probability is determined by $P_{\text{block}}^d$ and with the packet size of 1000, the blocking probability is determined by $P_{\text{block}}^c$. With moderate packet sizes the blocking probability is determined by both probabilities and hence, the simulated and theoretical results do not match perfectly. Nevertheless, according to our results these approximations do not significantly under- or overestimate the performance of MMAC in any case and hence, the use of these approximates is justifiable for adequate analysis.

Finally, effect of additional cycles can be formulated as

$$D_{\text{block}} = (P_{\text{block}}^d + P_{\text{block}}^c - P_{\text{block}}^d P_{\text{block}}^c) T_c.$$  \hfill (20)

and thus, the average access delay of MMAC is given by

$$D = E[D_0] + D_{\text{block}}.$$  \hfill (21)

and the throughput is

$$S = g_a T \cdot \frac{e^{-g\tau}}{3 - 2 e^{-g\cdot\tau}} \cdot (1 - P_{\text{block}}).$$  \hfill (22)

C. Synchronized MAC (SYN-MAC)

We use SYN-MAC [12] as an example of common hopping approaches and the same delay-throughput analysis applies to parallel rendezvous schemes as well. SYN-MAC exploits periodic hopping and resource reservations can be done only for the current channel to avoid the multi-channel hidden node problem. Therefore, the performance of SYN-MAC can be estimated similarly to single-channel systems by reducing the arrival rate of packets due to the utilization of multiple channels simultaneously. General operation of SYN-MAC on a single channel is demonstrated in Figure 6. For analysis purposes, we assume that all generated packets have to wait until the next resource reservation interval before competing for resources and data transmissions start precisely at the end of contention windows.

Resource reservation interval is divided into multiple small time slots ($\tau$) and to avoid collisions, each transmitter chooses a random backoff value from a given fixed window $\omega$. In other words, SYN-MAC exploits UB. Length of the contention period is $T_s = \omega \tau$ and we set $T_s = 10$ since this should give good results in general according to [12]. Consequently, in order to validate the assumption of Poisson arrivals, retransmitted packets are delayed over several contention windows randomly in simulations. Moreover, we denote the total length of a cycle by $T_c = T + T_s$.

Arrival rates have to be scaled to correspond to the operation of SYN-MAC. Naturally, the arrival rate is inversely proportional to the number of channels $N$. Moreover, packets generated during the packet transmission time $T$ will stack up. Hence, in case of SYN-MAC we scale arrival rates as follows

$$g_a = g \cdot \frac{\omega + T}{T / \omega} \cdot \frac{1}{N} = g \cdot \frac{(\omega + T)\omega}{T \cdot N}.$$  \hfill (23)

Now, in case of SYN-MAC the latency of a successful transmission is simply

$$E[D_0] = T_s + T_s^2,$$  \hfill (24)

on average. Contrary to other approaches, the induced latencies because of collisions or if a channel is sensed as busy are equal in SYN-MAC. If resource request messages collide or the channel is sensed as busy, a delay of $T_s$ will be added always. Thus, the delay due to $R$ retransmissions is simply

$$D_e = \sum_{i=1}^{R} T_s = T_s \cdot R,$$  \hfill (25)

and the amount of retransmissions on average is given by

$$E[R] = \frac{P_b + P_e}{P_s}.$$  \hfill (26)

Finally, we can find out the average access delay as follows

$$D = E[D_0] + T_s \cdot E[R]$$

$$= E[D_0] + T_s \left(1 - \frac{P_s}{P_s}\right)$$

$$= T_s \left(2 + \frac{P_e}{P_s}\right), \quad P_s > 0,$$  \hfill (27)

and the throughput is

$$S = g_a T \cdot \frac{(T_s/T) e^{-g\cdot\tau}}{1 + (T_s/T) - e^{-g\cdot\tau}}.$$  \hfill (28)

The probabilities for SYN-MAC are derived in Appendix C. Again, we compare our theoretical results with simulation results and the outcome is illustrated in Figure 7. With large packets ($T \geq (N - 1)T_s$) theoretical and simulated results