Time Management in the High Level Architecture

Richard M. Fujimoto
Professor

Computational Science and Engineering Division
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0765, USA

http://www.cc.gatech.edu/~fujimoto/

Outline

- Overview of time management services
- Time constrained and time regulating federates
- Related object management services
- Time Advance Request (TAR)
- Next Event Request (NER)
- Lookahead

HLA Message Ordering Services

The baseline HLA provides two types of message ordering:

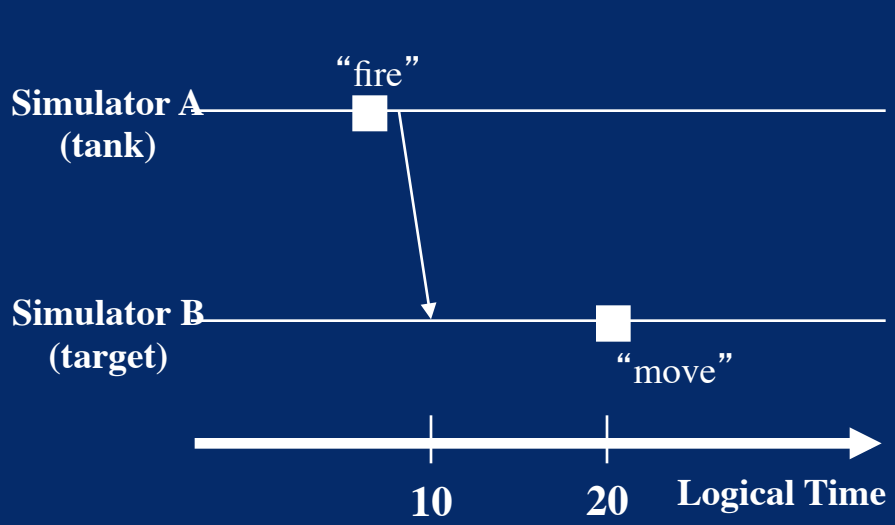
- ***receive order (unordered):*** messages passed to federate in an arbitrary order
- ***time stamp order (TSO):*** sender assigns a time stamp to message; successive messages passed to each federate have non-decreasing time stamps

Property	Receive Order (RO)	Time Stamp Order (TSO)
Latency	*low*	*higher*
reproduce before and after relationships?	*no*	*yes*
all federates see same ordering of events?	*no*	*yes*
execution repeatable?	*no*	*yes*
typical applications	*training, T&E*	*analysis*

- receive order minimizes latency, does not prevent temporal anomalies
- TSO prevents temporal anomalies, but has somewhat higher latency

Time Synchronized Delivery

Consider interconnecting two sequential, discrete event simulators



event

message

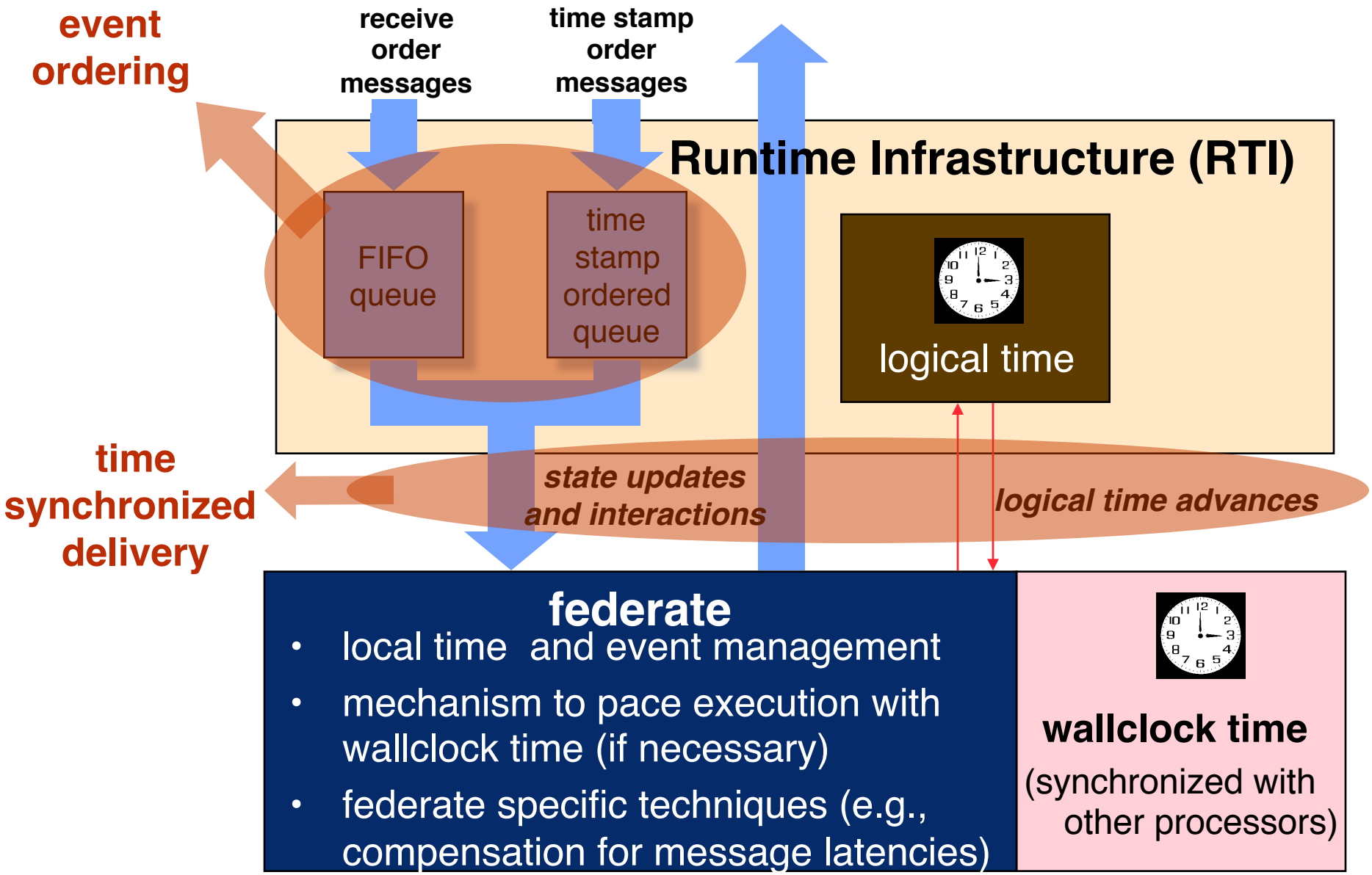
1. A sends TSO message to B w/ time stamp 10
2. B advances to logical time 20
3. Message arrives in B's past

In the HLA, logical time is synonymous with simulation time

Logical time advances by each simulator must be properly managed to ensure no simulator receives a message in its past.

HLA Time Management (TM) services define a protocol for federates to advance their logical time; RTI will ensure TSO messages are not delivered in a federate's past

HLA Time Management Services



Time Regulating and Time Constrained Federates

Federates must declare their intent to utilize time management services by setting their *time regulating* and/or *time constrained* flags

- Time regulating federates: can send TSO messages
  - Can prevent other federates from advancing their logical time
  - Enable Time Regulation … Time Regulation Enabled †
  - Disable Time Regulation
- Time constrained federates: can receive TSO messages
  - Time advances are constrained by other federates
  - Enable Time Constrained … Time Constrained Enabled †
  - Disable Time Constrained
- Each federate in a federation execution can be
  - Time regulating only (e.g., message source)
  - Time constrained only (e.g., Stealth)
  - Both time constrained and regulating (common case for analytic simulations)
  - Neither time constrained nor regulating (e.g., DIS-style training simulations)

† indicates callback to federate

Related Object Management Services

Sending and Receiving Messages

- Update Attribute Values … Reflect Attribute Values †
- Send Interaction … Receive Interaction †

Message Order (*Receive Order or Time Stamp Order*)

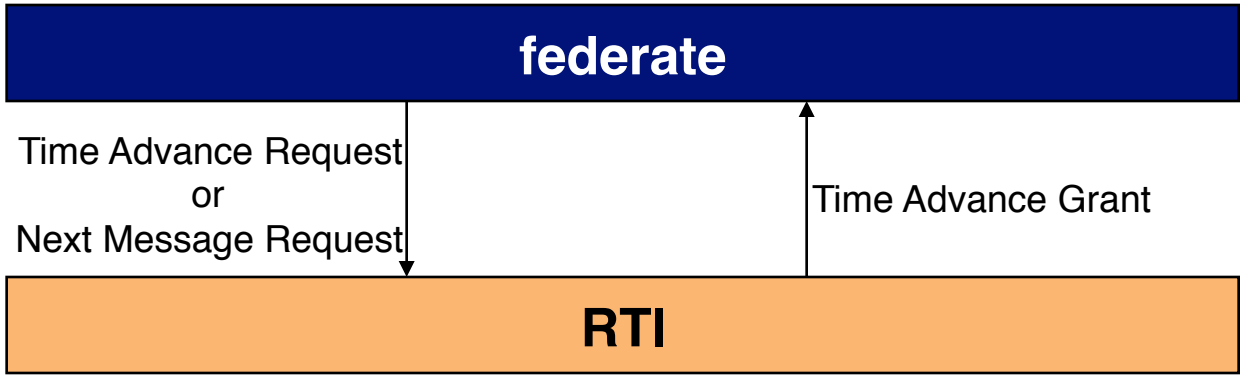
- *Preferred Order Type:* default order type specified in "fed file" for each attribute and interaction
- *Sent Message Order Type:*
  - TSO if preferred order type is TSO and the federate is time regulating and a time stamp was used in the *Update Attribute Values* or *Send Interaction* call
  - RO otherwise
- Received Message Order Type
  - TSO if sent message order type is TSO and receiver is time constrained
  - RO otherwise

† indicates callback to federate

HLA Time Management (TM) Services

HLA TM services define a protocol for federates to advance logical time; logical time only advances when that federate explicitly requests an advance

- Time Advance Request: *time stepped federates*
- Next Message Request: *event stepped federates*
- Time Advance Grant: RTI invokes to acknowledge logical time advances



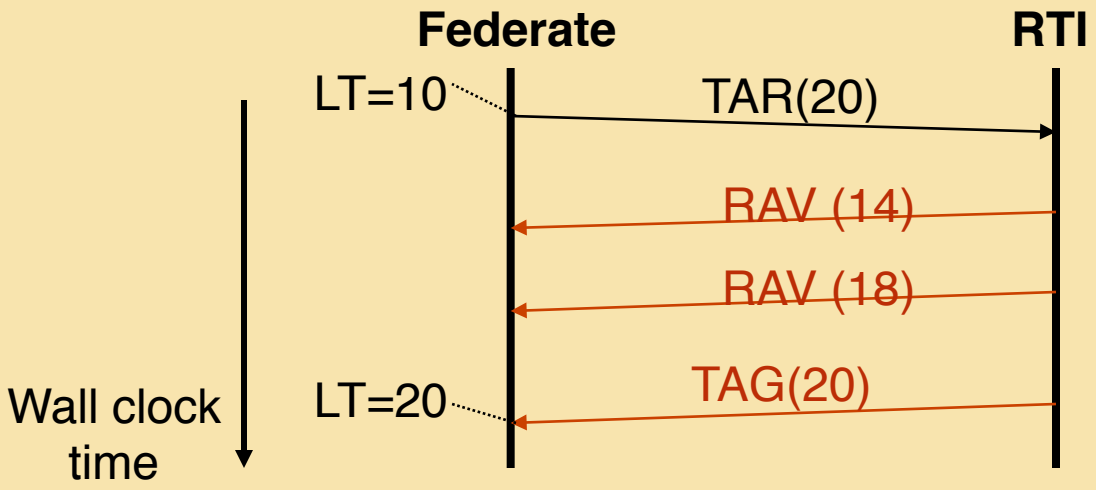
If the logical time of a federate is T, the RTI guarantees no more TSO messages will be passed to the federate with time stamp < T

Federates responsible for pacing logical time advances with wallclock time in real-time executions

Time Advance Request (TAR)

- Typically used by time stepped federates
- Federate invokes Time Advance Request (T) to request its logical time (LT) be advanced to T
- RTI delivers all TSO messages with time stamp ≤ T
- RTI advances federate's time to T, invokes Time Advance Grant (T) when it can guarantee all TSO messages with time stamp ≤ T have been delivered
- Grant time always matches the requested time

Typical execution sequence



TAR: Time Advance Request
RAV: Reflect Attribute Values
TAG: Time Advance Grant

Federate calls in black
RTI callbacks in red

Code Example: Time Stepped Federate

sequential simulator

```
T = current simulation time
While (simulation not complete)
    update local simulation state
    T = T + ΔT;
End-While
```

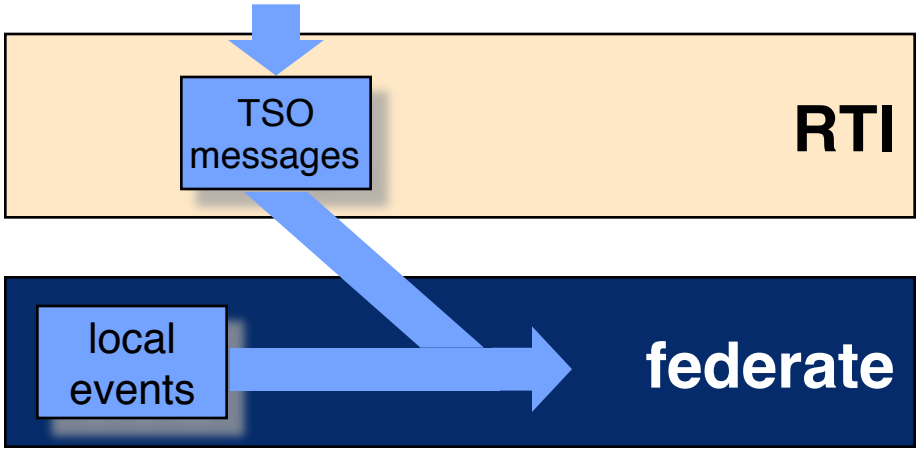
federated simulator

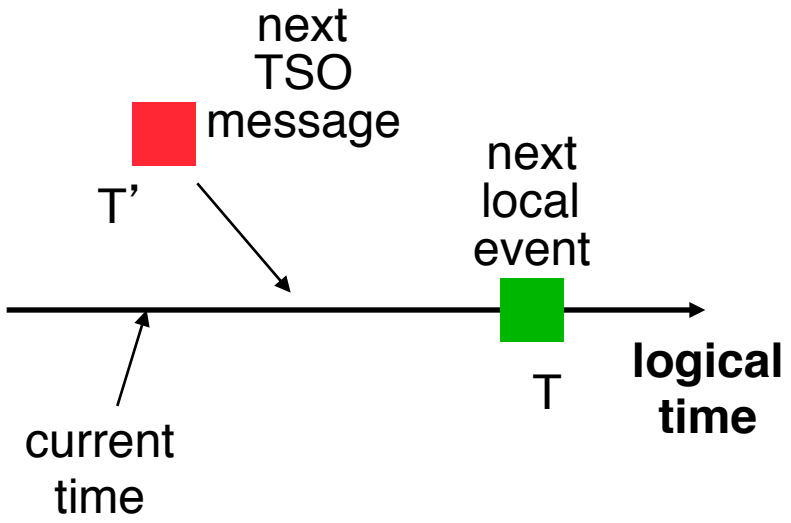
```
While (simulation not complete)
    update local simulation state
    UpdateAttributeValues (…)
    PendingTAR = TRUE;
    TimeAdvanceRequest(T+ ΔT)
    while (PendingTAR) Tick*(…);
    T = T + ΔT;
End-While

/* the following federate-defined
    procedures are called by the RTI */
Procedure ReflectAttributeValues (…)
    update local state

Procedure TimeAdvanceGrant (…)
    PendingTAR = False;
```

* Tick is only used in single threaded RTI implementations

Next Message Request (NMR)

- Typically used by event stepped federates
- Goal: process all events (local and incoming TSO messages) in time stamp order





Federate: next local event has time stamp T

- If no TSO messages w/ time stamp < T, advance to T, process local event
- If there is a TSO message w/ time stamp T' ≤ T, advance to T' and process TSO message

Next Message Request (NMR)

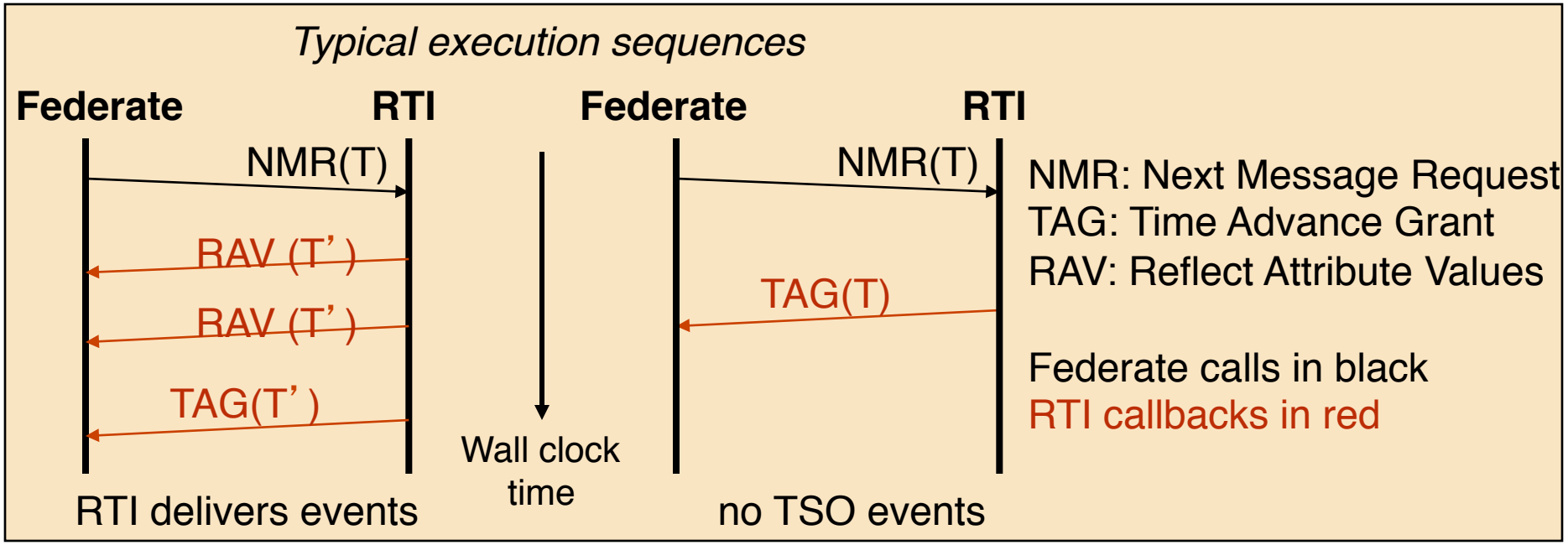
Federate invokes Next Message Request (T) to request its logical time be advanced to time stamp of next TSO message, or T, which ever is smaller

If next TSO message has time stamp T' ≤ T

- RTI delivers next TSO message, and all others with time stamp T'
- RTI issues Time Advance Grant (T')

Else

- RTI advances federate's time to T, invokes Time Advance Grant (T)



Code Example: Event Stepped Federate

sequential simulator

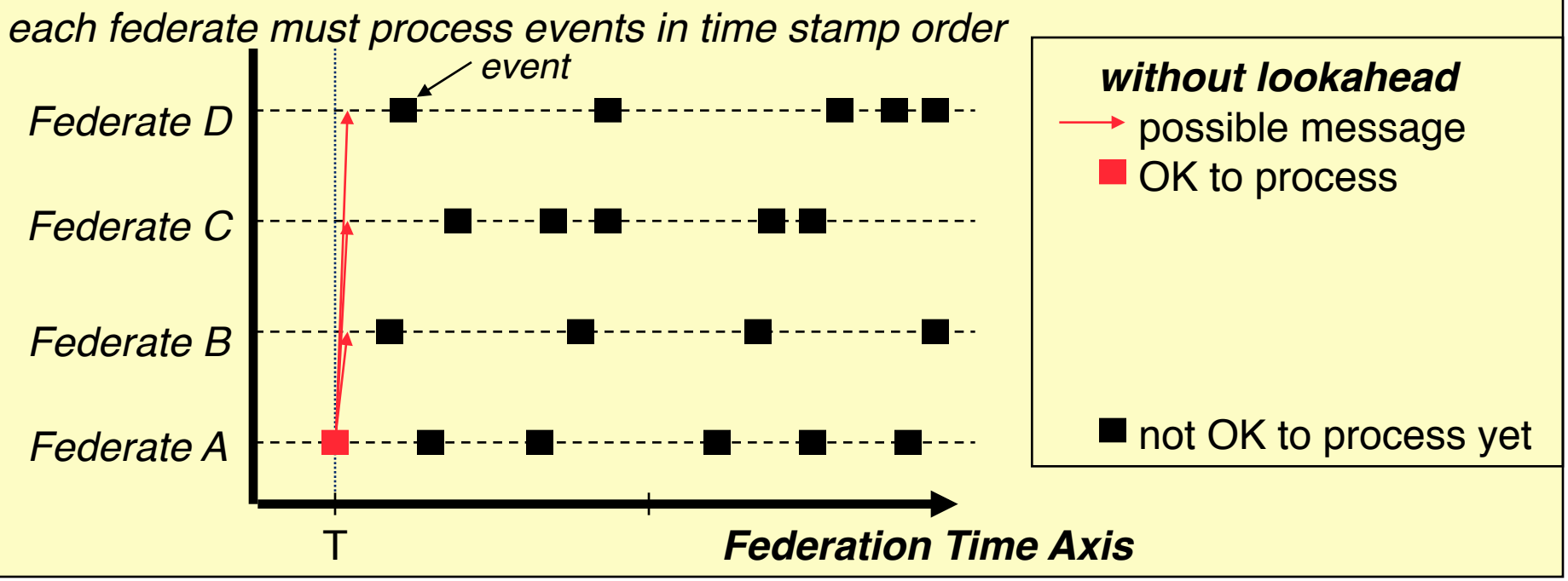
```
T = current simulation time
PES = pending event set

While (simulation not complete)
    T = time of next event in PES
    process next event in PES
End-While
```

federated simulator

```
While (simulation not complete)
    T = time of next event in PES
    PendingNMR = TRUE;
    NextMessageRequest(T)
    while (PendingNMR) Tick(…);
    process next event in PES
End-While

/* the following federate-defined
    procedures are called by the RTI */
Procedure ReflectAttributeValues (…)
    place event in PES

Procedure TimeAdvanceGrant (…)
    PendingNMR = False;
```

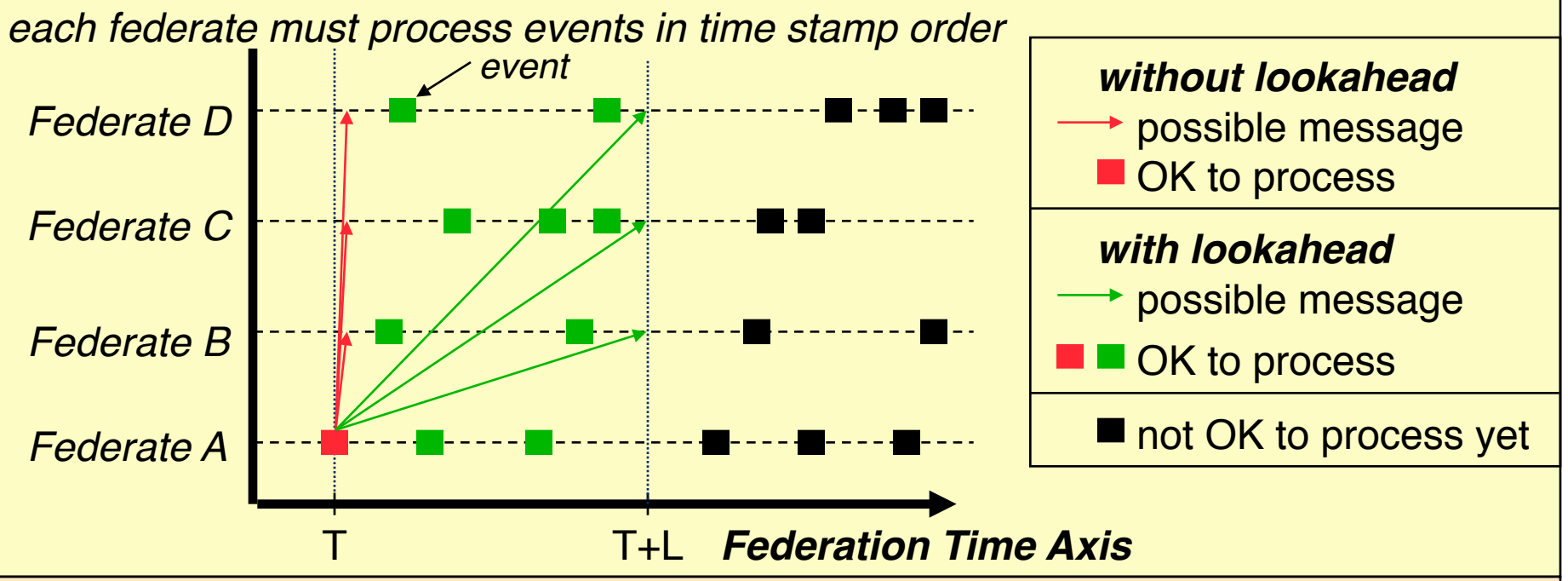
Lookahead

NMR: concurrency limited to events containing exactly the same time stamp



Lookahead

NMR: concurrency limited to events containing exactly the same time stamp

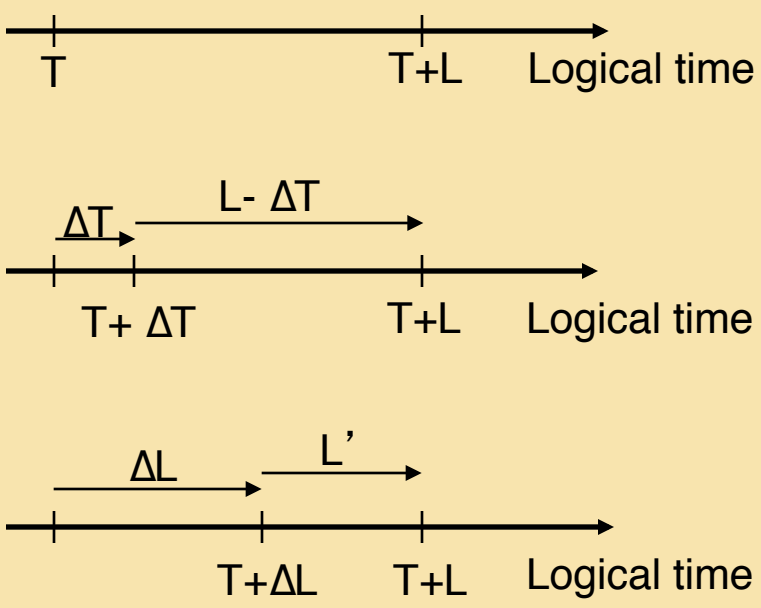


Each federate using logical time declares a lookahead value L; any TSO message sent by the federate must have a time stamp ≥ the federate's current time + L

Lookahead is necessary to allow concurrent processing of events with different time stamps (unless optimistic event processing is used)

Lookahead in the HLA

- Each federate must declare a non-negative lookahead value
- Any TSO sent by a federate must have time stamp at least the federate's current time plus its lookahead
- Lookahead can change during the execution (*Modify Lookahead*)
  - increases take effect immediately
  - decreased do not take effect until the federate advances its logical time



1. Current time is T, lookahead L
2. Request lookahead decrease by ΔL to L'

3. Advance ΔT, lookahead, decreases ΔT

4. After advancing ΔL, lookahead is L'

Federate/RTI Guarantees

Federate at logical time T (with lookahead L)

- All outgoing TSO messages must have time stamp ≥ T +L (L>0)

Time Advance Request (T)

- Once invoked, federate cannot send messages with time stamp less than T plus lookahead

Next Message Request (T)

- Once invoked, federate cannot send messages with time stamp less than T plus the federate's lookahead unless a grant is issued to a time less than T

Time Advance Grant (T) (after TAR or NMR service)

- All TSO messages with time stamp less than or equal to T have been delivered

Summary of Time Management Services

- Time Advance Request (T) [TAR(T)]
  - Federate$_i$ calls TAR to request logical time to advance to T
  - Federate$_i$ unconditionally guarantees all subsequent messages it generates will have time stamp ≥ T+L_i
  - typically used by time-stepped federates
- Next Message Request (T) [NMR(T)]
  - T is typically the time stamp of the next event local to the federate
  - Federate$_i$ calls NMR(T) to request next message with time stamp ≤ T
    - RTI delivers all events at time T' (T' ≤ T) if there are any, and advances LT_i to T', or
    - if there are no such events, LT_i is advanced to T
  - typically used by event driven federates
  - by invoking NMR(T) Federate$_i$ conditionally guarantees it will not generate any new messages with time stamp <T+L_i if it does not receive additional TSO messages with time stamp < T
- Any federate can interleave calls to TAR and NMR

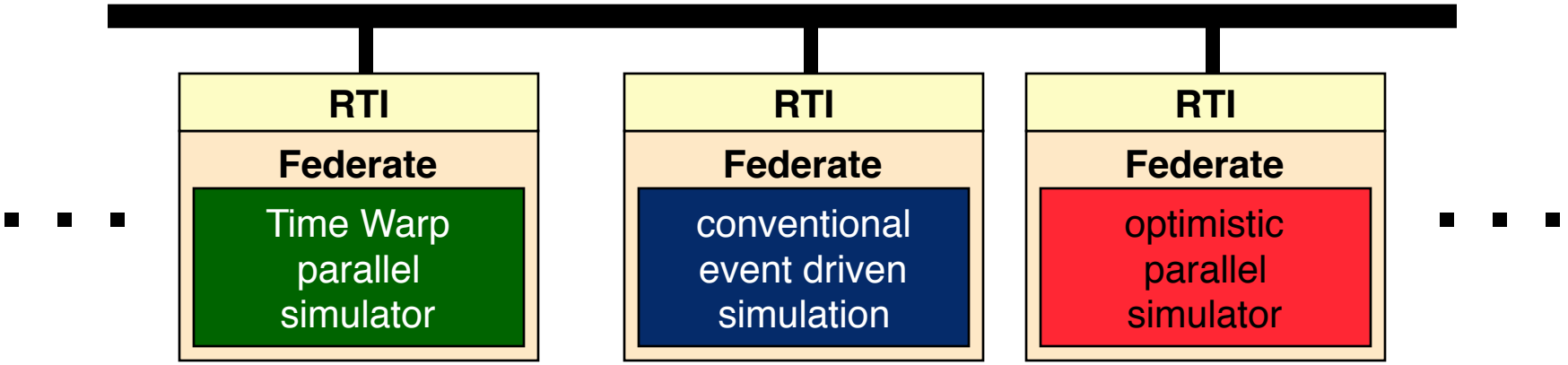
Synchronization Transparency

- HLA intended to support interoperability among different, autonomous simulators
- Within a single federation, different federates may use different time advancement mechanisms
  - Time stepped
  - Event stepped
  - Optimistic
- *Synchronization transparency*: Each federate need not know the time advancement mechanism used by other federates

Optimistic Federates

optimistic federates:

- allow events to be processed out of time stamp order
- use a roll back mechanism to erase incorrect computations
- may send erroneous messages; unsend via event cancellation (*anti-messages)*



synchronization transparency (federates need not know what synchronization mechanism is used locally by other federates) implies:

- allow optimistic federates to receive events that might later be cancelled
- prevent conservative federates from receiving events that might later be cancelled
- mechanisms required for event cancellation, GVT computation

Support For Optimistic Synchronization

- $LBTS_i$ = lower bound on time stamp of messages that might later be delivered to federate i
- *committed events*: events in RTI_i with time stamp < $LBTS_i$; will not be later canceled
- RTI only passes committed events to conservative federates
- optimistic federates can request delivery of uncommitted events **(FlushQueue)**

- state saving and restoration implemented locally within each federate
- cancel erroneous messages via **Retract** mechanism
  - RTI deletes canceled message if not yet delivered to federate
  - RTI forwards Retract operation to federate if it already delivered message (could cause secondary rollback, additional anti-messages)
  - also used by conservative federates (e.g., to model preemption)

Global Virtual Time (GVT):

- **LBTS** provides lower bound on time stamp of future messages
- rollback caused by receiving/canceling messages “in the past”
- GVT =min(LBTS, time stamp of local events+lookahead); compute locally

Summary

- HLA time management designed to support interoperability of simulations with different time advance mechanisms
  - Time stepped federates
  - Event-driven federates
- Time management services include services to order messages (time stamp ordered delivery) and mechanisms to advance simulation time
- Time regulating/constrained used to “turn on” time management
- Per federate lookahead supported
- Time management transparency and optimistic federates are supported