3GPP TSG RAN WG1
Meeting #2
22-25th, February, 1999
Yokohama, Japan

TSGR1#2(99)094

Source:	**Philips**
Title:	**Initial Response to Comments and Questions on FAUSCH from ARIB members**
Agenda item:	
Document for:	

Summary

This document is a first response to the questions raised by ARIB members on FAUSCH. More complete answers may be given later.

Original text is in *italics.*

3. Comments and questions

3.1. Collision free

It is mentioned that one of the significant advantage of FAUSCH over RACH is collision free. However, we can realize near collision free system using muti-time offset and signatures, as stated Tdoc L1 675/98. Therefore, it seems this advantage is hardly effective.

The number of collisions on RACH will be significant for high loading, which could occur in some scenarios described in Tdoc L1 008/99. The capacity of the RACH channel could be increased with additional Gold Codes, but the receiver complexity would be increased significantly. In Tdoc L1 675/98 is suggested that the envisaged packet traffic would be carried by a continuous DCH, avoiding the need to load RACH (or use FAUSCH). In this case the overhead of maintaining the DCH between packets will be significant and system capacity reduced.

On the other hand, if hardware resources for RACH and FAUSCH are limited (i.e. the number of multi-time offset and signature to be used is very lower than maximal number.), we have to consider collision problem. In this case, connections which applied FAUSCH are happy because of collision free environment. However, services which applied RACH are not happy, because hardware resources for RACH decreases and collision probability increases. As a results, it is estimated that total collision probability might increase because of lack of statistical multiplexing effect.

The balance of resources allocated to RACH/FAUSCH will depend on the traffic. If a significant fraction of traffic is supported using FAUSCH, the collision probability will not be increased. As an example, let us consider the case where all the resource is granted to RACH. For a given number of active users (say 100), the collision probability will have some value (say x). Suppose that the frame is divided so that 50% of the resource is allocated to RACH and 50% to FAUSCH. If the users are divided in the same way, then 50 of the users use RACH and 50 use FAUSCH. The collision probability for the 50 RACH users is unchanged (x), as both the resource and number of users have been reduced by the same amount. However, the 50 FAUSCH users have no collision, which will be particularly beneficial if the more delay sensitive applications preferentially use FAUSCH. Even considering all the users, the average collision probability is reduced by half (to x/2). In practice, more than 50% of the traffic could be allocated to FAUSCH. If the traffic were such that it would not benefit from FAUSCH (e.g. little or no uplink packet data), then all the resource could be allocated to RACH by the operator.

3.2 Eb/No point of view

It is mentioned that FAUSCH has an significant advantage from Eb/Io point of view.
In Tdoc L1 001/99, FAUSCH which "access sequence be correctly detected" and RACH which "Preamble and data payload is correctly detected" are compared. In this situation, RACH can carry 100 to several hundred information bits, while FAUSCH seems to need extra procedure, such as "granted by FACH", "DCH sync. establishment" and so on. Taking account these overhead for FAUSCH, I suppose that the difference in resultant of Tdoc L1 001/99 becomes smaller.

The Eb/No is better considered by looking at Tdoc L1 623/98. Tdoc L1 001/99 mainly considers the small difference in results due to the inclusion of coding gain on the RACH.
It is true that the FAUSCH needs some extra procedure compared with sending a packet in the message part of the RACH. In some cases it may be true that the advantage of better Eb/No in the uplink for FAUSCH is offset by the need to transmit a message in the downlink on FACH (although downlink overheads cannot be directly compared with uplink overheads). Therefore we consider that FAUSCH is suitable as an alternative to RACH to establish a DCH for packet transmission.
Thus, in Tdoc 623/98 the most useful comparison is (in Figures 1 and 2) between

(1) RACH/ ”granted by FACH”/ ”DCH sync establishment”

and

(2) FAUSCH/ ”granted by FACH”/ ”DCH sync establishment”

Here the procedures are essentially the same in both cases. This is the case where SMG2 L1 group recognized a performance advantage in Eb/No.

Considering these overhead, FAUSCH seems to have advantages in transmitting relatively longer messages, even though, there is a description in your mail on February 8 that says "the most beneficial application appears ....'up to a few hundred bits'".

We believe that FAUSCH/FACH/DCH is preferable to RACH/FACH/DCH for packets up a few hundred bits. Furthermore it is preferable to the use of message part of RACH for all packet sizes (considering only the uplink Eb/No).

3.3 Initial transmission power of downlink DPCH

For decision of initial transmission power of downlink DPCH, it is beneficial to use path loss which is calculated using P-CCPCH received power. RACH can transmit the latest measurement value of P-CCPCH received power just before initial transmission of DPCH.
On the other hand, FAUSCH cannot transmit any higher layer information. It is difficult to set suitable initial transmission power of DPCH. Therefore, It seems that employing FAUSCH cause longer synchronization establishment time of DPCH or excess initial transmission power, compared with using RACH .

Currently the RACH specification does not include the transmission of such information. One problem is the RACH limited payload size. If power information were transmitted after the GRANT message on FACH, the FAUSCH and RACH become the same in this regard.

3.4 Hardware and testing complexity

It is mentioned that additional complexity caused by employing FAUSCH is very small. However, it is clear that hardware complexity with FAUSCH will not be equal or less than that without FAUSCH.
We should recognize that the numbers of testing items will increase and configuration of test tool will be complicated, also.

It is generally true that any additional feature will add complexity. However, this additional complexity is small for FAUSCH, certainly in comparison with other features under discussion in RAN TSG WG1.

3.5 Delay of power ramping process
If some of the access slots are FAUSCH slots, it means that power ramping process of RACH will be delayed, since idle period between two consecutive RACH preambles will be longer, if it has to jump sometimes over some FAUSCH slots.

If power ramping is used in RACH (at least as currently described), then only alternate access slots are used by any one access attempt. For RACH the average delay will include at least 5ms for random selection of an access slot in the 10ms frame, and 2.5ms for each ramping step needed. If the frame has a single partition in the centre (as in Figure 3 of Tdoc 007/99), the average delay for RACH becomes 2.5ms for selection of an initial access slot, and an average of around 5.6ms per ramping step. So if up to three ramping steps were needed the average delay would be increased by rather less than 7ms. This is small compared with typical delays required for messages to or from L2/3 (30ms has been suggested). In any case FAUSCH can be incorporated with out adding any delay if every second “RACH” slot is allocated to FAUSCH (as in Tdoc L1 007/99 Figure 4).

3.6 Delay of standardization process
There are many issues related to FAUSCH in L1 and L23. It seems the discussion for these issue will consume very long time in each WG. We worry about delay of standardization process.

FAUSCH has only impact on limited parts of L1 and L2/3. For L2/3 it only affects the 1st step in requesting a new DCH in connected mode and assignment/release of the time offset at the association/deassociation phase. Algorithms for offset time assignment etc are an implementation issue. Most of the necessary discussion has already taken place in ETSI SMG2. FAUSCH could be an optional feature in products.

3.7 Time offset management
When and how, UE will release its uniquely allocated "time offset"?
Is it the only case when handover or registration to another NodeB is successfully accomplished?

I wonder there should be some schemes that can detect "disappeared UE" even when no available "release" message has come to RRC. According to these consideration, FAUSCH seems to need some extra-complexed scheme to manage "time offset" effectively, doesn't it?

Assignment of time offset will be triggered when the UE enters Node B connected mode and release of time offset will be triggered when the UE leaves Node B connected mode. Detection of whether the UE is entering or leaving Node B connected mode must be done in any case. For FAUSCH the RRC needs only to mark that one (or more) time offsets are now reserved for the UE. The release could be by time-out as well as explicit signalling (when leaving the Connected mode). This is similar to the case for packet transmission by discontinuous DCH, where a time-out is also required if the UE “disappears”. FAUSCH does not need a complex solution.

3.8 Timing of the time offset assignment
By what message, UE requires to NodeB to allocate new "time offset" when it arrives at new NodeB area? Is it always done with L3 registration message, or can be done alone itself?

We recommend the use of the registration message.
Strictly, the UE would only need to request a new time offset allocation at a new Node B if it wishes to send data by FAUSCH/FACH/DCH. This could be done using the RACH. Using the RACH/FACH/DCH method for packet transmission would also need similar signalling when arriving at a new Node B.

In case, "time offset allocation req." can be send it alone, it will lead extra traffic to do it, even when newly arrived NodeB is suffering "heavy load".

Load management for RACH has been the subject of some recent discussion in SMG2 L1 group, and broadcast of persistence parameters was proposed. This mechanism might be generally relevant for preventing overloading by new UE's arriving at a Node B (and not just for FAUSCH users).

3.9 Confirmation of the time offset assignment
Are there any "CONFIRMATION RESPONSE message for TIME OFFSET ALLOC" from UE to RNC when UE receives "TIME OFFSET ALLOC" message? If so, which is used for the CONFIRMATION RESPONSE message transmission, RACH or DPCH after FAUSCH?

Time offset allocation is performed during association, and therefore would use the association procedure. Only need an additional information element for the association handshake message.

3.10 Performance of FAUSCH
In doc.[1], "signature number 8" is applied to FAUSCH, while in doc.[2], "RACH preamble exhibits large sidelobes" is pointed out. There seems to be some inconsistency between them. Why "signature" can become applicable to FAUSCH?

The sidelobes might be significant for the worst case PRACH signature. This was the reason that a Barker sequence was proposed at one point in the discussion. However, in order to achieve better commonality with the PRACH, the best available signature was adopted instead (which has correlation sidelobes almost as good as the Barker sequence).

With latest condition of FAUSCH, which is merged with RACH scheme, how's the advantage of FAUSCH looks like? How many UE can be served and how's the overhead?

The original advantages of FAUSCH are maintained with the merged scheme (reduced collision risk, better Eb/No). The number of UE's which can be served depends on how much of the frame is allocated to FAUSCH, and the separation needed between access slots, which is scenario dependent. However, as indicated in Tdoc L1 008/99 Annex A, in a cell with radius 1.5km up to about 500 slots could be allocated in a 10ms frame, assuming a single Gold code dedicated to FAUSCH. If the frame is partitioned approximately equally between RACH and FAUSCH, using the same Gold Code, it should be possible to support up to around 200 UE's (in connected mode).

Conclusion

FAUSCH should be retained since it offers performance benefits with minor additional complexity.