Network Coding (NC)

CITHN2002 – Summer 2024

Prof. Dr.-Ing. Stephan Günther

Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

Chapter 3: Link quality and metrics

Link quality estimation

- Exponentially weighted moving average (EWMA)
- Mean-EWMA (M-EWMA)
- Window-Mean-EWMA (WM-EWMA)
- Rate-adaptive link quality estimation (RALQ)
- Comparison

Metrics

- Hop count
- Estimated transmission count (ETX) [3]
- Estimated optimal transmission count (EOTX) [1]
- ETX metric for a single link
- Quality-aware redundancy scheme

Bibliography

Chapter 3: Link quality and metrics

Link quality estimation

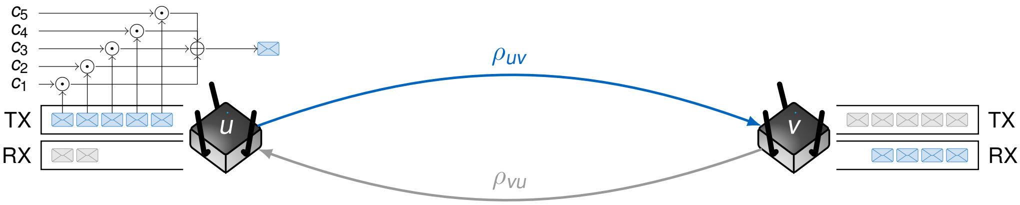
- Exponentially weighted moving average (EWMA)
- Mean-EWMA (M-EWMA)
- Window-Mean-EWMA (WM-EWMA)
- Rate-adaptive link quality estimation (RALQ)
- Comparison

Metrics

Bibliography

Link quality estimation

We consider a two-node lossy coded packet network with asymmetric link qualities $0 \leq \rho_{uv}, \rho_{vu} \leq 1$:



- Given link qualities, nodes can proactively transmit redundancy to compensate losses.
- Even without network coding, metrics may be based on the actual link quality.

Problem: How to reliably estimate the link qualities ρ_{uv} and ρ_{vu}?

- ρ is time-variant
- node u cannot measure ρ_{uv} directly, but $v \in N(u)$ can do that (programming exercise)

Link quality estimation

Assumptions:

- Packets transmitted by some node u carry a (per-node) sequence number $s \in \{0, 1, \dots, s_{\max}\}$ that is incremented by 1 per packet.
- Each neighboring node $v \in N(u)$ keeps track of the last sequence number observed from u.
- We denote by s_k the sequence number of the k-th packet transmitted by u that is received by a specific $v \in N(u)$[1].
- With each packet a specific neighbor $v \in N(u)$ overhears, the amount of packets transmitted by u but missed by v is given as

  $$z_k = (s_k - s_{k-1} - 1) \bmod (s_{\max} + 1),$$

  where the modulo operations takes care of wrap arounds of sequence numbers.

[1] Note that for a given transmitter u both k (number of received packets from u) and the corresponding sequence number s_k depend on the neighbor $v \in N(u)$ that we consider.

[2] Actually, there is nothing wrong. There are different definitions of modulo operations over $\mathbb{Z}$.

Link quality estimation

Assumptions:

- Packets transmitted by some node u carry a (per-node) sequence number $s \in \{0, 1, \dots, s_{\max}\}$ that is incremented by 1 per packet.
- Each neighboring node $v \in N(u)$ keeps track of the last sequence number observed from u.
- We denote by s_k the sequence number of the k-th packet transmitted by u that is received by a specific $v \in N(u)$[1].
- With each packet a specific neighbor $v \in N(u)$ overhears, the amount of packets transmitted by u but missed by v is given as

$$z_k = (s_k - s_{k-1} - 1) \bmod (s_{\max} + 1),$$

  where the modulo operations takes care of wrap arounds of sequence numbers.

Warning:

- The `%` operator in C yields the wrong[2] result.
- To avoid that, we can either
  - shift the sequence numbers by $s_{\max} + 1$, i. e., $z_k = (s_k - s_{k-1} + s_{\max}) \bmod (s_{\max} + 1)$, or
  - use unsigned integers and rely on the automatic wrap around.

---

[1] Note that for a given transmitter u both k (number of received packets from u) and the corresponding sequence number s_k depend on the neighbor $v \in N(u)$ that we consider.

[2] Actually, there is nothing wrong. There are different definitions of modulo operations over $\mathbb{Z}$.

Link quality estimation

- When node v overhears the k-th packet from one of its neighbors $u \in N(v)$, it can calculate the total number of successfully received and lost packets

$$p_k = p_{k-1} + 1 \quad \text{and} \quad q_k = q_{k-1} + z_k .$$

- The success probability (link quality) can then be updated to

$$\overline{\rho}_{uv}[k] = \frac{p_k}{p_k + q_k} .$$

- If ρ_{uv} is time-invariant, we obviously have $\overline{\rho}_{uv}[k] \to \rho_{uv}$ for $k \to \infty$.

Link quality estimation

- When node v overhears the k-th packet from one of its neighbors $u \in N(v)$, it can calculate the total number of successfully received and lost packets

$$p_k = p_{k-1} + 1 \quad \text{and} \quad q_k = q_{k-1} + z_k.$$

- The success probability (link quality) can then be updated to

$$\overline{\rho}_{uv}[k] = \frac{p_k}{p_k + q_k}.$$

- If ρ_{uv} is time-invariant, we obviously have $\overline{\rho}_{uv}[k] \to \rho_{uv}$ for $k \to \infty$.

What if ρ_{uv} is time-variant?

- $\overline{\rho}_{uv}[k]$ badly reflects variations in time.
- Short changes would not have much influence.

Exponentially weighted moving average (EWMA)

Idea

1. Increase the link quality for each packet successfully overheard.
2. Decrease the link quality for each packet missed.

For each event (success or loss), we update the old estimator according to

$$\hat{\rho}[n] = \mathrm{EWMA}[n] = \mathrm{EWMA}[n-1]\alpha + (1-\alpha)E[n], \quad \forall n \geq 1, 0 \leq \alpha \leq 1,$$

where $E[n]$ is 1 if the n-th packet transmitted was also received, and 0 otherwise. Note that n denotes the number of packets transmitted, *not* the number of packets received by some node.

Exponentially weighted moving average (EWMA)

Idea

1. Increase the link quality for each packet successfully overheard.
2. Decrease the link quality for each packet missed.

For each event (success or loss), we update the old estimator according to

$$\hat{\rho}[n] = \mathrm{EWMA}[n] = \mathrm{EWMA}[n-1]\alpha + (1-\alpha)E[n], \quad \forall n \geq 1, 0 \leq \alpha \leq 1,$$

where $E[n]$ is 1 if the n-th packet transmitted was also received, and 0 otherwise. Note that n denotes the number of packets transmitted, *not* the number of packets received by some node.

Problem

- We (from the perspective of a receiving node) can in general not differentiate between packet loss and no transmission in the first place.
- We see that only after receiving a packet.

The EWMA is therefore updated after receiving the k-th packet and determining the number of lost packets z_k according to

$$\hat{\rho}[k] = \mathrm{EWMA}[k] = \mathrm{EWMA}[k-1]\alpha^{z_k+1} + (1-\alpha), \quad \forall k \geq 1, 0 \leq \alpha \leq 1.$$

Exponentially weighted moving average (EWMA)

Idea

1. Increase the link quality for each packet successfully overheard.
2. Decrease the link quality for each packet missed.

For each event (success or loss), we update the old estimator according to

$$\hat{\rho}[n] = \mathrm{EWMA}[n] = \mathrm{EWMA}[n-1]\alpha + (1-\alpha)E[n], \quad \forall n \geq 1, 0 \leq \alpha \leq 1,$$

where $E[n]$ is 1 if the n-th packet transmitted was also received, and 0 otherwise. Note that n denotes the number of packets transmitted, *not* the number of packets received by some node.

Problem

- We (from the perspective of a receiving node) can in general not differentiate between packet loss and no transmission in the first place.
- We see that only after receiving a packet.

The EWMA is therefore updated after receiving the k-th packet and determining the number of lost packets z_k according to

$$\hat{\rho}[k] = \mathrm{EWMA}[k] = \mathrm{EWMA}[k-1]\alpha^{z_k+1} + (1-\alpha), \quad \forall k \geq 1, 0 \leq \alpha \leq 1.$$

Properties

- Tends to oscillations
- Chosing α depends on the packet rate

Mean-EWMA (M-EWMA)

Idea: similar to the ordinary EWMA, but updates are done only after receiving a fixed amount of $\delta \geq 1$ packets:

$$\bar{z}[k] = \begin{cases} z_k & \text{for } k \bmod \delta = 1, \\ \bar{z}[k-1] + z_k & \text{otherwise,} \end{cases}$$

$$\text{M-EWMA}[k] = \begin{cases} \text{M-EWMA}[k-1]\alpha + (1-\alpha)\dfrac{\delta}{\delta + \bar{z}[k]} & \text{for } k \bmod \delta = 0,\, 0 \leq \alpha \leq 1, \\ \text{M-EWMA}[k-1] & \text{otherwise.} \end{cases}$$

For $k = 1$ the M-EWMA reduces to the ordinary EWMA estimator.

Mean-EWMA (M-EWMA)

Idea: similar to the ordinary EWMA, but updates are done only after receiving a fixed amount of $\delta \geq 1$ packets:

$$\overline{z}[k] = \begin{cases} z_k & \text{for } k \bmod \delta = 1, \\ \overline{z}[k-1] + z_k & \text{otherwise,} \end{cases}$$

$$\text{M-EWMA}[k] = \begin{cases} \text{M-EWMA}[k-1]\alpha + (1-\alpha)\dfrac{\delta}{\delta + \overline{z}[k]} & \text{for } k \bmod \delta = 0, 0 \leq \alpha \leq 1, \\ \text{M-EWMA}[k-1] & \text{otherwise.} \end{cases}$$

For $k = 1$ the M-EWMA reduces to the ordinary EWMA estimator.

Properties

- Tends less to oscillation (effectively a low-pass filter for the EWMA)
- Update intervals are larger
- Both δ and α depend on the packet rate

Window-Mean-EWMA (WM-EWMA)

Idea: similar to M-EWMA, but update the estimator based on fixed time intervals $\Delta t > 0$.

- Determine the number of received and missed packets $p[\tau]$ and $z[\tau]$ within the time interval $\tau = [t - \Delta t, t)$.
- Afterwards, update the estimator according to

$$\text{WM-EWMA}[\tau] = \text{WM-EWMA}[\tau - \Delta t]\alpha + (1 - \alpha)\frac{p[\tau]}{p[\tau] + z[\tau]}, \quad \forall \tau > 0, 0 \leq \alpha \leq 1.$$

Window-Mean-EWMA (WM-EWMA)

Idea: similar to M-EWMA, but update the estimator based on fixed time intervals $\Delta t > 0$.

- Determine the number of received and missed packets $p[\tau]$ and $z[\tau]$ within the time interval $\tau = [t - \Delta t, t)$.
- Afterwards, update the estimator according to

$$\text{WM-EWMA}[\tau] = \text{WM-EWMA}[\tau - \Delta t]\alpha + (1 - \alpha)\frac{p[\tau]}{p[\tau] + z[\tau]}, \quad \forall \tau > 0, 0 \leq \alpha \leq 1.$$

Problem

What if no packets were received within a given interval τ?

Window-Mean-EWMA (WM-EWMA)

Idea: similar to M-EWMA, but update the estimator based on fixed time intervals $\Delta t > 0$.

- Determine the number of received and missed packets $p[\tau]$ and $z[\tau]$ within the time interval $\tau = [t - \Delta t, t)$.
- Afterwards, update the estimator according to

$$\text{WM-EWMA}[\tau] = \text{WM-EWMA}[\tau - \Delta t]\alpha + (1 - \alpha)\frac{p[\tau]}{p[\tau] + z[\tau]}, \quad \forall \tau > 0, 0 \leq \alpha \leq 1.$$

Problem

What if no packets were received within a given interval τ?

Properties

- Quite stable
- Requires a minimum expected packet rate
- Implementation of time intervals is tricky

Window-Mean-EWMA (WM-EWMA)

How to choose α?

- If the update is triggered by events, e. g. every successfully overheard packet or after overhearing a certain number of packets, α depends on the packet rate:
  - For high packet rates, $\alpha \approx 0.98$ might be a good choice.
  - But that leads to very slow adaptations if the packet rate significantly drops.
- Generally, updates based on regular time intervals are preferable.

How to choose Δt?

- Within a time interval $[t_i - \Delta t, t_i)$ there should be a reasonable number of packets.
- Given a beacon interval of 0.2 ms, $\Delta t = 2$ s might a meaningful choice.
- Note that α must be adapted accordingly to give new estimates sufficient weight.

Rate-adaptive link quality estimation (RALQ)

In order to work properly, all approaches so far either

- tend to oscillation,
- depend on the packet rate, or
- require a minimum packet rate.

Can we do better?

Rate-adaptive link quality estimation (RALQ)

In order to work properly, all approaches so far either

- tend to oscillation,
- depend on the packet rate, or
- require a minimum packet rate.

Can we do better?

- It is easy to maintain all-time counters for received and missed packets, namely p_k and q_k, respectively.
- However, the long-term link quality $\overline{\rho}_k = p_k/(p_k + q_k)$ is not time-variant.

Rate-adaptive link quality estimation (RALQ)

In order to work properly, all approaches so far either

- tend to oscillation,
- depend on the packet rate, or
- require a minimum packet rate.

Can we do better?

- It is easy to maintain all-time counters for received and missed packets, namely p_k and q_k, respectively.
- However, the long-term link quality $\overline{\rho}_k = p_k / (p_k + q_k)$ is not time-variant.

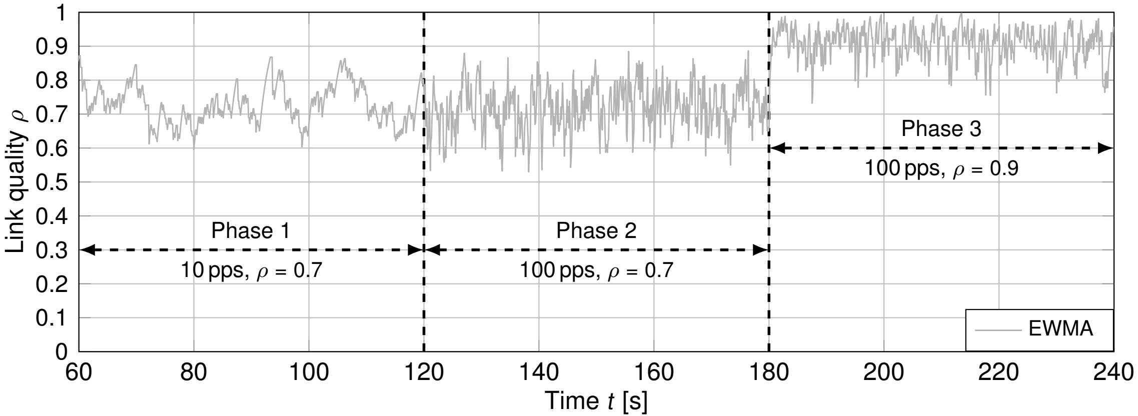
Idea: keep only a limited history of packet events, i. e., make p_k and q_k time-dependent.

- Choose a time constant $\tau > 0$ that determines the speed of decay.
- Given the point t_k in time when the k-th packet was received, weigh packet counters

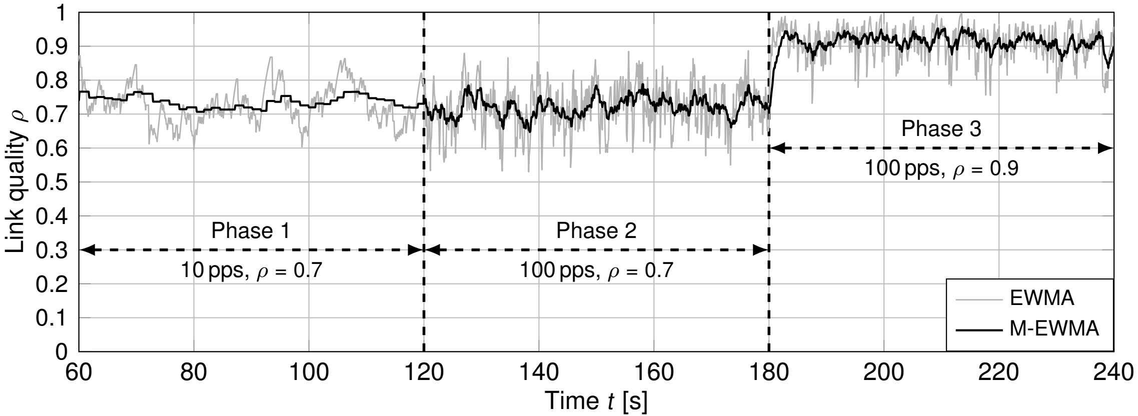
$$p_k(t) = (p_{k-1}(t_k) + 1)e^{-\tau(t-t_k)} \qquad q_k(t) = (q_{k-1}(t_k) + z_k)e^{-\tau(t-t_k)}, \quad \forall k \geq 1, t \geq t_1.$$

$\Rightarrow$ New samples after a communication pause have more influence.

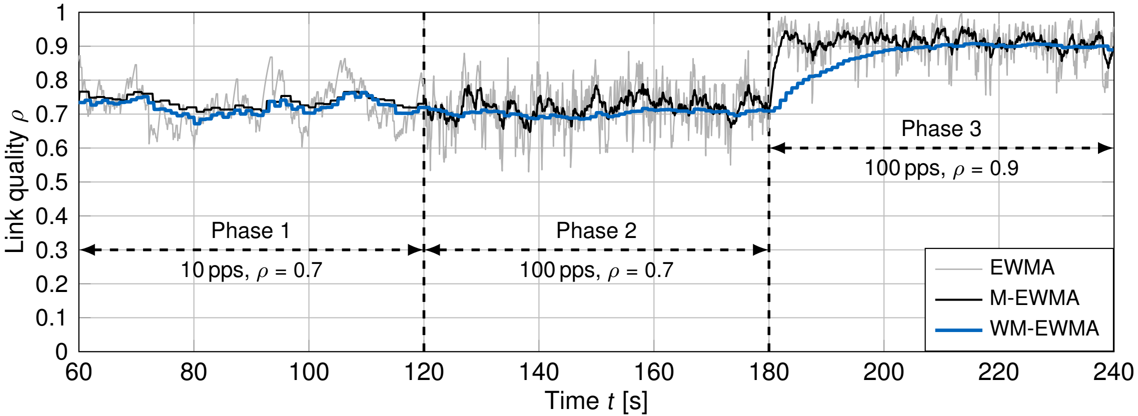
Comparison



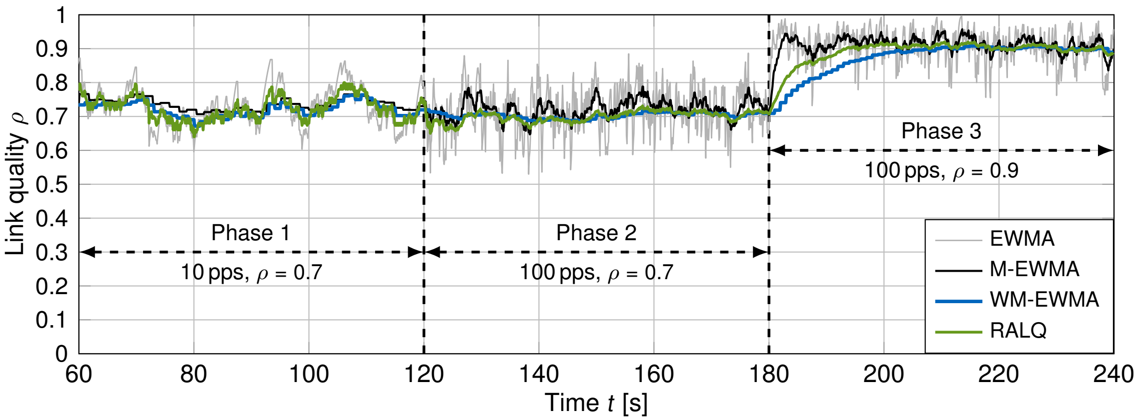
Characteristics	EWMA	M-EWMA	WM-EWMA	RALQ
Independent of packet rate	✕			
No minimum packet rate required	O			
Robust against oscillation	✕			
Speed of convergence	O			
Efficient implementation	✓			



Characteristics	EWMA	M-EWMA	WM-EWMA	RALQ
Independent of packet rate	✗	✗		
No minimum packet rate required	O	O		
Robust against oscillation	✗	O		
Speed of convergence	O	O		
Efficient implementation	✓	✓		



Characteristics	EWMA	M-EWMA	WM-EWMA	RALQ
Independent of packet rate	✗	✗	✓	
No minimum packet rate required	○	○	✗	
Robust against oscillation	✗	○	✓	
Speed of convergence	○	○	○	
Efficient implementation	✓	✓	○	



Characteristics	EWMA	M-EWMA	WM-EWMA	RALQ
Independent of packet rate	✗	✗	✓	✓
No minimum packet rate required	○	○	✗	✓
Robust against oscillation	✗	○	✓	✓
Speed of convergence	○	○	○	✓
Efficient implementation	✓	✓	○	✓

Preliminaries and assumptions

- We denote a link between nodes i and j as ij.
- A metric defines the costs[3] c_{ij} for such a link.
- We denote a route as $r_{ij,jk,kl\dots}$.
- A metric can be additive, multiplicative, convex, e. g. :

$$c_{r_{ij,jk}} = c_{ij} + c_{jk} \qquad \textit{(additive)}$$

$$\text{or} \quad c_{r_{ij,jk}} = \max\{c_{ij}, c_{jk}\} \qquad \textit{(convex)}$$

- We denote the set of routes between two nodes i and j as R_{ij}.
- The distance d_{ij} between two nodes i and j are the costs of the best route $r \in R_{ij}$, i. e.,

$$d_{ij} = \min\{c_r \mid r \in R_{ij}\}.$$

[3] In general, lower costs means a better route.

Hop count

A route r is rated by its hop count, i. e., the number of links between source and destination.

- The costs of a link ij are always $c_{ij} = 1$.
- The hop count metric is additive.
- The distance d_{ij} between two nodes i, j is the number of hops along the shortest path between i, j.

Hop count

A route r is rated by its hop count, i. e., the number of links between source and destination.

- The costs of a link ij are always $c_{ij} = 1$.
- The hop count metric is additive.
- The distance d_{ij} between two nodes i, j is the number of hops along the shortest path between i, j.

A common extension is to allow for arbitrary, postive link costs (weights) $c_{ij} \geq 1$.

- The metric is no longer solely hop count.
- Links may be weighted according to their speed, delay, or success probability.

Estimated transmission count (ETX) [3]

Assumptions

- We assume a network with hop-to-hop acknowledgements.
- A route is rated by the estimated number of transmission that are necessary to successfully transfer a packet.
- A retransmission is made if
  - the frame is not received or
  - the acknowledgment is not received.

The ETX for a single link *ij* is therefore

$$\mathrm{ETX}_{ij} = c_{ij} = \frac{1}{(1-\varepsilon_{ij})\cdot(1-\varepsilon_{ji})}.$$

- The ETX metric is additive.
- The distance d_{ij} between two nodes i, j is the estimated number of transmissions along the best path between i and j.

Estimated transmission count (ETX) [3]

Variations of the ETX metric

No acknowledgements

- We need only the unidirectional erasure rates:

$$\mathrm{ETX}_{ij} = c_{ij} = \frac{1}{1 - \varepsilon_{ij}}$$

- We assume each received frame triggers a forwarding transmission.
- The number of frames the source has to transmit is the multiplicative ETX metric.
- But the forwarders also transmit frames.
- Therefore, the metric is multiplicative and additive, e. g.:

$$\mathrm{ETX}_{r_{ij,jk,kl}} = c_{ij} c_{jk} c_{kl} + c_{jk} c_{kl} + c_{kl}$$

MORE [2]

- Has no hop-to-hop acknowledgements for data frames, but injects redundancy
- Therefore, uses only unidirectional erasure rates (see above)
- This metric is just additive

Estimated optimal transmission count (EOTX) [1]

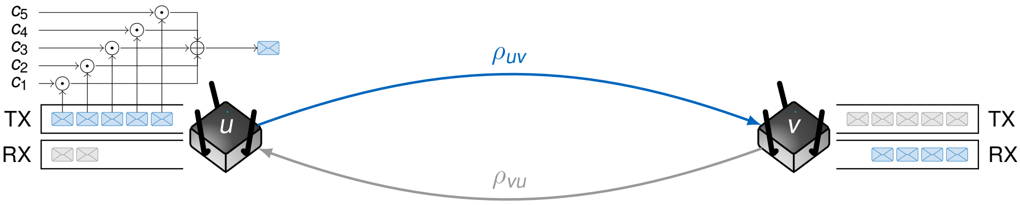
If nodes do opportunistic overhearing, i. e., may accept and forward data for which a node was not chosen as next hop by the transmitter, even suboptimal routes may be used:

- The EOTX for a single link ij is $\mathrm{EOTX}_{ij} = \mathrm{ETX}_{ij}$.
- The distance d_{ij} between two nodes i, j is defined as

$$d_{ij} = \frac{1}{1 - \prod_{k<i} \varepsilon_{ik}} + \sum_{k<i} d_{kj}(1 - \varepsilon_{ik}) \prod_{l<k} \varepsilon_{il}, \text{ where} \quad (1)$$

  the operator $k < i$ means that k is closer to the destination than i and $\epsilon_{ij} = 1 - \rho_{ij}$ denotes the erasure probability on the link ij.
  - The first summand of (1) is the expected number of packets that i has to transmit s. t. at least one node closer to j receives the packet.
  - The second summand represents the total amount of packets all nodes closer to j than i have to transmit (note the recursion through d_{kj}), provided that no other node l that is even closer to j has received the transmission directly from i.

We consider the two-node network depicted below:



- We assume that no acknowldgements are transmitted.
- We preclude random linear dependencies.
- Given a block of N packets, node u would thus transmit N/ρ_{uv} coded packets on average.

Question: What is the probability that for a specific block of N packets, transmission of N/ρ_{uv} is sufficient for decoding?

ETX metric for a single link

Let X denote the random variable indicating the number of coded packets received by v. Then, the probability that at least N packets were received given that n packets have been transmitted is given by

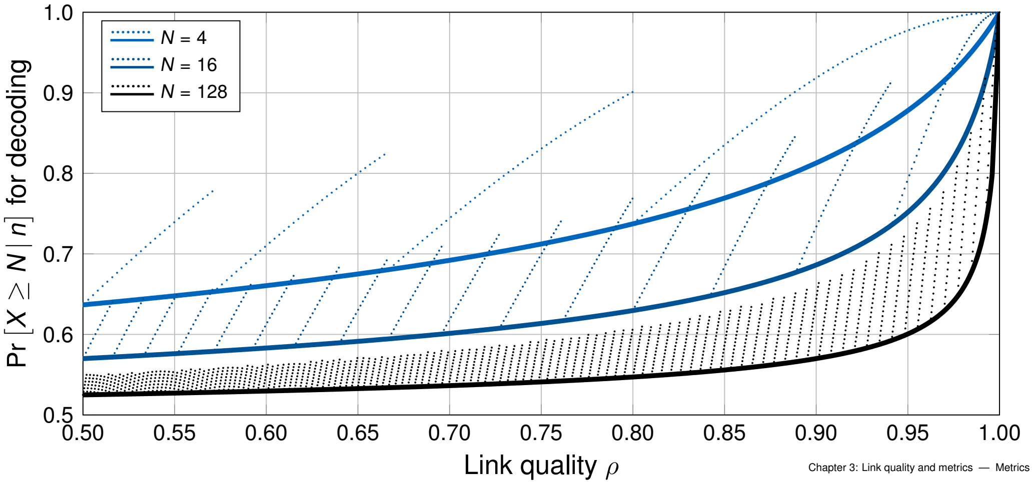
$$\Pr\left[X \geq N \mid n\right] = \begin{cases} 0 & n < N, \\ \sum_{i=N}^{n} \binom{n}{i} \rho^{i} (1-\rho)^{n-i} & \text{otherwise.} \end{cases} \quad (2)$$

Solving (2) for $n = \frac{N}{\rho}$ and $n' = \left\lceil \frac{N}{\rho} \right\rceil$ (since we cannot send fractions of packets) yields a surprisingly low probability ...

ETX metric for a single link

Decoding probability for a block size of $N \in \{4, 16, 128\}$ and link qualities $0.5 \leq \rho \leq 1.0$.

- Solid plots show results for $n = \frac{N}{\rho}$, i. e., fractional packets are allowed.
- Dotted plots show results for $n = \left\lceil \frac{N}{\rho} \right\rceil$, i. e., only full packets can be sent and n is thus rounded above.



Quality-aware redundancy scheme

Idea: Specify a decoding probability θ and transmit the minimum number of coded packets

$$n^* = \min_{n \geq N} \quad \text{s.t.} \ \Pr\left[X \geq N \mid n\right] \geq \theta. \tag{3}$$

Quality-aware redundancy scheme

Idea: Specify a decoding probability θ and transmit the minimum number of coded packets

$$n^* = \min_{n \geq N} \quad \text{s.t. } \Pr\left[X \geq N \mid n\right] \geq \theta. \tag{3}$$

Problem: Equation 3 does not take the reliability of our estimator for ρ into account.

Quality-aware redundancy scheme

Idea: Specify a decoding probability θ and transmit the minimum number of coded packets

$$n^* = \min_{n \geq N} \quad \text{s.t.} \; \Pr\left[X \geq N \mid n\right] \geq \theta. \tag{3}$$

Problem: Equation 3 does not take the reliability of our estimator for ρ into account.

We can do even that:

- The rate-adaptive link quality estimation is based on the history of packet losses $\boldsymbol{z}_k = [z_1, z_2, \ldots, z_k]^T$.
- We can therefore restate (3) as

$$n^* = \min_{n \geq N} \quad \text{s.t.} \; \Pr\left[X \geq N \mid n, \boldsymbol{z}_k\right] \geq \theta. \tag{4}$$

- The history $\boldsymbol{z}_k$ is implicitly available through updates of $p_k(t)$ and $q_k(t)$.
- The larger the number of samples $p_k + q_k$, the more reliable the estimate of ρ.

Quality-aware redundancy scheme

Idea: Specify a decoding probability θ and transmit the minimum number of coded packets

$$n^* = \min_{n \geq N} \quad \text{s. t. } \Pr\left[X \geq N \mid n\right] \geq \theta. \tag{3}$$

Problem: Equation 3 does not take the reliability of our estimator for ρ into account.

We can do even that:

- The rate-adaptive link quality estimation is based on the history of packet losses $\boldsymbol{z}_k = [z_1, z_2, \ldots, z_k]^T$.
- We can therefore restate (3) as

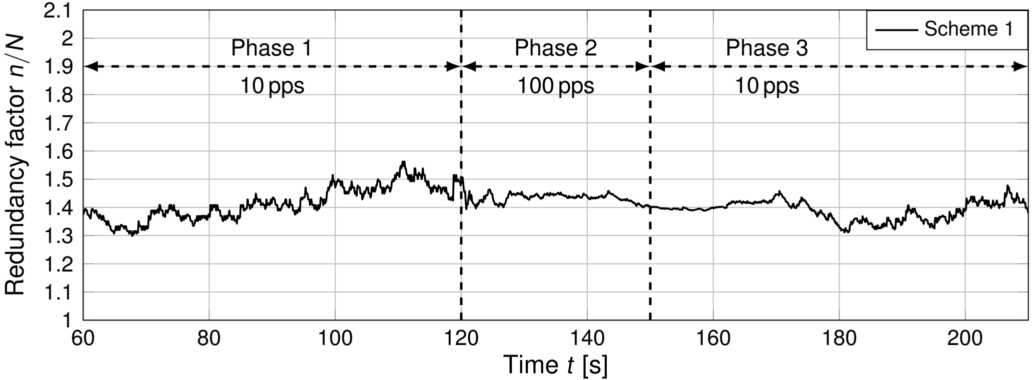
$$n^* = \min_{n \geq N} \quad \text{s. t. } \Pr\left[X \geq N \mid n, \boldsymbol{z}_k\right] \geq \theta. \tag{4}$$

- The history $\boldsymbol{z}_k$ is implicitly available through updates of $p_k(t)$ and $q_k(t)$.
- The larger the number of samples $p_k + q_k$, the more reliable the estimate of ρ.

The conditional probability from (4) for the optimization problem is given by (no proof [4])

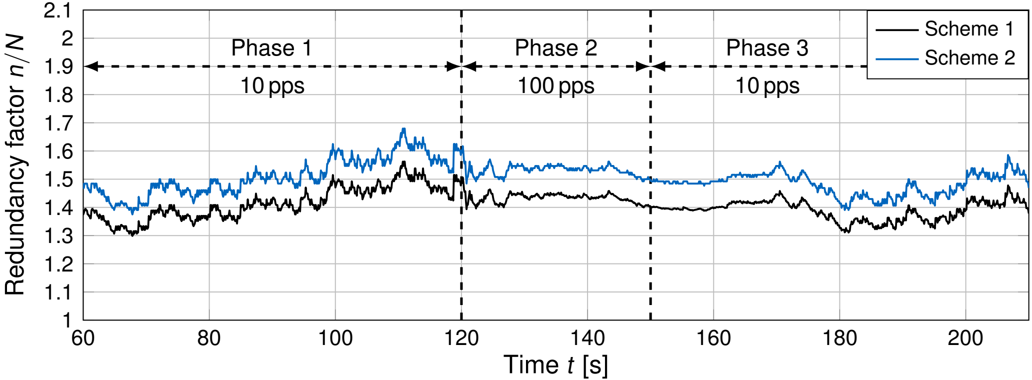
$$\Pr\left[X \geq m \mid n, \boldsymbol{z}_k\right] = 1 - \sum_{i=0}^{m-1} \prod_{j=1}^{i} \frac{p+j}{p+q+j+1} \prod_{j=i+1}^{n} \frac{j(q+j-i)}{(p+q+j+1)(j-1)}.$$

Quality-aware redundancy scheme



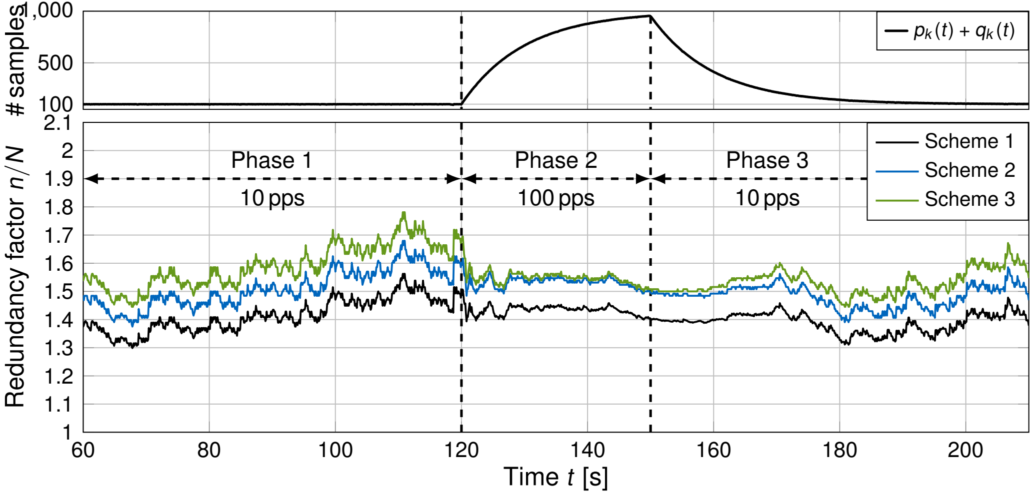
Characteristics	Scheme 1 (ETX)	Scheme 2	Scheme 3
Ensures decoding probability θ	✗		

Quality-aware redundancy scheme



Characteristics	Scheme 1 (ETX)	Scheme 2	Scheme 3
Ensures decoding probability θ	✗	✓	

Quality-aware redundancy scheme



Characteristics	Scheme 1 (ETX)	Scheme 2	Scheme 3
Ensures decoding probability θ	✗	✓	✓

Quality-aware redundancy scheme

- At the moment, this scheme only works for single links.
- To make it a fully useable metric, we have to extend it similar to the ETX and EOTX metric.

⇒ Open research question

Bibliography

[1] S. Chachulski.
Trading Structure for Randomness in Wireless Opportunistic Routing.
M.sc. thesis, Massachusetts Institute of Technology, 2007.

[2] S. Chachulski, M. Jennings, S. Katti, and D. Katabi.
Trading Structure for Randomness in Wireless Opportunistic Routing.
In *ACM SIGCOMM*, pages 169–180, 2007.

[3] D. De Couto, D. Aguayo, J. Bicket, and R. Morris.
A High-Throughput Path Metric for Multi-hop Wireless Routing.
Wireless Networks, 11(4):419–434, Jul. 2005.

[4] M. Leclaire, S. M. Günther, M. Lienen, M. J. Riemensberger, and G. Carle.
Rate Adaptive Link Quality Estimation for Coded Packet Networks.
In *IEEE International Conference on Local Computer Networks (LCN)*, Dubai, UAE, Nov. 2016.