

Multicast network capacity and how to achieve it

Leandros Tassiulas

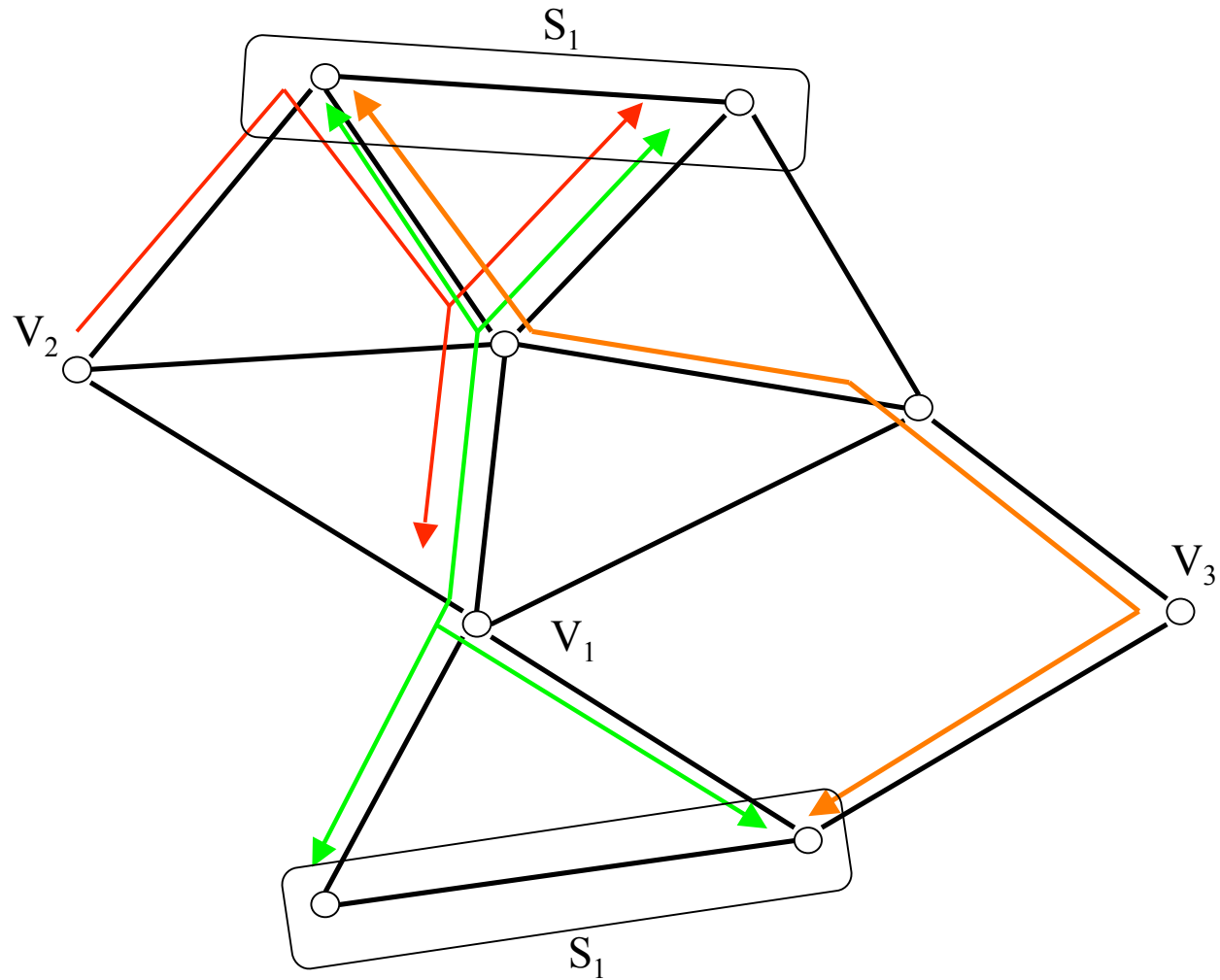
UNIVERSITY OF THESSALY
VOLOS /
GREECE
www.inf.uth.gr/~leandros

Outline

- Multicast network model
- Capacity characterization – segregated flows (no network coding)
- Achieving network capacity
 - multicast flow scheduling and prioritization
 - load balancing
- Complexity issues- implementation
- Wireless links, effect on capacity characterization
- Dynamically varying topology

Multicast Network

- Consider N multicast sessions $(v_1, S_1), (v_2, S_2), \dots, (v_N, S_N)$
 - v_n : Information Source
 - S_n : Group of intended destinations for information source v_n
- τ_n : Collection of directed trees rooted at v_n with leaves ending in the set of nodes S_n that may carry session n traffic
- τ_n may include
 - All multicast trees rooted at v_n with leaves terminating in S_n
 - Some pre-selected multicast trees.



**One multicast tree per session is depicted,
there are three sessions**

Network capacity

a_n traffic generation rate of session n

$\alpha = (a_n, n = 1, 2, \dots, N)$ traffic load vector

Capacity region C includes all feasible load vectors α

The traffic of session n is split to several trees in τ_n
that may support session (v_n, S_n)

Necessary and sufficient throughput feasibility condition

A collection of traffic rates a_n , $n = 1, 2, \dots, N$ is feasible if there exist a traffic splitting a_n^m , $m = 1, 2, \dots, M_n$ for each session n ,

$$a_n = \sum_{m=1}^{M_n} a_n^m$$

such that the capacity condition is satisfied

$$\sum_{n=1}^N \sum_{m=1}^{M_n} (a_n^m T_n^m) \leq C$$
$$C = (C_e : e \in E)$$

C_e : Capacity of link e

T_n^m : The m^{th} multicast tree that may carry session n traffic represented by a binary indicator vector $T_n^m = (t_e : e \in E)$

Verifying feasibility NP-hard, Steiner tree packing problem

Source of complexity:

The potentially huge number of trees supporting a multicast session

If we wish to exploit all the capacity out of the network we need to consider them all

Alternatively, we pre-select a family of multicast trees and the network operates such that all the capacity supportable by that family is exploited

Achieving capacity

Static approach

Given a load vector

Solve the throughput feasibility linear program

Split each session traffic to trees in proportions indicated by the solution

Support traffic of each tree by segregated BW allocation (TDMA)

Dynamic approach

Without knowing the load vector

Prioritize and schedule the traffic of each tree within the network

Balance appropriately session traffic load to the trees at the edge

Such that traffic load is supported if feasible

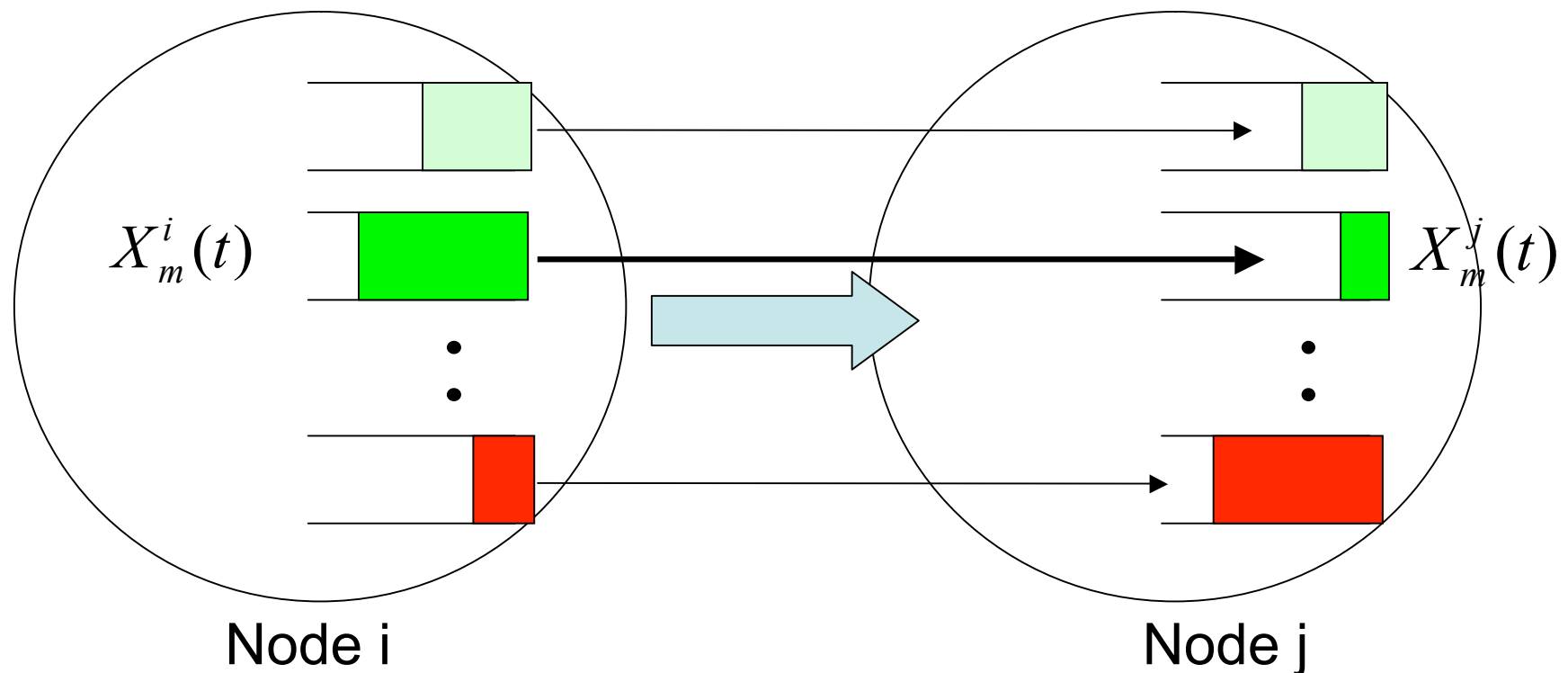
Backpressure and per link priority scheduling

Inside the network: unicast virtual circuit network

$X_m^i(t)$: VC m backlog at node i

Transmit a packet of VC m for which $X_m^i(t) - X_m^j(t)$

is maximum among all eligible VC's



Backpressure and per link priority scheduling Inside the network

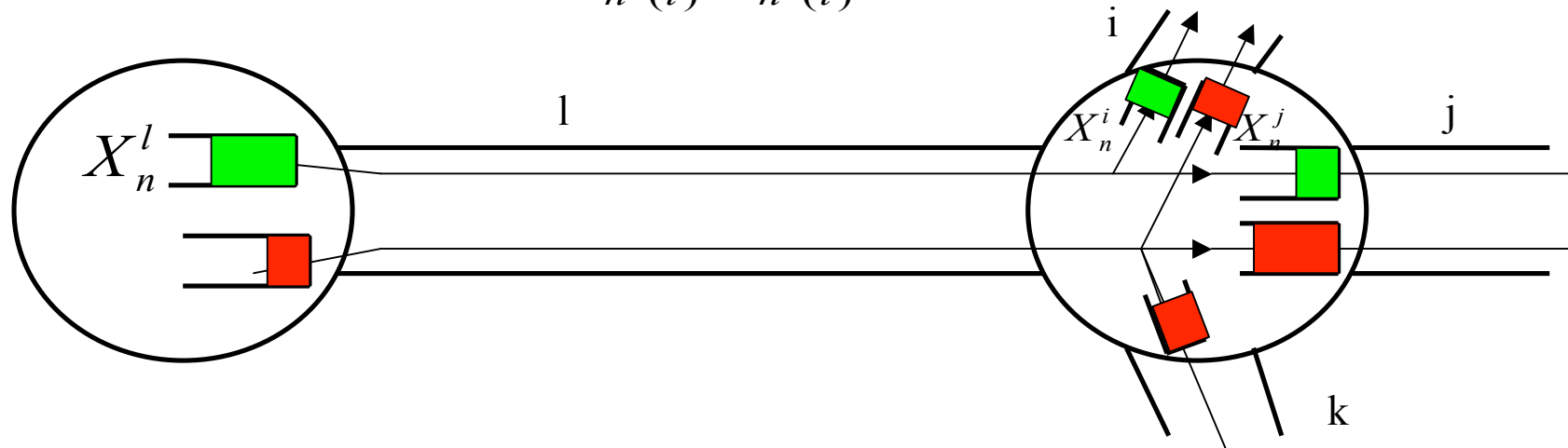
- $X_n^l(t)$: Backlog of tree n traffic in front of link l

$$W_n^l(t) = X_n^l(t) - \max_{k \text{ is a descendent of } l} X_n^k(t)$$

- $W_n^l(t)$: Weight (backlog gradient) of tree n at link l
- $b_n^l W_n^l(t)$: Priority index of tree n through link l .

$$n^l(t) = \arg \max_n b_n^l W_n^l(t)$$

if $b_{n^l(t)}^l W_{n^l(t)}^l(t) < -T_l$ then idle



Traffic splitting among trees at the source: Load balancing

Rule 1: at the source node the traffic is assigned to the multicast tree with minimum local backlog

Rule 2: at the source node the traffic is assigned to the multicast tree with minimum weight, where the weight of a tree is the sum of the weights of its links and the weight of a link is the maximum traffic backlog through the link.

The combination of the link scheduling prioritization scheme with either of the load balancing rules for traffic assignment achieve maximum throughput

Wireless mobile network model

- Collection of **wireless** nodes **moving** over a terrain
- Nodes **control** transmission **power**, **access decision** (transmit, don't transmit, which code (in CDMA) etc.), other **physical layer parameters** represented collectively by vector $I(t)$
- The environment changes as well due to mobility of the nodes and the environment itself; **“topology”** $S(t)$
- $C_{ij}(t) = C_{ij}(S(t), I(t))$: rate of bit pipe from i to j at t
- $C(t)$ communication topology at time t determined partly by environment $S(t)$ (uncontrollable), physical and access layer decisions $I(t)$ (controllable)

Throughput capacity at the access layer

- **Access Control vector $I(t)$** represents the selection of various access and physical layer parameters at t
- **Access Control policy** designates $I(t)$, $t=1,2,\dots$ $I(t)$ in A where A the collection of all possible access control vectors

Capacity region $C(s)$ for fixed topology state s includes all rate vectors realized by any access policy

$C(s)$ the convex hull of $\{C(s,I): I \text{ in } A\}$

Capacity region C the expectation of $C(s)$ with respect to the stationary distribution of topology process $S(t)$ i.e.

$$C = \{C: C = E[C(s)], C(s) \in C(s)\}$$

Necessary and sufficient throughput feasibility condition

A collection of traffic rates a_n , $n = 1, 2, \dots, N$ is feasible if there exist a traffic splitting a_n^m , $m = 1, 2, \dots, M_n$ for each session n ,

$$a_n = \sum_{m=1}^{M_n} a_n^m$$

such that the capacity condition is satisfied

$$\sum_{n=1}^N \sum_{m=1}^{M_n} (a_n^m T_n^m) \leq C$$

C belongs to the link capacity region

T_n^m : The m^{th} multicast tree that may carry session n traffic represented by a binary indicator vector $T_n^m = (t_e : e \in E)$

Verifying feasibility NP-hard, Steiner tree packing problem (at least)

Dynamic network control and mobility

- Multicast trees might break due to mobility
- The transient traffic needs to be accommodated somehow
- Instead of talking about trees, each packet carries in its header the group of destinations
- In every packet replication it is as if a packet is routed from one multicast session to a new one that corresponds to the new set of destinations