Maximising the Utility of Virtually Sliced Millimetre-Wave Backhauls

Rui Li, Chaoyun Zhang, Paul Patras, Pan Cao, and John Thompson

School of Informatics and School of Engineering
The University of Edinburgh

COSENERS 2018

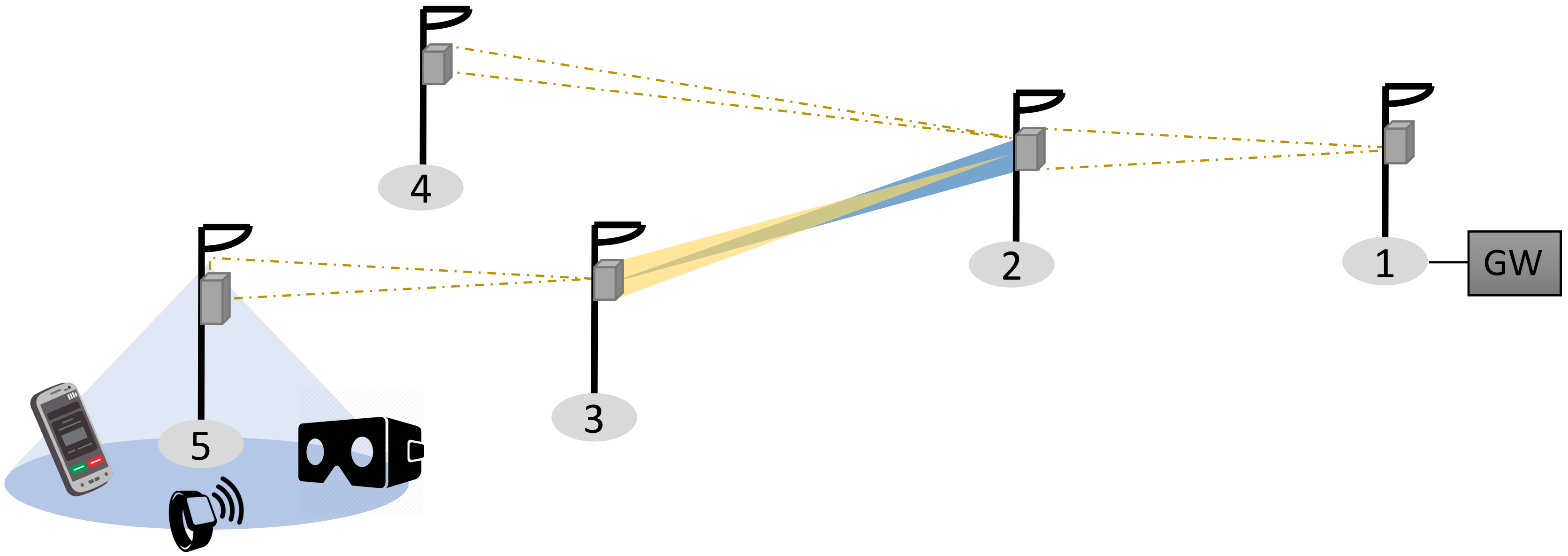
5G Networks

- Use cases with distinct performance requirements
  - Bandwidth: UHD video streaming and AR/VR
  - Delay: Autonomous vehicles and remote medical care

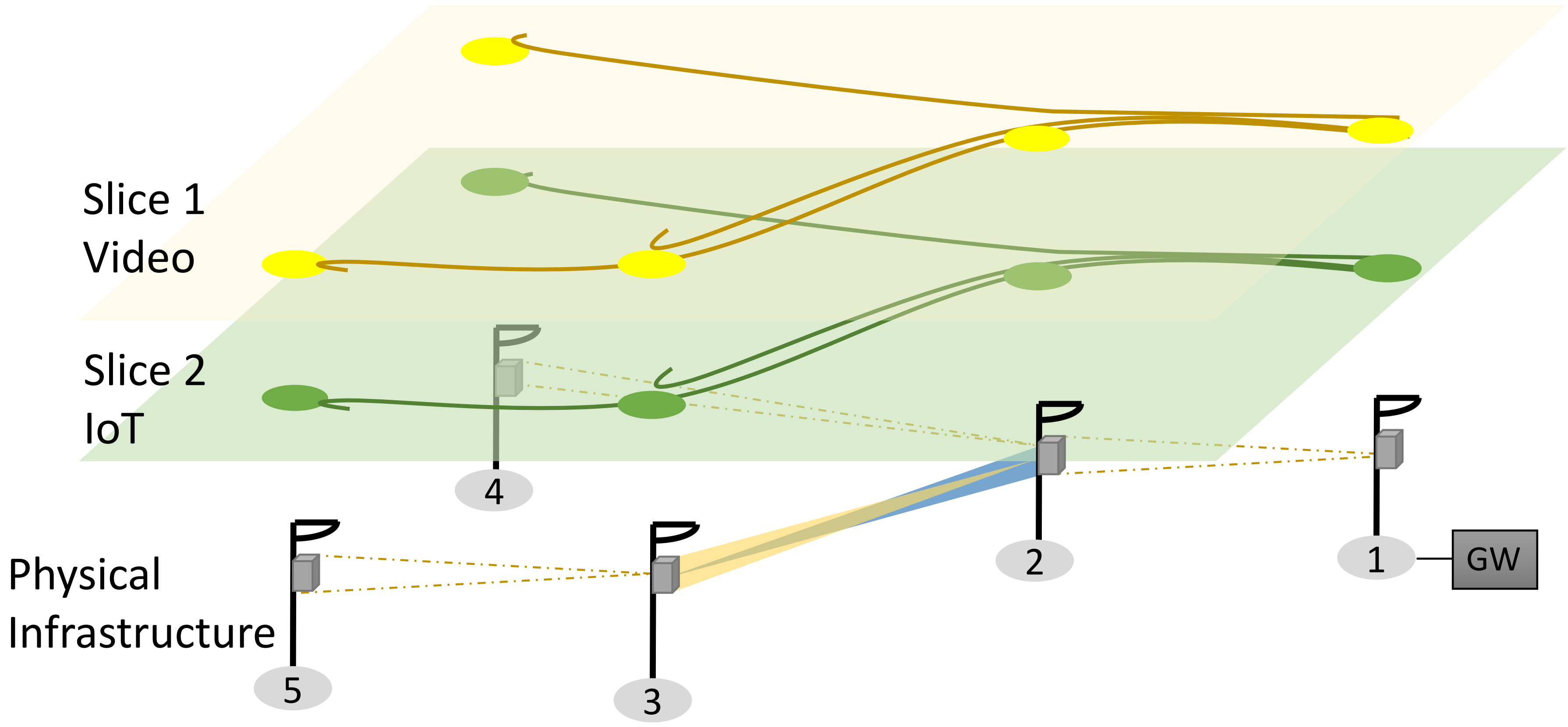
- Network slicing
  - Partition physical infrastructure into logically isolated networks

- Network densification
  - Millimetre-wave (mm-wave) enables multi-Gbps link rates
  - Tangible backhauling solution

Mm-Wave Small Cell Backhaul



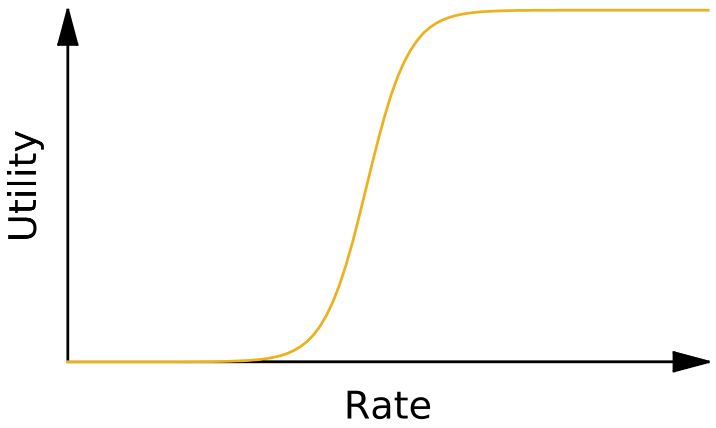
Virtually Sliced Mm-wave Backhaul Network



- Resource Allocation: Rate $r_{i,j}$ for flow $f_{i,j}$
- To meet the service requirements and to maximise resource utilitsation

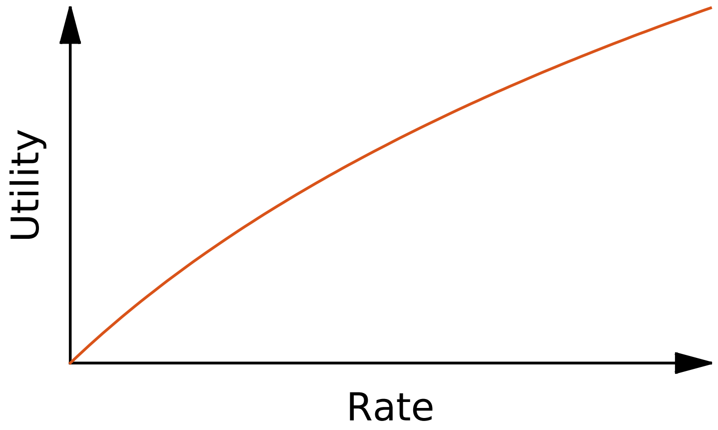
Utility Functions for Different Applications

QoS (AR/VR)



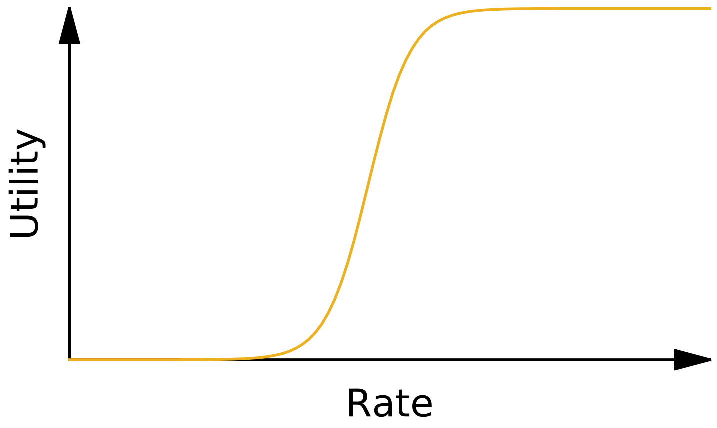
Sigmoid: $U_{sig}(r) = \frac{1}{1+e^{-\alpha_1(r-\beta_1)}}$

Best-effort (IoT)



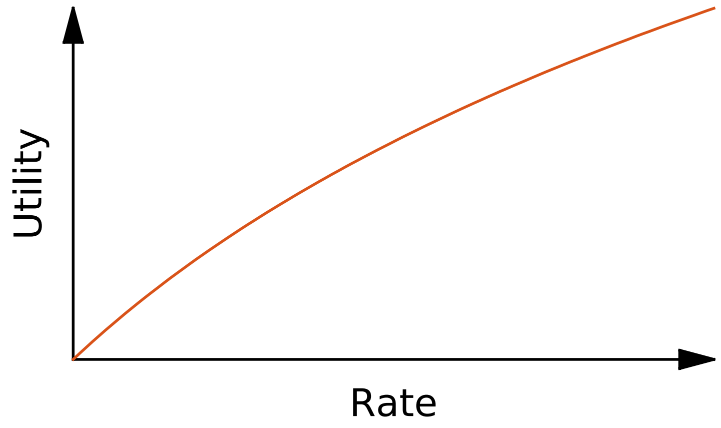
Logarithm $U_{log}(r) = \log(\alpha_2 r + \beta_2)$

QoS (AR/VR)



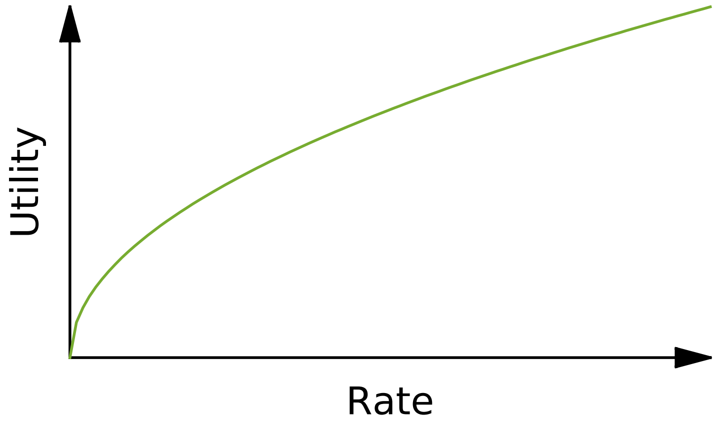
Sigmoid: $U_{sig}(r) = \frac{1}{1+e^{-\alpha_1(r-\beta_1)}}$

Best-effort (IoT)



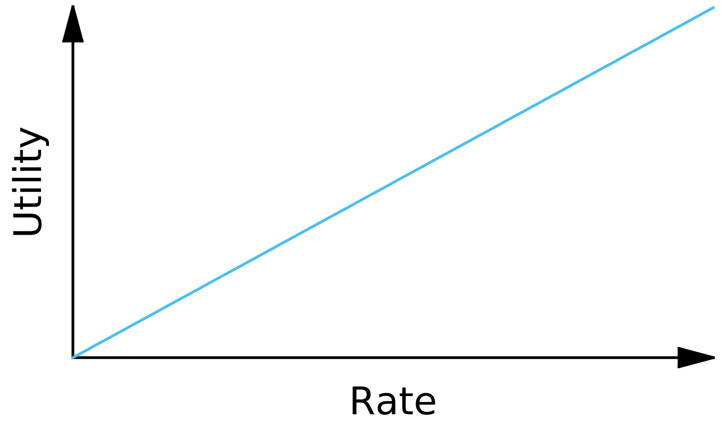
Logarithm $U_{log}(r) = \log(\alpha_2 r + \beta_2)$

Delay sensitive (Tele-Operation)



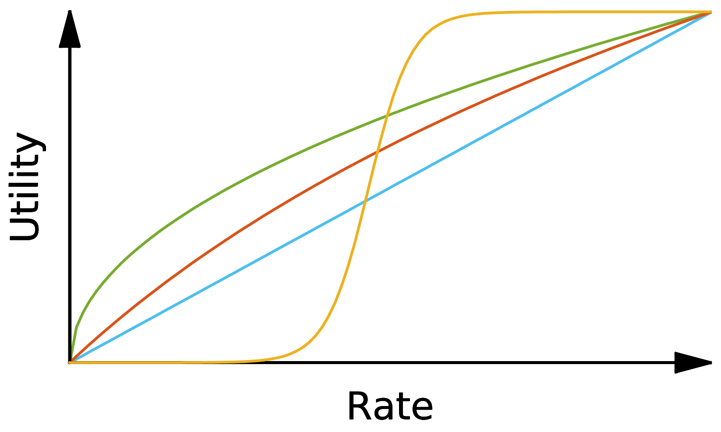
Polynomial: $U_{ply}(r) = \alpha_3(r^{\beta_3})$

Revenue (other)



Linear $U_{lnr}(r) = \alpha_4 r + \beta_4$

Utility Framework



Utility Framework

Sigmoidal: $U_{sig}(r) = \frac{1}{1+e^{-\alpha_1(r-\beta_1)}}$

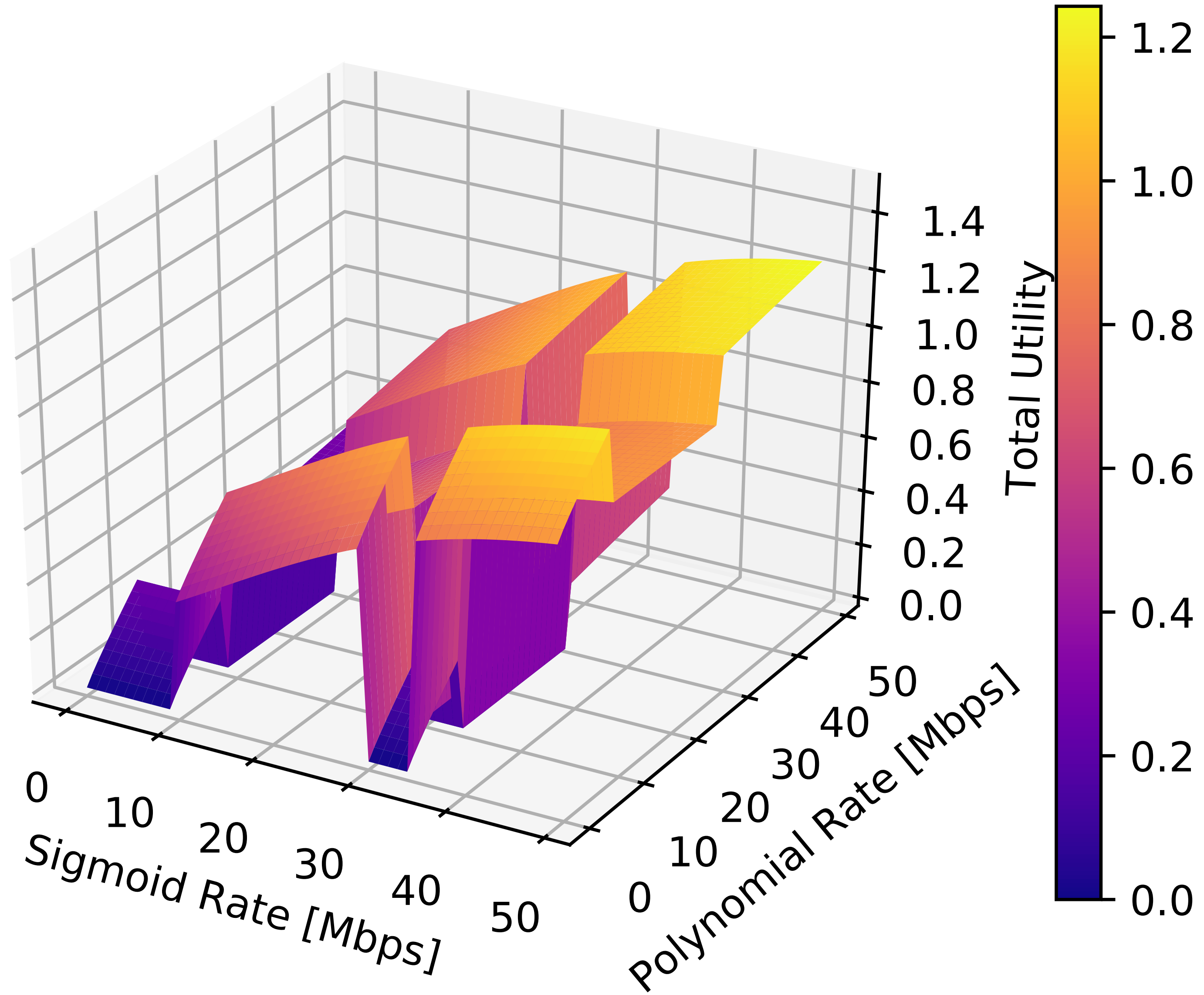
Polynomial: $U_{ply}(r) = \alpha_2(r^{\beta_2})$

Logarithmic: $U_{log}(r) = \log(\alpha_3 r + \beta_3)$

Linear: $U_{lnr}(r) = \alpha_4 r + \beta_4$

$$\left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \max \sum U_i(r_{i,j})$$

Combining these together *in a simple scenario…*

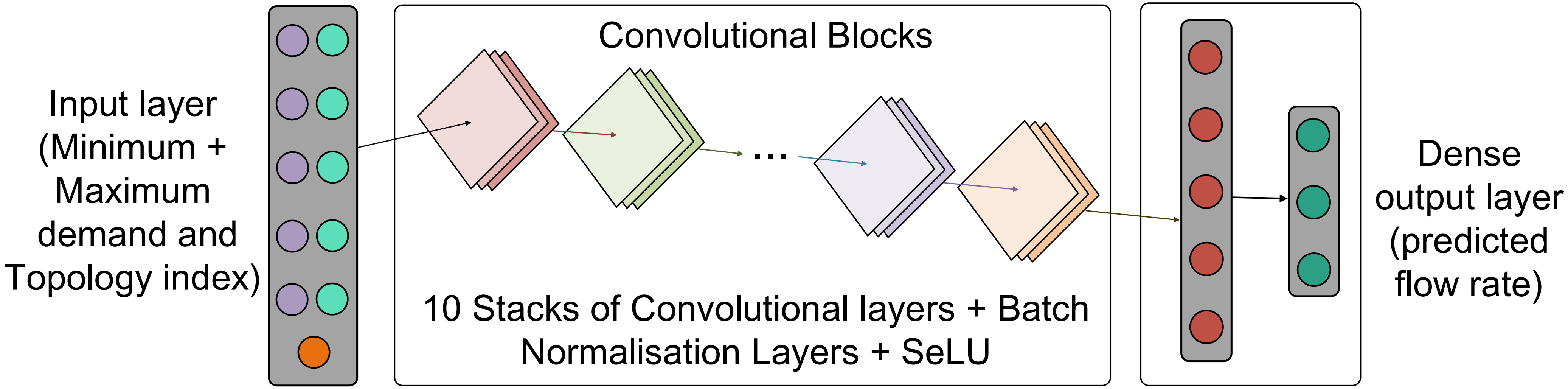


Utility Maximisation

- High-dimensional problem, highly non-convex
- Global search is time consuming
- Heuristic method can solve but sub-optimal

- Learn the correlation between flow demands and optimal allocations

Convolutional Neural Network (CNN) Solution



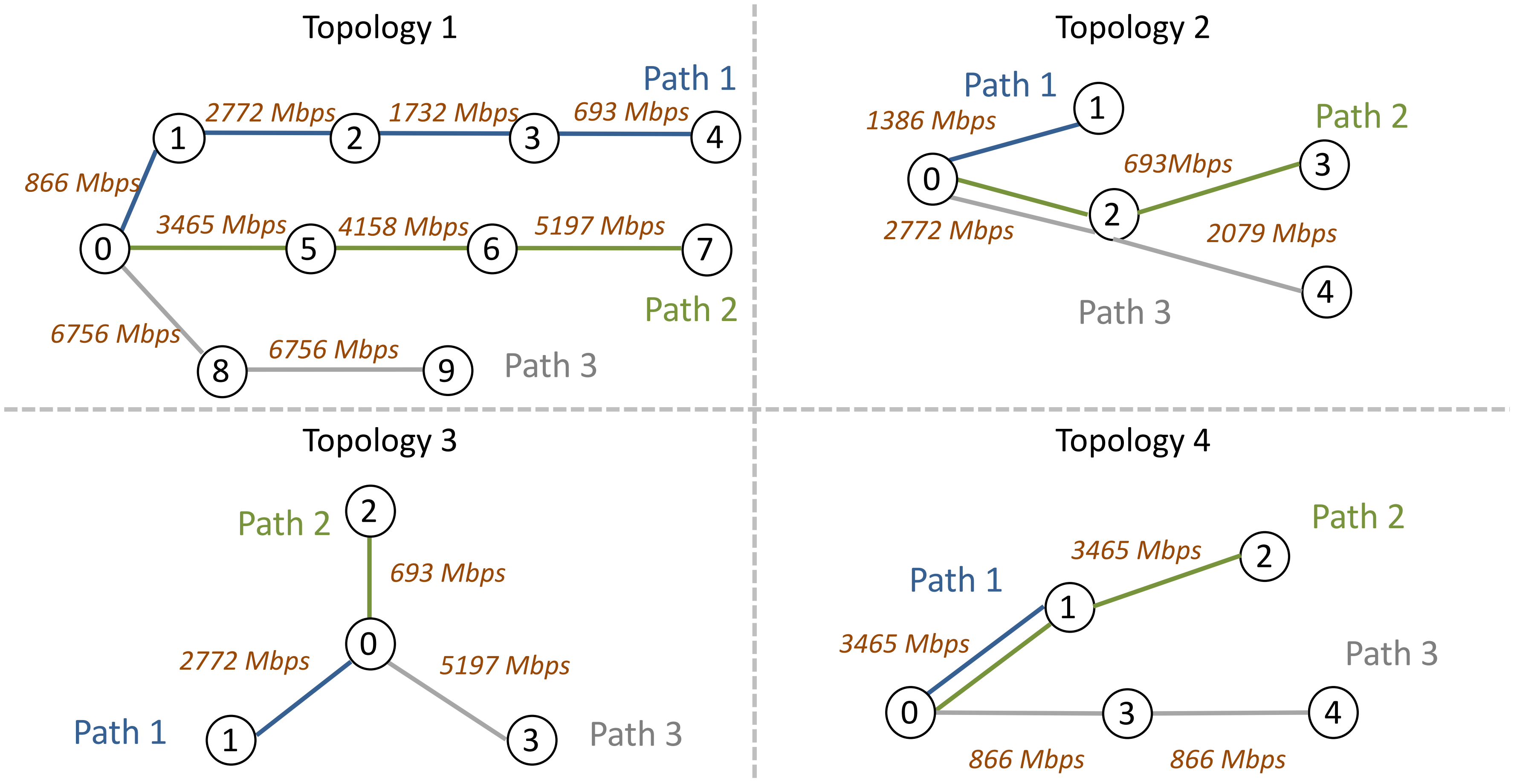
Numerical Evaluation - Methods

- 10,000 data points
  - Flow demands ($\mathrm{d}_{i,j}$) and minimum service rates ($\delta_{i,j}$)
  - Optimal solutions obtained from Global Search (GS) *
  - Training performed on GPU and inference on CPU

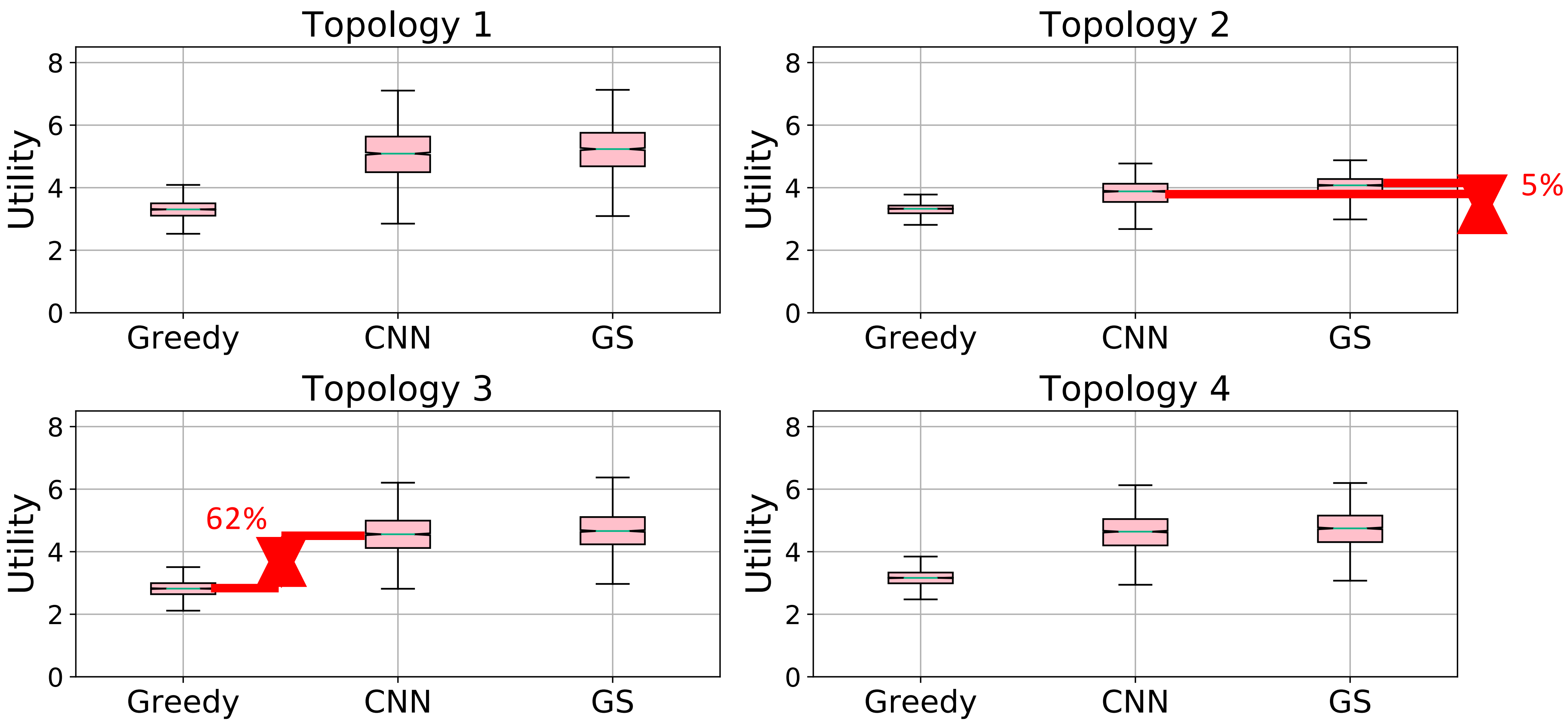
- Benchmark greedy solution
  - Supposed to work fast

* Optimality of GS is proven in Z. Ugray et al. Scatter search and local NLP solvers: *A multistart framework for global optimization*. Journal on Computing, 19(3), 2007.

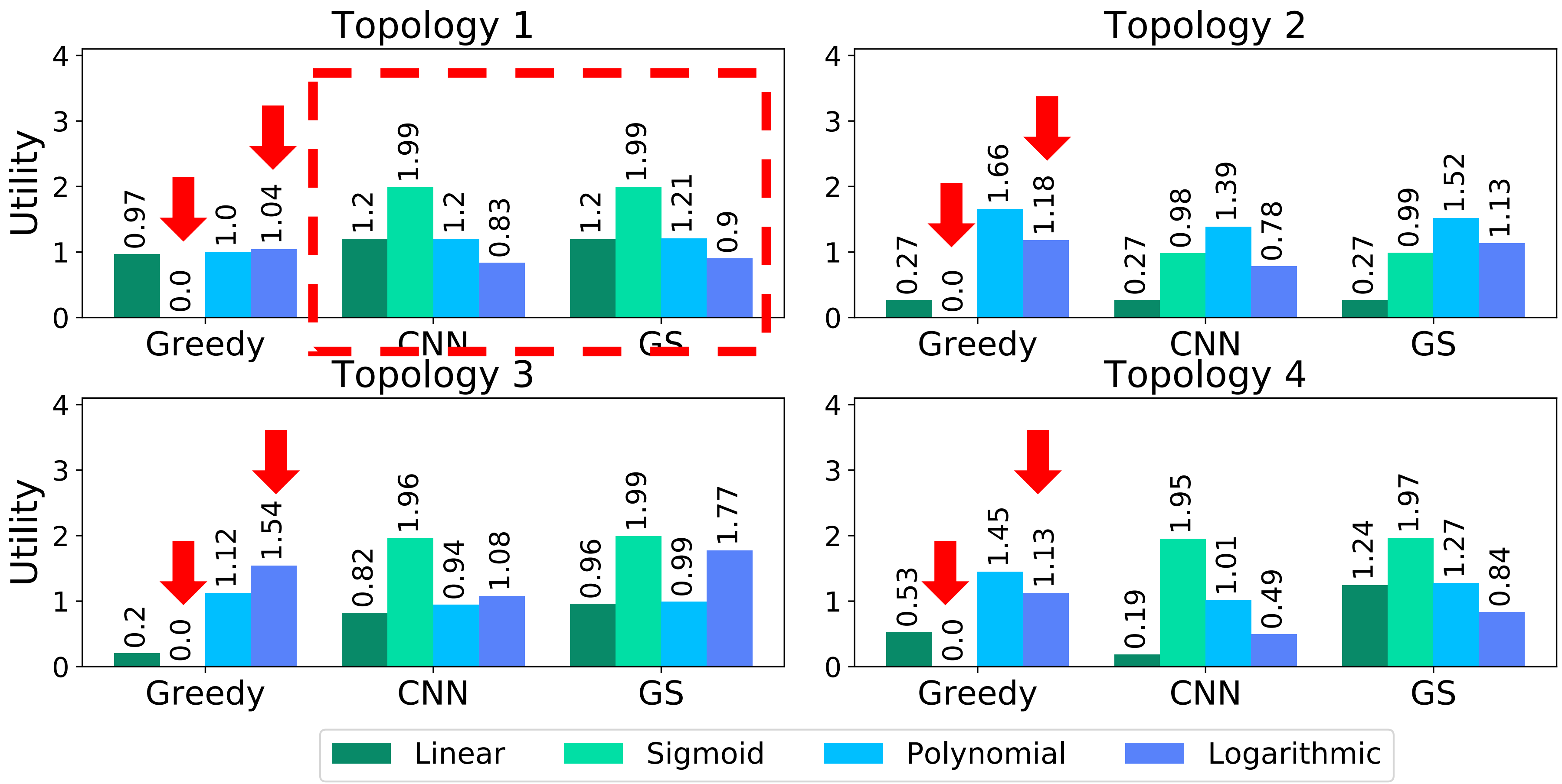
Numerical Evaluation - Topologies



Results: Total Utility Distribution



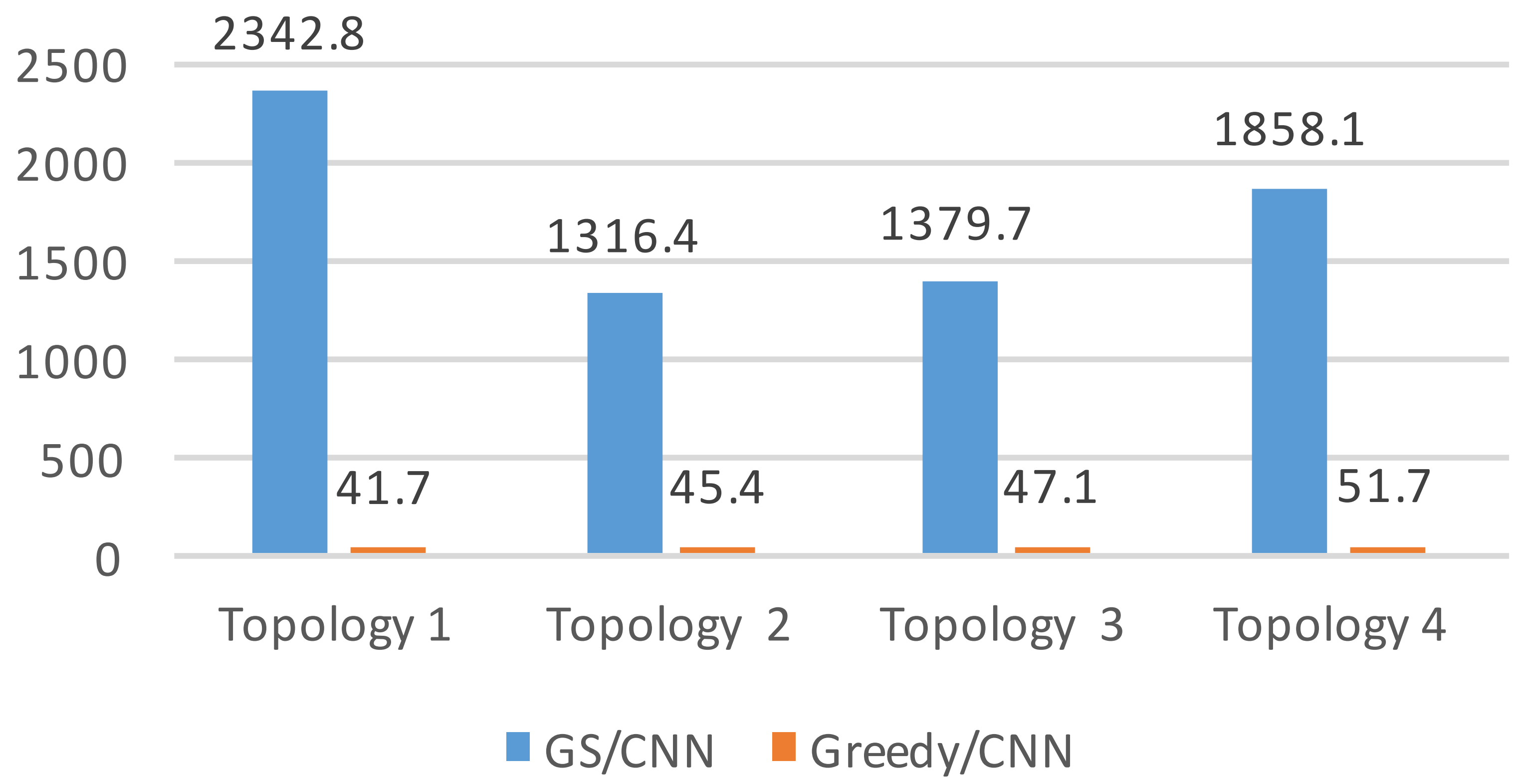
Results: Utility per Traffic Type



Computation Time

2000x faster than GS

50x faster than Greedy



CNN	0.0036s	0.0035s	0.0025s	0.0026s

Conclusions

- A general utility framework encompasses all known types of utility functions, and formulate an utility optimisation problem
- CNN achieves close-to-optimal solution, and makes rapid inference
- Suitable for 5G with real-time and dynamic requirements


рахмат
danke
謝謝
ngiyabonga
teşekkür ederim
spas
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
thank you
gracias
хвала
kiitos
dankie
nanni
nandri
dhanyavad
asante
manana
bedankt
bayarlalaa
gracie
hvala
maururu
köszönöm
aciu
dankon
akun
obrigada
tenki
enkosi
murakoze
djiere dieuf
tau
mochchakkeram
дякую
mamnun
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
chokrane
sobodi
dėkuji
sagolun
sukriya
kop khun krap
taiku
arigatô
takk
dakujem
trugarez
obrigado
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
terima kasih
rahmet
shukriya
mercè
мерси
kam sah hamnida
rahmat
ευχαριστώ
diolch
dhanyavadagalu
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci