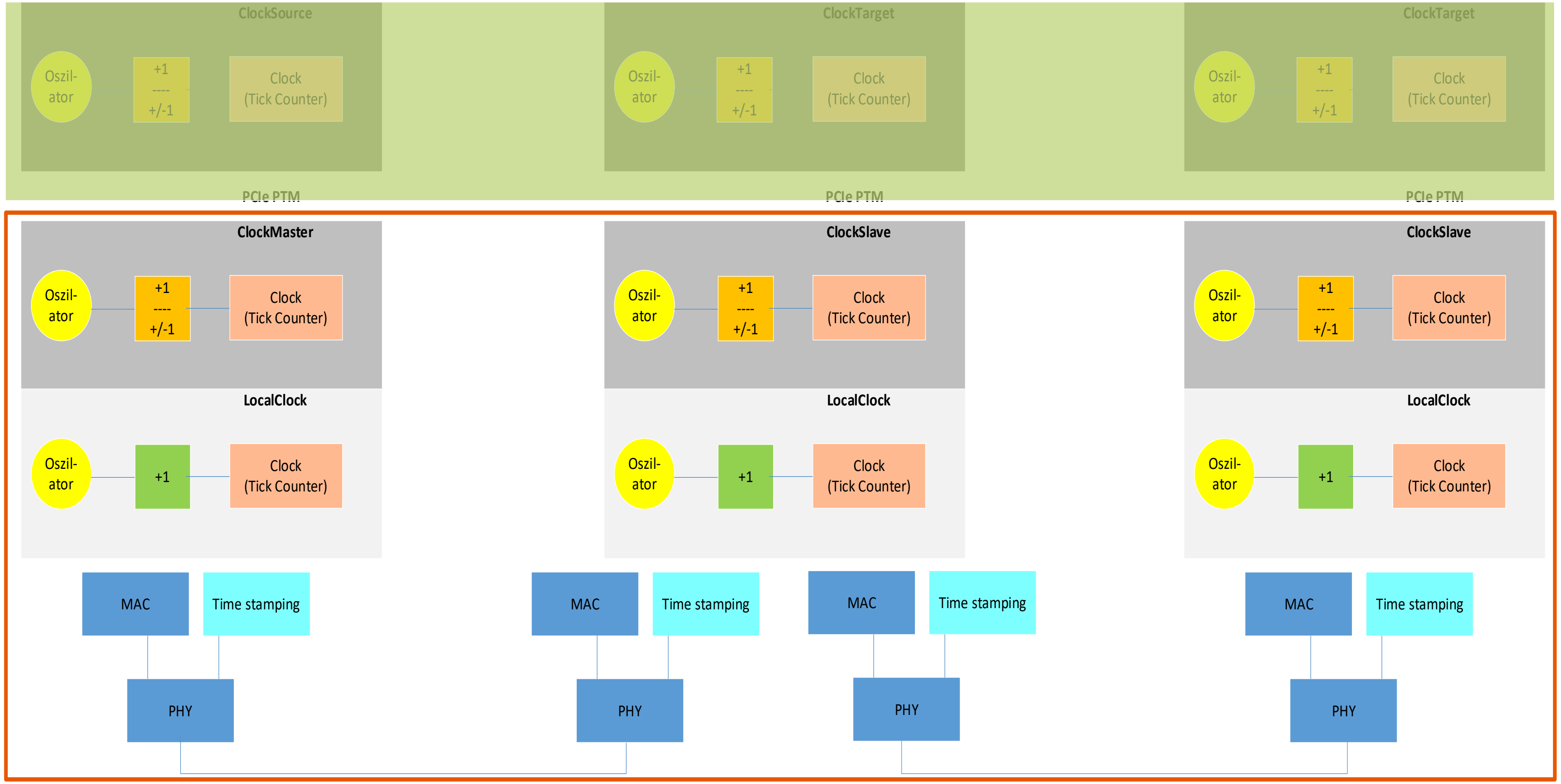
GrandMaster Frequency-Drift and its influence on the RR and MasterTime estimation at Slave Elements

Dragan Obradovic, Siemens AG

Line Topology

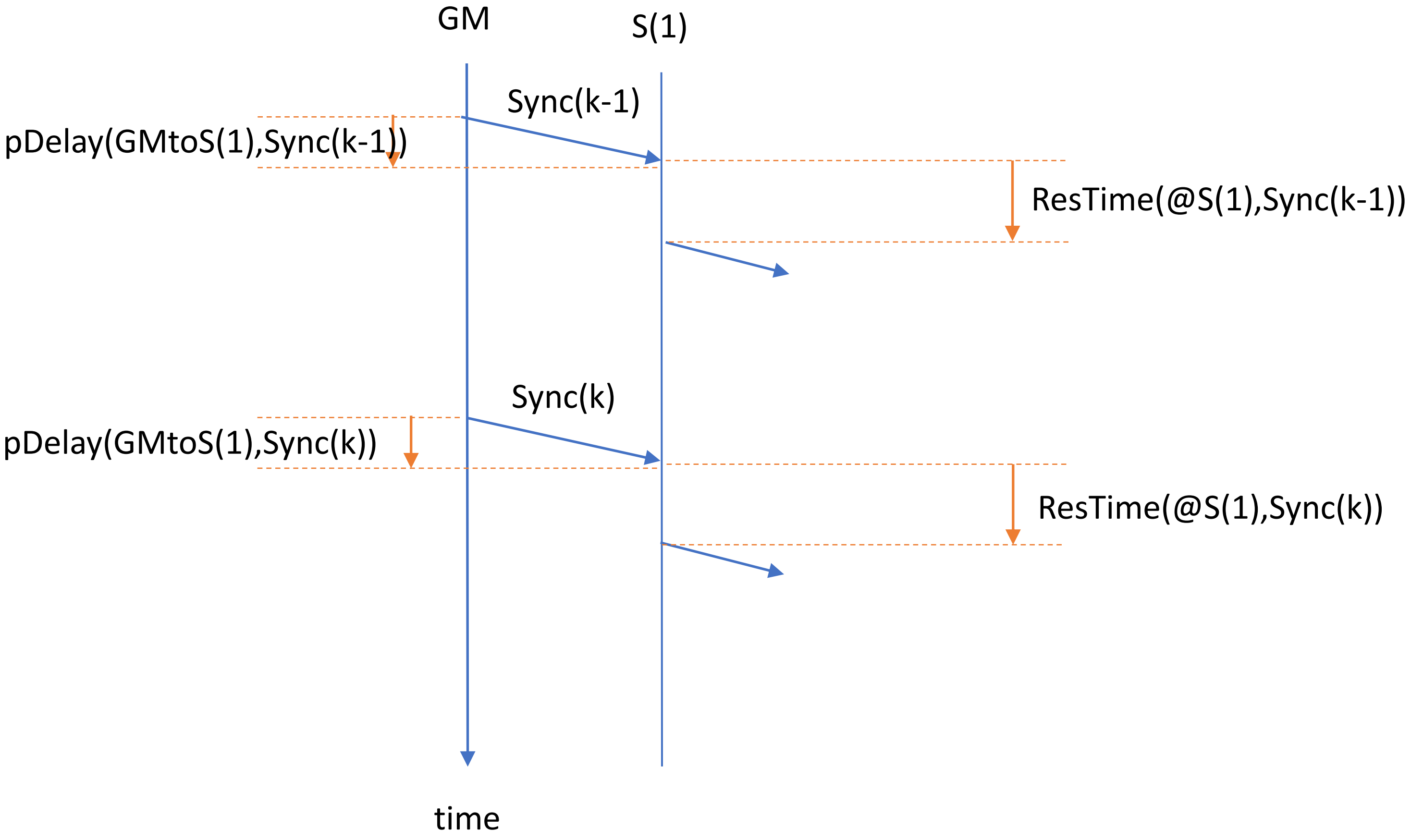


Propagation Delays of Sync Messages: Frequency Drift at GM only

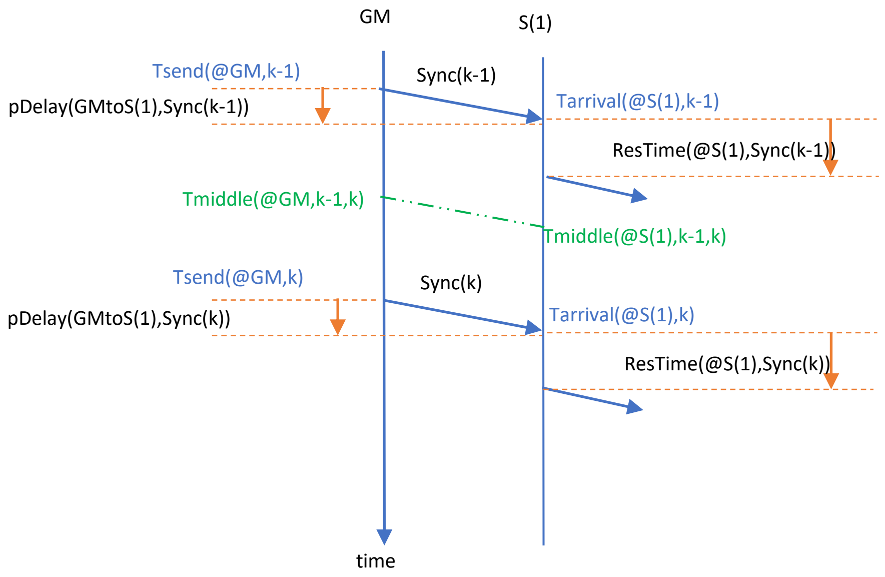
Assumptions:

- ClockMaster is equivalent with the LocalClock at GM.
- At S(1) we are looking at the calculation of RR and the MasterTime_Estimate of the Residence Time which will be forwarded to the next Slave element (S(1) is a relay element)
- LocalClock at GM drifts linearly with the rate "a"
- LocalClocks of S(1) and all other Slave elements have constant frequencies
- All nominal frequencies of LocalClocks in all elements are the same

→ Since the frequency of GM is changing (linearly drifting), the <u>RR(GM,S(1)) is a function of time</u> (this obviously holds for NRR too)!



Propagation Delays of Sync Messages: Frequency Drift at GM only



Some facts about linearly drifting frequencies and the calculation of RR and its usage:

1) $RR(@S(1),t) = \frac{freqGM(t)}{freqS(1)(t)} = \frac{freqGM(t_0)+aGM*(t-t_0)}{freqS(1)(t_0)+aS(1)*(t-t_0)}$

➔ The true RR is a function of time “t”! We always have to specify the time “t” RR refers to when we estimate it

2) Let us estimate RR at S(1) using the Sync messages (k-1) and k.

$RRest(@S(1), k-1, k) =$

$$= \frac{\left[freqGM\left(T_{send}(@GM,k-1)\right) + aGM * \frac{(T_{send}(@GM,k) - T_{send}(@GM,k-1)}{2}\right] * (T_{send}(@GM,k) - T_{send}(@GM,k-1))}{\left[freqS(1)\left(T_{arrival}(@S(1),k-1)\right) + aS(1) * \frac{(T_{arrival}(@S(1),k) - T_{arrival}(@S(1),k-1)}{2}\right] * (T_{arrival}(@S(1),k) - T_{arrival}(@S(1),k-1))}$$

$$\approx \frac{freqGM(T_{middle}(@GM,k-1,k))}{freqS(1)(T_{middle}(@S(1),k-1,k))}$$

➔ The estimate RRest really corresponds to the true RR in the middle of the (absolute) time interval between sending at GM and receiving Sync messages at S(1). But since we have only the local timeframes, we use them to find the middle points.

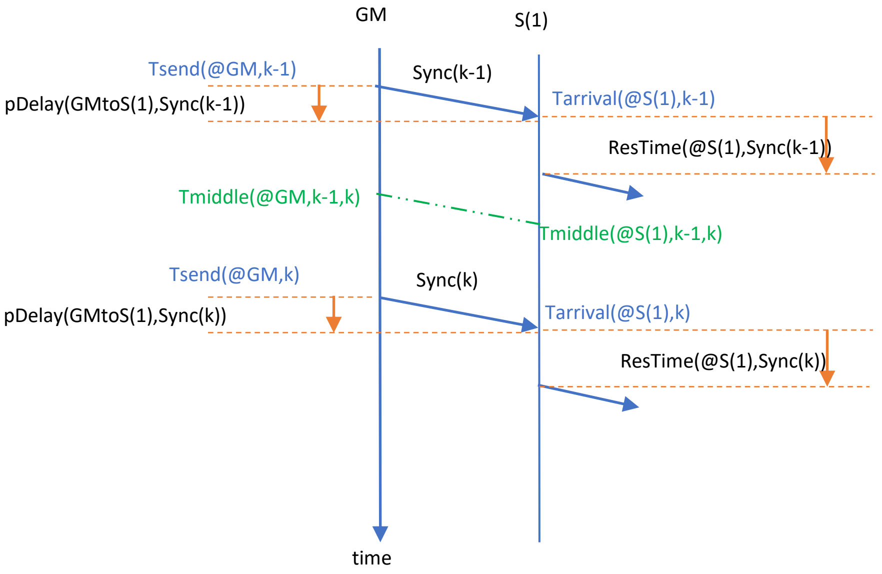
Propagation Delays of Sync Messages: Frequency Drift at GM only

Question:

Given a time interval ResTime(@S(1),Sync(k)) in the LocalClock at S(1) timeframe, define the "best" RRest(@S1, Tbest) to convert this ResTime to the Master-Time timeframe

→ The solution is the RRest corresponding to the middle of that Residence-Time interval at S(1)

$$RRest(@S(1),Tbest) \approx RRest\left(@S(1),\frac{Restime(@S(1),Sync(k))}{2}\right)$$

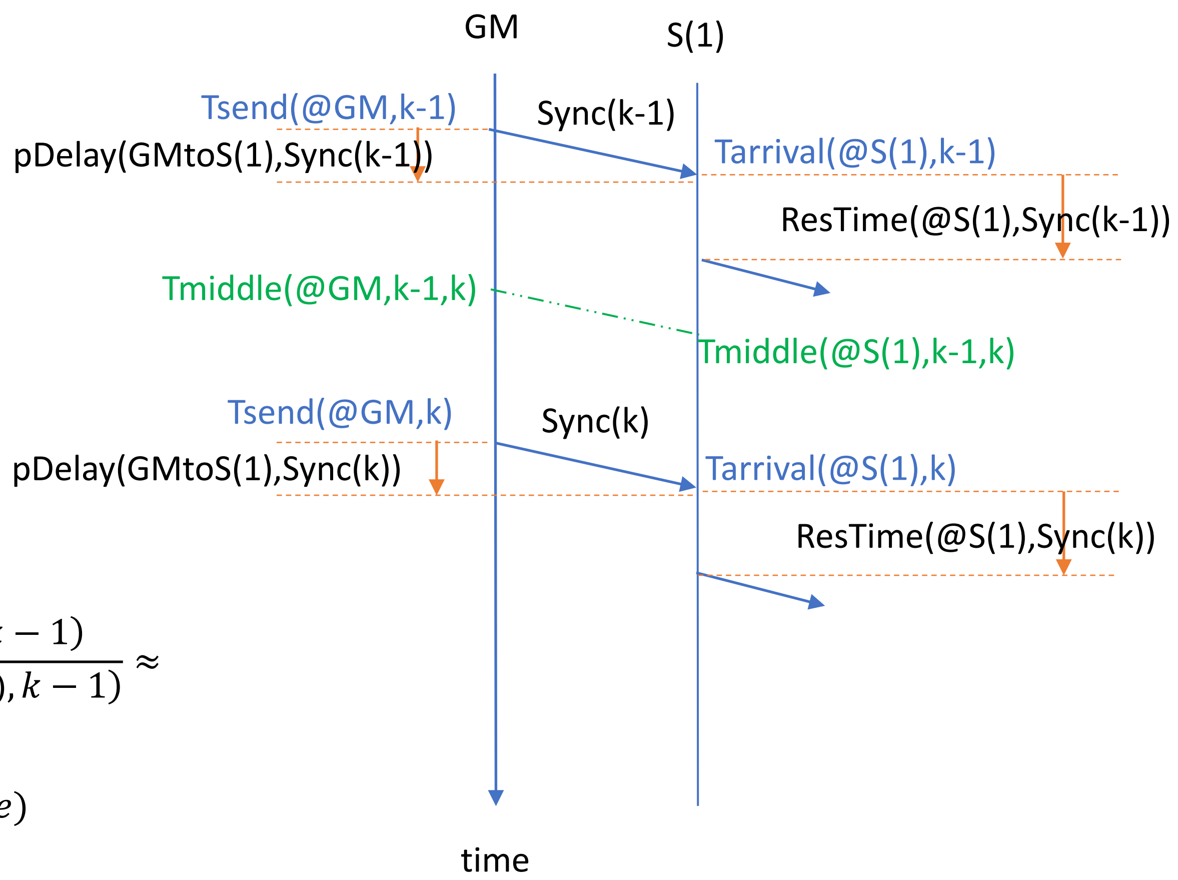


- ➔ If we could calculate RR(@S(1)) at any time, we should calculate in the middle of the Residence-Time in the LocalClock timeframe, in order to optimally convert the ResTime interval into the Master-Time timeframe.
- ➔ But we can calculate RR only at the arrival-instance of a Sync message, and at that point it has already "aged" for the half of the last Tsync interval. We need a prediction mechanism to map it to the middle of the Residence-Time interval!

Propagation Delays of Sync Messages: Frequency Drift at GM only

Estimated RR (and NRR) at S(1) using Sync messages (k-1) and (k) corresponds in the case of linear frequency drift pretty much to the ratio of the frequencies at the middle time point between sending and arriving of these two Sync messages

→ At the time instant Tarrival(@S(1),k) the estimate of RR (and NRR) at S(1) is already old by time interval ≈(Tsync/2 + pDelay(GMtoS(1))

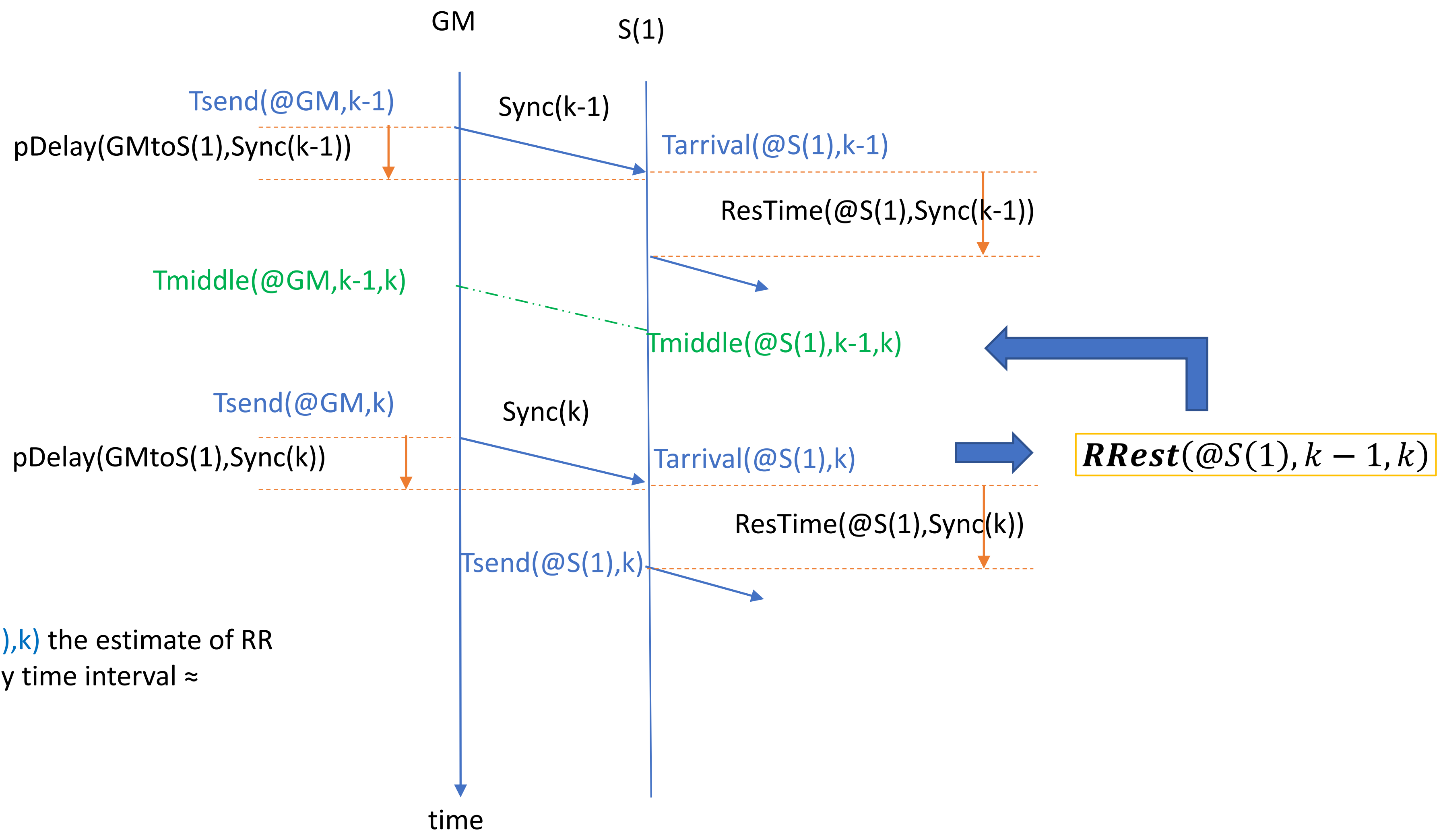


$$RRest(@S(1),k-1,k) = \frac{T_{send}(@GM,k) - T_{send}(@GM,k-1)}{T_{arrive}(@S(1),k) - T_{arrive}(@S(1),k-1)} \approx$$

$$\approx \frac{freq_{GM}\left(T_{middle}(@GM,k-1,k)\right)}{freq_{S(1)}\left(T_{middle}(@S(1),k-1,k)\right)} = RR(Tmiddle)$$

Question: when will $RRest(@S(1),k-1,k)$ be used?

Propagation Delays of Sync Messages: Frequency Drift at GM only



→ At the time instant Tarrival(@S(1),k) the estimate of RR (and NRR) at S(1) is already old by time interval ≈ Tsync/2 + pDelay(GMtoS(1))

→ But when will it be needed?

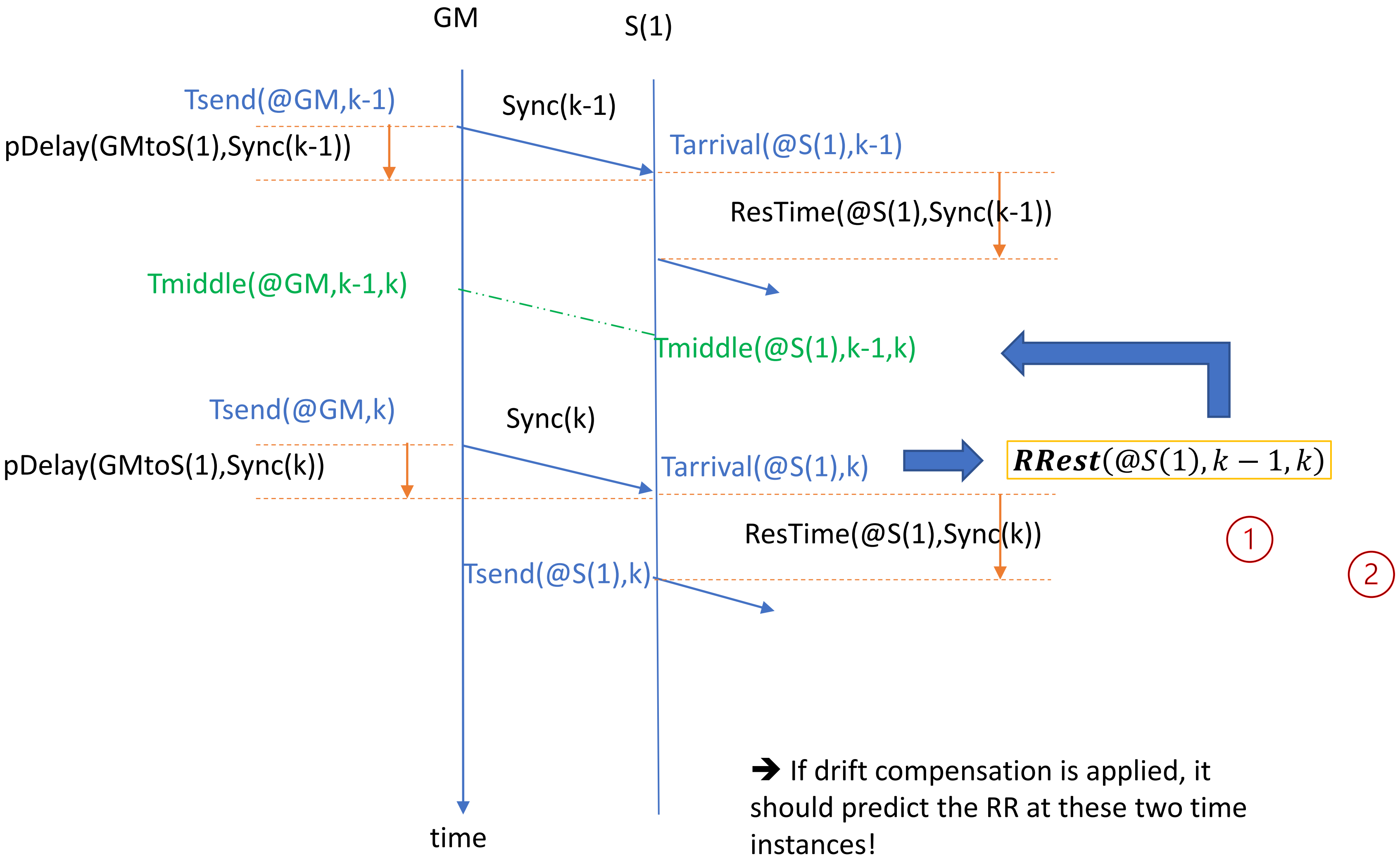
Propagation Delays of Sync Messages: Frequency Drift at GM only

Question: when will $RRest(@S(1),k-1,k)$ be used?

Answer:

- 1 To scale the ResidenceTime at S(1) of Sync(k), which is in the LocalClock time frame, into the GM-Time frame. Optimally, we need RRest in the middle of the ResTime(@S(1), Sync(k))
- 2 To forward it with a Sync message going from S(1) to S(2); Optimally, we need RRest at the time instant Tsend(@S1,k).

BUT: our RRest corresponds to Tmiddle(@S(1),k-1,k)



➔ If drift compensation is applied, it should predict the RR at these two time instances!

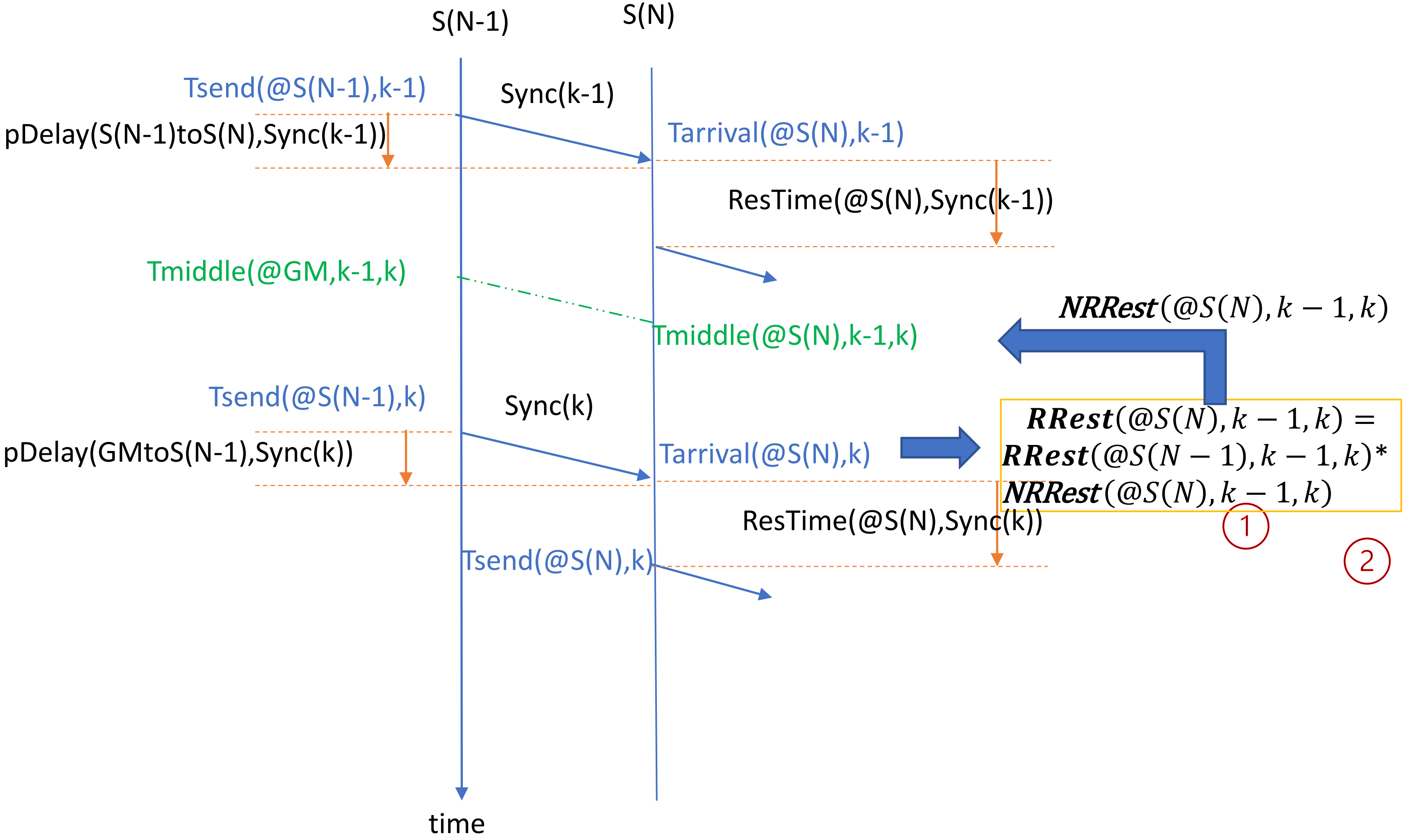
Propagation Delays of Sync Messages: Frequency Drift at GM only

Question: What happens between two Slave elements?

Answer: At Tarrival(@S(N),k):

- $\boldsymbol{RRest}(@S(N-1),k-1,k)$ is delivered by the Sync(k). If the drift compensation was done at S(N-1), its “age” is only the pDelay, which can be ignored in most cases.

- $\boldsymbol{NRRest}(@S(N),k-1,k)$ is calculated, but its age correspond to Tmiddle(@S(N),k-1,k). It should be drift-compensated for the time interval (Tarrival(@S(N),k)-Tmiddle(@S(N),k-1,k)), so that it corresponds to the same time as the $\boldsymbol{RRest}(@S(N-1),k-1,k)$.



S(N-1)

S(N)

Tsend(@S(N-1),k-1)

Sync(k-1)

pDelay(S(N-1)toS(N),Sync(k-1))

Tarrival(@S(N),k-1)

ResTime(@S(N),Sync(k-1))

Tmiddle(@GM,k-1,k)

Tmiddle(@S(N),k-1,k)

$\boldsymbol{NRRest}(@S(N),k-1,k)$

Tsend(@S(N-1),k)

Sync(k)

pDelay(GMtoS(N-1),Sync(k))

Tarrival(@S(N),k)

$$\boldsymbol{RRest}(@S(N),k-1,k) = \boldsymbol{RRest}(@S(N-1),k-1,k) * \boldsymbol{NRRest}(@S(N),k-1,k)$$

ResTime(@S(N),Sync(k))

Tsend(@S(N),k)

time

Propagation Delays of Sync Messages: Frequency Drift at GM only

Question: What happens between two Slave elements?

Drift-Compensation steps:

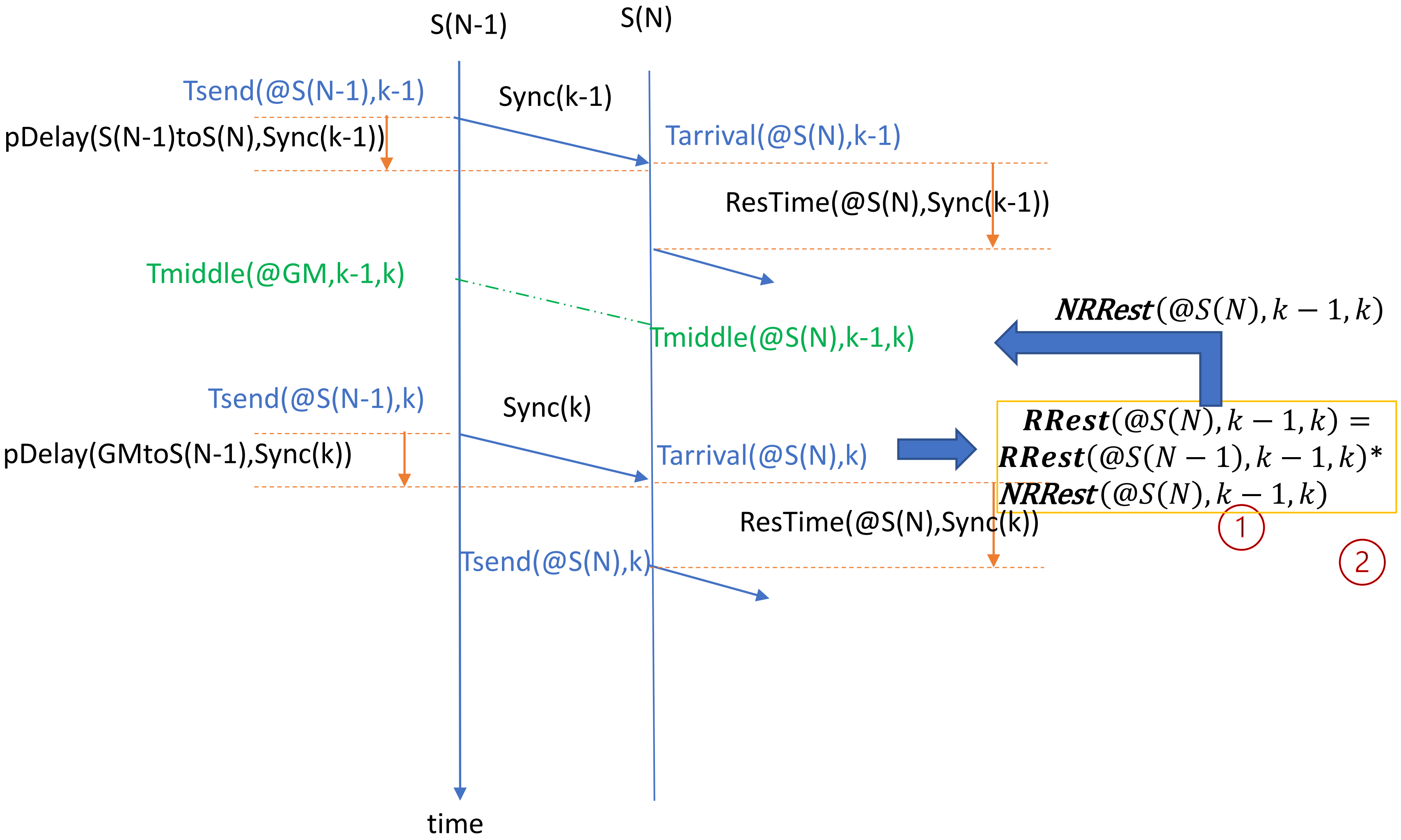
1) Drift-compensate $\boldsymbol{NRRest}(@S(N), k-1, k)$ so that it corresponds to Tarrival(@S(N),k). This requires an estimate of the drift rate of NRRest after receiving Sync(k).

.

2) Calculate $\boldsymbol{RRest}(@S(N), k-1, k)$ by using the above compensated NRRest and the RRest received by Sync(k).

3) Use the above calculated $\boldsymbol{RRest}(@S(N), k-1, k)$ to estimate its drift rate. Use the latter to predict RRest values at time points ① and ②

.



S(N-1)

S(N)

Tsend(@S(N-1),k-1)

Sync(k-1)

pDelay(S(N-1)toS(N),Sync(k-1))

Tarrival(@S(N),k-1)

ResTime(@S(N),Sync(k-1))

Tmiddle(@GM,k-1,k)

$\boldsymbol{NRRest}(@S(N), k-1, k)$

Tmiddle(@S(N),k-1,k)

Tsend(@S(N-1),k)

Sync(k)

pDelay(GMtoS(N-1),Sync(k))

Tarrival(@S(N),k)

$$\boldsymbol{RRest}(@S(N), k-1, k) = \boldsymbol{RRest}(@S(N-1), k-1, k) * \boldsymbol{NRRest}(@S(N), k-1, k)$$

ResTime(@S(N),Sync(k))

①

②

Tsend(@S(N),k)

time

Propagation Delays of Sync Messages: Frequency Drift at GM only

Compensation steps at S(N):

1) At arbitrary Slave element „N“, after receiving a new Sync message, we need to:
   - Estimate the “raw” NRRest
   - Estimate the drift rate of ***NRRest*** (by using their raw values)
   - Compensate the age of the current ***NRRest*** so that it corresponds to the time instant Tarrival (Sync(k))

2) Calculate the ***RRest(@N)*** by multiplying the ***RRest(@N-1)*** with the above compensated ***NRRest***

3) Estimate the drift of the ***RRest*** calculated in 2)

4) Use the above estimated drift of ***RRest*** to calculate its compensated (predicted) values needed for:
   - Converting the ResidenceTime interval in LocalClock to the Master-Time with the ***RRest*** corresponding to the „middle“ of this interval
   - Providing ***RRest(@N)*** which will be sent by the Sync message Sync(k) to the next element S(N+1). This ***RRest(@N)*** should correspond to the time of forwarding Sync(k) to „N+1“ element.

Propagation Delays of Sync Messages: Frequency Drift at GM only

What is gained and at which price:

1) If drifts remain linear with the constant rate over the time that is needed for Sync(k) to reach the last element, and if the estimates of all drifts (of NRRest and RRest at all intermediate elements) are accurate enough (t.b.d.), then the estimates of the RR and ResidenceTime in the Master-Time timeframe at the element N is (“significantly”) more accurate than without the compensation

2) BUT, the influence of delays (Residence Times, pDelays and also Tsync) up to the element “N” is actually moved from RRest to the estimates of the **drifts** of RRest! The delay (age) of a drift-estimate at the element “N” is equal to the total delay a Sync message needs to get to “N” (plus an extra delay depending on the estimation algorithm)

3) Every error in the drift estimate is amplified by the prediction horizon length “dt”: RRest_compensated(t+dt) = RRest(t)+RRdrift_est*dt. The noise in the system plays an important role too.

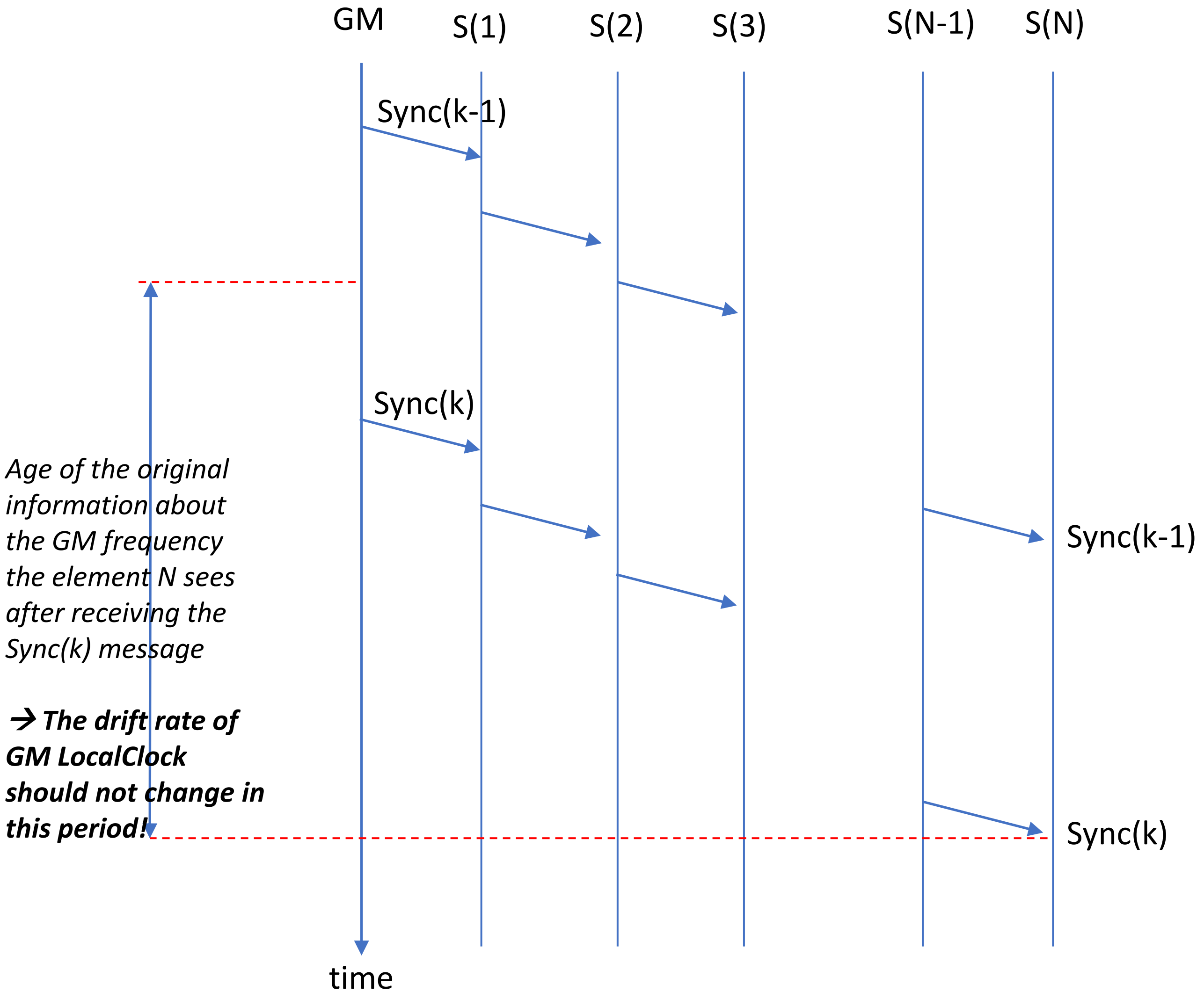
4) The drift estimate of RR at “N” depends on the drift estimates at all previous elements. Hence, we have a situation that could lead to “gain peaking” if the filters used for the drift estimation are not appropriately chosen. This should be investigated.

NRR as a mean for calculating RR

- **Rate Ratio RR at the element “N” is a product of NRRs between pairs of elements between the GM and N**

- The estimate of RR at the element “N” by receiving of the Sync-Message “j” (the same can be defined at times of sending this Sync message) is equal to:

- $RRest(GM, N, @Time = Receiving\ of\ Sync(k)) = \prod_{j=2}^{N} mNRR(k, k-1, @Time = Receiving\ of\ Sync(k)\ at\ Element\ j)$



Age of the original information about the GM frequency the element N sees after receiving the Sync(k) message

→ The drift rate of GM LocalClock should not change in this period!