|  |  |  |  |
| --- | --- | --- | --- |
| **TC MC Multiplexing** | **Creator = CSS Area** | **Editor = John** | **Reviewer = ??** |
| **Status: 191024** | | | |

| **Issue Short Title** | **Issue Description/Discussion** | **Source** | **Status** | **O/C** |
| --- | --- | --- | --- | --- |
| **add resource status** |  | **correspondence between WH and JP** | **Agreed and done** | **C** |
| **add "fwd" prefix** | **currently just "TC MC Mux" in registry** |  | **Agreed and done** | **C** |
| **support for Unified SLP?** | **Can this support USLP (possibly with some tweak\_ or do we need a separate Unified version of the FR? Tentative approach - treat as a separate FR/FR Set**  **WH: I suggest that we first develop USLP FRs and then check if we can merge them with the others and if such merging would be beneficial, e.g. because at least semantically most parameters are identical**  **190318 JP – I have in fact put placeholder FRs in for the Forward USLP FRs the FR Ref Model Tech Note, and have fleshed out the configuration parameters for the MC Mux and VC Mux FRs. The confusing thing to me is what gets into the “top” of the Fwd USLP stack: will SLE Space Packet be extended to support packets destined for USLP variable-length frames? What about packets destined for USLP fixed-length frames? Or is this the time to consider a more-general forward Space Packet service based on CSTS?**  **Alternatively, we might be able to change the “TC” and “AOS” designations to “VLF” (Variable Length Frame) and “FLF” (Fixed Length Frame) on the existing FR Sets and let them handle USLP. I have not looked into this.**  **190415 WH – I like the latter idea, but I have not looked into it to a reasonable level of detail. From a more practical perspective I doubt that we would be able to get the funding required to develop a generalized Forward Space Packet service suitable to operate over TC AOS and USLP type of links.**  **190722 WH – As discussed and agreed both by the CSTS and the SM WGs we shall keep the FRs supporting the USLP space link protocol separate from the TC FRs.** | **JP** | **Agreed not to handle USLP and other space link protocols by means of common FR types.** | **C** |
| **configures and reports** | **The descriptions of parameters for which Configured = true should begin "This parameter configures and reports"** | **correspondence between WH and JP** | **Agreed and done** | **C** |
| **Physical Channel Name?** | **TC SDLP BB lists this as managed parameter. Do we really need it? If so, the questions related to syntax and semantics apply.**  **WH: I think that syntax and semantic are clear when looking at the space link protocol books. What at least in my mind is not that clear is the question to which FR we should really assign this parameter.**  **190318 JP: We agree that it refers to the symbol stream and not to the RF link. For the forward 401 and 415 carriers, this is one-for-one. So for the fwd 401 and 415 carriers the channel names are redundant with the connectivity information inherent in the configuration profiles, since each carrier FR instance is connected to one and only one fwd sync and chnl encoding FR instance. The forward physical channel names therefore are unnecessary for the forward link FR chains, and we could eliminate them from the fwd 401 and 415 carrier FRs as well as the TC and FLF sync and channel coding FRs.**  **I also agree with your proposal to also keep it one-for-one for return 401 carriers, and so likewise the physical channel name is redundant with configuration profile information connecting the rtn 401 carrier FR with the single rtn sync and chnl decoding FR instance. However, the return physical channel name is still useful for cross-referencing from the fwd link for the source of CLCWs.**  **Finally, return 415 (i.e., TDRSS) carriers do allow separate data streams on the I & Q channels, so when we get around to specifying the fwd 415 carrier FR I propose that the rtn 415 carrier FR be the one that attaches the channel names to the channels. For consistency I believe that the Physical Channel Name should be specified in the carrier FR.**  **All instances of Physical Channel Name in FRs other than the carrier FRs are references to the physical channels consumed or supplied by the carrier FRs. My feeling is that it is not necessary to include these names beyond the sync and channel coding FRs because the configuration profiles essentially carry this information. E.g., because the config profile specifies that FwdTcMcMux FR instance M is connected to FwdTcPlopSyncAndChnlEncode FR instance X, and FwdTcPlopSyncAndChnlEncode FR instance X operates on the Physical Channel named Z, then we also know that FwdTcMcMux FR instance M is associated with Physical Channel Z. Let’s discuss this.**  **190415 WH – If I understand all of the above correctly, I think that we are in agreement that (disregarding for the moment CCSDS 415) knowing a given symbol stream and knowing the “connectivity” of the symbol stream to the carrier we know the Physical Channel Name and that name applies both to the symbol stream and the carrier. Therefore, in order to avoid redundancy, we should assign this parameter only to the FR specifying the carrier. The same consideration then applies to return links.**  **I also agree that once we get to specify the CCSDS 415 FRs, we assign the parameter that specifies the two Physical Channel Names to the carrier FR.**  **I have moved the parameter specifying the Physical Channel Name accordingly.**  **190722 WH – In the meantime I got the verbal response from Gian Paolo that he agrees with our approach to handling this parameter.** | **correspondence between WH and JP** | **Approach agreed.** | **C** |
| add fwdTcMcMuxMaxChnlAccessUnitSize parameter | WH: For the time being, this FR deals exclusively with TC frames. Therefore I don't understand the need to add any parameters related to CADUs. The FR has a parameter that specifies the maximum TC frame length.  190318 JP – This has nothing to do with CADUs. The TC Sync and Channel Coding book calls the primitive by which the TC SDLP (i.e., the TC MC Mux procedure) submits the set of frames and the associated Repetitions parameter the *ChannelAccess.request* primitive (CCSDS 231.0, A4.2). I coined the name *TC channel access unit* as the name for the set of frames that the MC Mux submits to the coding layer via the ChannelAccess.request primitive. The TC SDLP book simply calls this group of frames a “data unit”, but 231.0 terminology. that term is overused and not specific so I chose to base the name on. Would the name TC Channel Access Frame Set (resulting in the parameter name fwdTcMcMuxMaxChnlAccessFrameSetSize) be less ambiguous?  190415 WH - I now understand where you are coming from. However, I disagree that such parameter should be added. There is no such managed parameter defined in 231.0. What is restricted is the resulting CLTU size and that parameter is already specified for this FR. To be discussed.  190722 WH – Hopefully I did not overlook a response form John on this issue. I’m still of the opinion that such parameter shall not be added and therefore left the FR in this respect alone.  190808 JP – I see that the latest frm file has a ‘fwdTcPlopSyncMaxCltuLength' parameter in the FwdTcPlopSyncAndChnlEncode FR. I agree that this parameter serves the purpose of both proposed fwdTcMcMuxMaxChnlAccessUnitSize parameter and concur that this specific parameter should not be added, as long as we can assume that whatever system is building the “channel access unit” (i.e., the set of frames that are passed down) also has access to the coding information from the FwdTcPlopSyncAndChnlEncode FR so that the max frame set size can be “backed out” of the max CLTU length. I consider this item to be closed |  |  | C |
| **add fwdTcMcMuxMaxNumFramesInAccessUnit parameter** | **WH: See row above.**  **190318 JP – See row above.**  **190415 WH – See row above.**  **190722 WH – (Still) see row above.**  **190808 – This is actually a different issue from the max CLTU size, it’s the number of frames in the CLTU. I see that the FwdTcPlopSyncAndChnlEncode FR has a 'fwdTcPlopSyncMaxNumberOfFramesPerCltu' parameter that theoretically suites this purpose. However, the FwdTcPlopSyncAndChnlEncode FR doesn’t know or care about how many frames are in the CLTU (nor does the FCLTU SLE service). I suggest that the 'fwdTcPlopSyncMaxNumberOfFramesPerCltu' parameter be moved the FwdTcMcMux FR and relabeled 'fwdTcMcMuxMaxNumberOfFramesPerCltu'.**  **190909 WH – I concur and have added this parameter as suggested. I therefore change the status of this item to “closed”.** |  |  | **C** |
| **validTcScids?** | **SANA FR Registry has a validTcScids parameter. What is the purpose of/concept behind this parameter? Proposal: it is not used to actively filter data, but rather as a configuration-check parameter to ensure that all units feeding it are properly configured.**  **WH: My intention when specifying the parameter was a filter that prevents Master Channels being passed on to the forward link although those Master Channels are not permitted. I certainly concede that if everything closer to the service user is correctly set up, this problem will not occur. FSP implements such filters.**  **190318 JP: At either the Spring or Fall 2018 meeting, we discussed whether each FR needed to have all of the information that it might work with or if it could be assumed to “share” information with other FRs with which it is unambiguously associated (e.g., by configuration profile). Holger expressed his desire to assume that data is shared so as to minimize the possibility of having conflicting data within the configuration as a whole, and I believe this was also the WG consensus. I am a proponent of the idea that whatever FRs “feed” the VC and MC muxes will vet the incoming frames, either by filtering \*their\* input (as in the case of FSP) or because they actually generate the VCs and MCs based on their own configuration information. So in my opinion the set of valid SCIDs is not needed to be explicitly stated for this FR. However, I also recognize that some ESLT implementations may not be fully integrated and some localized checking may be desired. However, we have also not specified any behavior when an invalid frame is encountered. I propose that we treat this as a local (i.e., Agency/Provider CSSS/ESLT) matter, in which additional behavior, configuration parameters, read-only parameters (e.g., numbers of frames rejected because of invalid SCID), and even events (e.g., frame rejected for invalid scid’) can be defined as Agency-specific extensions under the Agencies branch (we still need to define the method for adding Agency-specific behavior and PEDs to CCSDS-standard FRs). Alternatively, we could possibly define optional configuration parameters that do not need to be set if the information can be gleaned from the surrounding configuration. Let’s discuss.**  **190415 WH – I agree that whenever possible we shall avoid the risk of conflicting configurations. The reason for proposing this parameter for the given FR was that I felt that such checks should be performed by the service production (of FSP in this case) which is represented by the FRs. In other words, FSP would delegate such checks to this FR. But I agree that we can also take the other approach to require that whatever FSP passes down to service production is “clean” in view of the applicable configuration. Then this parameter will be a parameter of the FSP provider FR. If that is the preferred option, I’ll move this parameter accordingly.**  **190722 WH – I have moved this parameter to the FSP provider FR type.** |  | **Parameter moved to the FSP provider FR type.** | **C** |
| **transfer frame version number?** | **TC SDLP has transfer frame VN as a managed parameter. However, since TC by definition is v1, this seems unnecessary.**  **WH: Agreed and that is why such parameter is not specified for this FR.** |  |  | **C** |
| **add fwdTcMcMuxBitRate param** | **Maximum bit rate accepted by the underlying coding sublayer.**  **WH: I don't understand the need for such parameter. Some kind of flow control will be needed between the FRs, but ultimately the bit rate is determined by the symbol rate configured for the Fwd401SpaceLinkCarrierXmit FR.**  **20190318 JP: I included this because the TC SDLP Blue Book lists it as a managed parameter for the physical channel, but as you point out that’s not for TC MC Mux “layer” to deal with. Agree to leave it out.** |  | **CLOSED on 2019-03-18** | **C** |
| **Add fwdTcMcMuxMaxNumRepetitions parameter** | **WH: While I agree that such parameters are needed, I have assigned them to the FwdTcPlopSyncAndChnlEncode FR. The reason for doing that is that then the same parameters can control the repetition both when the input are MC frames or CLTUs that carry a single frame.**  **20190318 JP: I agree that we don’t need to add this. This is a remnant from the time that I wasn’t sure whether FRs could “share” information (we have declared that they can) or need to be self-contained** | **TC SDLP Blue Book** | **CLOSED on 20190318** | **C** |
| Concept for repetition and frame aggregation information? | Process of deciding what frames can be bundled together into the same CLTU with the same repetition number is potentially complex but is not addressed in TCSDLP book. It may get even more convoluted with MC and VC multiplexing. We need a concept for how this might be done, and how that affects the managed parameters associated with it.  WH: I believe that the FwdTcPlopSyncAndChnlEncode FR parameters cover all needs, but please let me know if I have overlooked something in this context.  20190318 JP: According to the TC Blue Books, it is the TC SDLP’s job (i.e., the TC MC Mux’s job) to pass the set of frames that will grouped into a single CLTU and to set the Repetitions parameter of the ChannelAccess.request primitive (CCSDS 231.0 A4.2). How does the TC Mux know, for example, to pass down one frame with Repetitions = 3 vs. send 3 frames with Repetitions = 1? Let’s discuss.  20190415 WH – CCSDS 231.0 takes in a way the easy way out in that the ChannelAccess.request primitive is restricted to a single user and it is clearly spelled out that the repetition refers to the number of times the CLTU will be transmitted that results from the set of frames handed over by means of the primitive. That means that the responsibility for the functioning of the repetition mechanism is left to the service user. For instance, it is up to the user that BD frames are not repeated which means BD and AD frames must not be in a common group of frames except if the repetition is set to 1. If we want to support what FSP specifies, we must further make sure that a given group of frames contains frames of one VC only and either only BC or only AD frames because the repletion may be configured differently for both. Bearing in mind that the way FSP supports the repetition parameter has been agreed with the SLS folks, I would think that we should stick to that.  So we are in agreement that we should support scenario c) as described in JP email dated March 26.  I’m not sure where to capture the behavior of the TC MC multiplexer, but it will be something like this:  If the incoming frame is a BD frame or the frame is annotated such that the number of repetitions is 1, then the TC MC Multiplexer may group these frames for inclusion in a single CLTU as long as the fwdTcMcMuxMaxNumberOfFramesPerCltu is not exceeded.  Incoming AD frames from the same VC may be grouped for inclusion into a single shared CLTU as long as the next frame is an AD frame from the same VC and fwdTcMcMuxMaxNumberOfFramesPerCltu is not exceeded. The resulting CLTU will be passed to the FwdTcPlopSyncAndChnlEncode FR as many times as specified by the number of AD frame repetitions for the given VC. As soon as a non-AD frame or a frame from a different VC is coming in, a separate CLTU is generated.  Incoming BC frames from the same VC may be grouped for inclusion into a single shared CLTU as long as the next frame is a BC frame from the same VC and fwdTcMcMuxMaxNumberOfFramesPerCltu is not exceeded. The resulting CLTU will be passed to the FwdTcPlopSyncAndChnlEncode FR as many times as specified by the number of BC frame repetitions for the given VC. As soon as a non-BC frame or a frame from a different VC is coming in, a separate CLTU is generated.  190722 WH – Although we’ll see some clean-up regarding the repetition option in the SLS books, we’ll probably won’t get something that one could consider to be the specification of the related service production behavior. But I don’t think that we’ll do ourselves a favor when we limit ourselves to what one can read from the SLS books. Rather, we should undertake to document the assumed service production behavior and the matching FR type specifications. Probably the to be magenta FR TN is the best home for the related service production specification. In the light of the above, the FwdTcMcMux does not need to be modified in this regard. |  | CLOSED on 191024 | C |
| **add fwdTcMcMuxResourceStatusChange event** |  |  | **Agreed and done** | **C** |
| **add fwdTcMcMuxSetControlParameters directive** | **WH: I don't think that a dedicated directive is needed. The parameters can be set as needed using the fwdTcMcMuxSetContrParams directive.**  **20190318 JP: Agreed. This parameter was misnamed SetTcMcMuxControlParameters in the previous versions of theFRM XML file. I should have said “change classifier” instead of “add”.** |  | **CLOSED 20190318** | **C** |
| **add fwdTcMcMuxDiscardDataUnits directive** |  |  | **Agreed and done** | **C** |
| **Change name to TcMcMux** | **Drop the “Fwd” so that this FR can be used in nodes other than ESLTs.**  **This item will be closed when the change has been made to the working FRM file** | **JP - 191024** |  | **O** |