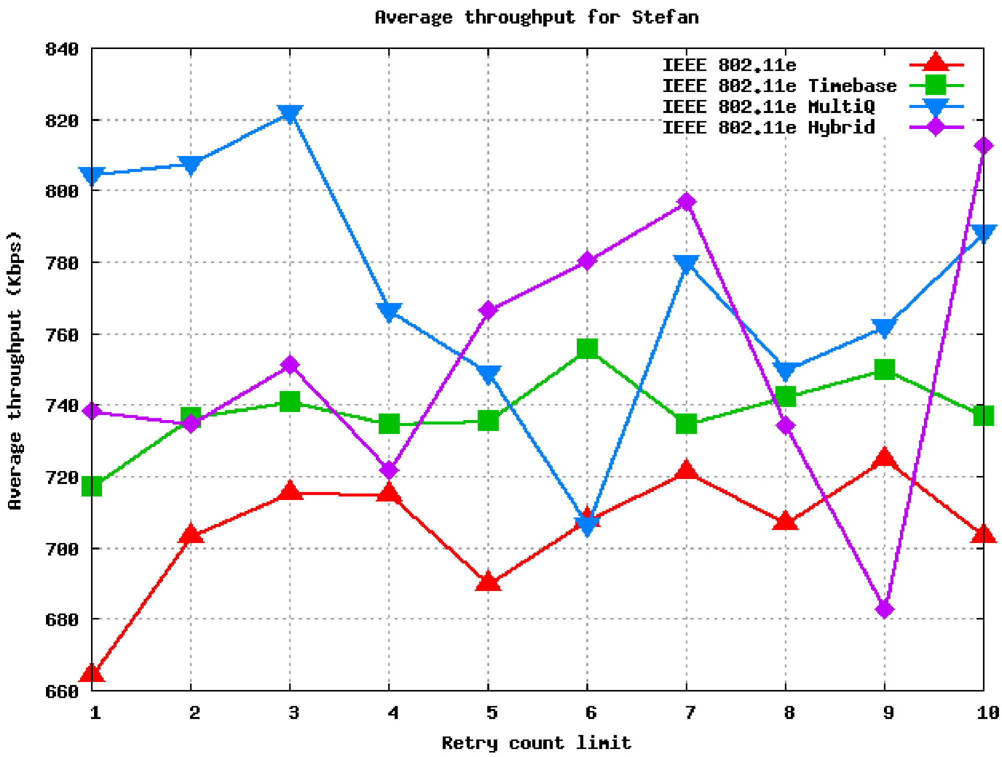

Figure 4.32: Average throughput against retry limit for Stefan.

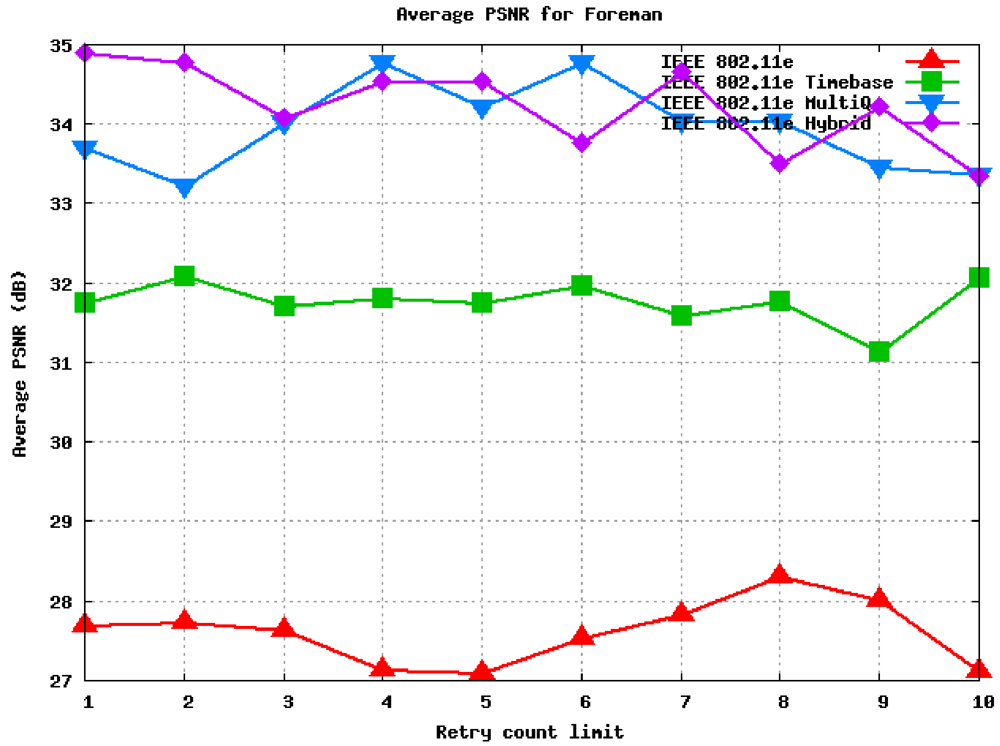

Figure 4.33: Average PSNR against retry limit for Foreman.

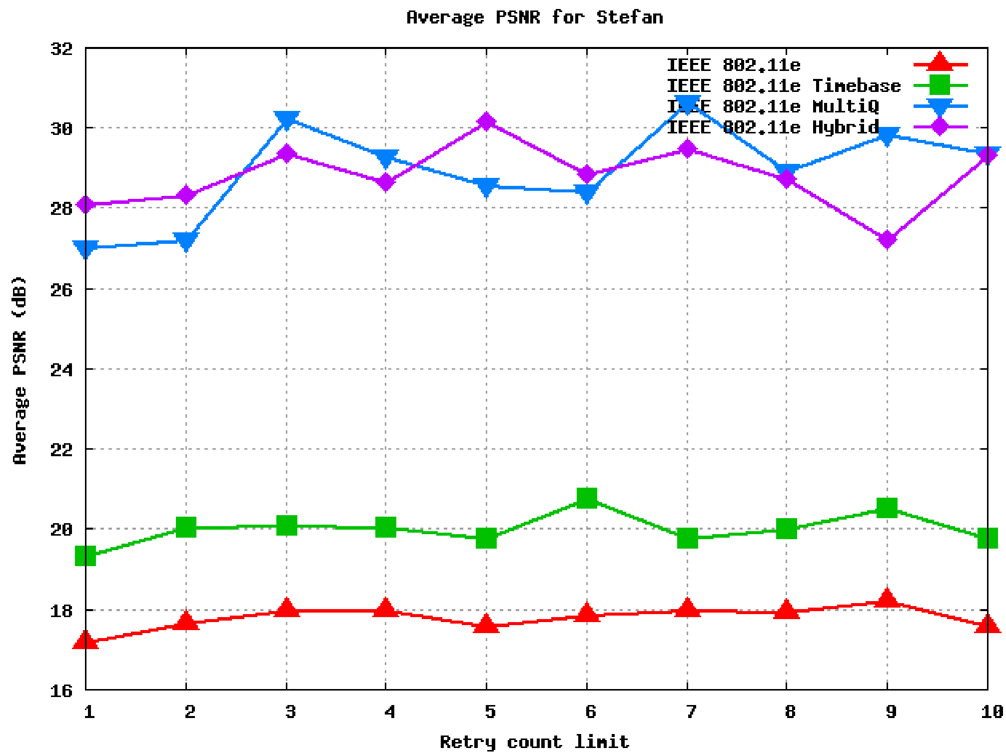

Figure 4.34: Average PSNR against retry limit for Stefan.