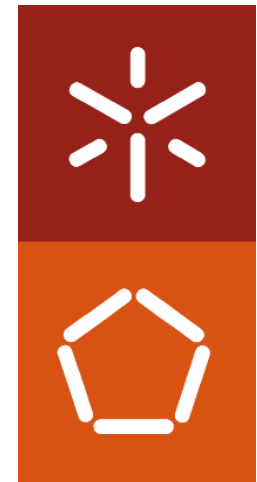


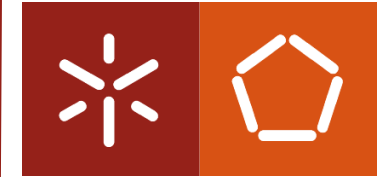
# Redes Tolerantes a Atrasos (DTN)

---

**Novos Paradigmas de Redes**  
**2023-2024**

2º Semestre





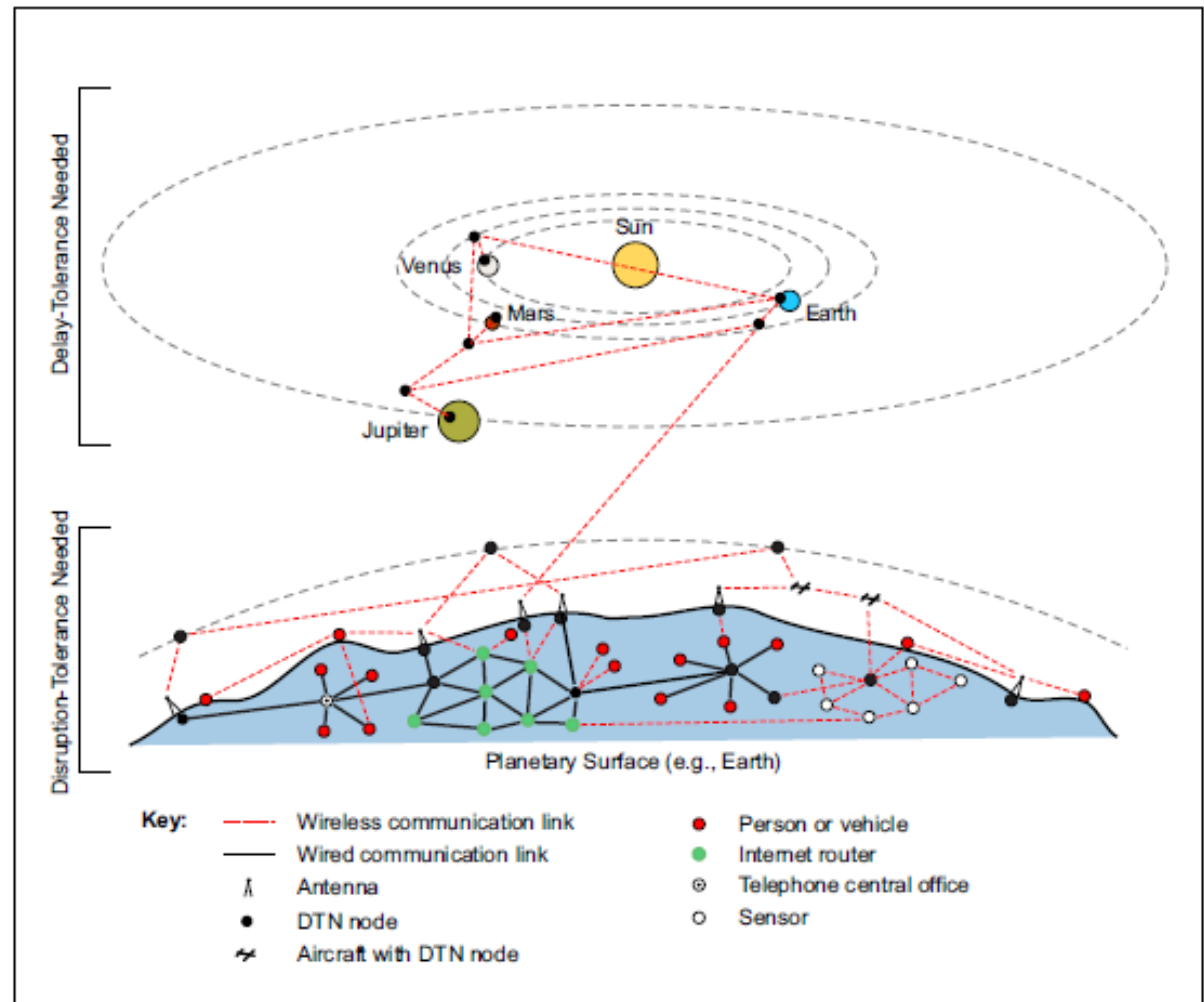
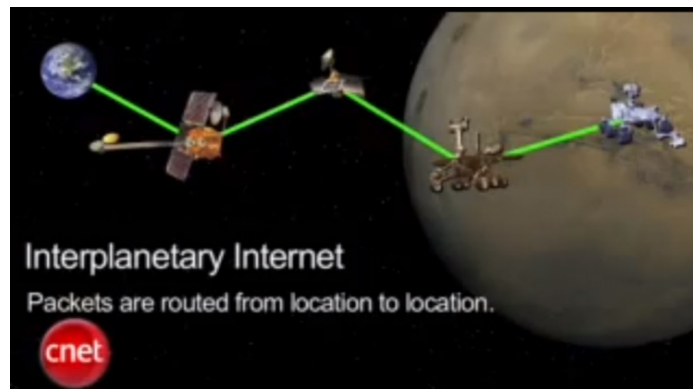
# Delay-Tolerant Networks

- Uma derivação das **MANET** (*Mobile Ad Hoc Networks*)
- Trata-se de uma rede onde não existe conectividade permanente entre todos os nós

Como encaminhar pacotes nestas circunstâncias?  
Problema é mais complexo se considerarmos a mobilidade!

- **O protocolo de encaminhamento é uma peça chave!**
  - Paradigma *store-carry-and-forward*

# DTN: A origem...



<https://www.nasa.gov/communicating-with-missions/delay-disruption-tolerant-networking/>  
A simplified Introduction to DTN: <https://www.youtube.com/watch?v=HV8CHoWP9-o>

# DTN: Aplicações (1)

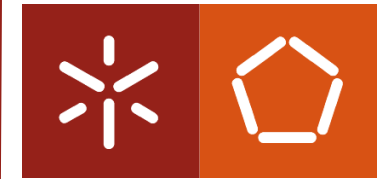


- **Conectividade de regiões em desenvolvimento:** Estudantes com laptop e sem infra-estrutura de rede de suporte que pretendem colaborar em projectos;
  - as mensagens podem ter de esperar na rede quando o aluno não estiver na escola
  - Conectividade externa em dois saltos:
    - Aproveitar os autocarros para carregar os pacotes para pontos de acesso com conectividade à Internet e vice versa (se necessário)
    - Enviar e receber DVD com dados pelo correio para ficheiros grandes

Ex: KioskNet  
(Índia e Cambodja)



# DTN: Aplicações (2)



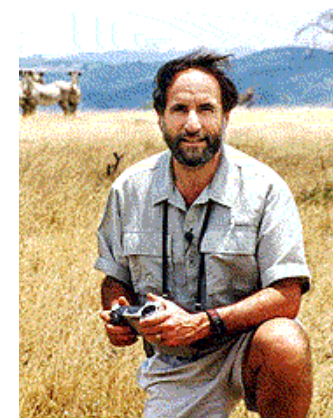
- Conectividade de regiões desconectadas regiões protegidas ou de conectividade não rentável
  - Parque natural da Lapónia (protegido pela UNESCO; não se pode instalar antenas...)
  - Esquimós...
  - Minas ... o melhor é levar os dados com os mineiros...

# DTN: Aplicações (3)



## ● Monitorização do ambiente e da vida animal

- Redes de sensores para estudo da vida animal (**ZebraNet**)
- The ZebraNet Wildlife Tracker (Princeton University), Kenya, 2004
- De Zebra para Zebra (equipadas com coleiras)
  - Capturar e guardar dados 24h/24h, georeferenciados por GPS
  - De vez em quando emitir um sinal e ver se há vizinhos(as) para aproveitar para trocar dados entre... os investigadores de vez em quando passam por lá também para coligir e mandar apagar dados

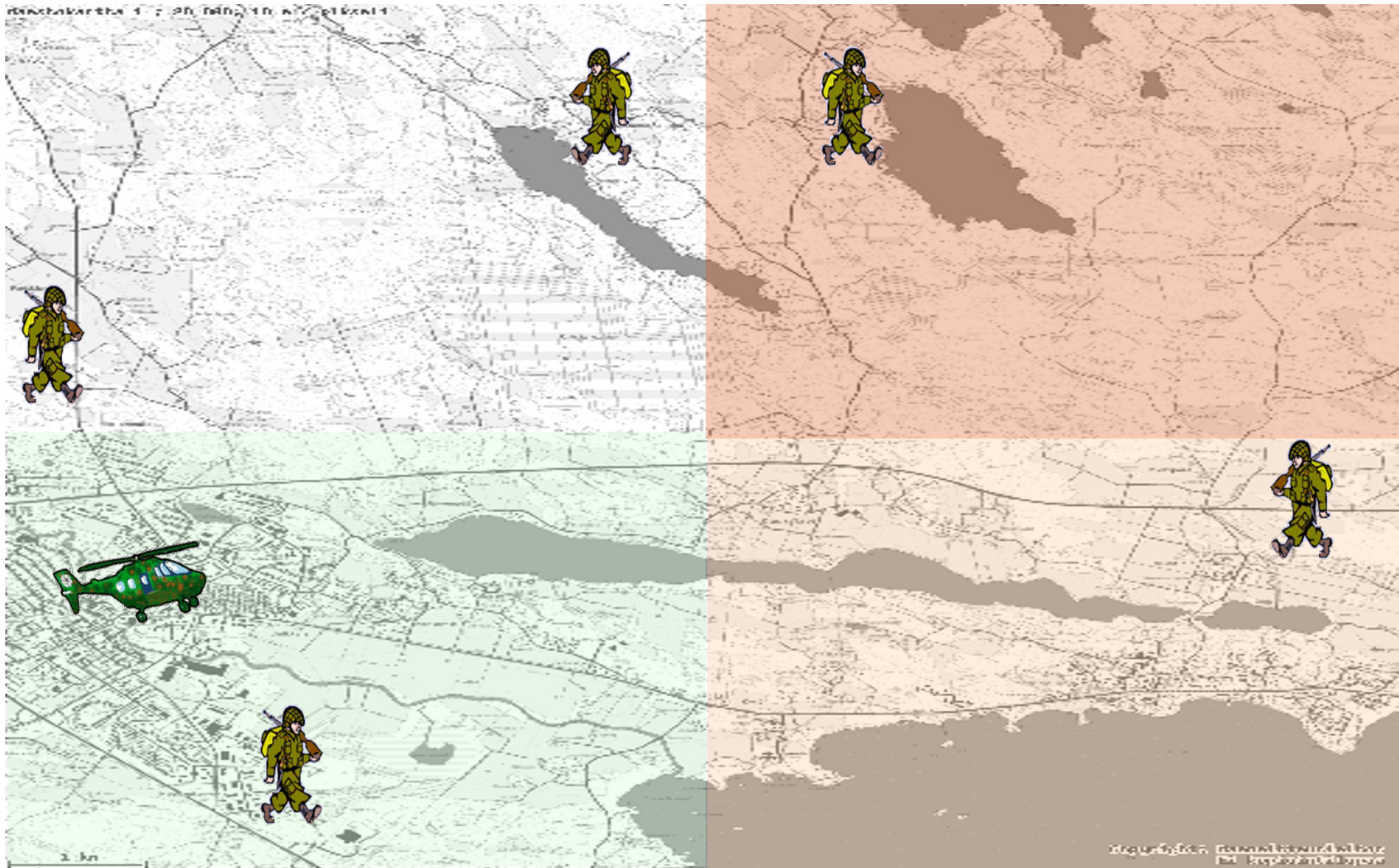




# DTN: Aplicações(4)



- **Aplicações militares:** militares em combate



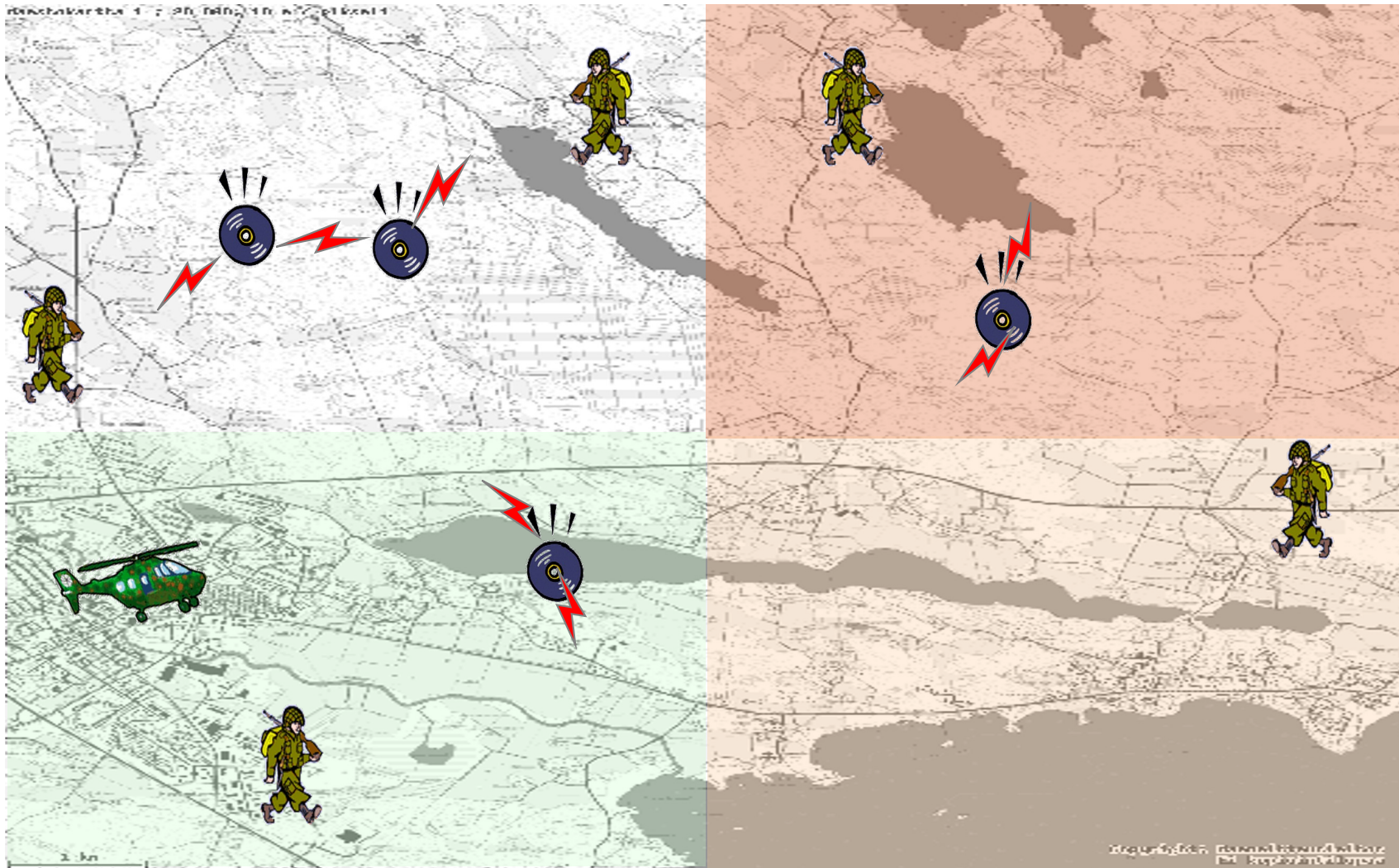
NPR 2023/2024 © Universidade do Minho



# DTN: Aplicações(4)



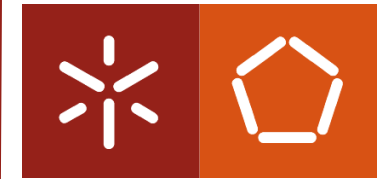
- **Aplicações militares:** DTN com *Throwboxes*



NPR 2023/2024 © Universidade do Minho



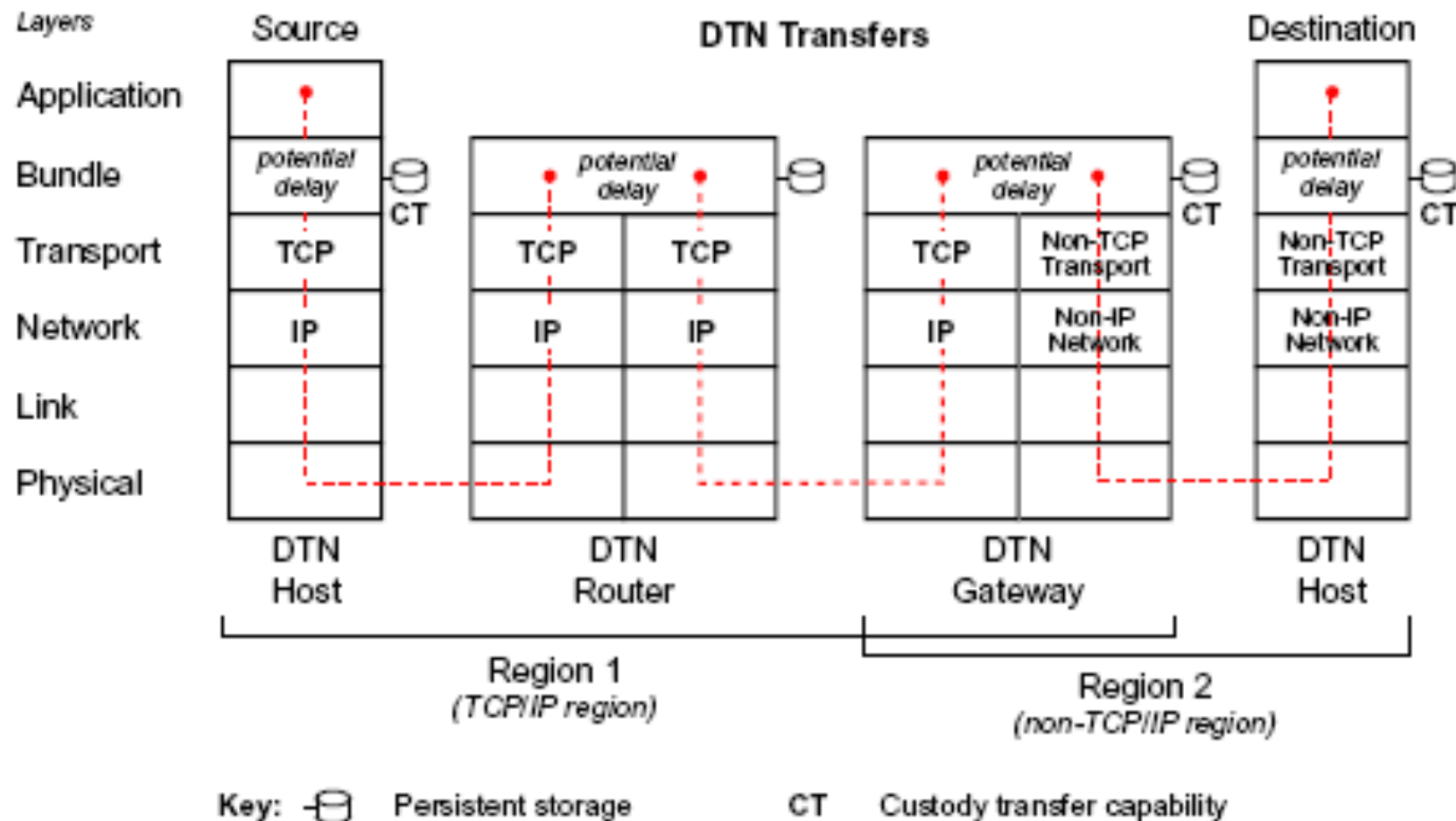
# DTN: Aplicações (5)



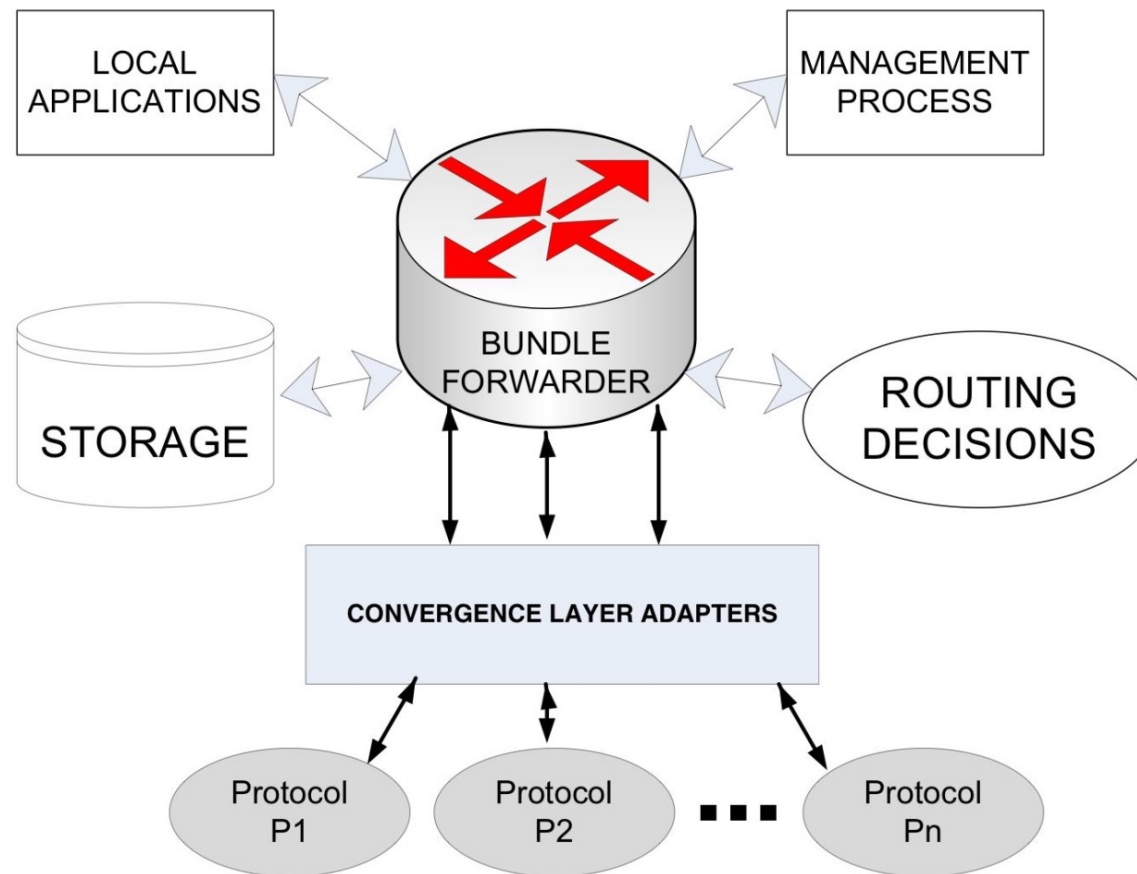
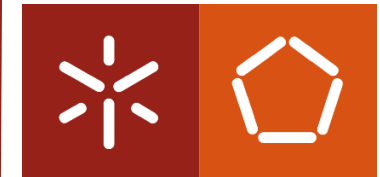
- Comunicações submarinas (*undersea*)
- VDTN: redes DTN veiculares
- Redes Urbanas ou Redes Sociais
  - Ex: *pocket switched networks*
- ...

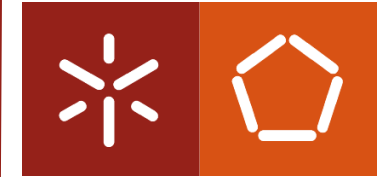
Qual a melhor forma de encaminhar pacotes nestas circunstâncias?

# DTN Architecture



# DTN Node

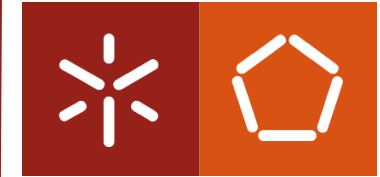




## ● Características das DTN

- Rede composta por nós que são sistemas computacionais ligados por links unidireccionais com alguns dos outros
- As ligações podem estabelecer-se ou desfazer-se pelas mais variadas razões
  - Obstruções, falha de energia, fora de alcance de transmissão, mobilidade, falhas da rede, etc...
- Quando a ligação entre dois nós se estabelece e está activa há uma oportunidade efectiva de enviar dados de um nó para outro – esta oportunidade designa-se por “**contacto**”
  - Pode existir mais que um contacto disponível entre dois nós
- Na teoria de grafos: *Multigrafo variável no tempo*



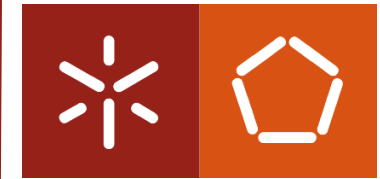


- **Características das DTN (continuação)**

- Pretende-se que seja sempre possível enviar dados entre quaisquer nós da rede aproveitando os contactos
- As mensagens podem ter de ser armazenadas temporariamente por longos períodos de tempo até haver um contacto

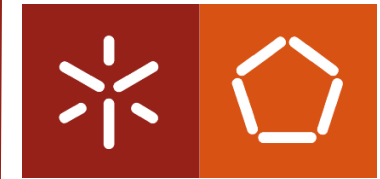
- **Tempo de transferência de uma mensagem de um nó para outro (4 componentes):**

1. **Tempo de espera** por contacto (desde que foi recebida até poder ser enviada ao próximo nó)
2. **Tempo na fila de espera** (tempo de esvaziamento da fila de mensagens com mais prioridade)
3. **Tempo de transmissão** (tempo necessário para enviar todos os bits da msg)
4. **Tempo de propagação** (tempo necessário para que um bit atravessasse a ligação)



## (1) Escalonamento dos contactos

- Das 4 componentes, o tempo maior é seguramente o tempo de espera por um contacto: pode variar entre segundos ou dias
- Contactos caracterizam-se quanto à sua previsibilidade:
  - **Escalonamento preciso** (ex: resultante de movimentos no espaço de objectos que podem ser calculados)
  - **Escalonamento aproximado** (ex: contactos podem ser menos precisos se ocorrerem ligeiros atrasos não previstos, como nos horários dos autocarros ou movimento de veículos na estrada)
  - **Escalonamento implícito** (movimento humano que segue muitas rotinas habituais)
  - **Escalonamento aleatório** (não se sabe quando há conectividade)



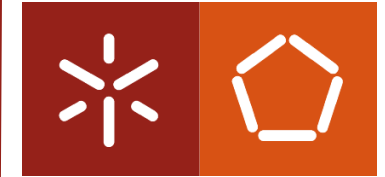
## (2) Capacidade dos contactos

- Que quantidade de dados é possível transferir em cada contacto?
- Depende da tecnologia usada na ligação e da duração do contacto!
- Mesmo que seja possível prever a duração dos contactos, a taxa de transmissão pode ser variável;
- Será importante para o *routing*?
- Pode-se ignorar a capacidade a menos que seja insuficiente para transferir uma mensagem, pois aí deve ocorrer fragmentação
- Boa capacidade pode ser preferida a outras características...

## (3) Espaço de armazenamento (buffers)

- As mensagens podem ter de permanecer muito tempo no nó...

# DTN: Principais Desafios



## (4) Capacidade de processamento

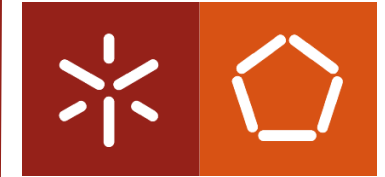
- Dispositivos pequenos, com CPU e memória muito limitados
- As estratégias de encaminhamento podem ser diferentes... pois podem não ser capazes de operações complexas

## (5) Energia

- As reservas de energia dos nós móveis podem ser limitadas
- A receção e envio de mensagens, bem como os procedimentos de encaminhamento, consomem energia
- Estratégias que enviem menos *bytes* ou façam menos cálculos permitem maior eficiência energética
- As estratégias de routing podem otimizar o consumo de energia



# DTN: Critérios de avaliação



## 1. Taxa de entrega

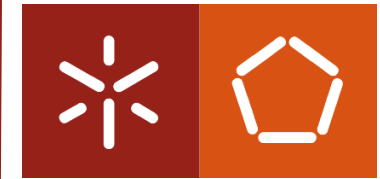
- Talvez a medida mais importante numa DTN: percentagem das mensagens geradas que é efectivamente entregue
- Nas DTN as mensagens podem não se “perder” mas a rede pode não as conseguir entregar num espaço de tempo útil

## 2. Latência

- Tempo desde que a mensagem é gerada até que é recebida
- Embora se espere que as aplicações possam tolerar atrasos grandes, é suposto beneficiarem de um tempo o mais curto possível

## 3. Transmissões

- Algumas estratégias transmitem mais mensagens que outras
- Medida da capacidade dos contactos que é consumida



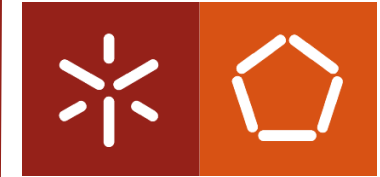
- Protocolos de *routing* convencionais não servem...

- Assumem que há sempre conectividade fim a fim e nas DTN tal não acontece
- A posição dos nós pode ser desconhecida se o seu padrão de mobilidade for desconhecido

- Várias abordagens/estratégias

- Podem ser comparadas em duas propriedades:

- Replicação - tentar aumentar a probabilidade de entrega aumentando o número de cópias das mensagens (mínimo de uma até o máximo de uma cópia por nó)
- Conhecimento – tentar aumentar o grau de conhecimento da rede e do escalonamento dos contactos (mínimo assumir que o nó só conhece os contactos activos, máximo conhece todo o escalonamento de contactos da rede)



- **Estratégias baseadas na inundação da rede (*flooding*)**

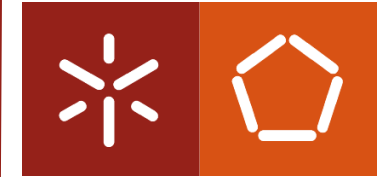
- (1) Encaminhamento “epidémico”** (Becker 2000)

- Não há nenhum conhecimento da rede
    - Pacotes espalham-se como um *virus* que contamina tudo:

Inundar todos os nós vizinhos alcançáveis com cópias de um pacote enquanto o nó de destino não obtiver uma cópia

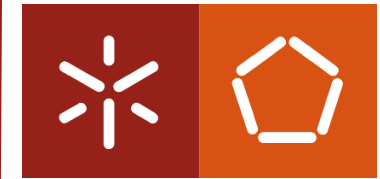
Cada nó mantém um vector de sumariação que constitui o índice dos pacotes que já viu; quando dois nós se encontram, trocam vectores entre si e pedem uma cópia dos pacotes novos;

# DTN: Encaminhamento



- **Esquemas epidémicos consomem muitos recursos:**
  - Largura de banda, buffers, energia do nó, etc
  - Consumo de recursos pode **degradar seriamente o desempenho**
- **É preciso controlar o *flooding*!**
- **Pode até acontecer das mensagens continuarem a ser reenviadas de nó para nó depois de chegar ao destino!**
- **Como lidar com isso?**





- **Autores propõem “*dead certificates*”**

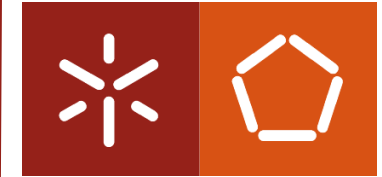
- Devem propagar-se também de forma epidémica... mas são mensagens muito mais pequenas
- O objectivo dos certificados é informar que a mensagem chegou ao destino e as cópias em buffer devem ser eliminadas imediatamente

- **Caso extremo na família de protocolos, que garante o tempo mínimo de entrega**

- Pacotes seguem sempre por todas as ligações possíveis

- **Aspecto mais crítico: espaço de armazenamento temporário (buffers)**

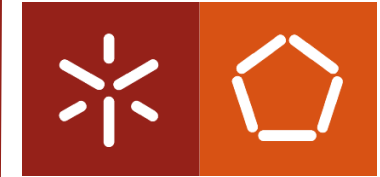
- Estratégias de gestão FIFO ou outras
- Ex: descarte de mensagens com base no histórico de probabilidade de sucesso na entrega



- Outras estratégias baseiam-se na limitação do número de réplicas das mensagens (tentando evitar uma cópia por nó)

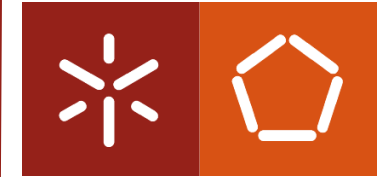
## (2) Contacto directo

- O emissor aguarda até que entre em contacto com o receptor, para lhe entregar a mensagem
- Só há uma única cópia da mensagem na rede
- Caso extremo inverso: não consome quase nenhuns recursos, mas diminui muito a probabilidade de entrega
  - Não escala: probabilidade de entrega tende para zero com aumento do número de nós na rede



## (3) Reenvio em 2 saltos (*two-hop relay*)

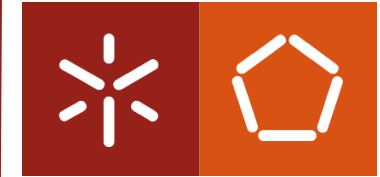
- O emissor copia a mensagem para os primeiros N nós com que contacta
  - Emissor armazena a mensagem até fazer N cópias
  - Nó que recebe a mensagem só entrega ao destinatário!
  - Número de cópias na rede até entrega:  $N + 1$
- Melhor probabilidade de entrega que o contacto directo
  - Aumentando o N diminui-se a latência e aumenta a probabilidade de entrega
  - Mas também é possível que as mensagens nunca sejam entregues
    - Basta não haver contacto dos nós intermédios com o destinatário
- Usada em estratégias de **mobilidade pro-activa**
  - nós de recolha que viajam pela rede em percursos planeados ou aleatórios para melhorar a conectividade de dois saltos



## (4) Reenvio em árvore (*Tree-Based Flooding*)

- O emissor copia a mensagem para os nós que contacta e diz-lhes também quantas cópias devem fazer
  - Árvores com profundidade 1 é equivalente ao reenvio em 2 saltos
- Muitas estratégias para limitar o número de cópias...
- Exemplo: cada nó pode copiar as vezes que quiser, mas a mensagem só atravessa no máximo N saltos (limita apenas a profundidade da árvore mas não a largura)
- Pode-se limitar também a largura impondo um máximo de cópias, além de um número máximo de saltos.

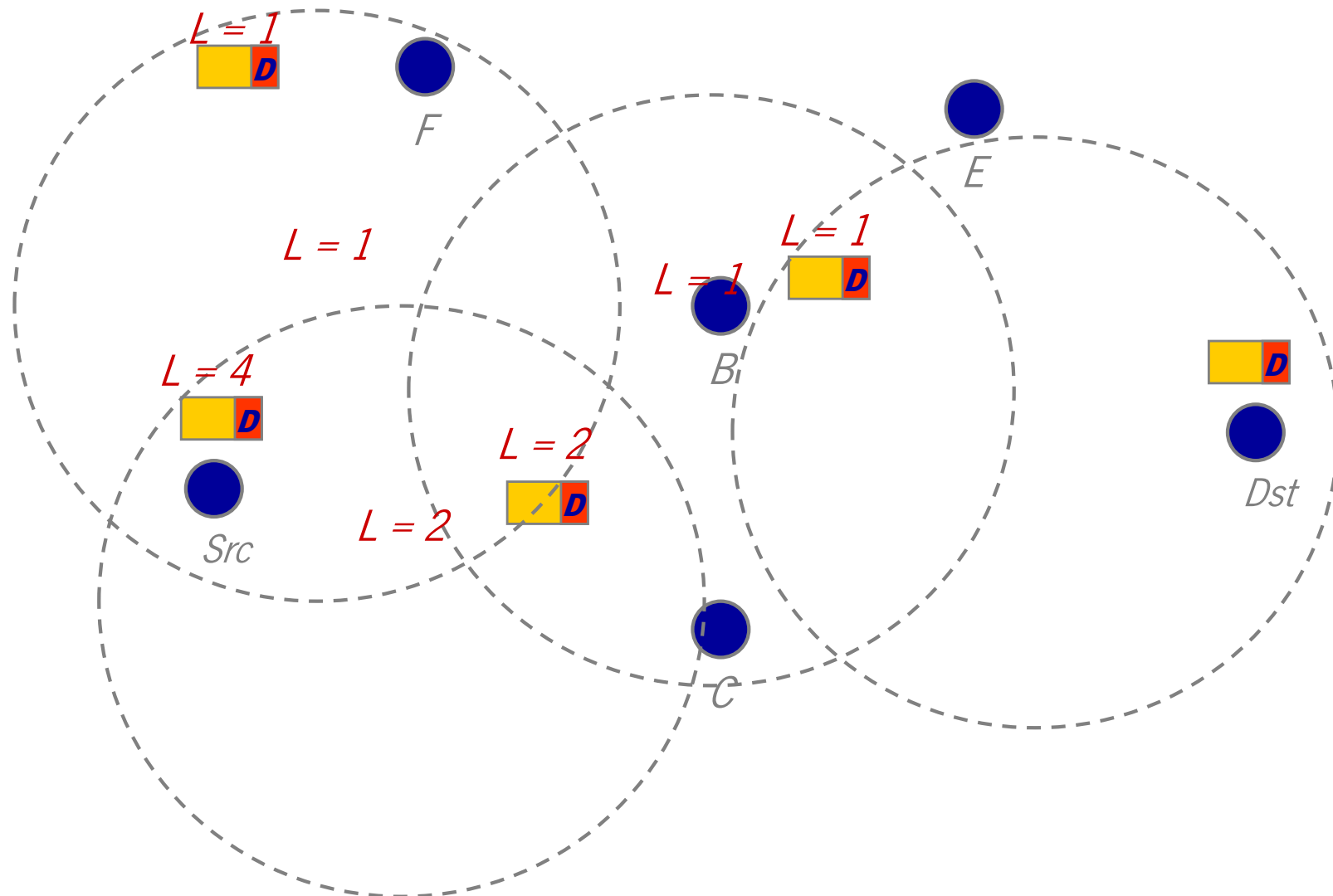


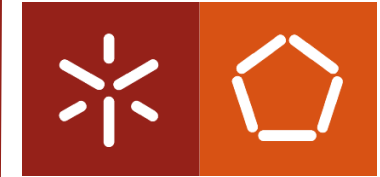


## (4) Reenvio em árvore (*continuação*)

- Mais complexo é o mecanismo para limitar o número de réplicas exactamente a N
  - Cada nó distribui sempre a responsabilidade de fazer metade das cópias actuais que tem para fazer para qualquer outro nó que contacta;
  - Este esquema é ótimo se as probabilidades de contacto inter-nós forem independentes e uniformemente distribuídas
- As estratégias em árvore conseguem entregar mensagens a múltiplos saltos de distância mas a afinação dos parâmetros pode ser complicada...

# DTN: Encaminhamento

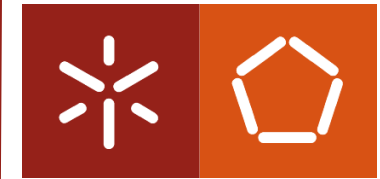




- **Estratégias baseadas em informação de reenvio**

- **Baseado na localização**

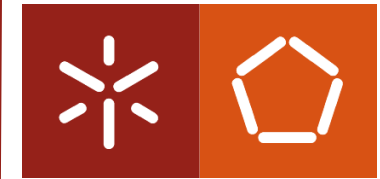
- Encaminhar na direção esperada do destinatário (obter informação de posição de algum modo, ex GPS)
- Vantagens: não são necessárias tabelas de encaminhamento, uma vez que um nó “só” necessita de conhecer a sua posição, a posição do destino e as posições dos (potenciais) *next hops*! Com base nessas posições um nó determina as distâncias em relação ao destino para tomar a melhor decisão de encaminhamento
- Desvantagens: pode falhar porque o facto de um nó estar próximo de outro não quer dizer que consigam comunicar (obstáculos); os nós estão em movimento e por isso a posição é um parâmetro muito dinâmico.



- **Estratégias baseadas em informação de reenvio (continuação)**

- **Baseado num gradiente**

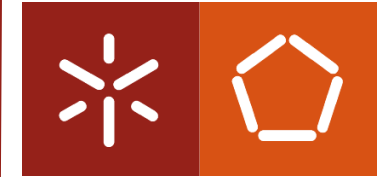
- Cada nó tem uma métrica associada que define a probabilidade desse nó entregar uma mensagem ao destino. O encaminhamento é feito com base nessa métrica, ou seja, encaminha se encontra um nó com maior probabilidade de fazer chegar a mensagem ao destino.
- Procura seguir pelo gradiente de melhoria do valor de uma função utilidade...
- Exige mais informação que a localização (cada nó tem que manter o valor da métrica para todos os destinos, o que faz com que a informação necessária ao cálculo da métrica tenha que ser propagada através da rede
- Alternativamente, usam-se parâmetros como o intervalo de tempo desde o último contacto com o nó destino, o número de vezes em que se contactou com o nó destino, etc.



- **Estratégias baseadas em informação de reenvio (continuação)**

- **Métricas atribuídas aos links**

- Mais próximo do encaminhamento “tradicional” mas com grandes problemas de adaptação às redes DTNs; As métricas mais comuns estão relacionados com o tempo de chegada ao destino (que se quer seja o mais cedo possível)
  - O tempo de atraso em cada ligação é influenciado por 4 tempos diferentes e independentes: tempo de transmissão, tempo de propagação, tempo em queue e tempo de espera por contacto
  - Para calcular o atraso em cada ligação é necessário conhecer o escalonamento dos contactos que influenciam o tempo de espera.
  - O tempo em queue é frequentemente ignorado para simplificar o algoritmo;
  - Se não houver informação precisa sobre o escalonamento de contactos podem usar-se aproximações com base num histórico.
  - Source Routing versus Per Hop Routing é outro aspeto relevante. O source routing pode ser uma boa opção se o escalonamento de contactos for conhecido à partida.



## Bibliografia

- Evan P.C. Jones and Paul A.S. Ward, Routing Strategies for Delay-Tolerant Networks, 2006.
- Fall, K. Farrell, S. DTN: an architectural retrospective, Selected Areas in Communications, IEEE Journal, June 2008 Volume: 26, Issue: 5
- RFC4838 - Delay-Tolerant Networking Architecture (2007)
- RFC 9171 – Bundle Protocol Version 7 (Janeiro 2022)
- RFC 9172 - Bundle Protocol Security (BPSec) (Janeiro 2022)
- <https://wiki.ietf.org/group/dtn/DtnUseCases>
- <https://datatracker.ietf.org/wg/dtn/documents/>