Reliability by Retransmissions in the Wireless World

Anastasios Giovanidis

Technical University of Berlin
Fraunhofer Heinrich Hertz Institute
Zuse Institute Berlin

EURECOM, Sophia-Antipolis
27. July 2010

Outline

Introduction

Why Retransmission Protocols?

- Wireless communication networks $\rightarrow$ Errors are unavoidable in the PHY bit pipe (noise, channel fading, interference).
- Error correction at the DLC Layer $\rightarrow$ virtual error-free link.
- ARQ (Automatic Retransmission reQuest) protocols trigger retransmission of erroneous messages (per hop/ end-to-end).

How does it work?

1. An error is detected at the receiver.
2. A binary ACK/NACK feedback link informs the transmitter.
3. If NACK: transmitter repeats effort.
4. If ACK: go to next packet.

Types of ARQ Protocols

- Stop-and-Wait: Transmitter idle while waiting for feedback.
- Go-Back-N: N packets held at sender equal to round-trip delay. If error, restart from erroneous packet.
- Selective-Repeat: Window of maximum number of packets sent without ACK. Errors retransmitted later (buffering, overhead cost).

Retransmissions are only activated when necessary $\rightarrow$
System throughput and delay improved compared to conservative FEC coding schemes.

Hybrid ARQ

Combination of FEC and Retransmissions to correct errors $\rightarrow$ more efficient and complex Hybrid ARQ protocols.

- Type-I H-ARQ: Vary the coding complexity per retransmission $\rightarrow$ Delay-Throughput Tradeoff.
- Type-II H-ARQ: Store erroneous efforts and combine to increase success probability step-wise (incremental code redundancy/ maximum ratio combining).

Cross-Layer Design

- Standard Cross-Layer Design combines information from the PHY and the MAC layer [YehCohen'03], [Neely et al.'03], [BerryGallager'02], [Goyal et al.'08], [BocheWiczanowski'04], [ZhouWunder'09], [Borkar et al.'05].
- The transmission rate is adapted to channel, buffer and traffic

$$\mu(h,u,\alpha) = f(h,\rho(h,u,\alpha)) \overset{(e.g.)}{=} \log(1+\rho(h,u,\alpha)\cdot h)$$

Cross-Layer Framework for ARQ

ARQ Cross-Layer Design $\rightarrow$ goodput is controlled.

$$g\left(h,\mathcal{I}\right) \quad = \quad \mu\left(h,\mathcal{I}\right)\cdot q\left(h,\mathcal{I},\rho\left(h,\mathcal{I}\right),\mu\left(h,\mathcal{I}\right)\right)$$

- Uses additional higher layer info $\mathcal{I}$ (buffer, arrivals, ACK/NACK, retransmission index, % dropping, contention).

Channel Knowledge

- Important issue: Availability of channel knowledge together with ACK/NACK [Uhlemann et al.'03], [Qui et al.'99], [Love et al.'08], [Tuninetti'07], [ZhangWasserman'02].
- Current thesis does **NOT** consider channel knowledge (outages) [BetteshShamai'00,'01,'06], [CaireTuninetti'01], [Huang et al.'05].

An ARQ protocol in isolation

Goodput Measures

- Previous definition of goodput ([AhmedBaraniuk'03]) $\rightarrow$ results from Renewal Theory.

$$\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t} R(\tau) = \frac{\mu}{\mathbb{E}[N_{ACK}]} = \mu \cdot q$$

- Condition for validity: The inter-renewal intervals are IID.
- Alternatively observe the process up to r-th success and assume it is IID only up to this point.

Short-Term Measures

- **Theorem 1**
  - Goodput up to first ACK

$$g_{s-t} = \mathbb{E}\left[\frac{\mu}{N_{ACK}}\right] = \mu\frac{q}{1-q}\log\left(\frac{1}{q}\right).$$

  - Up to r successful messages

$$g_{s-t}^{(r)} = \mu \cdot \mathbb{E}\left[\frac{r}{N_{ACK}^{(r)}}\right]$$

  - As $r \to \infty$

$$\lim_{r\to\infty} g_{s-t}^{(r)} = \mu \cdot q \quad (\leq g_{s-t})$$

Varying Success per Retransmission

- Fixed control strategy per retransmission $q(Z_t)$.
  Transition probability matrix

$$\mathbf{P}_{\{Z_t\}} = \begin{pmatrix} q_1 & p_1 & 0 & 0 & 0 & \cdots \\ q_2 & 0 & p_2 & 0 & 0 & \cdots \\ q_3 & 0 & 0 & p_3 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$

- Applies generally to ARQ protocols including Hybrid. E.g. INR

$$p_k = \frac{\mathbb{P}\left(\sum_{j=1}^{k} \log\left(1+\rho_j h_j\right) < \mu\right)}{\mathbb{P}\left(\sum_{j=1}^{k-1} \log\left(1+\rho_j h_j\right) < \mu\right)}$$

- [CaireTuninetti'01], [Badia et al.'08]

Reliable Protocols

- **Definition**

  Instantaneous system goodput after an ACK with μ constant and n inter-renewal interval

$$g_{inst} := \frac{\mu}{n}$$

An ARQ protocol is called reliable if and only if there exists a strictly positive goodput $\tilde{g} > 0$, such that

$$\mathbb{P}\left\{g_{inst} \leq \tilde{g}\right\} = 0$$

Conditions for Reliability

- **Theorem 2**
  An ARQ protocol is reliable if and only if the chain $\{Z_t\}$ is ergodic.
- **Theorem 3**
  The following two sufficient conditions hold:
  - If $\lim_{k\to\infty} \sup\left(1 - q_k \cdot k\right) < 0$ then the chain is ergodic.
  - If $\lim_{k\to\infty} \inf\left(1 - q_k \cdot k\right) > 0$ then the chain is non-ergodic.

Truncation

- Reliability considers infinite ARQ chain $\rightarrow$ unrealistic.
- Truncated Protocols $\rightarrow$ avoid high delay at the cost of dropping.
- Consider a reward-cost process

$$Y_n = \sum_{k=1}^{n} R_k \cdot \mathbf{1}_{\{ACK \ in \ k\}} - \sum_{k=1}^{n} D_k - \delta_n \cdot \mathbf{1}_{\{no \ ACK\}}$$

- Find truncation step n^* such that

$$\mathbb{E}\left[Y_{n^*}\right] = V := \sup_{\tau \in \mathcal{C}, 1 \leq \tau < \infty} \mathbb{E}\left[Y_\tau\right].$$

Optimal Stopping

- The solution follows the Principle of Optimality

$$n^* := \min\{n : Y_n = V_n,\ n \in \mathbb{N}_+\}, \quad V_n = \max(Y_n, \mathbb{E}[V_{n+1}|\mathcal{F}_n]).$$

- **Theorem 4** (Optimal Stopping Rule)
  Continue retransmissions until:
  - Either An ACK is received
  - Or The inequality

$$q_n \quad \leq \quad \frac{D_n - \delta_{n-1} + \delta_n}{R_n + \delta_n}$$

  is satisfied for the first time.

A Single Queue incorporating ARQ in Service

Delay-Dropping Tradeoffs

- [BerryGallager'02], [BetteshShamai'06], [Goyal et al.'08]
- Find the optimal policy to drop depending on queue length
- Aim: minimize the long-term average cost

$$\limsup_{N\to\infty} \frac{1}{N}\mathbb{E}_{\pi}\left[\sum_{n=1}^{N} u_n + \delta A_n \left(1 - q\left(Z_n\right)\right)\right]$$

- $A_n = 0$ continues retransmissions
- $A_n = 1$ breaks the cycle (drops).

Structural Properties

- **Theorem 5** (Threshold in k-axis - Retransmission Effort)
  There exists a threshold state $\left(l, \hat{k}_l\right)$ for each queue length l, s.t.
  - drop is optimal for $k \geq \hat{k}_l$
  - continue is optimal for $k < \hat{k}_l$
- **Theorem 6** (Threshold in l-axis - Queue Length)
  There exists a threshold queue length $l^{th}_{\pi^*}$, s.t.
  - for $l \geq l^{th}_{\pi^*}$, $\forall k$, drop is always optimal.

$$l^{th}_{\pi^*} \leq \delta \max_k q_{k+1} + (1 - \lambda)$$

- Let queue evolve with one-shot service after $l^{th}_{\pi^*}$ ([Neely'09]).

Downlink of a cellular system with ARQ

Downlink ARQ Model

- [TassiulasEphremides'93], [Neely et al.'03], [BocheWiczanowski'04]
- Base Station keeps N queues + power controlled ARQ - one per AT.
- Total power constraint $\rho_1 + \ldots + \rho_N \leq P_{tot}$.
- Interference is considered. Errors occur due to outages.

$$q_n\left(\vec{\rho}\left(\mathcal{I}\right), \mu_n\right) = \mathbb{P}\left(SINR_n\left(\vec{\rho}\left(\mathcal{I}\right)\right) \geq e^{\mu_n} - 1\right)$$

- Success probability controlled by power vector.
- Use higher layer Information $\mathcal{I}$.

Stability Region and Power Policy

- **Theorem 7**
  If the system of N queues is overflow stable under some policy $\rightarrow$ the input rate vector $\vec{\lambda} \in \Lambda_D$:

$$\Lambda_D = \textbf{co} \bigcup_{\vec{\rho} \in \mathcal{P}(P_{tot})} \left\{ \vec{\lambda} : \forall n, \lambda_n \leq \mu_n q_n(\vec{\rho}) \right\}$$

- **Theorem 8**
  The system of N queues is overflow stable under some policy $\pi^* \in \Pi$, if the input rates belong to the above region. The policy π^*:

$$\vec{\rho}^*(t) = \arg\max_{\vec{\rho} \in \mathcal{P}} \sum_{n=1}^{N} u_n(t)\, q_n(\vec{\rho})$$

Stability and Power Control in MANETs with per hop ARQ

MANET with per hop ARQ

- [TassiulasEphremides'92], [Neely et al.'08], [Chen et al.'06]
- Wireless Network with N nodes and $N \cdot (N-1)$ wireless links.
- Data flows enter at source nodes and removed at destinations.
- Each node chooses rate and power to transmit to neighboring links.
- Causes interference to neighboring nodes.
- Holds one queue per commodity flow.
- The goodput transmitted per link equals

$$g_l\left(\vec{\rho}, \mu_l\right) := \mu_l \cdot q_l\left(\vec{\rho}, \mu_l\right)$$

Stability Region

- **Theorem 9**
  The capacity region Λ is the set of all $\vec{\lambda} = (\lambda_1, \ldots, \lambda_S)$ s.t. there exist multicommodity goodput flow variables $\{g_l^d\}$, satisfying
  - $g_l^d \geq 0$
  - $\sum_{l:e(l)=n} g_l^d + \lambda_n^d \leq \sum_{k:b(k)=n} g_k^d$
  - $\vec{g} \in \Gamma$.

$$\Gamma = \mathbf{co} \bigcup_{\vec{\rho} \in \mathcal{P}} \left\{ \vec{g} : \forall l, \; g_l \leq \max_{\mu_l \in \mathcal{M}} \mu_l \cdot q_l\left(\vec{\rho}, \mu_l\right) \right\}$$

Network Utility Optimization

- Each flow is associated with a utility function. Solve the NUMP:

$$\begin{array}{ll} \mathbf{max}_{\lambda_s \geq 0,\ g_l^d \geq 0} & \sum_{s \in \mathcal{S}} U_s\left(\lambda_s\right) \\ \mathbf{subject\ to} & \sum_{l: e(l)=n} g_l^d + \lambda_n^d \leq \sum_{k: b(k)=n} g_k^d \ \forall n, d \\ & \vec{g} \in \Gamma \end{array}$$

Decomposition

1. The input rate control problem

$$\sum_{s \in \mathcal{S}} \max_{\lambda_s \geq 0} \left\{ U_s\left(\lambda_s\right) - \nu_s \lambda_s \right\}$$

2. The scheduling problem (w_l is the back-pressure weight)

$$\max_{\vec{g} \in \Gamma} \sum_{l} w_l \cdot g_l$$

The Scheduling Problem

- Fix rates & each node chooses a single end-node per hop.
  $\rightarrow$The problem is relaxed in a pure power control problem

$$\max_{\vec{\rho}} \sum_{n} w_n \cdot \mu_n q_n \left(\vec{\rho}, \mu_n\right)$$

Distributed Pricing Algorithm

- Algorithm solves a supermodular game with N power players

$$\max_{\rho_n} \quad \omega_n \mu_n \log\left(q_n\left(\rho_n, \vec{\rho}^*_{-n}, \mu_n\right)\right) + \rho_n \sum_{m \neq n} \hat{\pi}_{m,n}$$

and $N \cdot (N-1)$ price players ([Huang et al.'05])

$$\hat{\pi}_{m,n} = \omega_m \cdot \mu_m \frac{\partial q_m\left(\vec{\rho}, \mu_m\right)}{\partial \rho_n} \frac{1}{q_n\left(\vec{\rho}, \mu_n\right)}$$

- Algorithm works for the family of supermodular success functions.

Possible future Extensions

Extensions of the Work

- Routing in networks with NACKs (per hop / end-to-end) as congestion indicator.
- Use of relays in a network to transmit more than one copy of a message through a different route.
- Utilization of the ACK/NACK feedback to learn the unknown channel fading.
- Control algorithms to max throughput / min delay in ALOHA (variation of the ARQ principle).
- ...

FIN