

UCX Request API

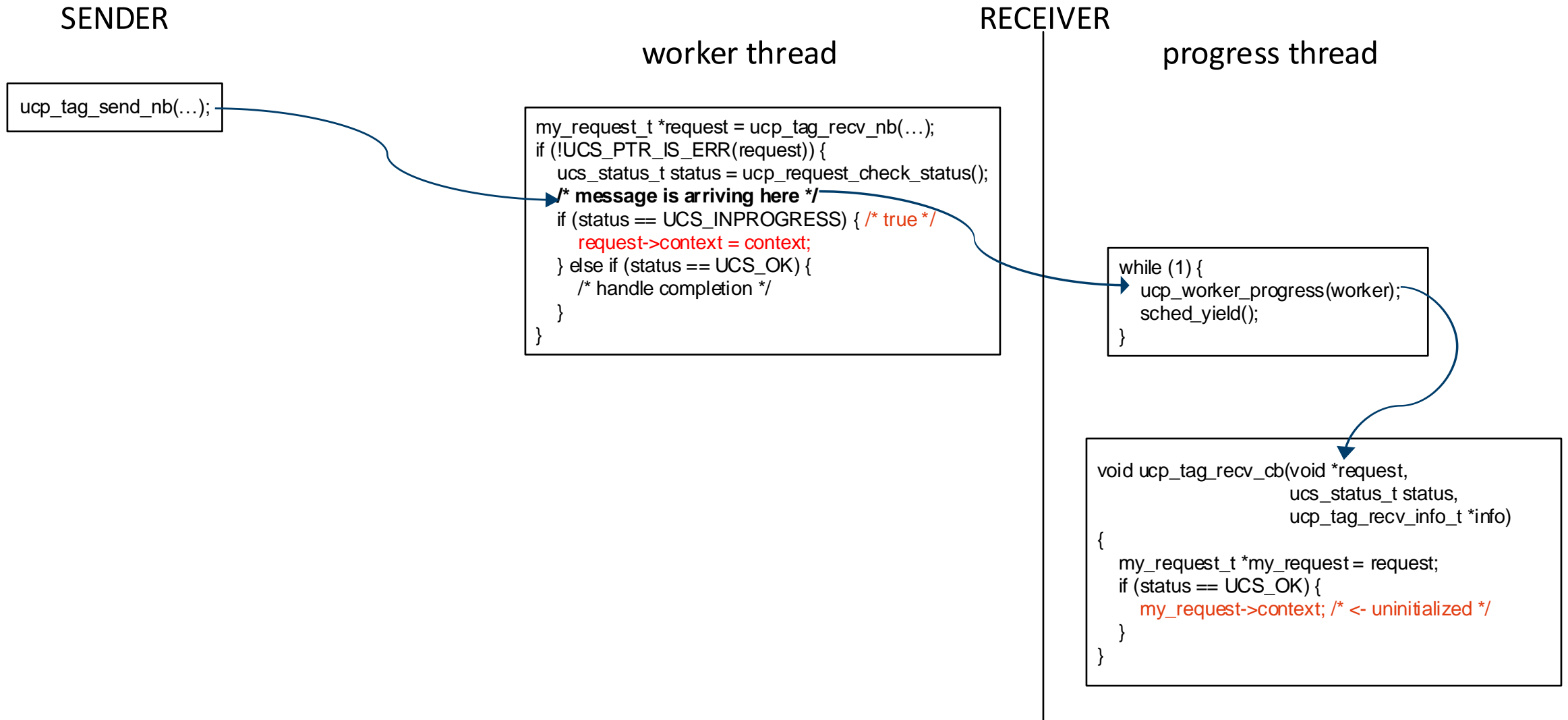


UCX TAG API

	Completion callback	External request
ucp_tag_send_nb	X	
ucp_tag_send_nbr		X
ucp_tag_send_sync_nb	X	
ucp_tag_recv_nb	X	
ucp_tag_recv_nbr		X

- Send and recv APIs are not consistent (immediate completion, callback invocation)
- Not possible to initialize request with *_nb APIs
- Can use _nbr APIs, but:
 - they do not have completion callback
 - fast allocator (memory pool/cache) is needed (or synchronous send/recv model)

MT problem



API Change (option 1): extend *_nbr APIs

- Define `ucs_status_ptr_t *ucp_request_alloc(const ucp_request_params_t *params)`
- Or just `ucs_status_ptr_t *ucp_request_alloc(void *completion_cb)`

```
my_request_t *request = ucx_request_alloc(my_comp_cb);  
if (request != NULL) {  
    request->my_preinitialized_field = ...; // Init before calling receive  
    ucp_tag_recv_nbr(..., request);  
}  
  
typedef struct {  
    uint64_t field_mask;  
    void *completion_cb;  
    uint16_t flags;  
} ucp_request_params_t;
```

- Use this request with *_nbr APIs
- Pros:
 - Can use callbacks with *_nbr API
 - Minimal API change
- Cons:
 - Extra UCX call
 - Applications may have their own allocator (can define `ucp_request_cb_set()` though)

API Change (option 2): new API calls

- Several options possible, e. g.:

```
ucs_status_ptr_t ucpv2_tag_recv_nb(ucp_worker_h worker, void *buffer, size_t count,  
                                   ucp_datatype_t datatype, ucp_tag_t tag, ucp_tag_t  
                                   tag_mask, ucp_request_t *request);
```

```
ucs_status_ptr_t ucpv2_tag_send_nb(ucp_ep_h ep, const void *buffer, size_t count,  
                                   ucp_datatype_t datatype, ucp_tag_t tag,  
                                   ucp_request_t *request);
```

- Pros:

- No extra UCX calls
- Generic, can be extended in the future

```
typedef struct {  
    uint64_t flags;  
    void *completion_cb;  
    void *context;  
} ucp_request_t;
```

- Cons:

- Another API for already existing functionality
- Extra overhead

```
typedef struct {  
    /* internal UCP fields */  
    ucp_request_t  
    user_space_request;  
} ucp_request_impl_t;
```