PRINCIPLES OF SELF-MODELING TOWARDS FAULT DIAGNOSIS FOR IMS NETWORKS AND SERVICES

Carole Hounkonnou (PhD), Eric Fabre

INRIA Rennes

INRIA evaluation seminar – March 22, 2012

Past experience in self-modeling for diagnosis

- Use-cases
  - recover fault propagations between (managed objects) of SDH networks
  - *idem* for Submarine Line Terminal Equipment

- Models+algorithms
  - interacting automata/Petri nets, handled as *dynamic systems* with concurrency
  - *self-modeling* = instantiate the (large) model from (few) generic component types
    effort on the design of generic comp. (expertise), instantiation by network discovery
  - model-based *distributed* event/alarm correlation *algorithms*, output = causality graph

Alcatel’s opinion: interesting for network auditing, but…

Outline

(ongoing work, not published)

1. Motivation
2. Structure of IMS networks and services
   - Main features: hierarchical, generic, and composable structures
   - Their architecture reveals dependencies between resources
3. Self-modeling as a support for fault localization
   - Model = pattern; Managed network = instance
   - Example
4. Reasoning with generic bayesian networks
5. Perspectives

Motivation

- Objectives
  - **joint fault management** for core network + access network(s) + IMS services over them
  - fault management = root cause, impact analysis, towards mitigation/healing suggestions

- Difficulties
  - **large scale** and complex system,
  - network structure **evolving** in time
  - learning approaches are inefficient: limited scope, not enough (failure) data

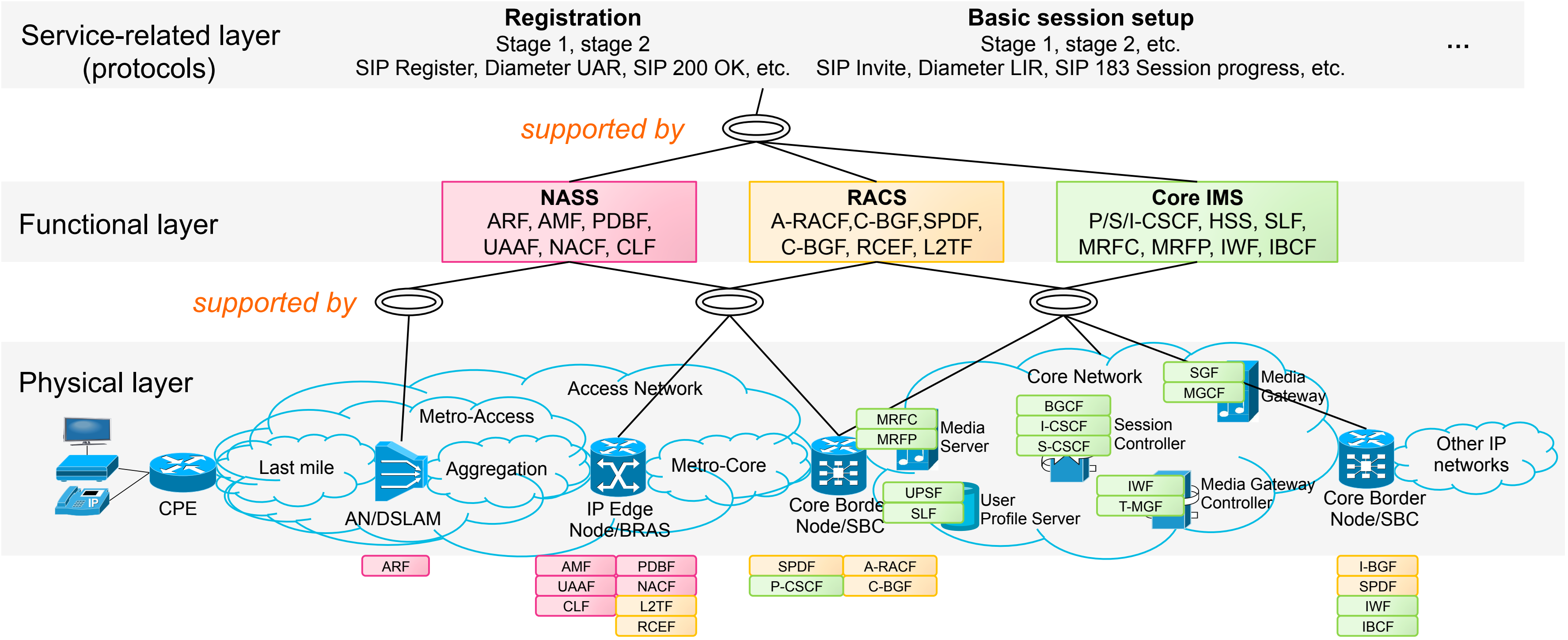
- Our approach
  - **model-based correlation**: fault location based on a model of resources dependencies
  - **self-modeling**: generic (cross-layer) patterns, instantiated on the fly
  - (distributed) **probabilistic reasoning**, on a dynamically updated model

Outline

1. Motivation
2. Structure of IMS networks and services
   - Main features: hierarchical, generic, and composable structures
   - Their architecture reveals dependencies between resources
3. Self-modeling as a support for fault localization
   - Model = pattern; Managed network = instance
   - Example
4. Reasoning with generic bayesian networks
5. Perspectives

Structure of IMS networks and services

- IMS networks and services are **hierachically** organized structures

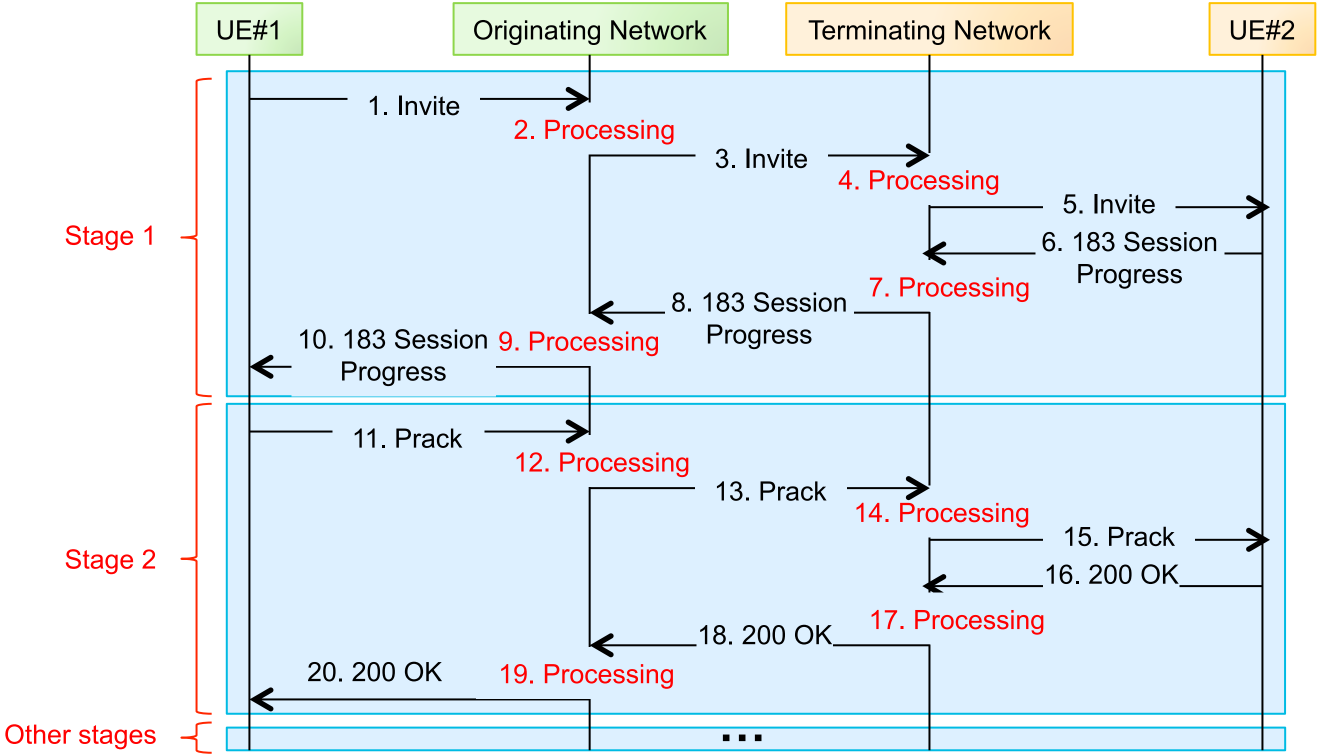


Structure of IMS networks and services

- Hierarchy in the service/protocol layer:
  - In time: procedures decompose as succession of stages (request/answer)
  - In space: each request/answer decomposes as a succession of operations of different granularities
- Dependency relations between « resources »:
  - « **supported by** » type of dependency between the service layer and the functional layer
  - Within the service layer: dependencies take the form of ***precedence***
    a phase or request must be completed successfully before another one starts

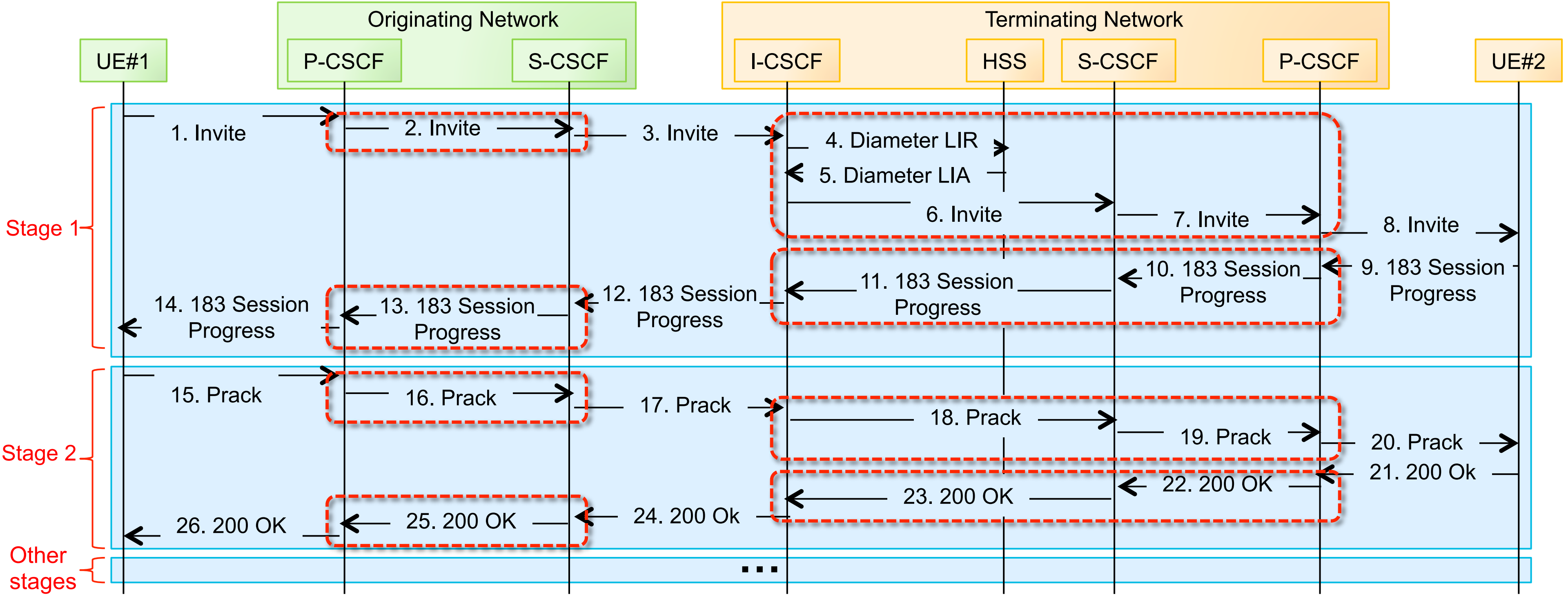
Structure of IMS networks and services

- Example: A basic session setup between two UEs



Structure of IMS networks and services

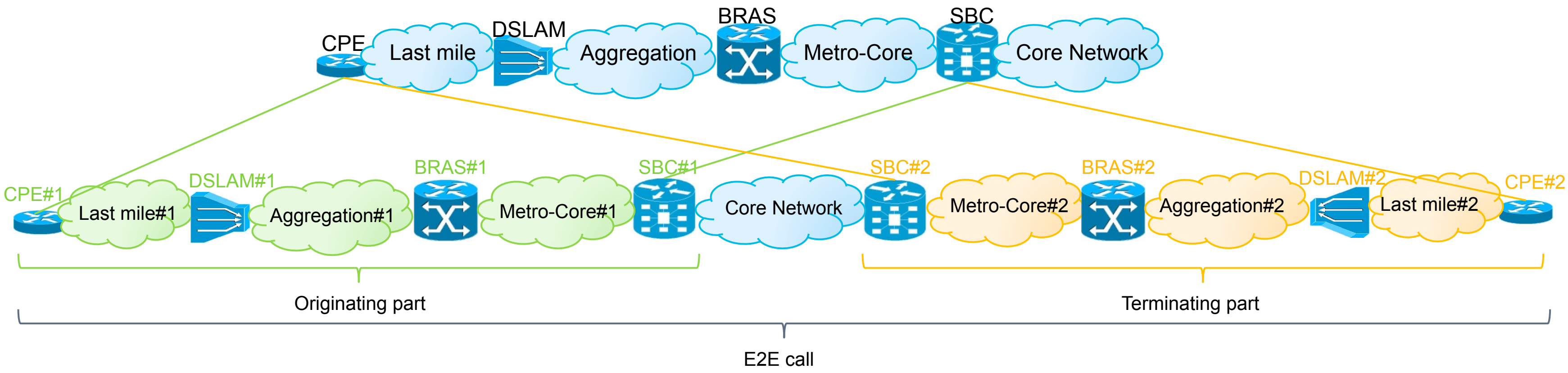
- Example: A basic session setup between two UEs



Structure of IMS networks and services

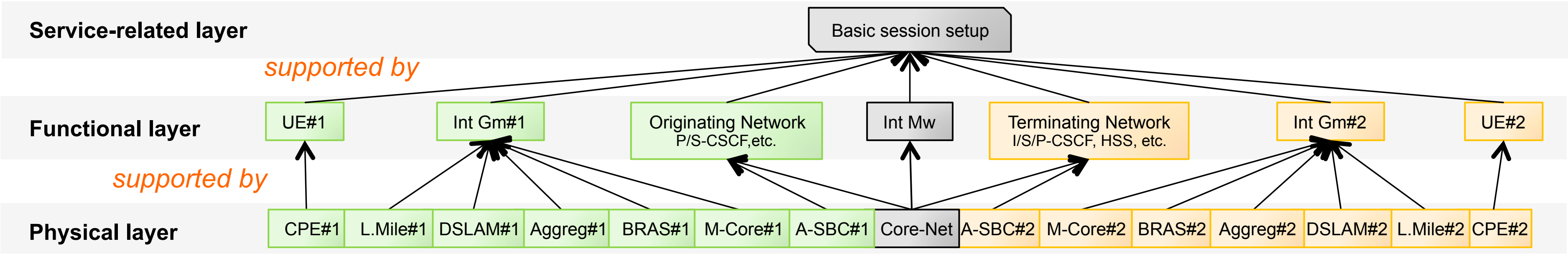
- Model segments are **composable**, i.e. can be assembled into larger structures

- Example: end-2-end call
  - Single granularity level of the hierarchy,
  - physical layer view.



Structure of IMS networks and services

- Example: end-2-end call (continued)
  - Elements of the 3 layers : session setup on top (service layer)
  - Single granularity represented within each layer
  - Inter-layer resources dependencies are displayed



Structure of IMS networks and services

Summary

- 3 (or 4) layers:
  - Physical / functional / service (protocols) / capabilities (high-level network properties)
- Features
  - hierarchical structures,
  - generic patterns, derived from standards and common practices
  - composable patterns
- Relevance for diagnosis (and impact analysis)
  - evidences the resources involved in some high level service
  - displays the (statistical) dependencies of resources: "*supported by,*" "*precedes*"
  - identifies observable elements: state of a component, or of a procedure)
  - may also include extra descriptions of malfunctions
    (failed requests, propagations of failures, effects of incorrect states…)

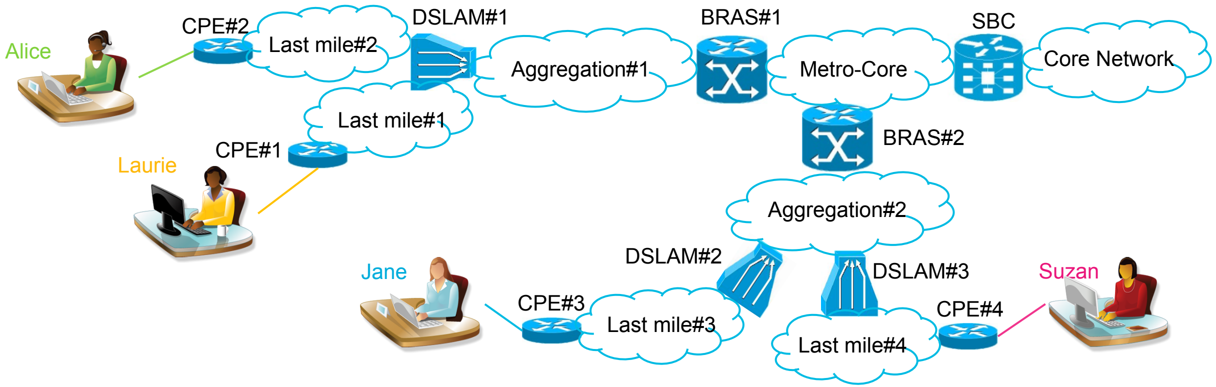
Outline

1. Motivation
2. Structure of IMS networks and services
   - Main features: hierarchical, generic, and composable structures
   - Their architecture reveals dependencies between resources
3. Self-modeling as a support for fault localization
   - Model = pattern; Managed network = instance
   - Example
4. Reasoning with generic bayesian networks
5. Conclusion

Working with a generic model

- Patterns:
  - A "generic" model defines typical structures, or ***patterns***
  - Example: connection of a user to the internet, physical layer, single granularity

- Instance:
  - In reality, one has to work on a network ***instance***
  - The patterns are instantiated many times, with overlapping parts

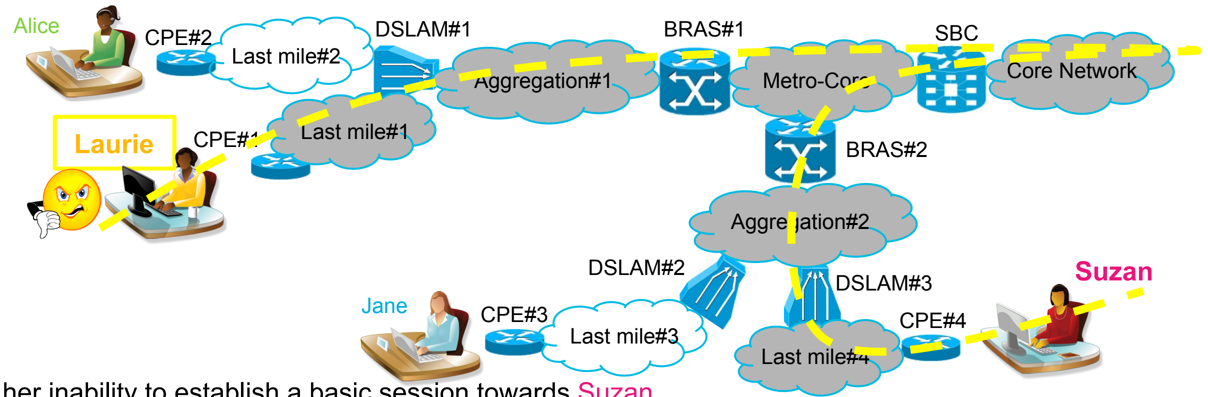


Solving a diagnosis problem

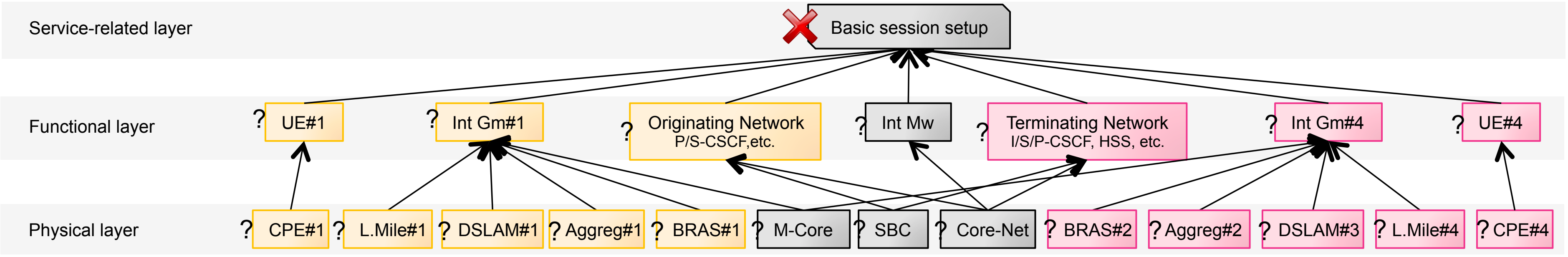
- Start point = request to explain some observed malfunction

- Method:
  1. Find/assemble the pattern that describes the resources used by this function (at some granularity level),
  2. Objective = identify the faulty resource in this pattern
  3. Locate the instance of this pattern in the model
  4. Collect observations within the instance to locate the faulty resource
  5. If not sufficient, explore other pattern instances with common resources, and collect their observations, until confidence in the explanation becomes sufficient

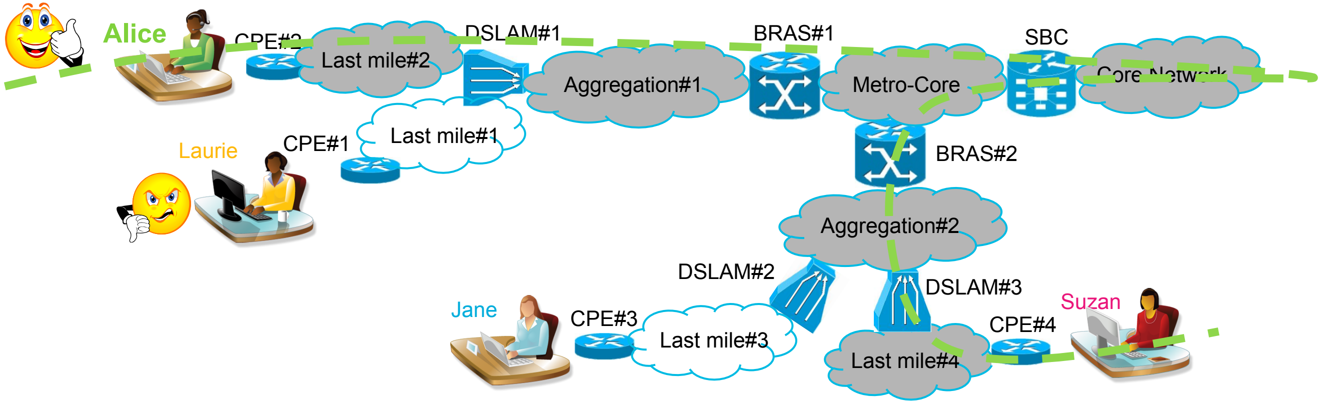
Example: explain Laurie's unability to call Suzan



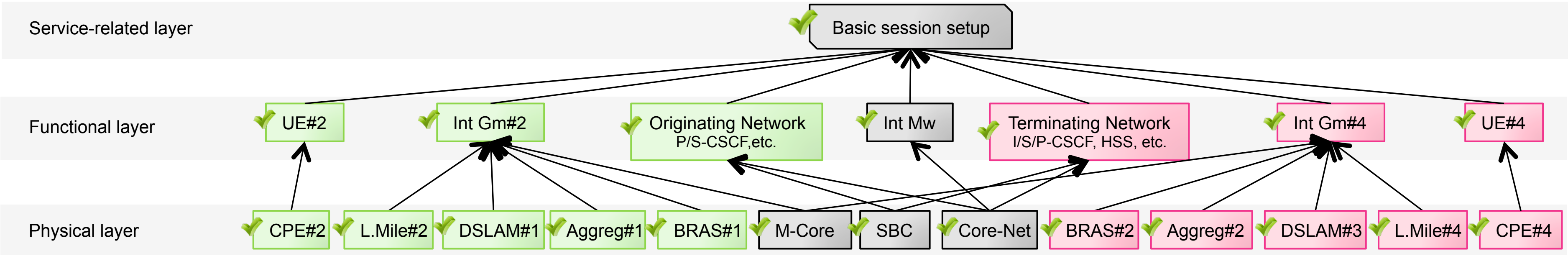
- Laurie complains about her inability to establish a basic session towards Suzan
- The generic dependency model corresponding to a basic session setup is retrieved and instantiated for a basic session setup from Laurie to Suzan
- In our example, the variable « Basic session setup » is observable and its state is « down » for Laurie
- The dependency model instance contains all the variables that can possibly explain Laurie's failure



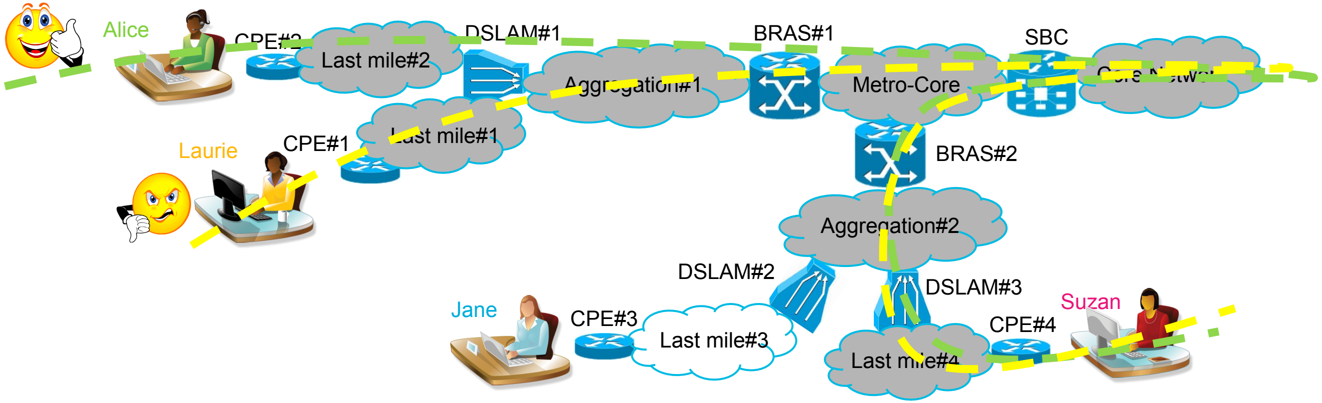
Example: explain Laurie's unability to call Suzan



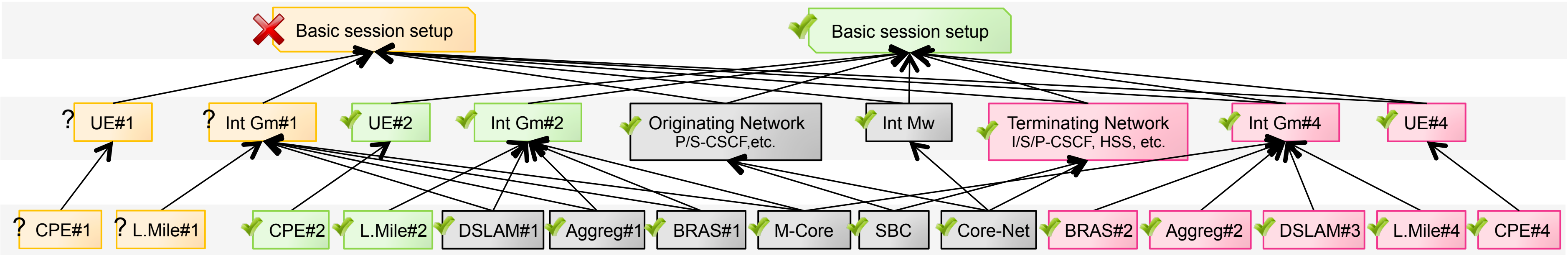
- Some of the possible causes for Laurie's failure are resources that are also shared with Alice
- So, the state of a session setup from Alice to Suzan can give us information about the actual state (« up » or « down ») of these resources
- Suppose that Alice can successfully establish a basic session with Suzan, the corresponding dependency model instance is given below



Example: explain Laurie’s unability to call Suzan



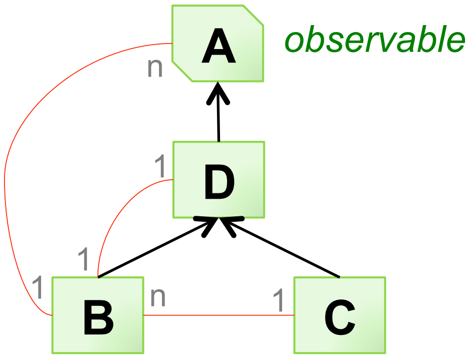
- Merging Alice’s and Laurie’s dependency model instances reduces the set of possible explanations for Laurie’s failure since it eliminates non faulty resources

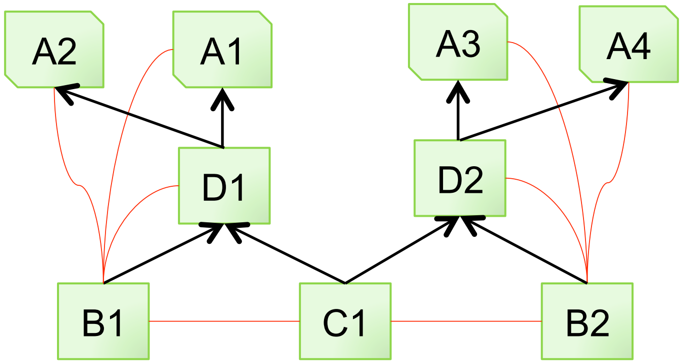


Outline

1. Motivation
2. Structure of IMS networks and services
   - Main features: hierarchical, generic, and composable structures
   - Their architecture reveals dependencies between resources
3. Self-modeling as a support for fault localization
   - Model = pattern; Managed network = instance
   - Example
4. Reasoning with generic bayesian networks
5. Conclusion

Generic Bayesian Network





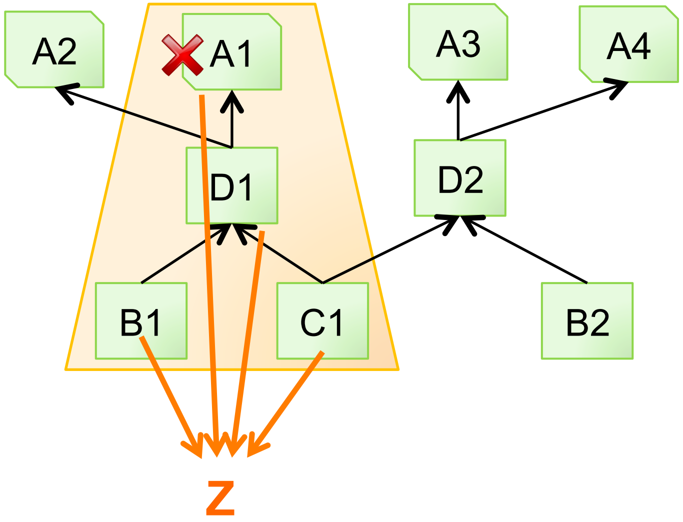
Patterns (≈ class diagrams)

- state variables of resources
- topological information "connected to"
- statistical dependency relations: conditional law of failure propagations
- individual fault probabilities
- defines variables that are observable or testable (possibly with cost)

Instance

- the full Bayesian network, ***possibly huge !***
- contains copies of pattern instances (instances of generic IMS resources)
- interaction by shared variables

Inference in a GBN



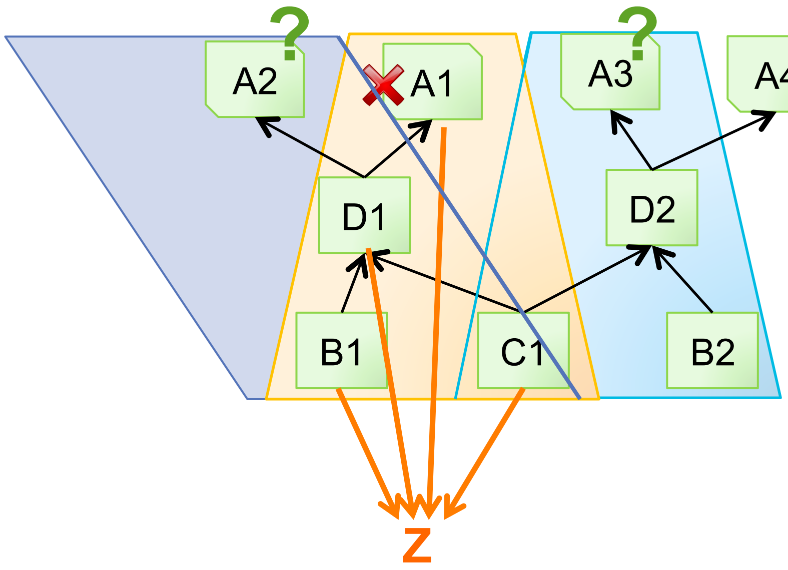
P(Z | A1,B1,C1,D1)

Method:

1. failure signaled : *A1 down*
2. find the pattern(s) identifying all its resources: the cause is there
3. build the locator variable Z: points towards the guilty resource (excludes multiple failures) Here Z can take 4 values, *A1,B1,C1* or *D1*
4. compute P(Z | A1=down, other observations), by standard BN inference, and decide on the most likely cause…

… if the entropy H(Z | observations) is small enough to decide

Inference in a GBN (cont.)



Model scope extension:

1. check for neighbour pattern instances that contain observable variables
2. select the most promising one:
H(Z|A1=down,A2) % H(Z|A1=down,A3)
standard BN inference
3. Go get observation/test value A2 (for ex.), compute H(Z|A1=down,A2=a2)
4. Either decide with Z on the most likely cause, if entropy small enough,
or proceed to a new extension

Outline

1. Structure of IMS networks and services
   - Main features: hierarchical, generic, and composable structures
   - Their architecture reveals dependencies between resources
2. Self-modeling as a support for fault localization
   - Model = pattern; Managed network = instance
   - Example
3. Reasoning with generic bayesian networks
4. Conclusion

Conclusion

Next step:

- define a relevant set of “patterns” + fault benchmarks for IMS networks
- formalize generic Bayesian networks and their algorithms
- implement and test

Following step:

- define ***hierarchical*** (generic) Bayesian networks
- inference should also be able to decide the *refinement* of the model instance, i.e. to get a finer view

Later:

- towards failure impact analysis