Part of proceedings of 23rd NMRG Meeting in Enschede, the Netherlands

University of Twente

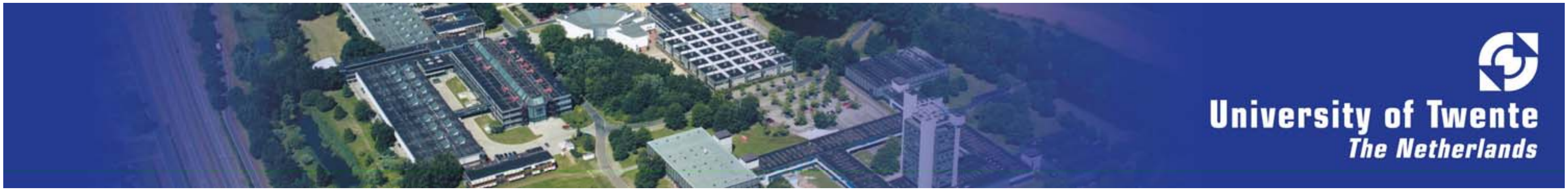
Periodicity of SNMP traffic

Gijs van den Broek

University of Twente
The Netherlands

Overview

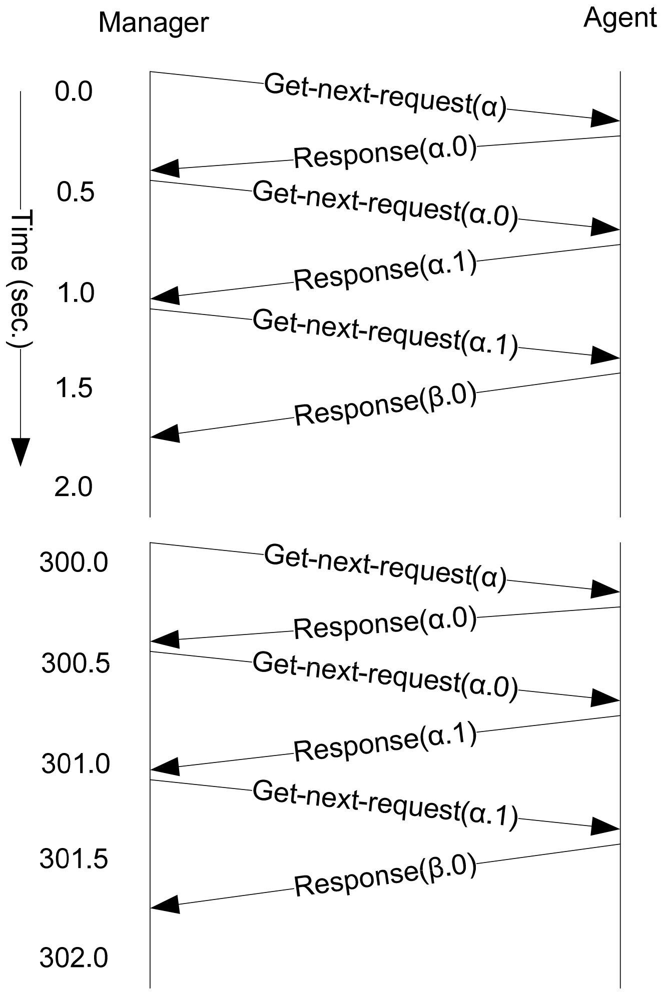
- Problem description
- General approach
- Brief algorithm description
- Results
- Conclusions



Problem description

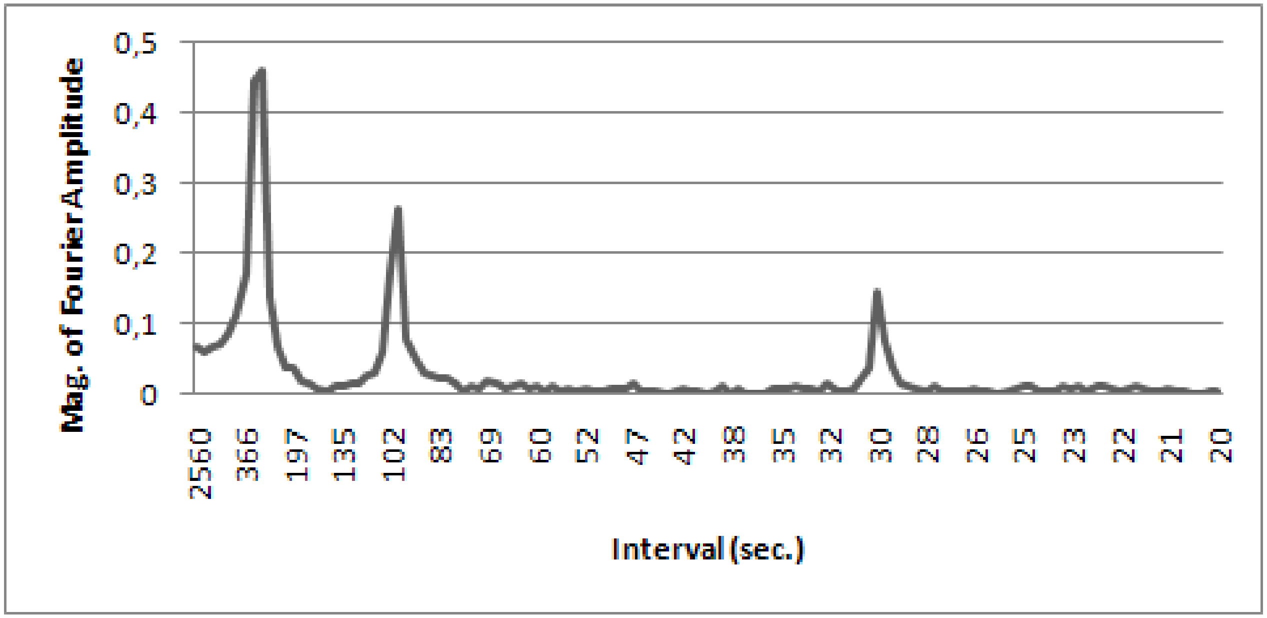
- SNMP traffic is widely in use
- Large SNMP trace files at our disposal
- How to perform research on just either (a)periodic SNMP traffic?
  - We need to separate periodic from aperiodic SNMP traffic first

Trace Example



- Which messages are periodic, which are aperiodic?

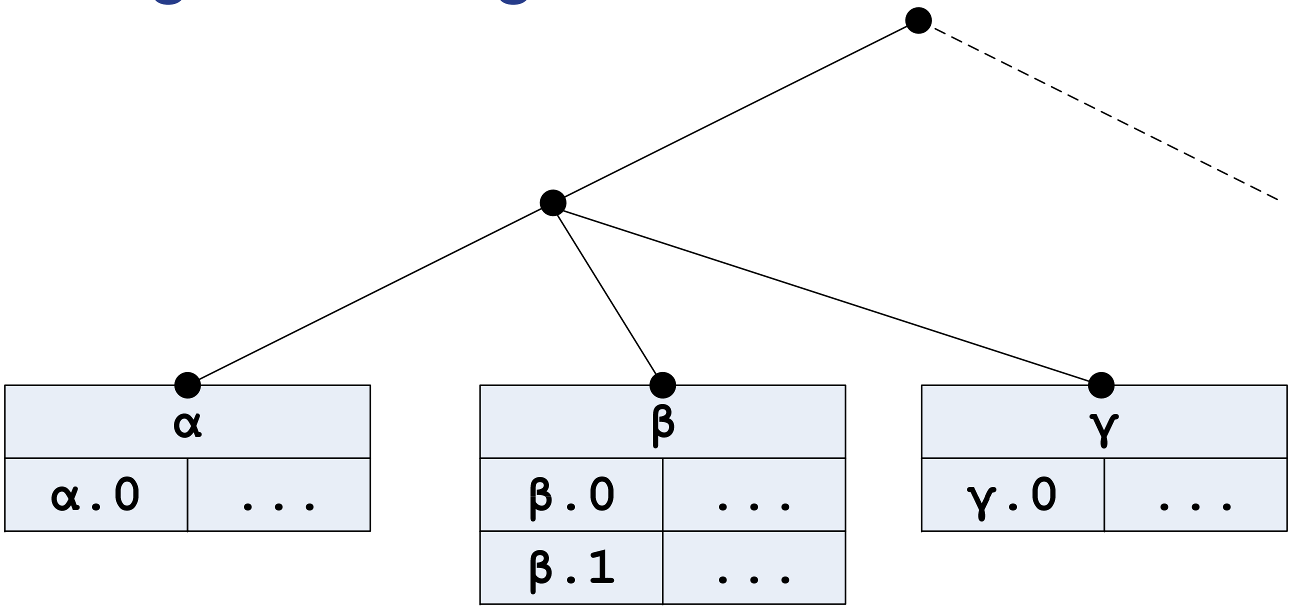
Approach 1: Fourier Analysis



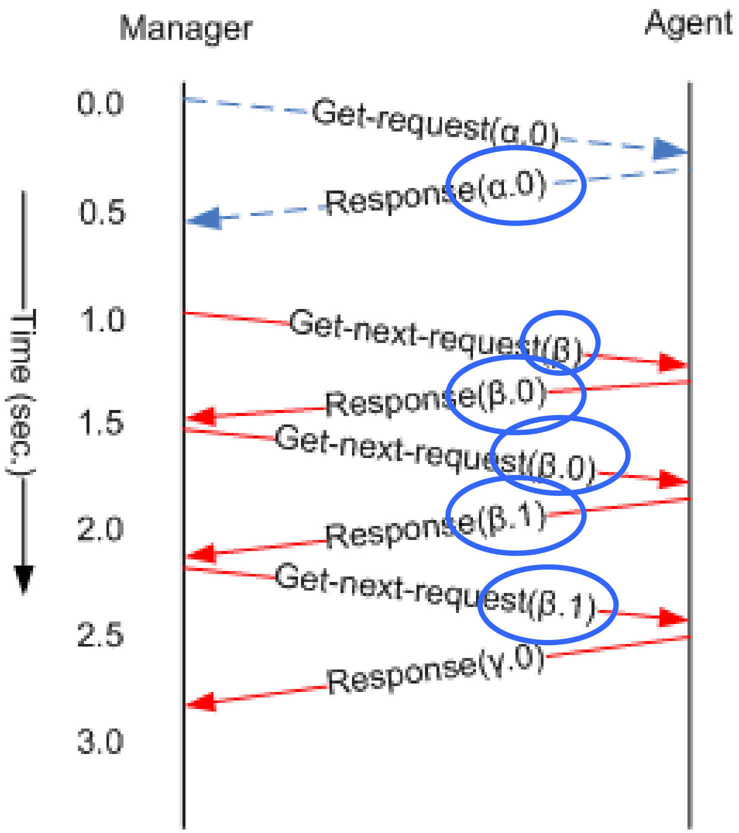
- Which messages are contributing to which peak?

Approach 2: Find patterns in related traffic

- Example scenario:
  - 1 manager set to poll 1 agent every x min.
  - 1 agent having a number of tables:



Approach 2: Find patterns in related traffic



Considerations

- Retransmissions
  - Referenced OIDs may be the same in different messages
- Table characteristics
  - May contain holes
  - Unequal column lengths
- OID insertion and position change
  - OID reference could be made unexpectedly halfway a table/column retrieval process
  - The order of OID references in a message may differ

Characteristics : Session (1)

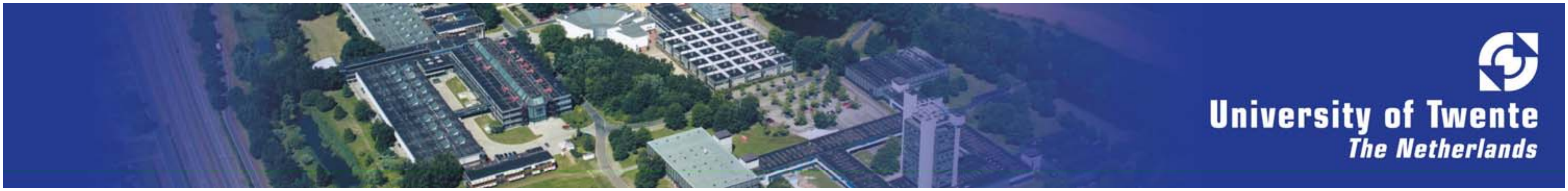
- One or more SNMP messages which are all **exchanged between exactly two network elements**, which are each identified through their IP address and respective port number.

Characteristics : Session (2)

- The messages other than responses in a session **all have the same operation type;**
- At least **one OID in every non-response is lexicographically the same** compared to an OID in chronologically last response to the previous non-response

Characteristics : Session (3)

- Every **non-response** must occur within a specifiable amount of time after the last listed response (if any) and must come from the same network element as the other non-responses of a session
- A **response** is considered part of a session if it occurred within a specifiable amount of time after an already listed non-response that has the exact same request ID

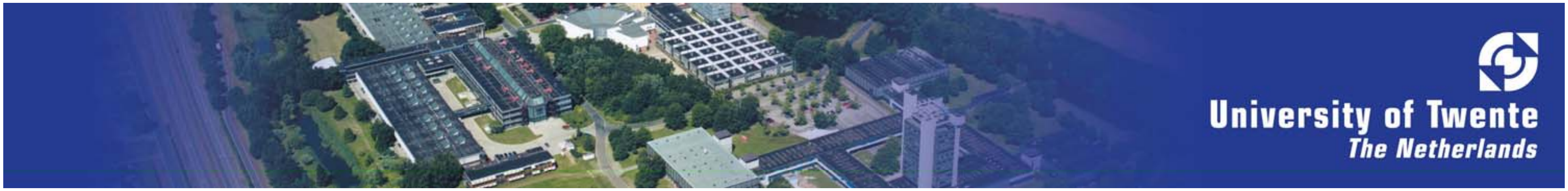


Characteristics : Session (4)

- Retransmission will be considered part of a session in the following cases:
  - **Response**: it contains a request ID that is equal to one of the already listed requests in that session and occurred within a specific amount of time after the respective original response message. The list of OIDs and the values may be different.

Characteristics : Session (5)

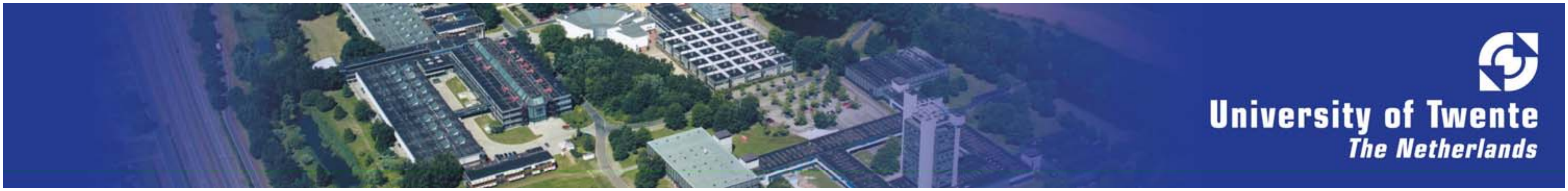
- **get, get-next, get-bulk, set, or inform request:** it contains a list of OIDs that is equal (though the order may be different) to one of the already listed requests in that session with the same list of OIDs. The request ID may be different**.**
- **trap or report**: it contains a list of OIDs that is equal (though the order may be different) to an already listed trap or report message respectively



Characteristics : Session [paper version]

A **session** is a set of one or more SNMP messages for which the following holds:

- Are all **exchanged between two hosts** and all non-response messages originate from just one of them;
- Are all of the **same operation type**, or are a response to a previous non-response of that session with a matching RequestID;
- At least **one OID in a non-response** message can also be found **in the last response message** to the previous non-response message (except for the first non-response message);
- Every message occurred **within a certain amount of time** after the previous message of that session.

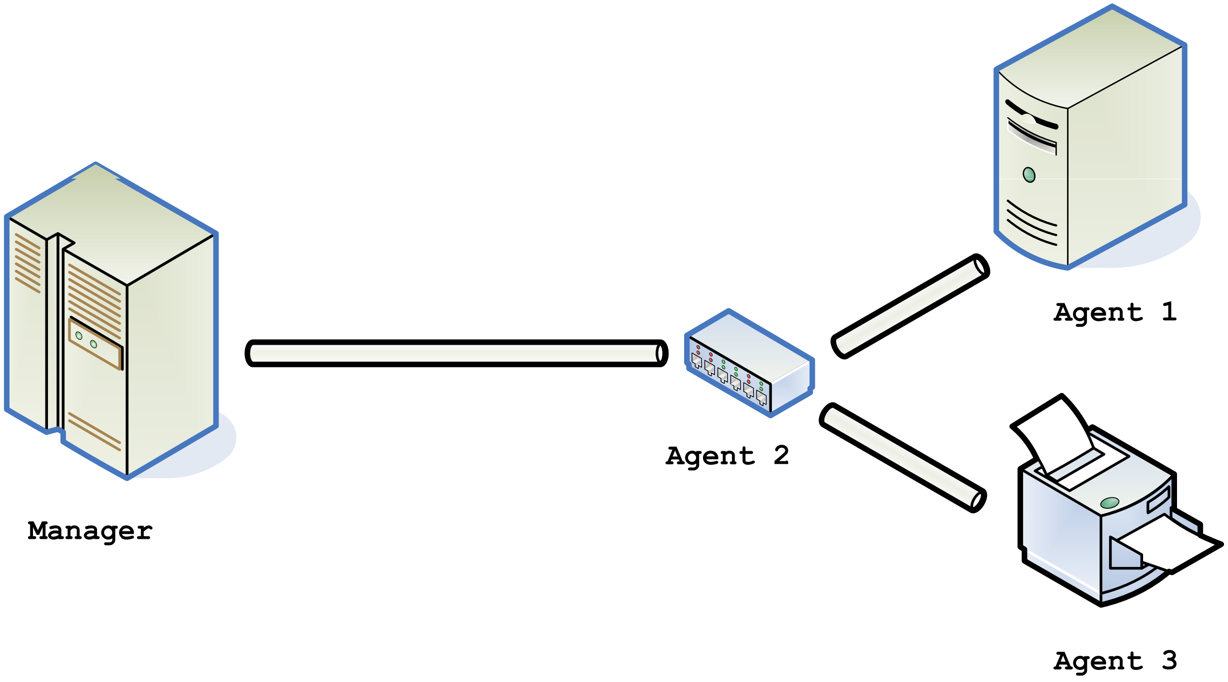


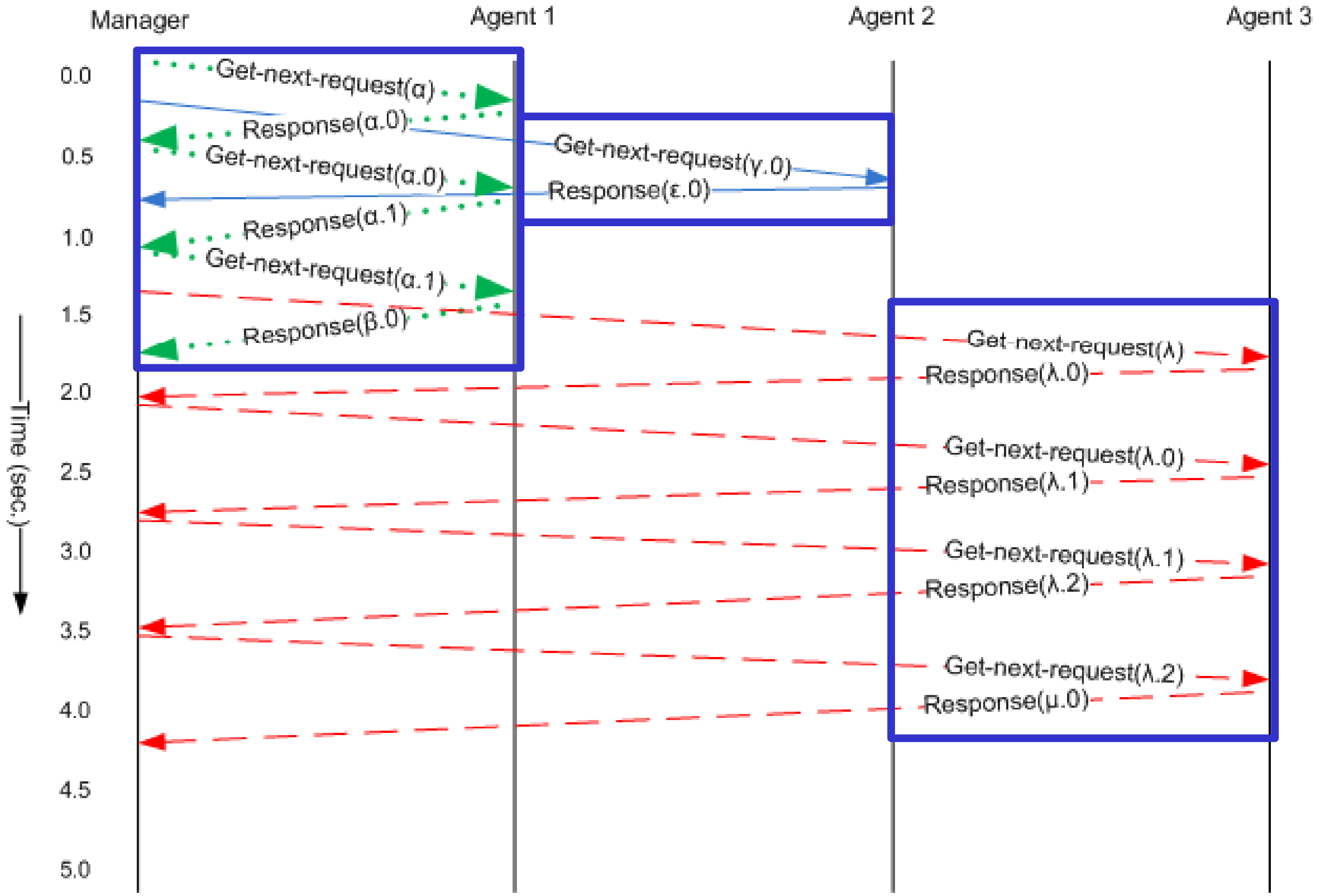
Problem description

- Sessions described so far only contain messages exchanged between two network elements
- Are there other (larger) relationships identifiable?

New scenario

- Consider a scenario of a single manager and multiple agents



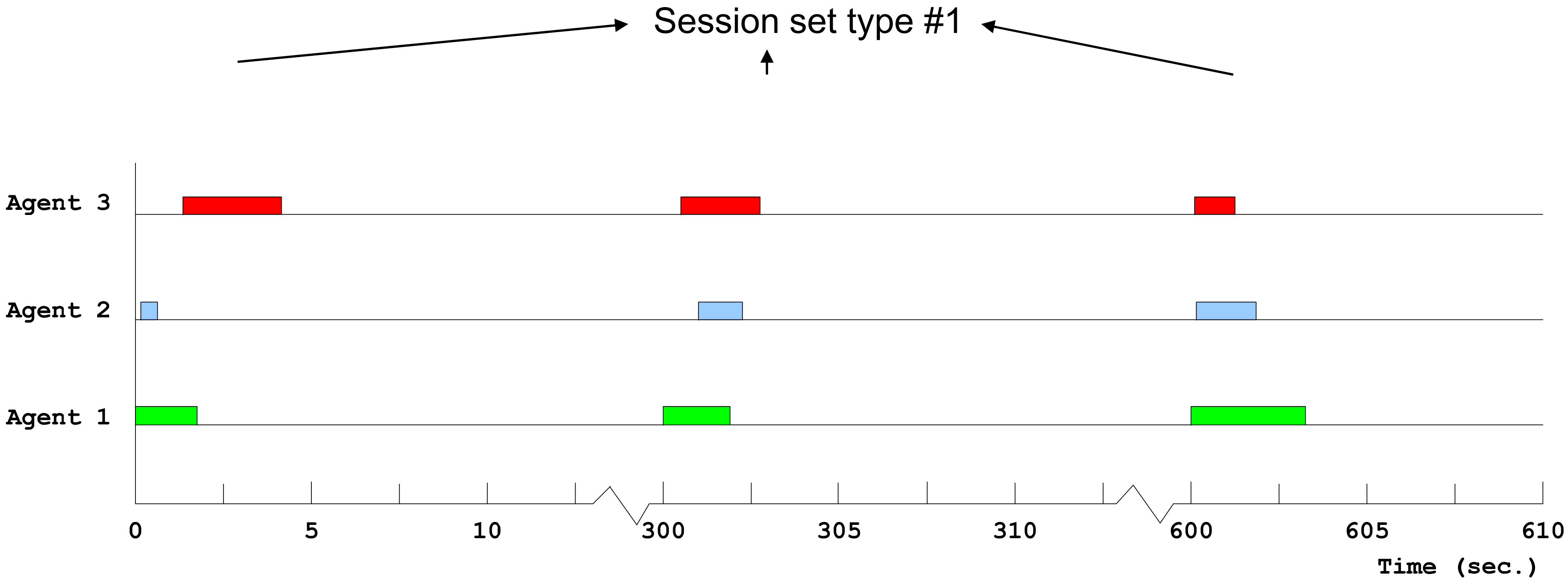

Manager
Agent 1
Agent 2
Agent 3
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Time (sec.)
Get-next-request(α)
Response(α.0)
Get-next-request(α.0)
Get-next-request(γ.0)
Response(ε.0)
Response(α.1)
Get-next-request(α.1)
Response(β.0)
Get-next-request(λ)
Response(λ.0)
Get-next-request(λ.0)
Response(λ.1)
Get-next-request(λ.1)
Response(λ.2)
Get-next-request(λ.2)
Response(μ.0)

Characteristics: Session Set

A **session set** encompasses one or more sessions that have the following characteristics:

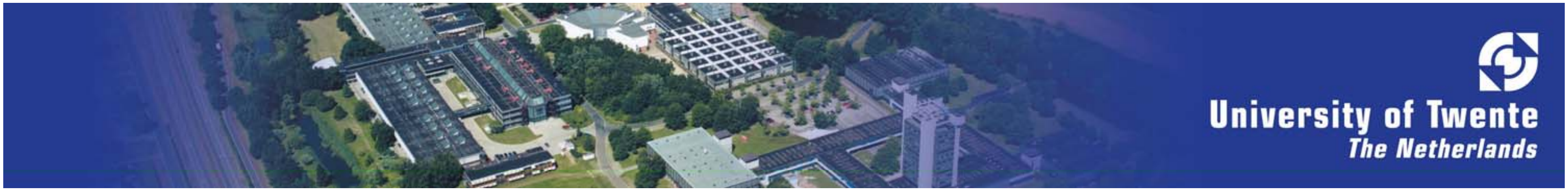
- Are all initiated by a single initiator;
- All occurred within an initiator specific time frame of each other;
- Are related in their occurrence to each other.

Session set type #1



Considerations

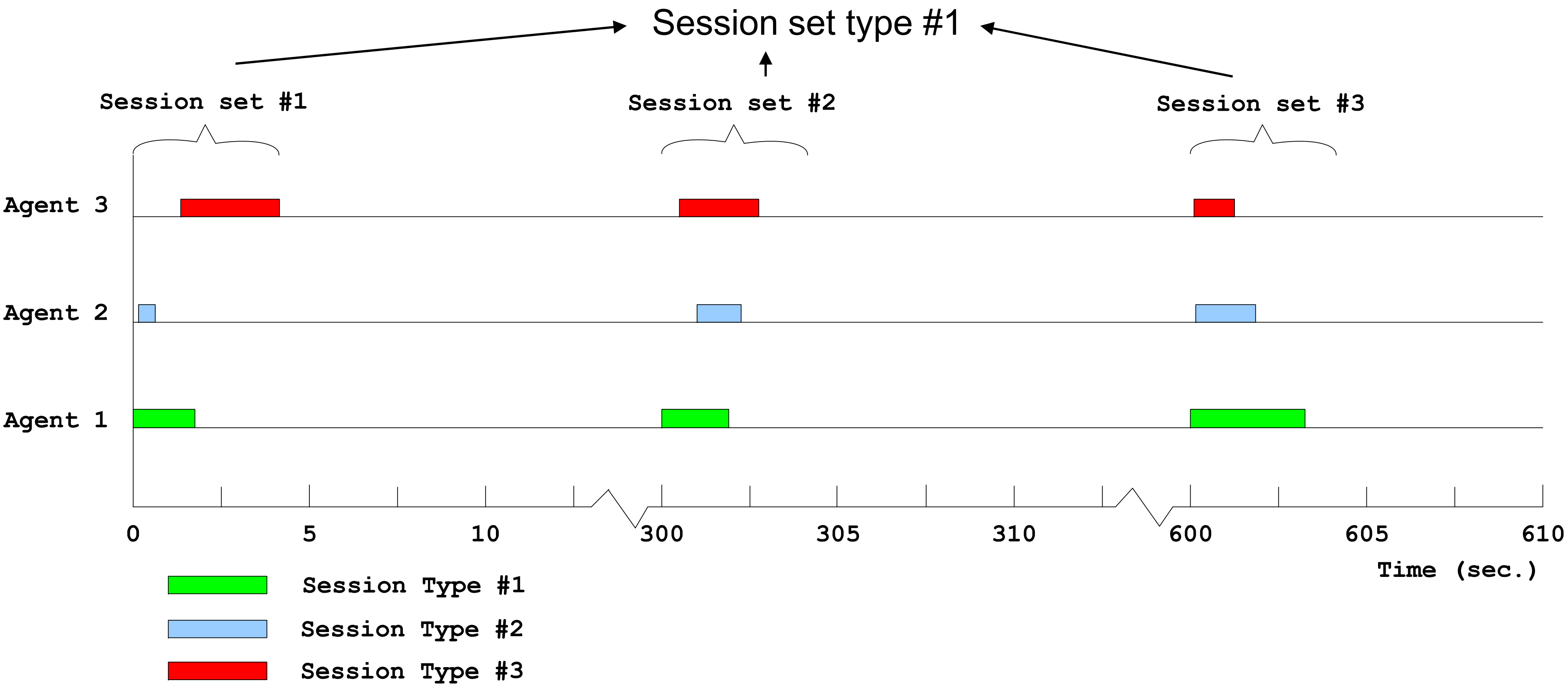
- Multiple initiators
  - All session sets of a session set type contain only sessions that were initiated by one single initiator
- Multiple managers operating from the same IP address
  - Session sets may contain multiple occurrences of a specific session type
- Irregularly occurring sessions
  - Certain sessions may not always occur in every session set

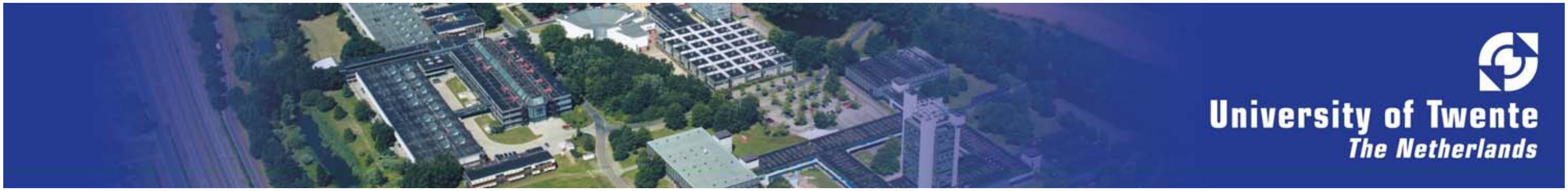


Characteristics: Session Set Type

A **session set type** involves one or more session sets which:

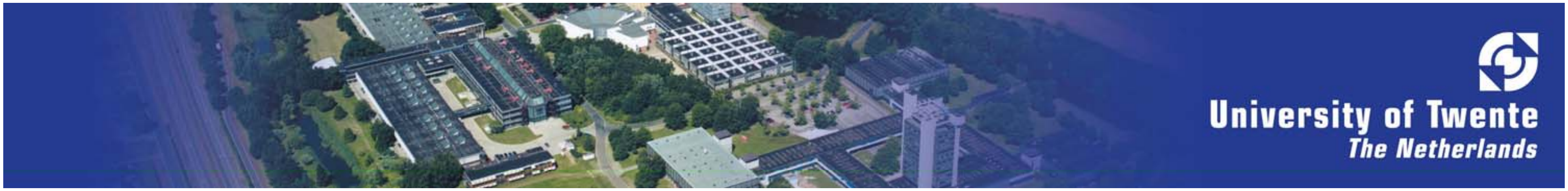
- Are all initiated by a single initiator;
- Are very similar to each other.


Session set type #1
Session set #1
Session set #2
Session set #3
Agent 3
Agent 2
Agent 1
0
5
10
300
305
310
600
605
610
Time (sec.)
Session Type #1
Session Type #2
Session Type #3



Considerations

- Different manager-agent relations
  - Different managers and/or agents may have different characteristics
- Different operations on the same table
  - Different operation types have different purposes
- Retransmissions within sessions
  - Should not influence the session type determination process

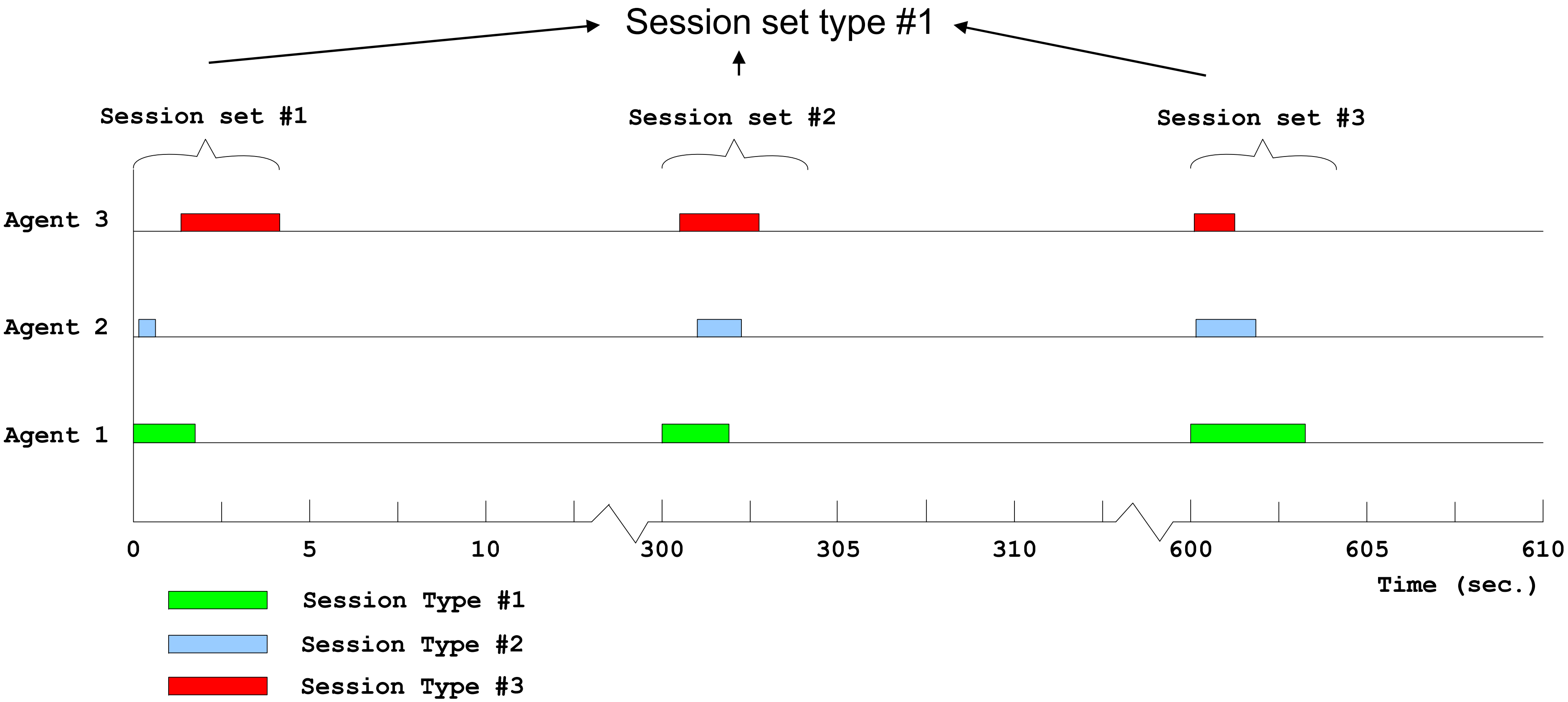


Characteristics: Session Type

A **session type** is a type mark that can be determined for every session. Furthermore, sessions with the same session type:

- Have occurred between the same two hosts;
- Are all initiated by the same single initiator;
- Have non-response messages that are of the same operation type;
- Have a common set of OID prefixes.

Complete Picture



Session set type #1

Session set #1

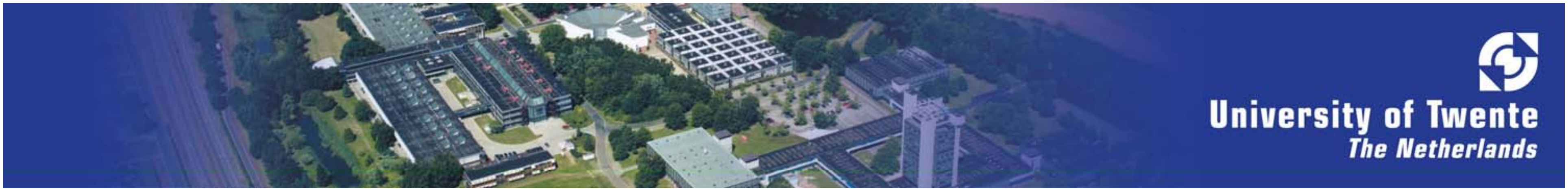
Session set #2

Session set #3

🟩 Session Type #1

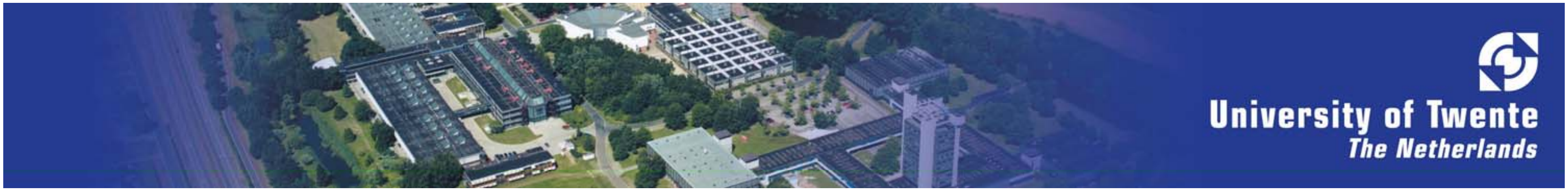
🟦 Session Type #2

🟥 Session Type #3



Traffic separation and interval detection

- Determine per session set type which session sets are (a)periodic;
- Determine intervals for periodic ones.



Considerations

- Timer related issues
  - Deviation in exact start time of periodic session sets
- Multiple intervals
- Trace holes
  - Interruption of trace recording
  - Initiator paused/changed settings

Algorithm steps

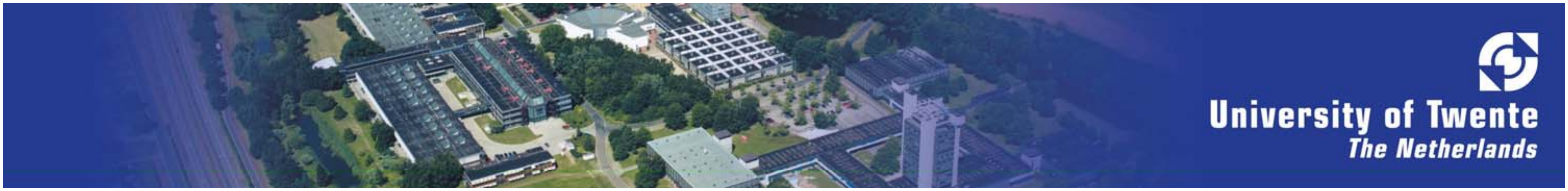
1. Find all sessions in an SNMP trace;
2. Determine session type for each session;
3. Find small sets of regularly occurring sessions, forming session sets. These session sets are of a specific session set type
4. Determine which session sets are (a)periodic (per session set type)

Time (sec.)	Source IP	Source Port	Dest. IP	Dest. Port	Operation type	Request ID	OIDs
0,35	A	1100	B	161	Get-next	1	**α**
0,49	B	161	A	1100	response	1	**α.0**
0,52	A	3852	C	161	Get-next	7	**γ**
0,55	A	1100	B	161	Get-next	2	**α.0**
0,61	C	161	A	3852	response	7	**γ.0**
0,70	A	3852	C	161	Get-next	8	**γ.0**
0,74	B	161	A	1100	response	2	**α.1**
0,78	C	161	A	3852	response	8	**ε.0**
0,83	A	1100	B	161	Get-next	3	**α.1**
0,88	B	161	A	1100	response	3	**β.0**

Session #1:
{0}1}3}6}8}9}

No sessions found yet

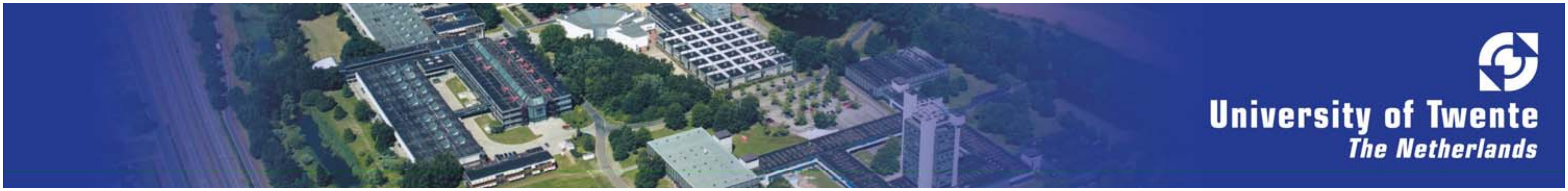
Session #2:
{2}4}5}7}



Operation type	Request ID	OIDs
Get-next-req.	1	α β
Response	1	α.0 β.0
Get-next-req.	2	α.0 β.0
Response	2	α.1 β.1
Get-next-req.	3	α.1 β.1
Response	3	β.0 γ.0

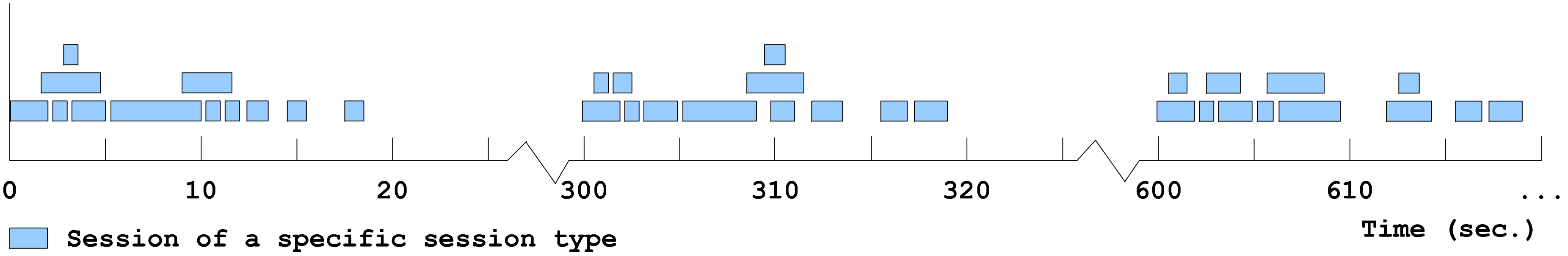
Found Referenced OIDs
α
β

Session type information	
Initiator IP Address	A
Initiator Port Number	1100
Other Party IP Address	B
Other Party Port Number	161
Operation Type	Get-next
Referenced OIDs	



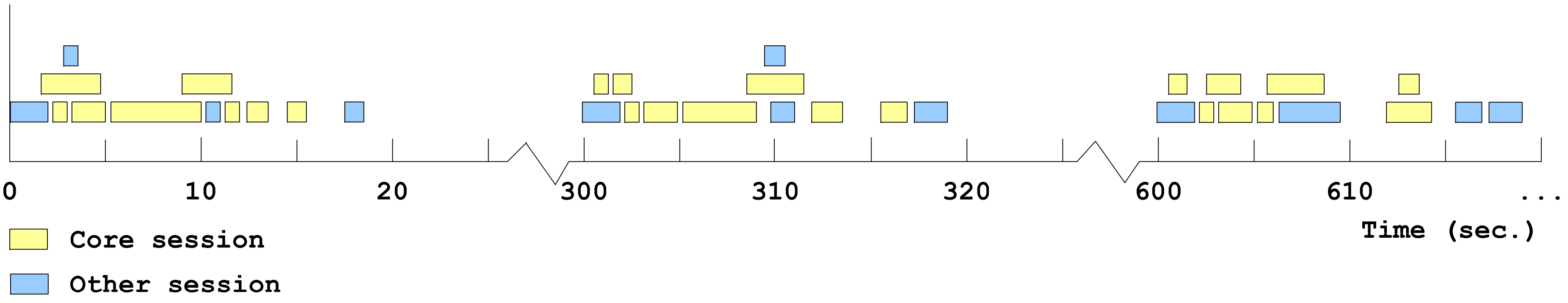
Finding related sessions

- Step 1: Split sessions in groups with equal initiator



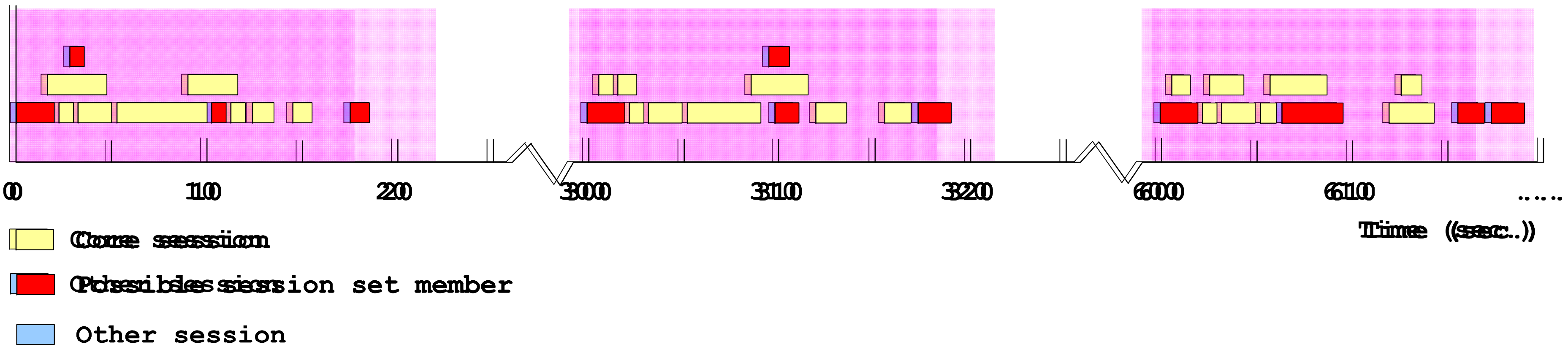
Algorithm description

- Step 2: Count session type occurrences
- Step 3: Find the largest group of session types that seem directly related (*core session type(s)*)



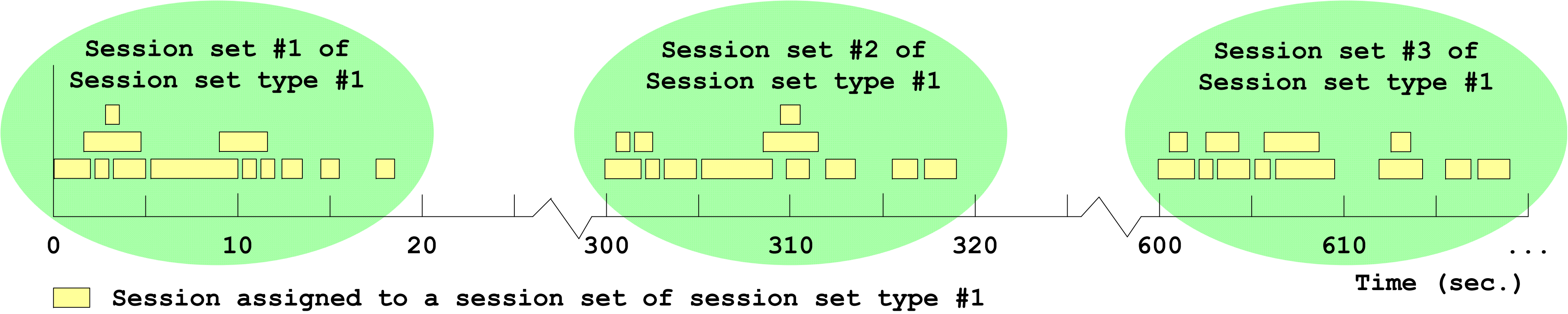
Algorithm description

- Step 4: Scan for other potential session set members



Algorithm description

- Step 5: Detect session types that occur often enough in these potential session sets
- Step 6: Determine session sets that will make up a single session set type



Session Set Start Time (sec.)	Last session start (sec.)	Last request (sec.)	Last response (sec.)	Number of sessions in session set	Session set
150,50	159,10	159,88	159,98	20	Session set #1
196,71	204,22	204,75	204,79	20	Session set #2
300,40					
449,93					
600,98	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
750,11	758,57	758,67	758,72	20	Session set #6

Interval	46,21 seconds
Next Lower Limit	232,92 seconds
Next Upper Limit	252,92 seconds

Session Set Start Time (sec.)	Last session start (sec.)	Last request (sec.)	Last response (sec.)	Number of sessions in session set	Session set
150,50	159,10	159,88	159,98	20	Session set #1
196,71	204,22	204,75	204,79	20	Session set #2
300,40	308,12	308,39	308,42	20	Session set #3
449,93					
600,98					
750,11					

Interval	149,90 seconds
Next Lower Limit	440,30 seconds
Next Upper Limit	460,30 seconds

Example results

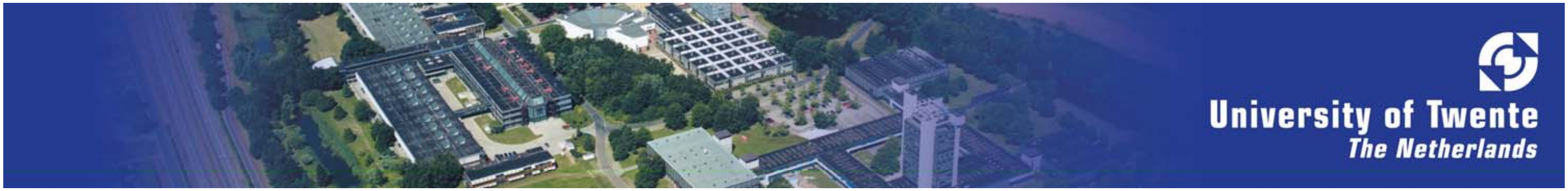
- Session sets numbers 1, 3, 4, 5 and 6 behave periodically with an estimated interval of ~150 seconds
- Session set #2 is considered aperiodic

Results

- L01t01
  - 100% of all session sets are aperiodic
- L03t02
  - >99% of all sessions are in periodically marked session sets (900 sec. interval)
  - <1% of all session sets are aperiodic
- L04t01
  - 92% of all sessions are in periodically marked session sets (300 sec. interval)
  - 8% of all sessions are in session sets that were marked ambiguous
- L05t01
  - >99% of all session sets are periodic (20 sec. interval)
  - <1% of all session sets are aperiodic

Conclusions

- This approach appears to be correct for a number of traces
- Still, there may be scenarios which require a different approach
  - Large gaps between sessions (>60 sec.) that appear to be part of the same session set



More information:

http://wwwhome.cs.utwente.nl/~broekjg/bsc

Q&A

-

Discussion