draft-ietf-core-dynlink

IETF 103

Recent

- Reference implementation for the conditional observe attributes – C/C++
- Some learning about the interactions between attributes
- Logic based expression using interval time bounds
- Learning from developing the OCF version

Definitions for notification conditions

```
//notifiable.c
bool notifiable( Resource * r ) {

#define BAND r->band
#define SCALAR_TYPE ( num_type == r->type )
#define STRING_TYPE ( str_type == r->type )
#define BOOLEAN_TYPE ( bool_type == r->type )
#define PMIN_EX ( r->last_sample_time - r->last_rep_time >= r->pmin )
#define PMAX_EX ( r->last_sample_time - r->last_rep_time > r->pmax )
#define LT_EX ( r->v < r->lt ^ r->last_rep_v < r->lt )
#define GT_EX ( r->v > r->gt ^ r->last_rep_v > r->gt )
#define ST_EX ( abs( r->v - r->last_rep_v ) >= st )
#define IN_BAND ( ( r->gt <= r->v && r->v <= r->lt ) ||
                  ( r->v >= r->gt && r->gt >= r->lt ) ||
                  ( r->v <= r->lt && r->lt <= r->gt ) )
#define VB_CHANGE ( r->vb != r->last_rep_vb )
#define VS_CHANGE ( r->vs != r->last_rep_vs )
```

Logic expression

```
  return (
    PMIN_EX &&
    ( SCALAR_TYPE ?
      ( ( !BAND && ( GT_EX || LT_EX || ST_EX || PMAX_EX ) ) ||
        ( BAND && IN_BAND && ( ST_EX || PMAX_EX) ) )
    : STRING_TYPE ?
      ( VS_CHANGE || PMAX_EX )
    : BOOLEAN_TYPE ?
      ( VB_CHANGE || PMAX_EX )
    : false )
  );
}
```

Next

- Add a state diagram for the interactions between attributes
- Incorporate feedback received
- Provide observe attributes as query parameters to the observe request
- Restructure the draft; introduce observe attributes first, then dynamic links, then binding table implementation
- Add implementation notes about link state tracking
- Implementation may reuse observers and updates