Multi-User ARQ

Peter Larsson and Niklas Johansson
Ericsson research, Ericsson AB, 164 80 Stockholm, Sweden.
{peter.j.larsson | niklas.j.johansson} @ericsson.com

***Abstract* — This paper presents a novel ARQ scheme for multiple unicast flows in a multi-user system, which enhances the aggregate throughput efficiency. The core idea is to exploit knowledge of previously overheard packets by unintended users, and perform wise selection and joint encoding of any retransmitted information. Intended users may decode the jointly encoded packet by exploiting the previously overheard packets. It is found that the throughput efficiency in a K-user system approaches, the fairly remarkable result of, unit value when K goes to infinity. Moreover, complexity and overhead of the scheme is studied and a link to (feedback free) network coding is established.**

Keywords; ARQ, Multi-user communication, Network Coding, Feedback.

I. Introduction

In an error prone communication channel, ARQ is traditionally used to enable (more) reliable communication. ARQ schemes have evolved from rudimentary retransmissions methods offering basic reliability, e.g. Stop-and-Wait (*SW*), Go-Back-N (*GBN*) and Selective-Repeat (*SR*), to become a vital performance enhancer of wireless systems of today through various Hybrid-ARQ schemes [1].

Many wireless communication systems of today, such as cellular systems, have a (physical layer) point-to-multipoint (*PMP*) structure, i.e. a sender communicating with multiple users $u_k \quad k \in \{1,...,K\}$. Despite this, classical ARQ schemes are solely based on a (link) point-to-point design, thereby failing to utilize the PMP structure.

In this paper, an ARQ (and scheduling) method denoted Multiuser ARQ (*MU-ARQ*) that exploits the PMP topology is introduced, e.g. see Fig. 1. The total throughput, feedback and addressing overhead, en/de-coding complexity are analyzed in a K -user system in identically, independently distributed (i.i.d.) channels with packet reception probability p. The main result is that throughput efficiency approaches one as $K \to \infty$ when $p \neq 0$. The throughput efficiency for i.i.d Rayleigh fading channels is also considered. Then, both major SNR and diversity gains are seen. In addition, a two-user case with non-identically i.d. channels (Rayleigh) is analyzed. For numerous users, the overhead is not negligible, whereas the en/de-coding complexity is low.

The rest of this paper is organized as follows. The proposed scheme is introduced in section II, analyzed in section III whereas related work is discussed in IV, and finally in V the paper is summarized.

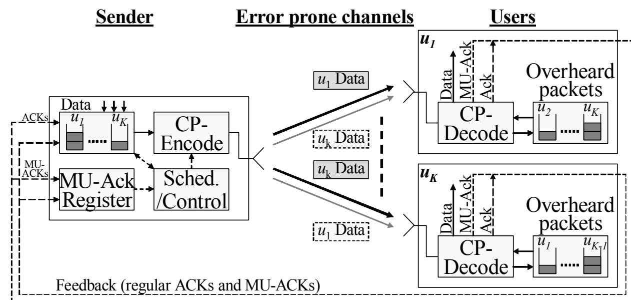

Figure 1. Multi-User ARQ System Architecture

II. Multi-User ARQ

The guiding observation towards the proposed method is that users often overhear each others information, due to the radio channels' broadcast characteristic, but traditional ARQ schemes fail to exploit this. In the following, it will be explained how this observation is used and why it is beneficial.

- Each user u_k, receives its own designated packets $D_k(n)$, with sequence number n, but also overhears and receives other users' packets $D_l(n), l \neq k$. These overheard packets, here called unintended packets, are temporarily stored. Then, each user u_k returns feedback on correctly received packets, i.e. regular acknowledgments (ACK) for designated packets and (MU- ACKs) for unintended packets.

- The sender registers the received MU-ACKs, and purges transmitted packets as regular ACKs are received. At subsequent transmission, the sender examines received MU-ACKs and pending packets to (re-)transmit and schedules what to send and to whom. The decision could be to transmit a new packet, a previously transmitted packet, or a so-called composite packet (CP). The CP is a packet where wisely selected multiple packets[a], not yet received by their designated users but overheard by other unintended users, are jointly encoded such that the length of the CP (payload) equals the largest of the multiple packets. For the encoding, a bit-wise XOR operation is used here (with zero-padding if needed). The individual headers of the multiple packets encoded in the CP are, for instance, appended after each other in a CP header.

- At a CP reception, a user identifies the encoded packets and uses its a priori information (i.e. unintended packets) to decode the CP (to greatest extent possible). For decoding,

[a] Wisely selected, such that if the CP is correctly received by any of the addressed receivers, the CP may be decoded.

bitwise XOR-ing is used that effectively cancels the impact of unintended packets. With only partial a priori information corresponding to packets encoded in a CP, the CP cannot be fully decoded. If so, partially decoded composite packet may also be stored and used as a priori information for decoding of forthcoming CPs. Information on received and decoded packet is, continuously, fed back to the sender.

- When a user u_k has received a designated packet, it is of no use for other users. Hence, the users discard such unintended packets, e.g. through expiration of an unintended packet time-to-live timer or purge message from the sender.

After this description of MU-ARQ, the function and benefit is exemplified through Fig. 2. Here, after the first two transmissions, u_1 and u_2 have not yet received their own (designated) packets, but each others (unintended) packets. The sender then schedule and transmit, based on feedback, a bitwise XOR encoded CP. There are two causes for the performance improvement. First, if both users receive the CP, both can decode their designated packet. This delivers two packets with one re-transmission, albeit with a lower probability as both receptions must be successful. Second, if either of the users receives the CP, it may extract its designated packet. The probability for this event is almost twice as large as one user receives one packet (for high reception probabilities). In all, the throughput efficiency for the CP can be shown to be twice that of a single user case.

III. Performance Analysis

In this section the throughput efficiency, overhead, as well as complexity of MU-ARQ are examined for i.i.d. channels. Also, throughput efficiency for $K=2$ with non-identical i.d. channels is studied. For analytical tractability, the scheme is simplified and the studied cases are limited, yet not implying that this is, or will be, the operation in practice.

A. Throughput Efficiency

Here, the throughput efficiency[b] for K users is examined with the assumptions of each user having identical; packet sizes, data rates, and i.i.d. reception (failure) probabilities p ($q=1-p$). Further, non-sequence number limited ARQ, error free feedback, and full buffers are assumed.

The idea behind the analysis is as follows. In a first phase, a large number of M' regular data packets are sent, i.e. large enough to allow a probabilistic analysis. Some packets are received by the designated users, and others are received by unintended users. Based on feedback, ACK and MU-ACK, the sender in a second phase forms and sends CPs only. In the CP phase, immediate feedback from a CP addressed user occurs when an intended packet is extracted from the CP. The goal is that after the CP phase, of in total M'' transmission, all packets received by any user in the first phase will be received by their designated user. The throughput efficiency is then the number of packets received by any user in the first phase, divided by the number of transmissions for both the first and the second phase. While phases are used in the analysis to ease the understanding, the result is not limited hereto. For instance, one may imagine an infinite number of partially overlapping two-phase periods, to see the analysis general applicability.

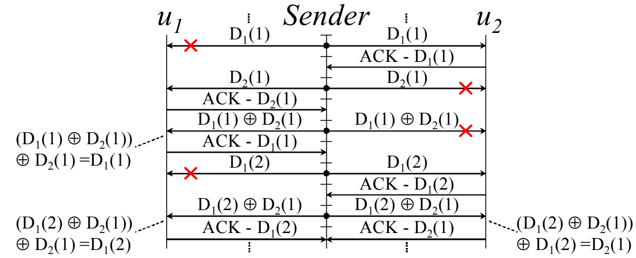

Figure 2. MU-ARQ example. K=2 users.

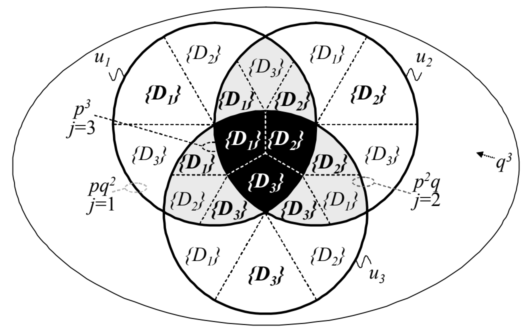

Figure 3. MU-ARQ Throughput analysis. K=3 users.

The situation after the first phase is exemplified for $K=3$ users in Fig. 3, which also exemplifies the model for the analysis. Here, three users u_1, u_2, u_3 are represented as partially overlapping circles. Each circle signifies the set of packets received by corresponding user, and the union of the circles is the set of packets received by any user, etc. As each user have identical source data rates, each user receives an equal share of packets designated for all users. In Fig. 3, the set $\{D_k\}$ (separated with dotted lines) symbolically indicates a set of received packets designated for user u_k . The sets $\{D_k\}$ indicated in bold corresponds to packets that are received by the intended user, whereas the sets $\{D_k\}$ not indicated in bold corresponds to packets that are received by an unintended user. For the forthcoming discussion it proves useful to note that regions of j partially overlapping circles, where packets are received by j users, has a multiplicity of K over j. In Fig. 3, three sets of regions are also depicted with white ($j=1$), grey ($j=2$), and black ($j=3$) shades. Each set comprises all regions of j partially overlapping circles and are henceforth denoted *level* j.

After the first phase, CPs shall be sent. A first task is then to identify suitable packets to jointly encode into a CP. Let's exemplify this process based on Fig. 3. In an overlapping region, e.g. between u_1 and u_2 , both users receive the same packets. Each may keep their designated packets, but unintended packets, i.e. for u_3 , need to be retransmitted. Similarly, overlapping regions may be found for u_2 and u_3 , as

[b] Delay and transmission order are not of primary concern.

well as u_1 and u_3, where packets to u_1 and u_2 respectively need to be retransmitted. A CP may be formed by using any three packets, but from each and every one of those regions. If any of three (addressed) users receive the CP, they will always be able to decode their missing designated packet. We call such CP, where each addressed user may decode its designated packet, an *optimal CP*. It turns out that with the model, as in Fig 3, optimal CPs can always be formed as long as the sets of unintended received packets are non-empty. Consider, e.g. if u_1 received a CP, a revised CP may be formed with a new packet for u_1 from the common region between u_2 and u_3. Finding optimal CPs are always possible for arbitrary K, by considering sets of packets on the overlap level j and selecting a group j users with overlapping regions. For each level j, there are K over j such groups. In the following, only optimal CP will, and need to, be considered.

After this introduction, the throughput efficiency is now derived and may be written as

$$T = \frac{N'}{M'+M''} = \frac{N'/M'}{1+M''/M'}, \tag{1}$$

where N' is the number of packets received by any user in the first phase, and M' and M'' are the number of sent packet in the first and second phase, respectively. The ratio N'/M'' is the probability of reception in the first phase, i.e.

$$N'/M' = 1-q^K. \tag{2}$$

The number of transmissions required in phase two is

$$M'' = \sum_{j=1}^{K} \frac{N'_j}{T''_j}, \tag{3}$$

where T''_j is the CP throughput efficiency on level j, and N'_j is the number of level j packets received in the first phase by unintended users.

When determining T''_j, it is first noted that as optimal CPs can be formed every time for any group at level j, the level j CP throughput efficiency T''_j is the same as the throughput efficiency for each CP transmission. Therefore, T''_j can be calculated as the multiplicity that i out of $j+1$ users receive a CP packet, times the probability of the event to happen, summed over all possible events. More concisely, this may be written

$$T''_j = \sum_{i=1}^{j+1} \binom{j+1}{i} i p^i q^{j+1-i} = (j+1)p. \tag{4}$$

To determine N'_j at level j in the first phase, it is noted that each such overlapping region, occur with probability $p^j q^{K-j}$, has a fraction $(K-j)/K$ of unintended received packets, and have multiplicity of K over j, which when combined gives

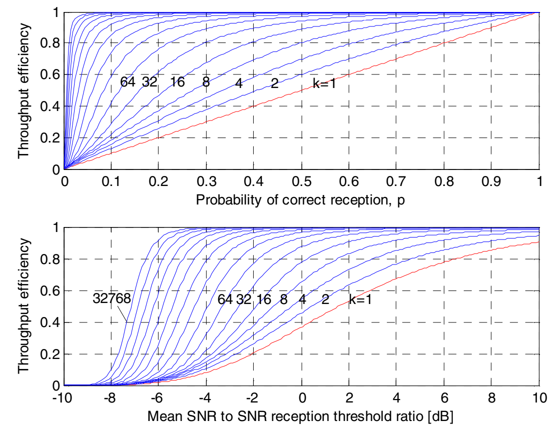

Figure 4. MU-ARQ throughput vs. reception probability (upper) and mean SNR to SNR reception threshold (lower). k=2^0,2^1,..,2^{15}

$$N'_j = M' \binom{K}{j} \frac{(K-j)}{K} p^j q^{K-j}. \tag{5}$$

By inserting (4) and (5) in (3), one gets

$$\frac{M''}{M'} = \sum_{j=1}^{K} \binom{K}{j} \frac{(K-j)}{K(j+1)} p^{j-1} q^{K-j} = \frac{q\left(1-q^K - Kpq^{K-1}\right)}{Kp^2}. \tag{6}$$

Now, with (6) and (1), the throughput efficiency is given as

$$T = \frac{1-q^K}{1+\dfrac{q}{Kp^2}\left(1-q^K-Kpq^{K-1}\right)}. \tag{7}$$

Many interesting observations can be made from (7), but the most important is

$$\lim_{K\to\infty} T = \begin{cases} 1 & \text{if } p \neq 0 \\ 0 & \text{if } p = 0 \end{cases}. \tag{8}$$

This tells us, the extraordinary result, that the throughput efficiency can approach one for an unreliable channel, if the number of users goes towards infinity. In contrast, classical SR-ARQ is upper limited to $T_{SR-ARQ} = Kp/K = p$, irrespective of number of users. Equation (7) is plotted versus the reception probability p in Fig. 4 (upper), illustrating the effect in (8).

One question is valid at this instance. Why is the throughput efficiency so good and approaches one with increasing K? One hint to the answer is to note that a genie directed transmitter, that always succeed, has a throughput efficiency upper limited to $1-q^K$ that approaches one when $K \to \infty$. A better answer is to note that as K increases, the overlapping regions with $j = K/2$, that have multiplicity K over $K/2$, soon dominate with increasing K. Briefly assuming that all regions are of this type and that $N'' = M'\left(1-q^K-p\right)$ packets need to be retransmitted with optimal CPs, it is found that $M'' = N''/T''_{K/2} \to 0$ when $K \to \infty$ and $p \neq 0$ since $T''_{K/2} = (K+1)p/2$ that intuitively explains (8).

As MU-ARQ can be used in wireless, the performance for some typical random channel, e.g. i.i.d. Rayleigh fading, is of interest.[c] The probability of correct reception is then given by the exponential distribution

$$p = \exp\left(-\frac{\Gamma}{\gamma}\right), \text{ when } \gamma \geq 0\,, \tag{9}$$

where γ is the average received signal to interference plus noise ratio (SINR), and Γ [d] is a SINR threshold above which the receiver correctly decodes a packet. With (9) in (7), the throughput efficiency is plotted vs. γ/Γ [dB] in Fig. 4 (lower). Two observations are that the SNR sensitivity and the degree of diversity increase, as K increases.

B. Overhead

In contrast to classical ARQ, more feedback will be sent here due to MU-ACKs. What is then a sound measure for feedback overhead? One viable measure is the average amount of positive feedback per sent packet (negative ACKs are omitted here). This is found by summing the amount of feedback sent in the two phases divided with the total number of transmissions. The most straightforward approach is to provide feedback for all received packets and CP in each phase, however this approach is not efficient[e]. Instead, it is recognized that a packet, received by a designated user in an overlapping region, see Fig. 3, do not need multiple feedbacks. This may be solved if regular ACKs for packets received by designated users are sent first, and then announced by the sender to suppress users from sending corresponding MU-ACKs. Subsequently, MU-ACKs for any other packets are fed back.

The average feedback overhead is the number of feedback messages divided by the number of transmissions over the two phases. In the first phase, feedback messages is counted one time for packets received by the intended receivers $M'p$, and then counted for all packets needing MU-ACK feedback messages. The number of MU-ACK feedback messages is determined as the sum over all levels j, where each level has multiplicity K over j overlapping regions, $(K-j)/K$ is the fraction of packets, j is the number of MU-ACK feedback messages per received unintended packets, and $p^j q^{K-j}$ is the probability of occurrence. In addition, the CPs as such incurs $M'(1-q^K-p)$ feedback messages. The overhead is then

$$\begin{aligned} O_{FB} &= \frac{M'\left(p + \sum_{j=1}^{K}\binom{K}{j}\frac{K-j}{K} j p^j q^{K-j} + \left(1-q^K-p\right)\right)}{M'+M''} \\ &= \frac{1-q^K+pq(K-1)}{1+M''/M'} \end{aligned}\,. \tag{10}$$

[c] Note that in reality, reception probabilities differ. Yet, this provides an upper bound of the throughput efficiency.

[d] A reasonable assumption for modern codes (LDPC and Turbo) and decoders, is a fixed threshold reception model.

[e] This overhead is $O_{FBfull} = \left(1-q^K+p(K-1)\right)/\left(1+M''/M'\right)$.

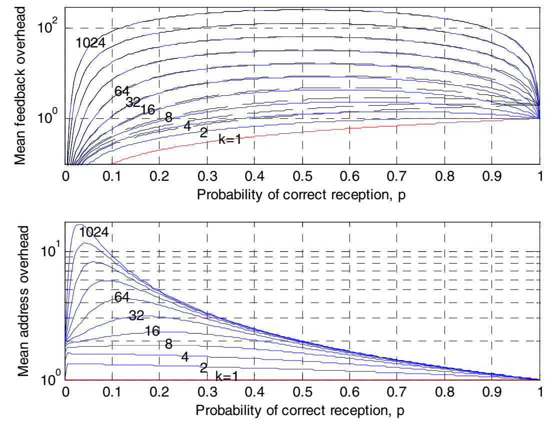

Figure 5. Mean feedback and address overhead. $k=2^0,2^1,..,2^{10}$

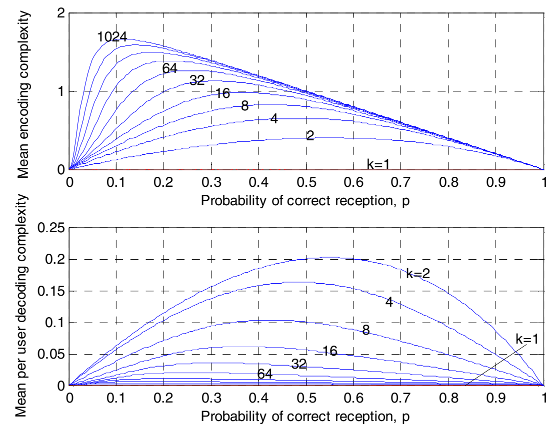

Figure 6. Mean encoding and per user decoding complexity. $k=2^0,2^1,..,2^{10}$

Note that some feedback overhead is in effect moved to the sender, due to feedback information announcement. This equals $p/(1+M''/M')$ and may, if desired, be added to (10).

The feedback overhead (10) vs. the reception probability is shown in Fig. 5 (upper) (the dotted lines includes the sender announcements). It is found that (10) peaks when $p = 0.5$ as $K \to \infty$, with $O_{FB} \approx K/4$. Clearly, the overhead is not insignificant. Yet, this loss must be weighted against the throughput efficiency gain, with packet and feedback sizes in mind.

Now, the average address (and sequence number) overhead is studied. A possible method, not necessarily the most efficient one, is that each CP carries all its $j+1$ addresses (sequence numbers etc.) to its designated users. The overhead is determined in a similar manner as the throughput efficiency analysis, but each packet and CP is weighted with the number of addresses in the header. The result is

$$\begin{aligned} O_{Addr} &= \frac{M'\left(1 + \sum_{j=1}^{K}\binom{K}{j}\frac{(K-j)}{K}\frac{(j+1)}{T_j} p^j q^{K-j}\right)}{M'+M''} \\ &= \frac{p^{-1}\left(1-q^K\right)}{1+M''/M'} = p^{-1}T \end{aligned}\,. \tag{11}$$

Fig. 5 (lower) shows (11) and that it increases when K increases and p decreases. For some values on K and p, as $O_{Addr} \approx p^{-1}$ when $K \to \infty$, the overhead appears too large and strategies need to be found to diminish this figure.

C. Complexity

In this section, average XOR encoding and decoding complexity for the CP will be addressed, in the order given.

Different CP encoding methods may be envisioned. Here, it is suggested that successfully delivered packets by a CP, is cancelled (by XOR-ing the just delivered packets) from the sent CP, and new packets are encoded (by XOR-ing the new packets) onto the CP in processing. In this way, each packet that is delivered with a CP requires two XOR operations (over the whole word). The average encoding complexity is then two times the number of packets left to be delivered in the second phase divided by the total number of transmissions, i.e.

$$C_{Enc} = \frac{2M'\left(1-q^K-p\right)}{M'+M''} = \frac{2\left(1-q^K-p\right)}{1+M''/M'}, \qquad (12)$$

with M''/M' as in (6). It is found that $C_{Enc} \le 2$, with equality at $p=0$ and $K \to \infty$, see Fig. 6 (upper).

For the CP decoding, different methods may also be imagined. One idea is that when a CP for level j and a group of j users is received, a user internal partial CP is built up for that level and group of users. This internal CP is XORed with a received CP, and a designated packet for the user can be retrieved. The internal CP is continuously modified and updated, through XOR – operations, based on newly received CPs. I.e. packet delivered to their designated users, are canceled from the internal CP, and any new packet coded in the CP is amended to the internal CP. As a result, the XOR complexity is, as in the encoding complexity analysis, two operations per outstanding packets from the first phase. While the encoding complexity was evaluated for a sender, it is instead reasonable to consider the decoding complexity per user, giving

$$C_{Dec} = C_{Enc}/K\,. \qquad (13)$$

It is seen in Fig. 6 (lower) and (13) that the decoding complexity is low (upper limited to $C_{Dec} \approx 0.2$ when $K=2$ at $p=0.55$) and decreasing to zero when $K \to \infty$. Note that with the en/de-coding methods studied here, low complexity is favored over memory requirements. Yet, even when using a costly method, i.e. doing an XOR operation for every packet contained in a CP, it can be shown (though not here) that the decoding complexity remains low (a peak develops at $p=0.5$, when $K \to \infty$ with $C_{Dec} \approx 0.25$).

D. Throughput efficiency for K=2 and p1 ≠ p2.

In the foregoing, identical reception probabilities were used. Now, MU-ARQ and classical SR-ARQ throughputs are compared, given $K=2$, arbitrary i.d. reception probabilities p_1, p_2, and identical per user packet delivery ratios.

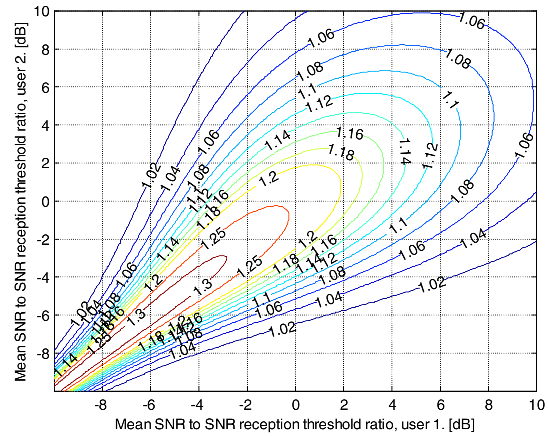

Figure 7. MU-ARQ to SR-ARQ Throughput efficiency gain in Rayleigh fading channels

In the first phase, x_1' and x_2', are the relative traffic shares for user u_1 and u_2, respectively and $x_1'+x_2'=1$. Each user has received a set of packets intended for the other user. Those packets should be delivered to the other user, where the number of transmission needed is determined by the throughput efficiency of the other user. The number of transmissions required in phase two to deliver all packets to user u_1 and u_2 respectively, are

$$M_1'' = x_2' M' p_1 q_2 / p_2\,, \text{ and } M_2'' = x_1' M' q_1 p_2 / p_1\,. \qquad (14a,b)$$

If the number of transmissions are set to be identical, i.e. $M_1''=M_2''$, then optimal CP transmissions may always be used in phase two. Hence, setting (14a) and (14b) equal yields

$$x_1' = \left(1+\frac{q_1}{q_2}\left(\frac{p_2}{p_1}\right)^2\right)^{-1}. \qquad (15)$$

Based on that $M'(1-q_1q_2)$ packets are received in the first phase, and with $M''=M_k''$ $k\in\{1,2\}$, (14) and (15) yields

$$T_{MUARQ2} = \frac{M'(1-q_1q_2)}{M'+M''} = \frac{p_1+p_2-p_1p_2}{1+\left(\frac{p_1}{q_1p_2}+\frac{p_2}{q_2p_1}\right)^{-1}}\,. \qquad (16)$$

Since the delivery ratio will be different from x_1' after phase two, the resulting delivery ratio may be calculated as

$$x_1 = \frac{x_1'p_1 + x_1'q_1p_2}{x_1'p_1 + x_1'q_1p_2 + x_2'p_2 + x_2'p_1q_2}\,. \qquad (17)$$

The SR-ARQ throughput efficiency may then be written

$$T_{SR-ARQ} = y_1p_1 + y_2p_2\,, \qquad (18)$$

where y_1 and y_2, with $y_1+y_2=1$, are the relative traffic shares for user u_1 and u_2, respectively. For a fair comparison of MU-ARQ and SR-ARQ throughput, the latter should have the same packet delivery ratio as the former. Setting the relative share of delivered data to u_1 in SR-ARQ equal to x_1'', y_1 can be solved for as

$$y_1 = \frac{x_1'' p_1}{p_1 + x_1''(p_2 - p_1)}. \quad (19)$$

By inserting (15) in (17), (17) in (19), (19) in (18), the SR-ARQ throughput efficiency can be calculated. In Fig. 7, a contour plot of the relative throughput efficiency gain for MU-ARQ over SR-ARQ in Rayleigh fading channels is shown as a function of γ_1/Γ_1 [dB] and γ_2/Γ_2 [dB] (extended to $K=2$ from (9)).

IV. RELATED AND FUTURE WORK

Clearly, MU-ARQ relates to regular ARQ, for which much prior work exists [1]. By now, the many differences are evident, vis-à-vis Feedback, Multi-user, en/de-coding.

Interestingly, a link exists to the emerging area of Network Coding (NC) [2]. In NC, data is routed in a network and allowed to be mixed with each other in the intermediate routers, resulting in increased network throughput. NC and MU-ARQ share the notion of coding (or mixing) packets with each other, but ample differences exist. In contrast to MU-ARQ, NC does (generally) not: consider feedback, operate in an error prone broadcast channel (like wireless), use retransmissions, exploit multiple users, strive for increased robustness through diversity, and consider multiple unicast flows. Instead, NC has mainly focused on broad-/multicasting in wired networks, and exploited network connectivity jointly with en/ de-coding.

Fountain Coding for broad-/multicasting (FC) and alike [3] may also be compared to MU-ARQ. FC is a rate-less data encoding scheme that produces new parity information until all users have decoded the sent data. This enables efficient multicasting in erasure channels where roughly ε more redundancy than data needs to be received. A clear difference is in the goal, i.e. multicast (FC) vs. multiple unicast (MU-ARQ). Also, in FC, feedback from each user is deferred until the full packet has been decoded. On the subtle side, FC throughput efficiency scales proportionally to the reception probability, whereas MU-ARQ $T=1$ as $K \to \infty$.

Many extensions to this study and scheme are foreseen. We are currently planning to extend this study to include; i) packet delay assessment, ii) an online schedule mechanism and its performance, iii) NACK extension and analysis of the resulting feedback overhead, iv) analyze complexity of finding appropriate packets for optimal CP creation, v) memory consumption analysis, and finally vi) its generalization to arbitrary non-identical i.d. reception probabilities and arbitrary number of users. Moreover, we are currently working on several enhancements of the proposed scheme. Examples of such are, but not limited to; i) improved throughput efficiency by exploiting the additional information provided by CPs that are received by only unintended users, ii) MU-ARQ may also be applied in multihop networks, exploiting overheard transmissions (thanks to the multiple transmission inherently caused by the multiple hops), iii) joint MU-ARQ and channel dependent scheduling is envisioned where multiple channel peaks to multiple users are used.

V. SUMMARY AND CONCLUSIONS

A new ARQ scheme, intended for multiple unicast flows, exploiting overheard packets was proposed and examined. It was found that throughput efficiency approaches unit value, when $K \to \infty$ and channels are i.i.d. and $p \neq 0$. Feedback and address overhead, as well as en/de-coding complexity issues were studied with the conclusion that; feedback overhead is a concern at large number of users (at $p \approx 0.5$), address overhead is only an issue at low reception probabilities, and the en/-decoding complexity appears low. Throughput efficiency evaluation in Rayleigh fading channel(s) revealed both significant SNR and diversity gains with increasing number of users. At the end, this work was linked to NC, differences to FC were pointed out, and several enhancements of the study and the scheme itself were highlighted.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] L. Shu, and D. J. Costello, Jr., “Error Control Coding: Fundamentals and Applications”, Englewood Cliffs, N.J., Prentice-Hall, 1983.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", IEEE Trans. on Information Theory, vol. 46, pp. 1204-1216, 2000

[3] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In Proceedings of ACM SIGCOMM, pages 56--67, 1998.