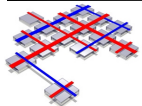


INTERVAL



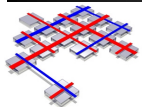
Real-time extensions to MSC-2000 and TTCN-3

Helmut Neukirchen



Institute for Telematics
University of Lübeck

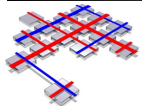
- **MSC-2000 extensions**
 - Alternatives
 - Loops
- **TTCN-3 extensions**
 - General Approach
 - An Example
- **Conclusion & Outlook**



INTERVAL

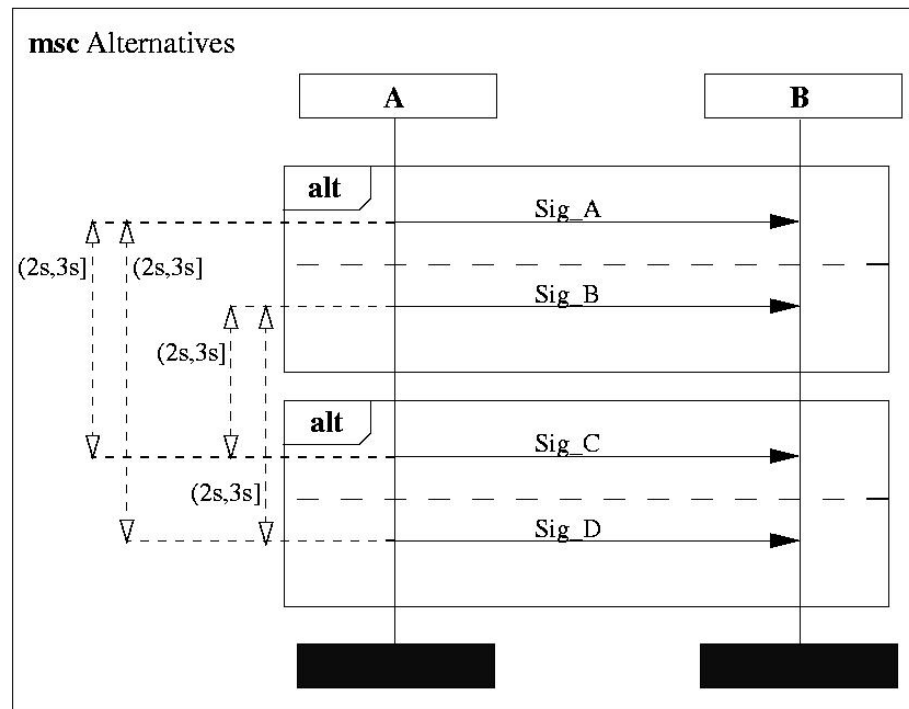
IST-1999-11557

MSC-2000 Extensions



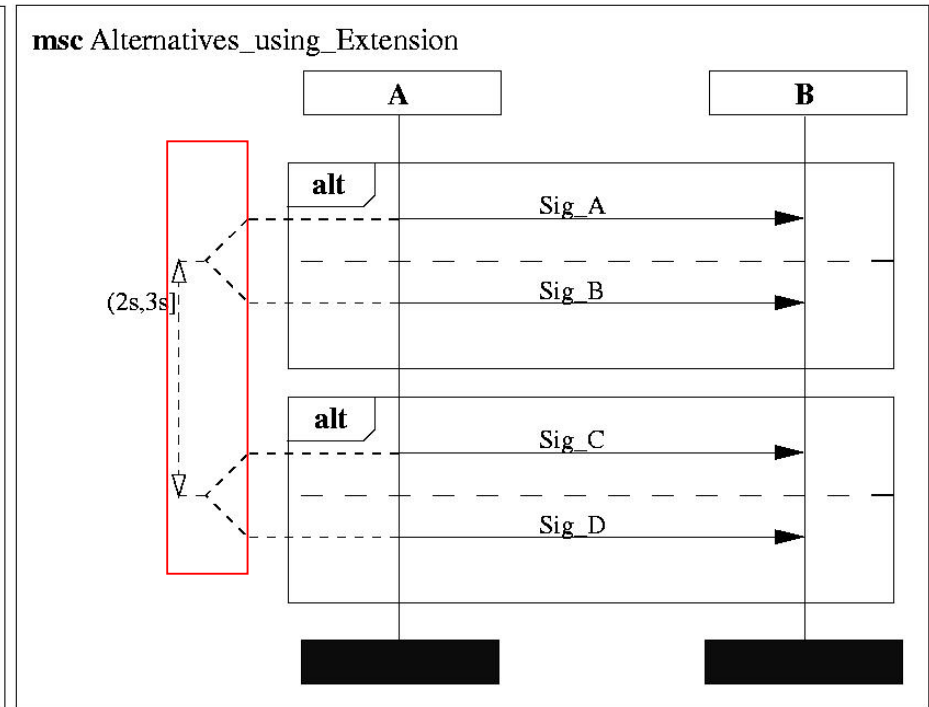
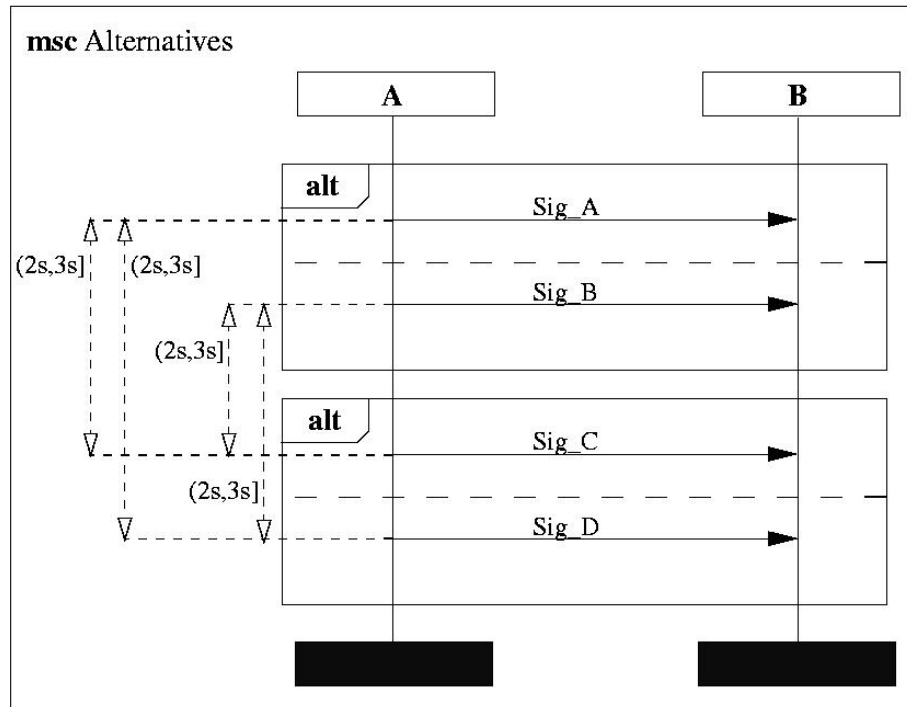
Institute for Telematics
University of Lübeck

- **Problem: Explosion of time intervals due to possible permutations**

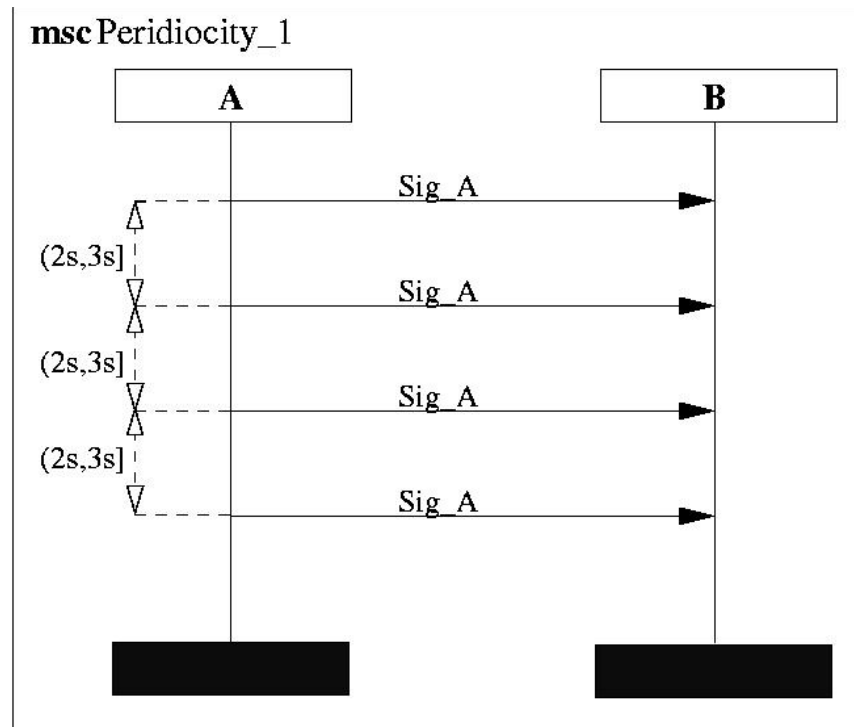


- **Problem: Explosion of time intervals**
due to possible permutations

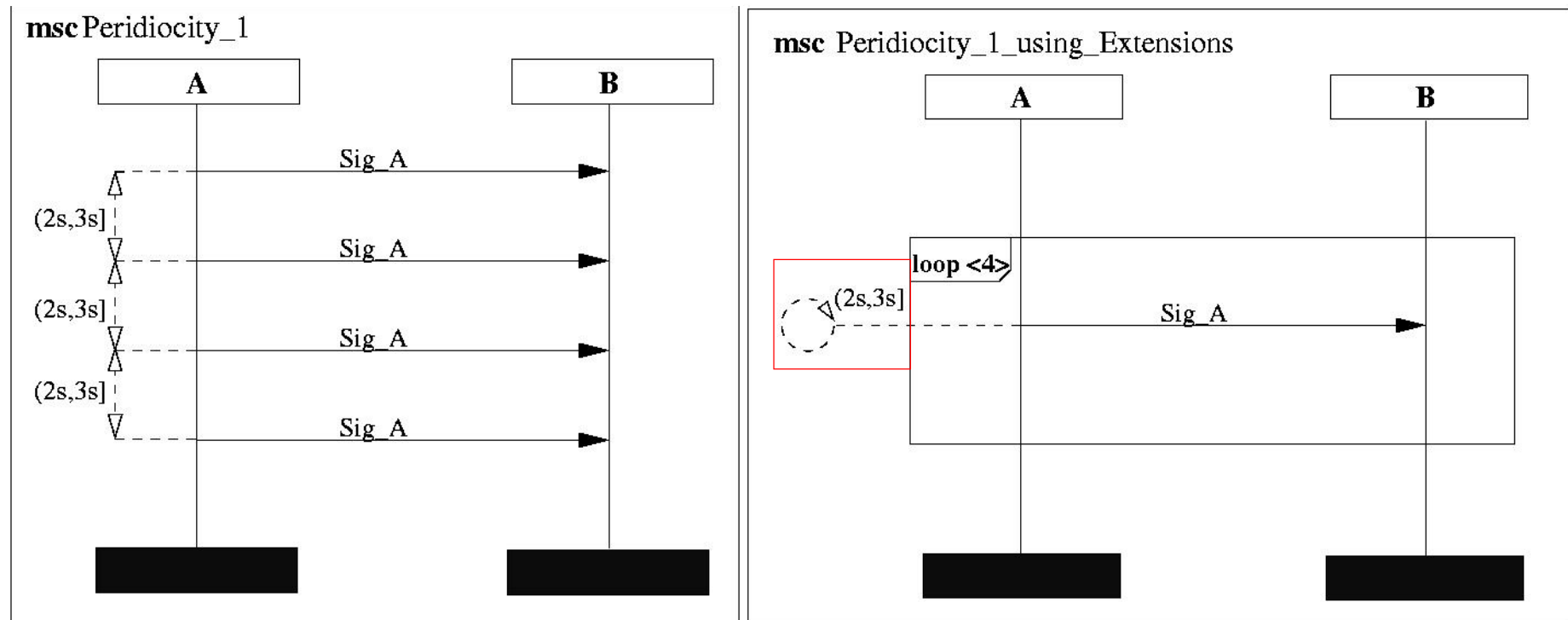
Extension:
Join dashed lines



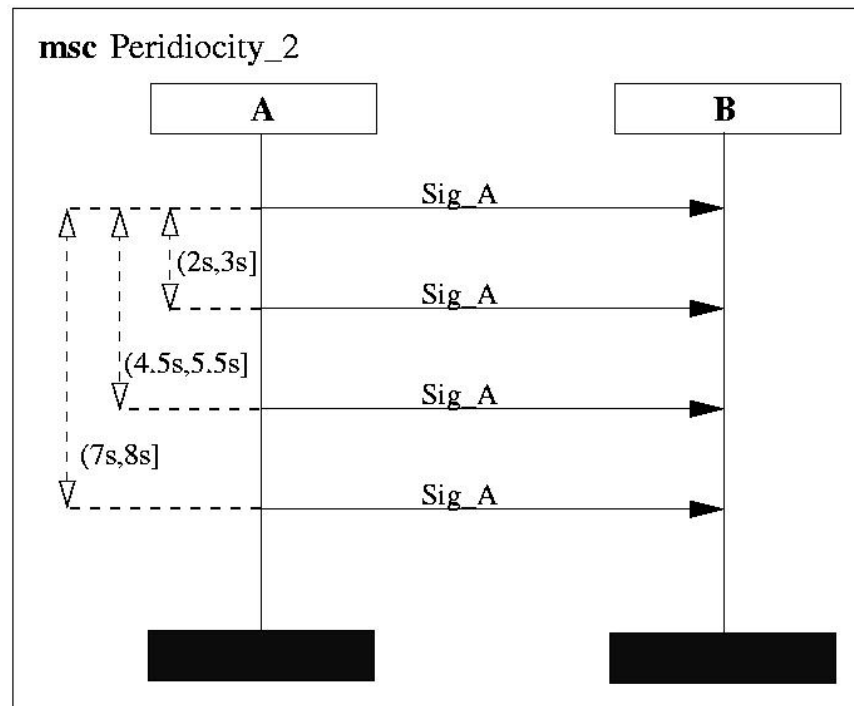
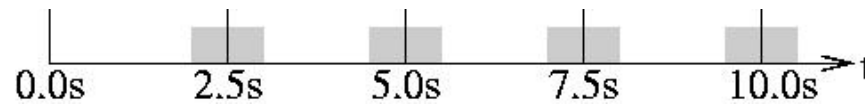
- **Problem: Periodic occurrence of events with a fixed frequency**



- **Problem: Periodic occurrence of events with a fixed frequency** **Extension: Add looped time interval**

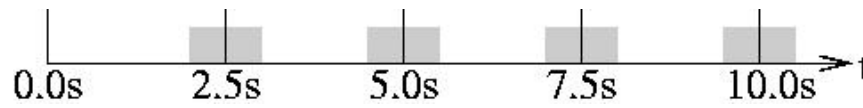


■ **Problem:**

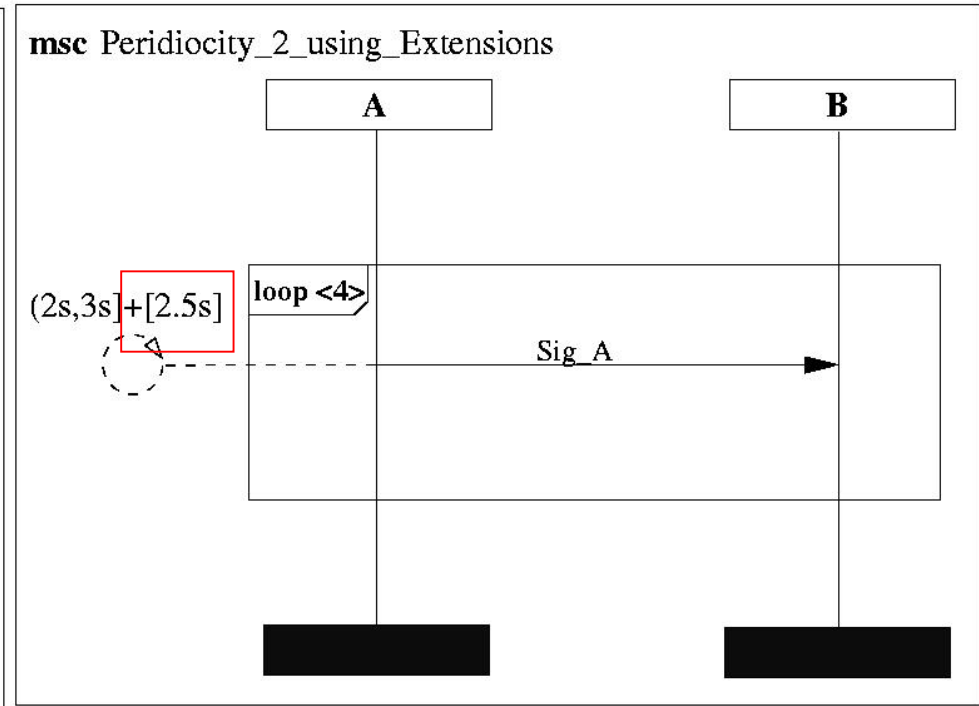
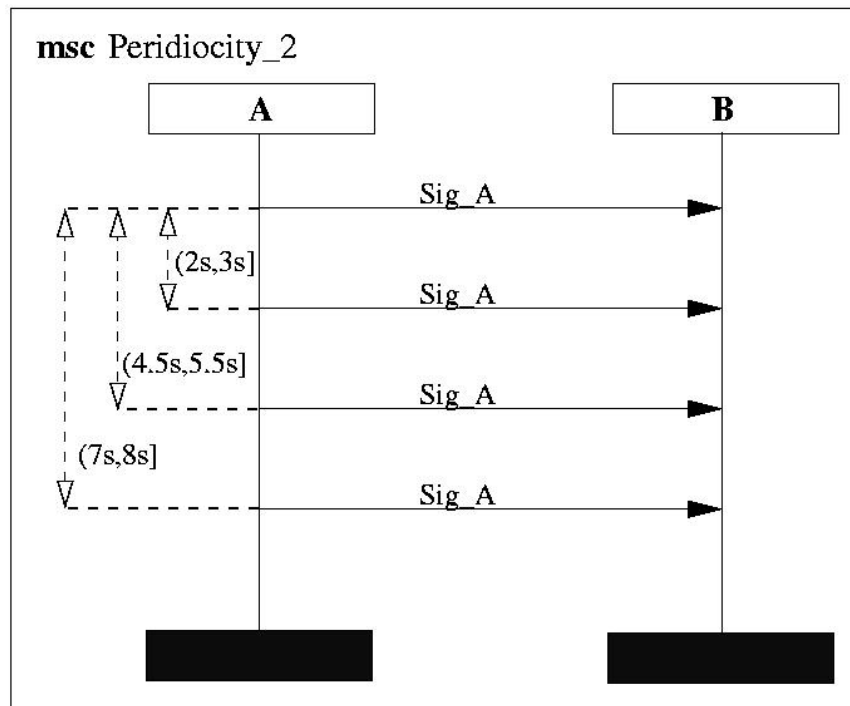


Periodicity (unbounded Jitter)

■ **Problem:**



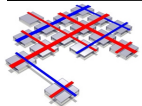
Extension:
add an offset



INTERVAL

IST-1999-11557

TTCN-3 Extensions

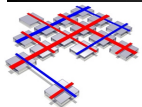


Institute for Telematics
University of Lübeck

■ **Approach:**

Separate functional and non-functional requirements:

- Instrument functional testcases to generate timestamps
- Real-time requirements are specified separately from the functional testcases



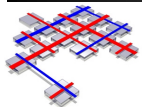
■ **Mathematical terms, which describe the relationship of timestamps t_i , e.g.**

- Response time: $\forall i : t_{bi} - t_{ai} < 10ms$
- Jitter: $\forall i : \left| \frac{\sum_{j=1}^n (t_{bj} - t_{aj})}{n} - (t_{bi} - t_{ai}) \right| < 1ms$

■ **Applying evaluation functions to the timestamps:**

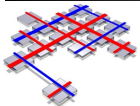
- Offline evaluation after execution of functional test cases
- Online evaluation for reactive real-time tests

■ **Assign a test verdict**



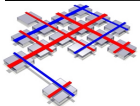
- **Necessary extensions:**
 - Read value of global clock
 - Dump structured timestamp objects to a log
 - Retrieve timestamps objects from the log

```
testcase Inres_100_Data_Transfer_Latency
{
  for (i:=1; i<=100; i:=i+1) // Send 100 data packages
  {
    ISAP1.send(IDATreq("data"));
    alt
    { [] MSAP2.receive(Medium_Data_Transfer(expected_num))
      {
        j:=1;
        MSAP2.send(Medium_Acknowledgement(expected_num));
        expected_num:=succ(expected_num);
      }
      [] MSAP2.receive(Medium_Data_Transfer(succ(expected_num)))
      { j:=j+1;
        MSAP2.send(Medium_Acknowledgement(succ(expected_num)));
      }
    }
  }
}
```



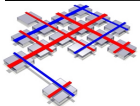
```
type record myTimestampType {time logtime, charstring id};

testcase Inres_100_Data_Transfer_Latency
{
  for (i:=1; i<=100; i:=i+1) // Send 100 data packages
  {
    ISAP1.send(IDATreq("data"));
    alt
    {
      [] MSAP2.receive(Medium_Data_Transfer(expected_num))
      {
        j:=1;
        MSAP2.send(Medium_Acknowledgement(expected_num));
        expected_num:=succ(expected_num);
      }
      [] MSAP2.receive(Medium_Data_Transfer(succ(expected_num)))
      {
        j:=j+1;
        MSAP2.send(Medium_Acknowledgement(succ(expected_num)));
      }
    }
  }
}
```



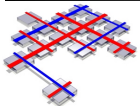
```
type record myTimestampType {time logtime, charstring id};

testcase Inres_100_Data_Transfer_Latency
{
  for (i:=1; i<=100; i:=i+1) // Send 100 data packages
  {
    timelog(myTimestampType:{now.get, "a"}); // log Timestamp
    ISAP1.send(IDATreq("data"));
    alt
    {
      [] MSAP2.receive(Medium_Data_Transfer(expected_num))
      {
        j:=1;
        MSAP2.send(Medium_Acknowledgement(expected_num));
        expected_num:=succ(expected_num);
      }
      [] MSAP2.receive(Medium_Data_Transfer(succ(expected_num)))
      {
        j:=j+1;
        MSAP2.send(Medium_Acknowledgement(succ(expected_num)));
      }
    }
  }
}
```




```
type record myTimestampType {time logtime, charstring id};

testcase Inres_100_Data_Transfer_Latency
{
  for (i:=1; i<=100; i:=i+1) // Send 100 data packages
  {
    timelog(myTimestampType:{now.get, "a"}); // log Timestamp
    ISAP1.send(IDATreq("data"));
    alt
    {
      [] MSAP2.receive(Medium_Data_Transfer(expected_num))
      {
        timelog(myTimestampType:{now.get, "b"}); // log Timestamp
        j:=1;
        MSAP2.send(Medium_Acknowledgement(expected_num));
        expected_num:=succ(expected_num);
      }
      [] MSAP2.receive(Medium_Data_Transfer(succ(expected_num)))
      {
        j:=j+1;
        MSAP2.send(Medium_Acknowledgement(succ(expected_num)));
      }
    }
  }
}
}
```



■ Log-File with timestamp objects:

```
{5.619, "a"}  
ISAP1 send IDATreq ( 0 )  
MSAP2 receive MDATind ( DT, one, 0 )  
{5.627, "b"}  
MSAP2 send MDATreq ( AK, one, 0 )  
{5.632, "a"}  
ISAP1 send IDATreq ( 0 )  
MSAP2 receive MDATind ( DT, zero, 0 )  
{5.641, "b"}  
MSAP2 send MDATreq ( AK, zero, 0 )
```

■ **Log-File with timestamp objects:**

{5.619, "a"}

{5.627, "b"}

{5.632, "a"}

{5.641, "b"}

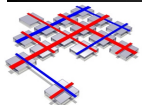
■ Evaluation function:

```
function latency(charstring idFirst, charstring idSecond, float
lowerbound, float upperbound) return verdicttype
{
  var myTimestampType t_a, t_b;
  var int i;
  for(i:=0; i<=100; i:=i+1)
  {
    timelog.search(myTimestampType:{*,idFirst}, i)->value t_a;
    timelog.search(myTimestampType:{*,idSecond}, i)->value t_b;
    if ((t_b.time-t_a.time<upperbound)
and (t_b.time-t_a.time>lowerbound))
      verdict.set(pass);
    else
      verdict.set(fail);
  }
  return verdict.get
}
```



■ Evaluation function:

```
function latency(charstring idFirst, charstring idSecond, float
lowerbound, float upperbound) return verdicttype
{
  var myTimestampType t_a, t_b;
  var int i;
  for(i:=0; i<=100; i:=i+1)
  {
    timelog.search(myTimestampType:{*,idFirst}, i)->value t_a;
    timelog.search(myTimestampType:{*,idSecond}, i)->value t_b;
    if ((t_b.time-t_a.time<upperbound)
and (t_b.time-t_a.time>lowerbound))
      verdict.set(pass);
    else
      verdict.set(fail);
  }
  return verdict.get
}
```



- **Real-time extensions for MSC-2000:**
 - Nearly syntactic sugar
- **Real-time extension for TTCN-3:**
 - Easy to use
 - Few changes to the semantics/ changes are orthogonal to the existing semantics
 - Open issues:
 - ▼ Interrupt concept for online evaluation
 - ▼ Non-functional test verdicts
 - ▼ Distributed testing: clock synchronisation

