Interplay of security and clock synchronization

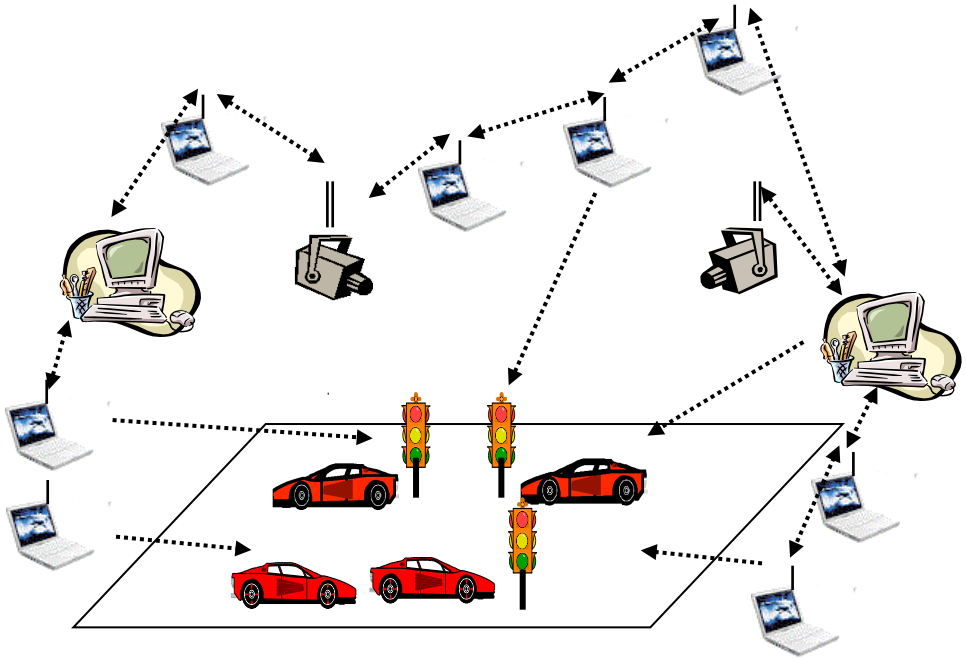
Yih-Chun Hu and P.R. Kumar

Dept. of Electrical and Computer Engineering, and
Coordinated Science Lab
University of Illinois, Urbana-Champaign

Sep 4-5, 2008

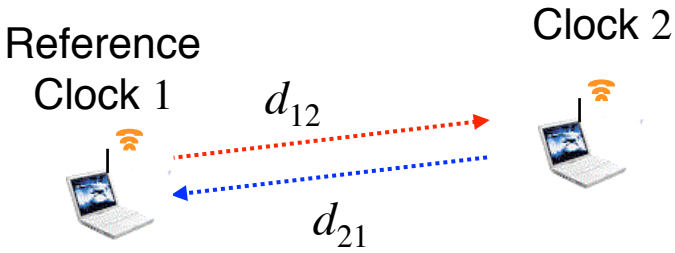
Clock synchronization over networks

- Knowledge of time is important in Networks
  - Communication network protocols
  - Sensor network applications
  - Networked control
- However no two clocks agree
- Several issues
- How to synchronize clocks in wireless networks?
- Can clock synchronization be helpful vis-à-vis security?
- And what about security of clock synchronization itself?

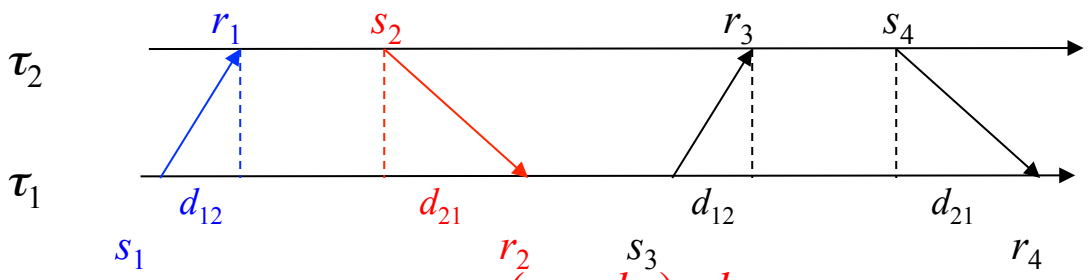


It is impossible to synchronize two clocks

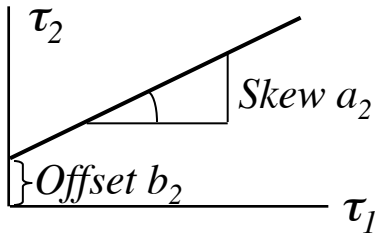
- **Theorem (Graham & K '04)**
  - It is impossible to determine $(d_{12}, d_{21}, a_2, b_2)$ through any packet exchanges



$$r_1 = a_2(s_1 + d_{12}) + b_2$$



$$s_2 = a_2(r_2 - d_{21}) + b_2$$



$$\begin{bmatrix} r_1 \\ s_2 \\ r_3 \\ s_4 \\ \dots \end{bmatrix} = \begin{bmatrix} s_1 & 1 & 0 & 1 \\ r_2 & 0 & -1 & 1 \\ s_3 & 1 & 0 & 1 \\ r_4 & 0 & -1 & 1 \\ \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} a_2 \\ a_2 d_{12} \\ a_2 d_{21} \\ b_2 \end{bmatrix}$$

Rank 3:
Cannot estimate 4 parameters

So what is determinable?

- **Theorem**

  i. The skew a_2 can be estimated correctly.

  ii. The round-trip delay $(d_{1j} + d_{j1})$ can be estimated precisely.

  iii. The sender can predict the receiver's time at which receiver receives a packet.

  iv. The offset is unknown. It represents one undeterminable degree of freedom.

  v. Delays are affine functions of the unknown offset.

  vi. By invoking causality, we can determine an interval in which the offset lies

Proof

$$\begin{bmatrix} r_1 \\ s_2 \\ r_3 \\ s_4 \\ \dots \end{bmatrix} = \begin{bmatrix} s_1 & 1 & 0 & 1 \\ r_2 & 0 & -1 & 1 \\ s_3 & 1 & 0 & 1 \\ r_4 & 0 & -1 & 1 \\ \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} a_2 \\ a_2 d_{12} \\ a_2 d_{21} \\ b_2 \end{bmatrix}$$

$$a_2^* := \frac{r_{1,2}^{(k)} - r_{1,2}^{(l)}}{s_1^{(k)} - s_1^{(l)}}$$

$$d_{12}^* := \frac{r_{1,2}^{(k)} - a_2^* s_1^{(k)}}{a_2^*} \qquad d_{21}^* := \frac{a_2^* r_{2,1}^{(l)} - s_2^{(l)}}{a_2^*}$$

$$\hat{d}_{12} \geq 0 \text{ and } \hat{d}_{21} \geq 0 \Rightarrow \hat{b}_2 \in [-a_2^* d_{21}^*, a_2^* d_{12}^*]$$

$$r_{1,2}^{(k)} = a_2 s_1^{(k)} + a_2 d_{12} + b_2 = a_2^* s_1^{(k)} + a_2^* d_{12}^*$$

Interplay between Clock Synchronization and Security

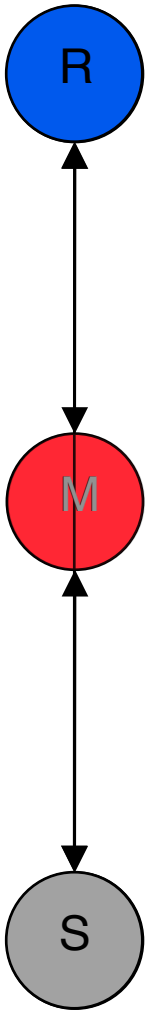
(Hu & K ’08)

Man in the middle attack

- What must a *Man in the Middle* do to remain undetected?
- What resources does a *Man in the Middle* need to remain undetected?
- How to challenge the *Man in the Middle*?

- Can we synchronize clocks in spite of the *Man in the Middle?*

- M provides a logical channel between S and R
  - M cannot decrypt any messages between S to R
  - M cannot alter any messages between S and R
  - M cannot create any fake messages between S and R
  - M can *occasionally* discard messages between S and R

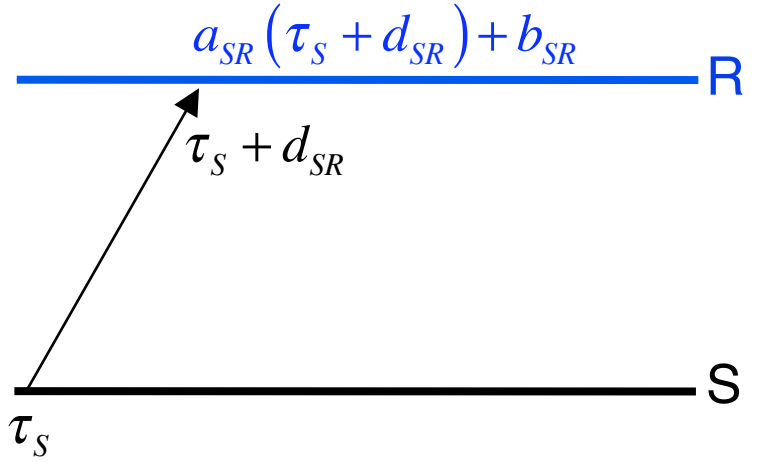


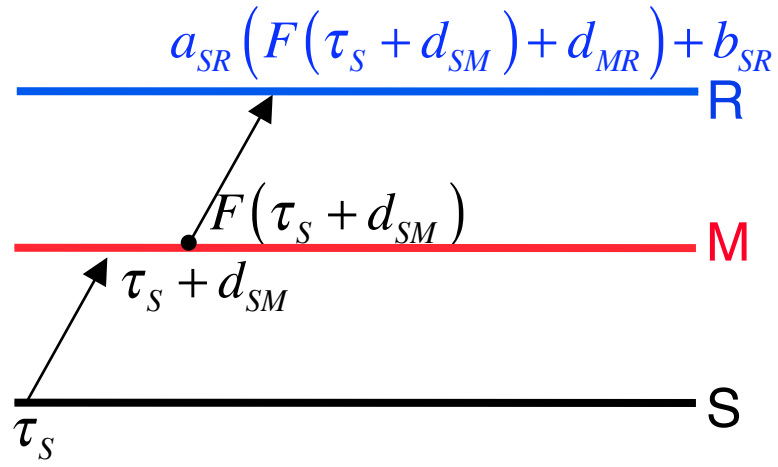
What can Man-in-the-Middle do?

Affine forwarding policy

- Without Man in the Middle
  - Time received is affine in τ_S
  - Coefficient a_{SR} is estimate of skew
- With Man in the Middle
- M's forwarding policy
  - Packet received at τ
  - Forwarded at $F(\tau)$
- Receipt time $a_{SR}\left(F\left(\tau_S + d_{SM}\right) + d_{MR}\right) + b_{SR}$

  has to be affine in τ_S
- So $F(\tau)$ has to be affine in τ

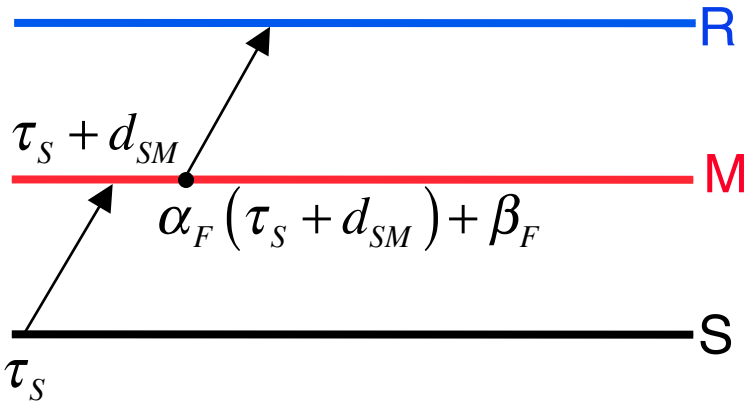




Expansionary affine forwarding policy

- Consider affine forwarding policy

$$F(\tau) = \alpha_F \tau + \beta_F$$



- Causality

- Forwarding packet can only take place *after* receiving packet

$$\alpha_F(\tau_S + d_{SM}) + \beta_F \geq \tau_S + d_{SM} \text{ for all } \tau_S$$

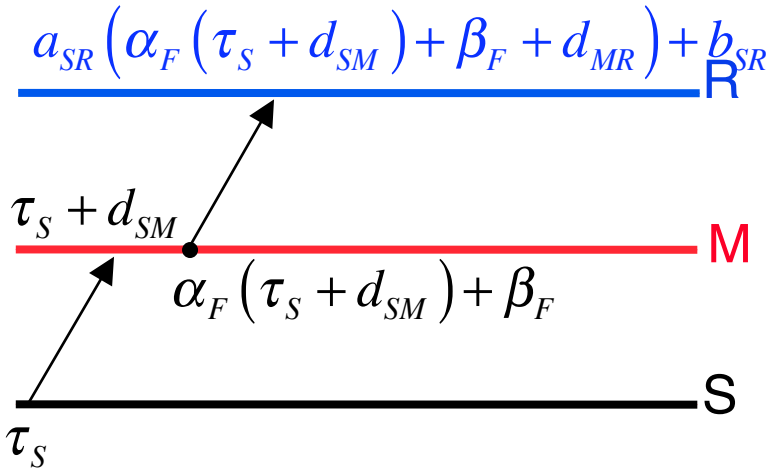
- So $\alpha_F \geq 1$

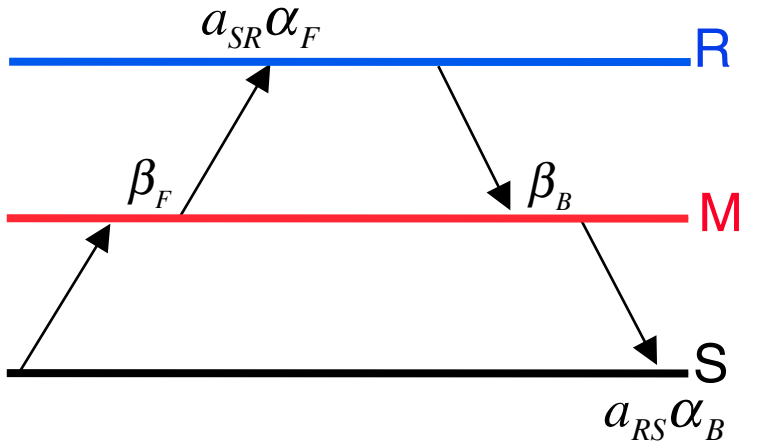
M can only add a *delay* to packets

- Estimate of skew = Coefficient of τ_S
- So skew estimate made by R with reference to S is $a_{SR}\alpha_F$
- *Backward* path skew estimate made by S with reference to R is $a_{RS}\alpha_B$
- But *product* of skew estimates has to be 1

$$a_{SR}\alpha_F a_{RS}\alpha_B = \alpha_F\alpha_B = 1$$

- But $\alpha_F \geq 1$ and $\alpha_B \geq 1$
- So $\alpha_F = \alpha_B = 1$
- Forwarding time is *pure delay*: $F(\tau) = \tau + \beta_F$

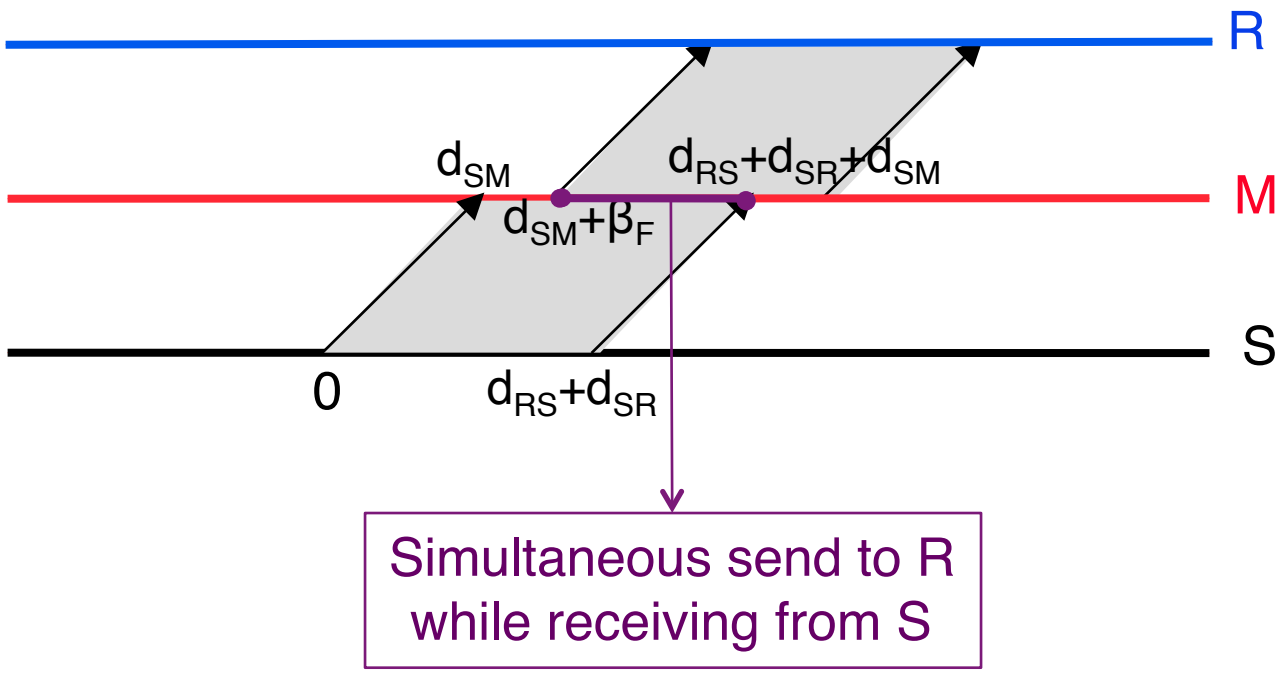




Detecting a Half-duplex Man-in-the-Middle

Detecting a Half-Duplex Man-in-the-Middle

- Send a long packet of duration greater than $d_{RS}+d_{SR}$



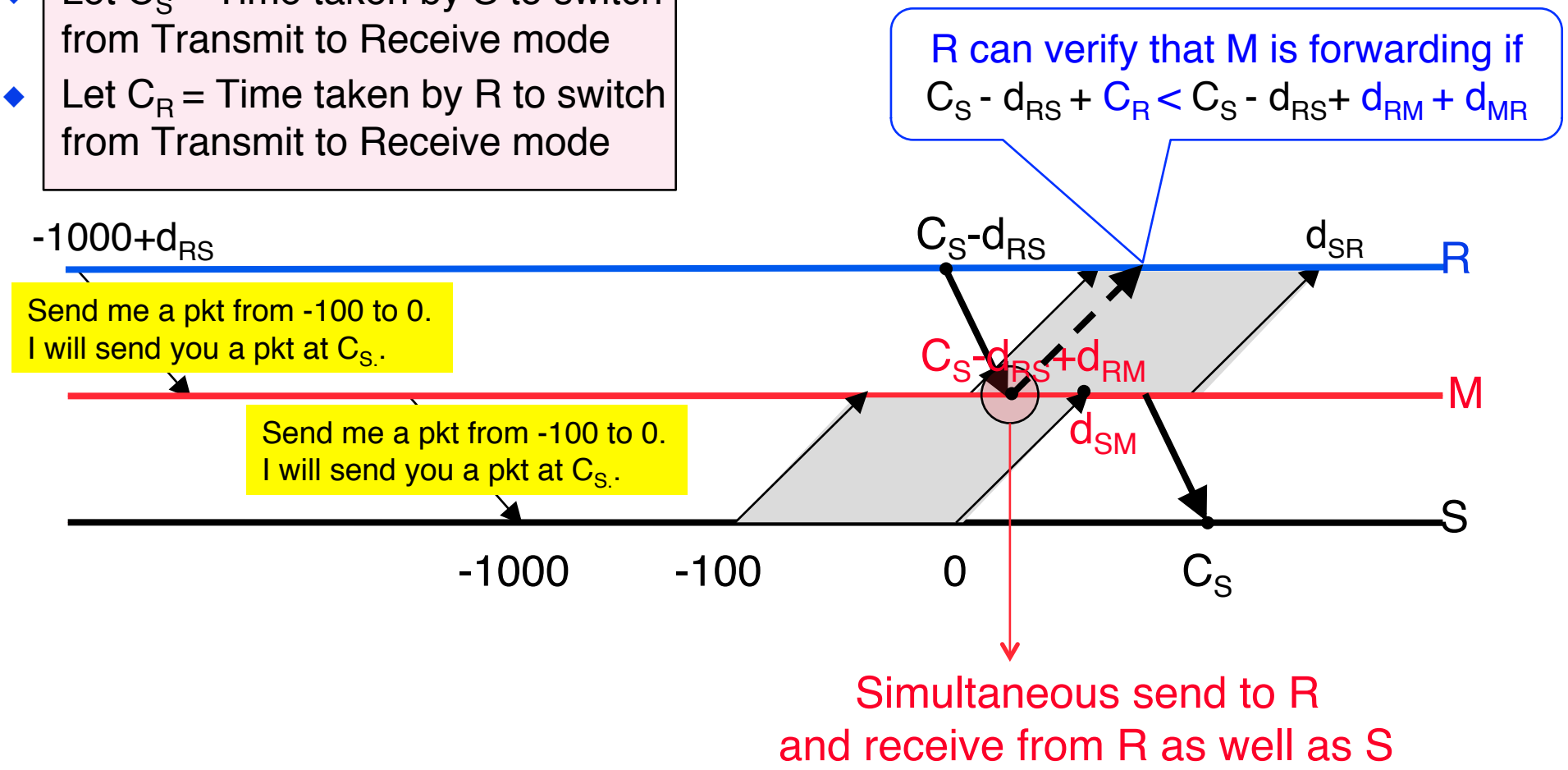
- M will need to simultaneously receive and send for a *positive* duration:

$$(d_{RS}+d_{SR}+d_{SM}) - (d_{SM}+\beta_F) = d_{RS}+d_{SR} - \beta_F > 0$$

Detecting a Full-Duplex Man-in-the-Middle

The Simultaneous Receive, Send, Receive Challenge (SRSR Challenge)

- Let C_S = Time taken by S to switch from Transmit to Receive mode
- Let C_R = Time taken by R to switch from Transmit to Receive mode



- If $C_R < d_{RM} + d_{MR}$ and $C_S - d_{RS} + d_{RM} < d_{SM}$ then M gets caught

The Switch Time condition for detecting the Man-in-the-Middle

- IF $C_R < d_{RM} + d_{MR}$ and $C_S - d_{RS} + d_{RM} < d_{SM}$

  OR

  $C_S < d_{SM} + d_{MS}$ and $C_R - d_{SR} + d_{SM} < d_{RM}$

  X, X → + → If both violated, then $C_S + C_R \geq d_{RS} + d_{SR}$

  then M gets caught in one of the two directions

- Suppose $C_R < d_{RM}+d_{MR}$ AND $C_S<d_{SM}+d_{MS}$
- Then $C_R+C_S < d_{RM}+d_{MR}+d_{SM}+d_{MS} < d_{SR} + d_{RS}$
- Hence EITHER $C_S - d_{RS} + d_{RM} < d_{SM}$ OR $C_R - d_{SR} + d_{SM} < d_{RM}$ holds
- Thus M gets caught in one of the two directions

When is Man-in-the-Middle impossible to detect?

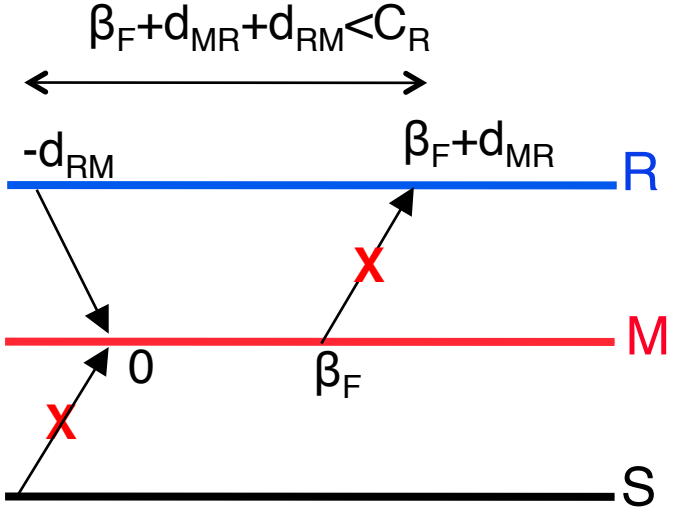
Impossibility condition for detecting Man-in-the-Middle

- **Theorem**
  - If $C_R > d_{MR}+d_{RM}$ OR $C_S > d_{MS}+d_{SM}$
  - Then Man-in-the-Middle can evade detection
- **Proof**
- Suppose $C_R > d_{MR}+d_{RM}$ wlog
- Consider the following *RS-Priority Detection Prevention Strategy* for M
- *Choose* forwarding delay $0 \le \beta_F < C_R - d_{MR} - d_{RM}$ for both R to S and S to R
- **Conflict resolution strategy**: Give *priority* to RS packets (from R to S)
  - When packets from both R and S are incoming, *listen only to R* and not S
  - When M needs to transmit to both R and S, *transmit only to S* and not R

RS Priority Detection Avoidance Strategy

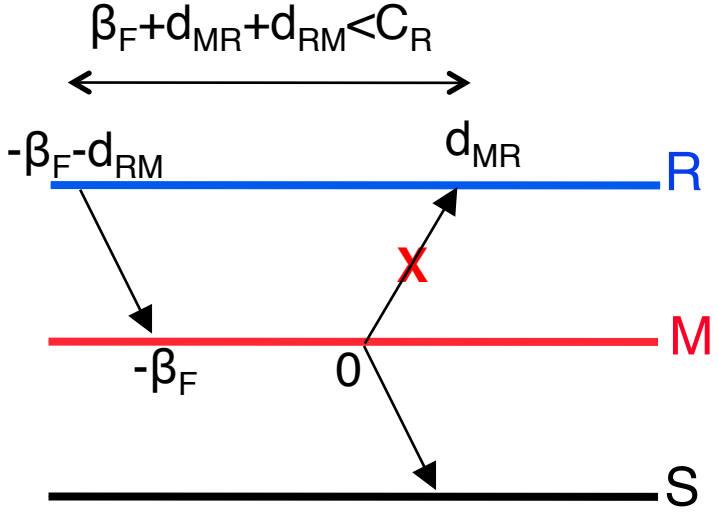
- All packets from R are received and sent to S
- We only need to consider packets from S to R: Recall $\beta_F < C_R - d_{MR} - d_{RM}$

- **Receiver conflict**
- R cannot check reception



- What M doesn't hear doesn't hurt

- **Transmitter conflict**
- R cannot check reception



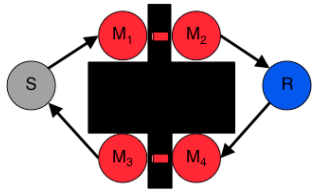
- What M doesn't transmit doesn't hurt

Man in the Middle Attack: Clocks, Detectability and Consequences

- **Theorem**
- A *Half Duplex Man-in-the-Middle* can always be detected

- A *Full Duplex Man-in-the-Middle* will be detected by the SRSR Challenge if both turnaround times are short: $C_R < d_{RM}+d_{MR}$ AND $C_S<d_{SM}+d_{MS}$
- The *Full Duplex Man-in-the-Middle* can avoid detection by:
  - An *RS-Priority Strategy with Low Forwarding Delay* (RSLFD) if $C_R > d_{RM}+d_{MR}$,
  - By an SRLFD Policy if $C_S > d_{SM}+d_{MS}$

- The *Double Full Duplex* Man-in-the-Middle can avoid detection
  - Two simultaneous transmitters, two simultaneous receivers with shielding between transmitters and receivers, and two tunnels



- Even when he can avoid detection
  - A Man-in-the-Middle can only *add a pure delay* in each direction
  - The delay should be *small enough* if Man-in-the-Middle is Full Duplex
  - So time-based applications are still *temporally consistent*

Thank you