OFDM (DMT) Bit and Power Loading for Unequal Error Protection

Werner Henkel and Khaled Hassan

School of Engineering and Science
International Universities Bremen (IUB)

11th International OFDM-Workshop, 2006

Outline

1. Motivations
   - Why Unequal Error Protection (UEP)?
   - Why UEP Physical Transport?
2. UEP: Bit-Loading
   - Previous Work
   - Proposed UEP Bit-Rate Maximization
3. Channel Model
   - Noise Environment
4. Simulation Results
   - UEP Performance: SER Analysis
   - Bit and Power Loading
5. Conclusions

Outline

1. Motivations
   - Why Unequal Error Protection (UEP)?
   - Why UEP Physical Transport?
2. UEP: Bit-Loading
   - Previous Work
   - Proposed UEP Bit-Rate Maximization
3. Channel Model
   - Noise Environment
4. Simulation Results
   - UEP Performance: SER Analysis
   - Bit and Power Loading
5. Conclusions

Realizing UEP

- Source encoders of some applications deliver data of different importance.
- The different error sensitivities of different communication devices, e.g., PDAs, laptops, $\cdots$.
- Matching the channel variations to enhance performance and throughput.

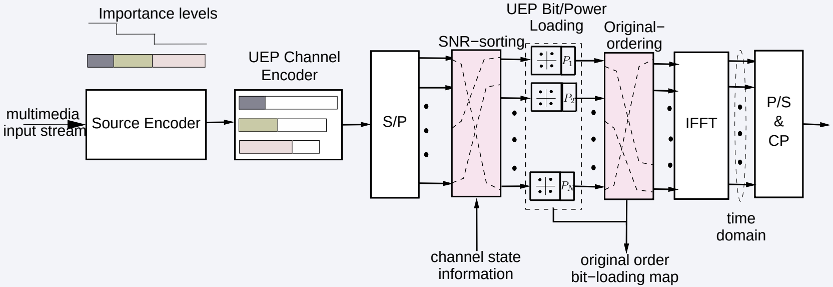
Realizing UEP

- Source encoders of some applications deliver data of different importance.
- The different error sensitivities of different communication devices, e.g., PDAs, laptops, $\cdots$.
- Matching the channel variations to enhance performance and throughput.

Realizing UEP

- Source encoders of some applications deliver data of different importance.
- The different error sensitivities of different communication devices, e.g., PDAs, laptops, $\cdots$.
- Matching the channel variations to enhance performance and throughput.

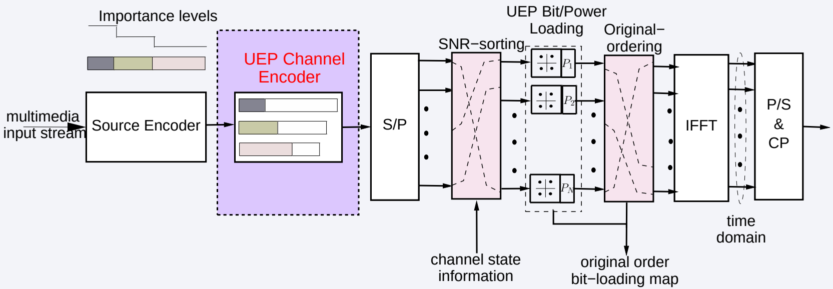
Advantages of UEP Physical Transport



Why UEP physical transport?

- Reduce effort and complexity
- Arbitrary performance steps

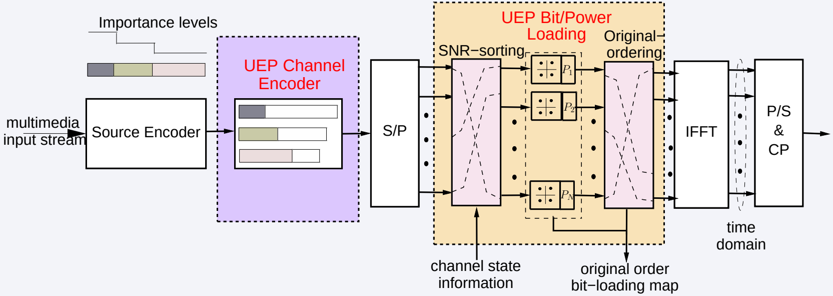
Advantages of UEP Physical Transport



Why UEP physical transport?

- Reduce effort and complexity
- Arbitrary performance steps

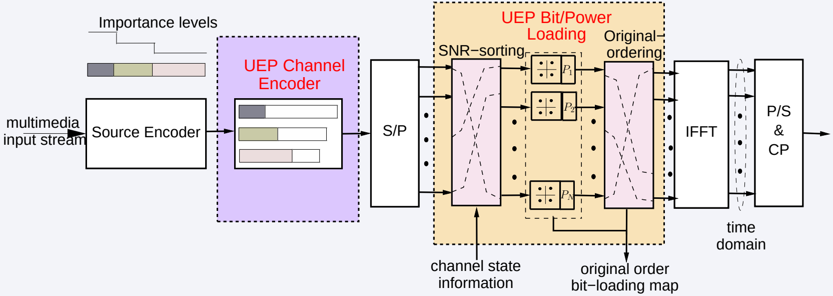
Advantages of UEP Physical Transport



Why UEP physical transport?

- Reduce effort and complexity
- Arbitrary performance steps

Advantages of UEP Physical Transport



Why UEP physical transport?

- Reduce effort and complexity
- Arbitrary performance steps

Outline

1. Motivations
   - Why Unequal Error Protection (UEP)?
   - Why UEP Physical Transport?
2. UEP: Bit-Loading
   - Previous Work
   - Proposed UEP Bit-Rate Maximization
3. Channel Model
   - Noise Environment
4. Simulation Results
   - UEP Performance: SER Analysis
   - Bit and Power Loading
5. Conclusions

Bit-Loading Algorithms

Bit-Loading solutions:

- Optimum: add bits to the locations of minimum incremental power, e.g.: Hughes-Hartogs and Campello
- Sub-optimum: based on Shannon capacity (Chow et. al.) or Lagrange-optimization (Fischer-Huber and Yu-Willson)

Bit-Loading Algorithms

Bit-Loading solutions:

- Optimum: add bits to the locations of minimum incremental power, e.g.: Hughes-Hartogs and Campello
- Sub-optimum: based on Shannon capacity (Chow et. al.) or Lagrange-optimization (Fischer-Huber and Yu-Willson)

Bit-Loading by Chow et. al.:

$$b_k = \log_2\left(1 + \frac{\text{SNR}_k}{\gamma}\right)$$

Bit-Rate Maximization Problem:

$$\max_{b \in Z}\left\{B_{\text{tot}} = \sum_{k=0}^{N-1} b_k\right\}$$

$$\text{subject to } \sum_{k=0}^{N-1} P_k(b_k) < P_T \ ,$$

Bit-Loading Algorithms

Bit-Loading solutions:

- Optimum: add bits to the locations of minimum incremental power, e.g.: Hughes-Hartogs and Campello
- Sub-optimum: based on Shannon capacity (Chow et. al.) or Lagrange-optimization (Fischer-Huber and Yu-Willson)

Bit-Loading by Chow et. al.:

$$b_k = \log_2\left(1 + \frac{\mathrm{SNR}_k}{\gamma}\right)$$

Quantization Error:

$$\begin{aligned} \hat{b}_k &= \lfloor b_k + 0.5 \rfloor_0^{b_{\max}} \\ \Delta b_k &= b_k - \hat{b}_k \end{aligned}$$

Modifications to Bit-Loading by Chow et. al.

Problem definitions

- N_g levels of protections with noise margins γ_j
- Noise margin step size $\Delta\gamma_j$
- Target-rates T_j for each class
- Over all target bit-rate B_T

Modified bit-loading:

$$\begin{aligned} b_{k,j} &= \log_2\left(1+\frac{\mathrm{SNR}_{k,j}}{\gamma_j}\right) \\ \hat{b}_{k,j} &= \lfloor b_{k,j}+0.5\rfloor_0^{b_{\max}} \\ \Delta b_{k,j} &= b_{k,j}-\hat{b}_{k,j} \end{aligned}$$

SNR-sorting

Modifications to Bit-Loading by Chow et. al.

Problem definitions

- N_g levels of protections with noise margins γ_j
- Noise margin step size $\Delta\gamma_j$
- Target-rates T_j for each class
- Over all target bit-rate B_T

Modified bit-loading:

$$\begin{aligned} b_{k,j} &= \log_2\left(1+\frac{\mathrm{SNR}_{k,j}}{\gamma_j}\right) \\ \hat{b}_{k,j} &= \lfloor b_{k,j}+0.5 \rfloor_0^{b_{\max}} \\ \Delta b_{k,j} &= b_{k,j}-\hat{b}_{k,j} \end{aligned}$$

SNR-sorting

Modifications to Bit-Loading by Chow et. al.

Problem definitions

- N_g levels of protections with noise margins γ_j
- Noise margin step size $\Delta\gamma_j$
- Target-rates T_j for each class
- Over all target bit-rate B_T

Modified bit-loading:

$$\begin{aligned} b_{k,j} &= \log_2\left(1+\frac{\text{SNR}_{k,j}}{\gamma_j}\right) \\ \hat{b}_{k,j} &= \lfloor b_{k,j}+0.5\rfloor_0^{b_{\max}} \\ \Delta b_{k,j} &= b_{k,j}-\hat{b}_{k,j} \end{aligned}$$

SNR-sorting

- SNRs have to be allocated to N_g levels.
- Allocate important data to weaker subcarriers to protect them against non-stationary noise.

Modifications to Bit-Loading by Chow et. al.

Problem definitions

- N_g levels of protections with noise margins γ_j
- Noise margin step size $\Delta\gamma_j$
- Target-rates T_j for each class
- Over all target bit-rate B_T

Modified bit-loading:

$$
\begin{aligned}
b_{k,j} &= \log_2\left(1+\frac{\text{SNR}_{k,j}}{\gamma_j}\right)\\
\hat{b}_{k,j} &= \lfloor b_{k,j}+0.5\rfloor_0^{b_{\max}}\\
\Delta b_{k,j} &= b_{k,j}-\hat{b}_{k,j}
\end{aligned}
$$

SNR-sorting

- SNRs have to be allocated to N_g levels.
- Allocate important data to weaker subcarriers to protect them against non-stationary noise.

Modifications to Bit-Loading by Chow et. al.

Problem definitions

- N_g levels of protections with noise margins γ_j
- Noise margin step size $\Delta\gamma_j$
- Target-rates T_j for each class
- Over all target bit-rate B_T

Modified bit-loading:

$$\begin{aligned} b_{k,j} &= \log_2\left(1+\frac{\mathrm{SNR}_{k,j}}{\gamma_j}\right) \\ \hat{b}_{k,j} &= \lfloor b_{k,j}+0.5\rfloor_0^{b_{\max}} \\ \Delta b_{k,j} &= b_{k,j}-\hat{b}_{k,j} \end{aligned}$$

SNR-sorting

- SNRs have to be allocated to N_g levels.
- Allocate important data to weaker subcarriers to protect them against non-stationary noise.

Modifications to Bit-Loading by Chow et. al.

Problem definitions

- N_g levels of protections with noise margins γ_j
- Noise margin step size $\Delta\gamma_j$
- Target-rates T_j for each class
- Over all target bit-rate B_T

Modified bit-loading:

$$\begin{aligned} b_{k,j} &= \log_2\left(1+\frac{\mathrm{SNR}_{k,j}}{\gamma_j}\right) \\ \hat{b}_{k,j} &= \lfloor b_{k,j}+0.5\rfloor_0^{b_{\max}} \\ \Delta b_{k,j} &= b_{k,j}-\hat{b}_{k,j} \end{aligned}$$

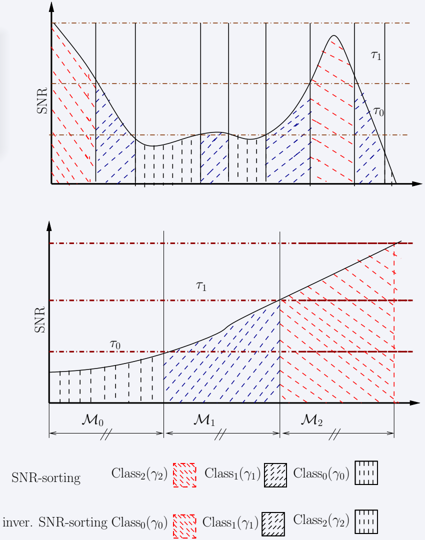
SNR-sorting

- SNRs have to be allocated to N_g levels.
- Allocate important data to weaker subcarriers to protect them against non-stationary noise.

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}_{ej}}$, and bit-loading

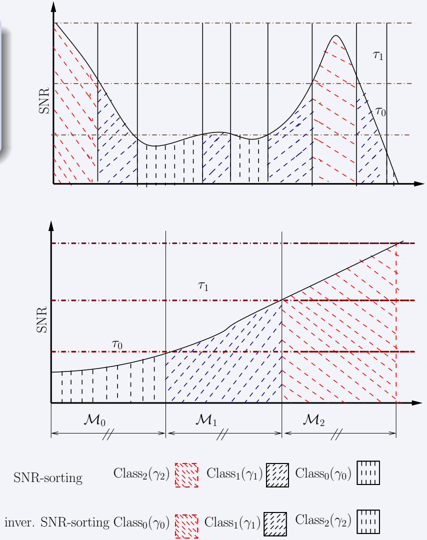
- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j \cdot \Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively,
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old} \cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}_{ej}}$



UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}_{ej}}$, and bit-loading

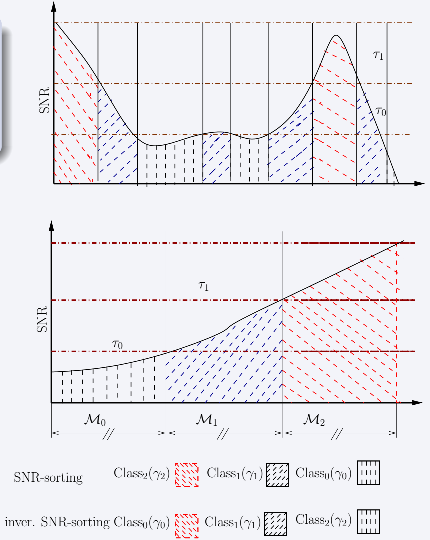
- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j \cdot \Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively,
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old} \cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}_{ej}}$


SNR
τ_1
τ_0
SNR
τ_1
τ_0
$\mathcal{M}_0$
$\mathcal{M}_1$
$\mathcal{M}_2$
SNR-sorting Class$_2(\gamma_2)$ Class$_1(\gamma_1)$ Class$_0(\gamma_0)$
inver. SNR-sorting Class$_0(\gamma_0)$ Class$_1(\gamma_1)$ Class$_2(\gamma_2)$

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}}_{ej}$, and bit-loading

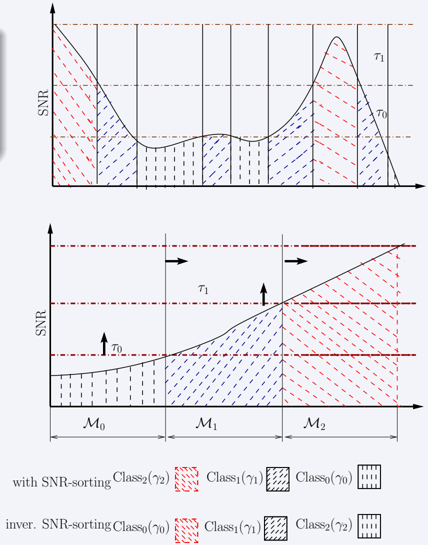
- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j \cdot \Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively,
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old} \cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}}_{ej}$


SNR
τ_1
τ_0
SNR
τ_1
τ_0
$\mathcal{M}_0$
$\mathcal{M}_1$
$\mathcal{M}_2$
SNR-sorting Class$_2(\gamma_2)$ Class$_1(\gamma_1)$ Class$_0(\gamma_0)$
inver. SNR-sorting Class$_0(\gamma_0)$ Class$_1(\gamma_1)$ Class$_2(\gamma_2)$

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}}_{ej}$, and bit-loading

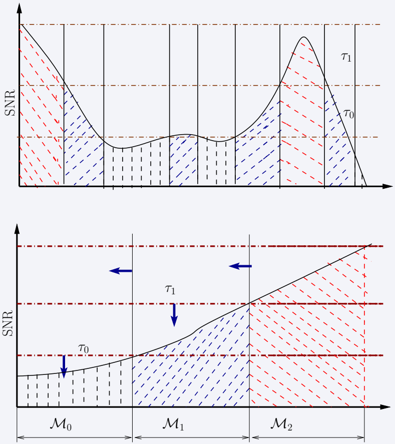
- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j \cdot \Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively, if $\sum_{k,j} b_{k,j} < T_j$
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old} \cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}_{ej}}$


SNR
τ_1
τ_0
SNR
τ_1
τ_0
$\mathcal{M}_0$
$\mathcal{M}_1$
$\mathcal{M}_2$
with SNR-sorting Class$_2(\gamma_2)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_0(\gamma_0)$ ▥
inver. SNR-sorting Class$_0(\gamma_0)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_2(\gamma_2)$ ▥

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}}_{ej}$, and bit-loading

- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j\cdot\Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively, if $\sum_{k,j} b_{k,j} > T_j$
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old}\cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}}_{ej}$



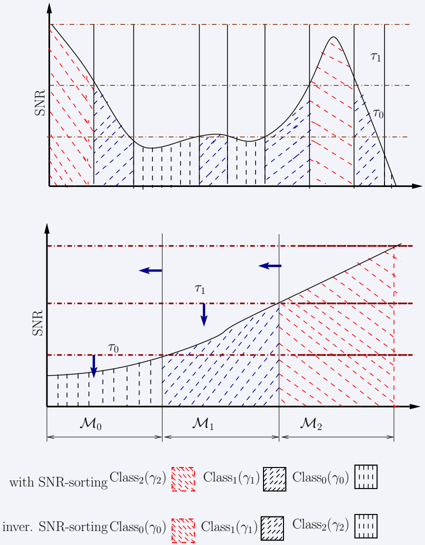
with SNR-sorting Class$_2(\gamma_2)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_0(\gamma_0)$ ▥

inver. SNR-sorting Class$_0(\gamma_0)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_2(\gamma_2)$ ▥

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}}_{ej}$, and bit-loading

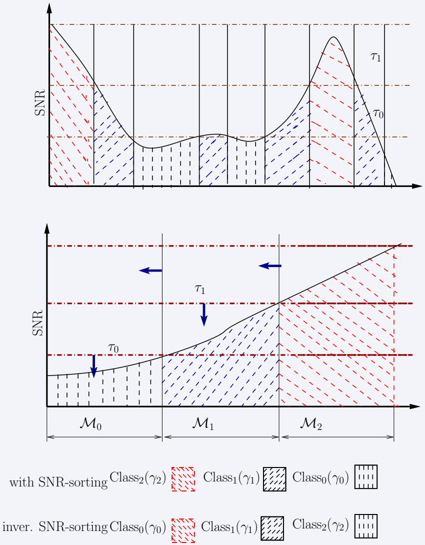
- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j \cdot \Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively, repeat unit $\sum_{k,j} b_{k,j} = T_j$ or maximum iteration
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old} \cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}}_{ej}$


SNR
τ_1
τ_0
SNR
τ_1
τ_0
$\mathcal{M}_0$
$\mathcal{M}_1$
$\mathcal{M}_2$
with SNR-sorting Class$_2(\gamma_2)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_0(\gamma_0)$ ▥
inver. SNR-sorting Class$_0(\gamma_0)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_2(\gamma_2)$ ▥

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}}_{ej}$, and bit-loading

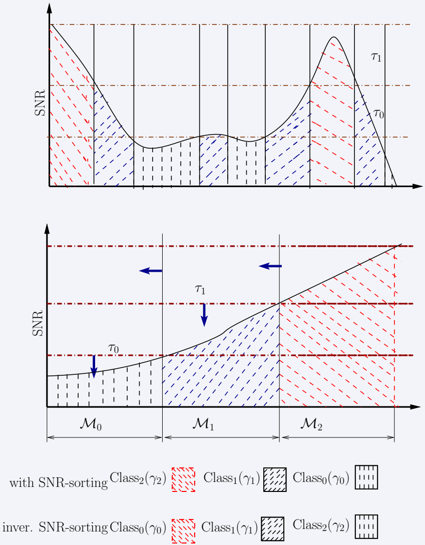
- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j \cdot \Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively, repeat unit $\sum_{k,j} b_{k,j} = T_j$ or maximum iteration
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old} \cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}}_{ej}$


SNR
τ_1
τ_0
SNR
τ_1
τ_0
$\mathcal{M}_0$
$\mathcal{M}_1$
$\mathcal{M}_2$
with SNR-sorting Class$_2(\gamma_2)$ Class$_1(\gamma_1)$ Class$_0(\gamma_0)$
inver. SNR-sorting Class$_0(\gamma_0)$ Class$_1(\gamma_1)$ Class$_2(\gamma_2)$

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}}_{ej}$, and bit-loading

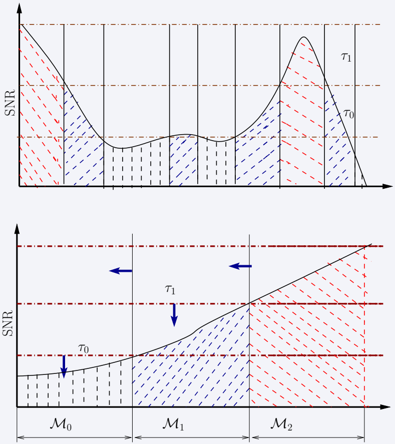
- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j \cdot \Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively, repeat unit $\sum_{k,j} b_{k,j} = T_j$ or maximum iteration
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old} \cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}}_{ej}$


SNR
τ_1
τ_0
SNR
τ_1
τ_0
$\mathcal{M}_0$
$\mathcal{M}_1$
$\mathcal{M}_2$
with SNR-sorting Class$_2(\gamma_2)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_0(\gamma_0)$ ▥
inver. SNR-sorting Class$_0(\gamma_0)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_2(\gamma_2)$ ▥

UEP Bit-Loading and SNR-Sorting Algorithms

Input: SNR$_{k,j}$ in k^{th} subcarrier of j^{th} class, N, N_g, B_T, T_j, and $\Delta\gamma$
Output: γ_j, average probability of error $\overline{\mathscr{P}}_{ej}$, and bit-loading

- Compute b_{kj} using γ_j ($\gamma_j = \gamma_0 - j\cdot\Delta\gamma$).
- Adjust $\mathscr{M}_j$ iteratively, repeat unit $\sum_{k,j} b_{k,j} = T_j$ or maximum iteration
- If $B_{tot} \neq B_T$, $\gamma_{0,new} = \gamma_{0,old}\cdot 2^{\frac{B_{tot}-B_T}{N}}$
- If $B_{tot} \neq B_T$ again, add or subtract bits according to $\Delta b_{k,j}$
- The power is allocated according to $\overline{\mathscr{P}}_{ej}$

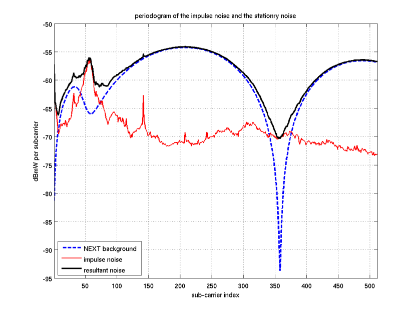


with SNR-sorting Class$_2(\gamma_2)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_0(\gamma_0)$ ▥

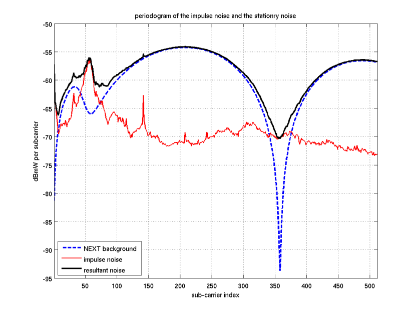
inver. SNR-sorting Class$_0(\gamma_0)$ ▨ Class$_1(\gamma_1)$ ▨ Class$_2(\gamma_2)$ ▥

Outline

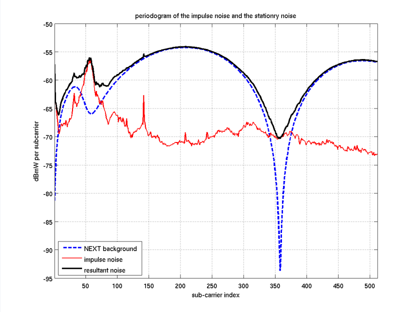
1. Motivations
   - Why Unequal Error Protection (UEP)?
   - Why UEP Physical Transport?
2. UEP: Bit-Loading
   - Previous Work
   - Proposed UEP Bit-Rate Maximization
3. Channel Model
   - Noise Environment
4. Simulation Results
   - UEP Performance: SER Analysis
   - Bit and Power Loading
5. Conclusions



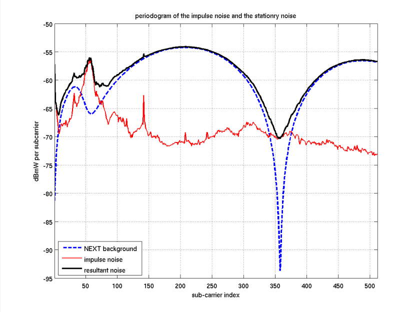
- ADSL2*plus* with 512 subcarriers is considered for this channel.
- A wireline cable of diameter 0.4 mm and 2 km length is assumed.
- A combination of T1 + HDSL NEXT and -130 dBm/Hz AWGN is used for the bit-loading.
- Additionally, real measured impulse noise is introduced after bit allocation.



- ADSL2*plus* with 512 subcarriers is considered for this channel.
- A wireline cable of diameter 0.4 mm and 2 km length is assumed.
- A combination of T1 + HDSL NEXT and -130 dBm/Hz AWGN is used for the bit-loading.
- Additionally, real measured impulse noise is introduced after bit allocation.



- ADSL2*plus* with 512 subcarriers is considered for this channel.
- A wireline cable of diameter 0.4 mm and 2 km length is assumed.
- A combination of T1 + HDSL NEXT and -130 dBm/Hz AWGN is used for the bit-loading.
- Additionally, real measured impulse noise is introduced after bit allocation.



- ADSL2*plus* with 512 subcarriers is considered for this channel.
- A wireline cable of diameter 0.4 mm and 2 km length is assumed.
- A combination of T1 + HDSL NEXT and -130 dBm/Hz AWGN is used for the bit-loading.
- Additionally, real measured impulse noise is introduced after bit allocation.

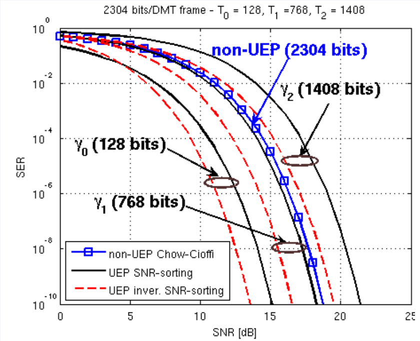
Outline

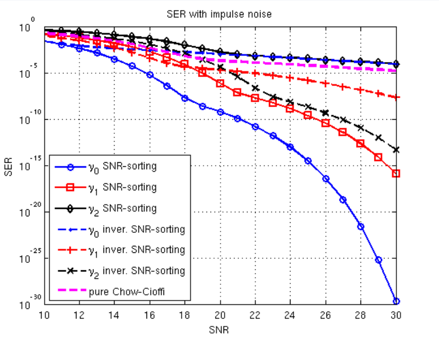
1. Motivations
   - Why Unequal Error Protection (UEP)?
   - Why UEP Physical Transport?
2. UEP: Bit-Loading
   - Previous Work
   - Proposed UEP Bit-Rate Maximization
3. Channel Model
   - Noise Environment
4. Simulation Results
   - UEP Performance: SER Analysis
   - Bit and Power Loading
5. Conclusions

SER for Stationary and Non-stationary Noise

SER for stationary noise

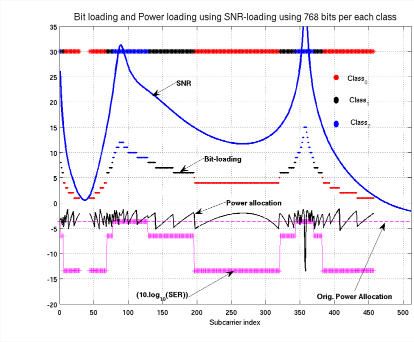
SER for non-stationary noise



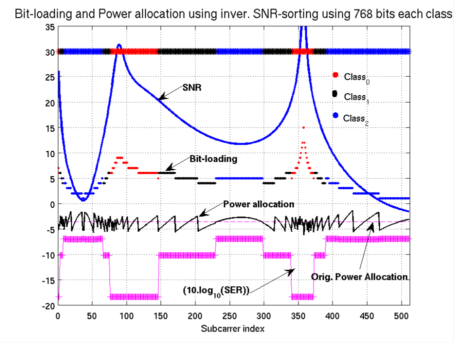


UEP bit-loading and power-allocation:

SNR-Sorting Scheme



Inverse SNR-Sorting Scheme



Outline

1. Motivations
   - Why Unequal Error Protection (UEP)?
   - Why UEP Physical Transport?
2. UEP: Bit-Loading
   - Previous Work
   - Proposed UEP Bit-Rate Maximization
3. Channel Model
   - Noise Environment
4. Simulation Results
   - UEP Performance: SER Analysis
   - Bit and Power Loading
5. Conclusions

Conclusions

- We described an UEP bit-allocation scheme as an extension of the algorithm by Chow et al..
- Allows arbitrary margin definitions and bit-rates according to the priorities.
- SNR-sorting will ensure that the high-priority class will still be well protected even under non-stationary noise.

Modified bit-loading:

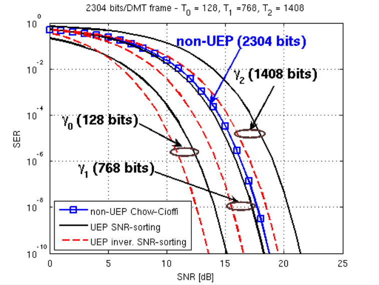
$$\begin{aligned} b_{k,j} &= \log_2\left(1+\frac{\mathsf{SNR}_{k,j}}{\gamma_j}\right) \\ \hat{b}_{k,j} &= \lfloor b_{k,j}+0.5\rceil_0^{b_{\max}} \\ \Delta b_{k,j} &= b_{k,j}-\hat{b}_{k,j} \end{aligned}$$

Open points:

Possible mixed allocation and hierarchical modulation.

Conclusions

- We described an UEP bit-allocation scheme as an extension of the algorithm by Chow et al..
- Allows arbitrary margin definitions and bit-rates according to the priorities.
- SNR-sorting will ensure that the high-priority class will still be well protected even under non-stationary noise.

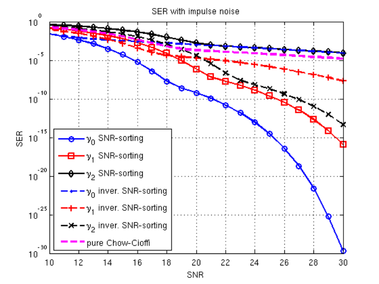


Open points:

Possible mixed allocation and hierarchical modulation.

Conclusions

- We described an UEP bit-allocation scheme as an extension of the algorithm by Chow et al..
- Allows arbitrary margin definitions and bit-rates according to the priorities.
- SNR-sorting will ensure that the high-priority class will still be well protected even under non-stationary noise.

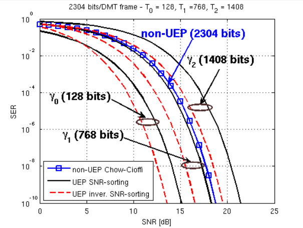


Open points:

Possible mixed allocation and hierarchical modulation.

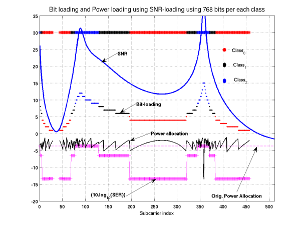
Conclusions

- We described an UEP bit-allocation scheme as an extension of the algorithm by Chow et al..
- Allows arbitrary margin definitions and bit-rates according to the priorities.
- SNR-sorting will ensure that the high-priority class will still be well protected even under non-stationary noise.



Open points:

Possible mixed allocation and hierarchical modulation.



Conclusions

- We described an UEP bit-allocation scheme as an extension of the algorithm by Chow et al..
- Allows arbitrary margin definitions and bit-rates according to the priorities.
- SNR-sorting will ensure that the high-priority class will still be well protected even under non-stationary noise.

Open points:

Possible mixed allocation and hierarchical modulation.

Thank you!