## Resource collision avoidance

* To mitigate the problem of resource collision, two modifications are proposed:
  + A new reserveQuery message
  + Additional hint information in the reserveFail message
  + A common structure is returned for both cases
* resourceQuery: new operation that has the responses: resourceQueryConfirmed and Error message
* Behaviour of reserve :
  + A reservation will succeed if resources are available within the constraint space specified in the reserveRequest message, no constraint space will be returned in the reserveConfirmed message, but the actual reserved resources are returned.
  + If a reservation fails, then the associated uPA will return in the reserveFail message, an alternative set of parameters based on the available resources in an **expanded or alternate** constraint space.
* Behaviour of the resourceQuery:
  + A constraint space needs to be specified in a resourceQuery message.
  + The associated uPA will return in a resourceQueryConfirmed message a set of parameters **within** the constraint space specified in the resourceQuery message.
* Definition of constraint space
  + Current service definition for a reservation is a 5-tuple: includes ingressSTP, egressSTP, capacity, startTime (optional – not included means now), endTime (optional – not included means forever).
  + The constraint space is built as a 3-dimensional cube i.e ingressSTP, egressSTP and capacity that is bounded within a specific start and end time.
  + Multiple constraint spaces can be returned. Constraint spaces can overlap each other.
* Behaviour of aggregator
  + The aggregator will return a constraint space that is the intersection of all child constraint spaces.
* AP: draft an NSI GFD requirement document describing the new NSI CS messages. This will include the schema for the constraint space for the NSI CS, for the EVTS service definition