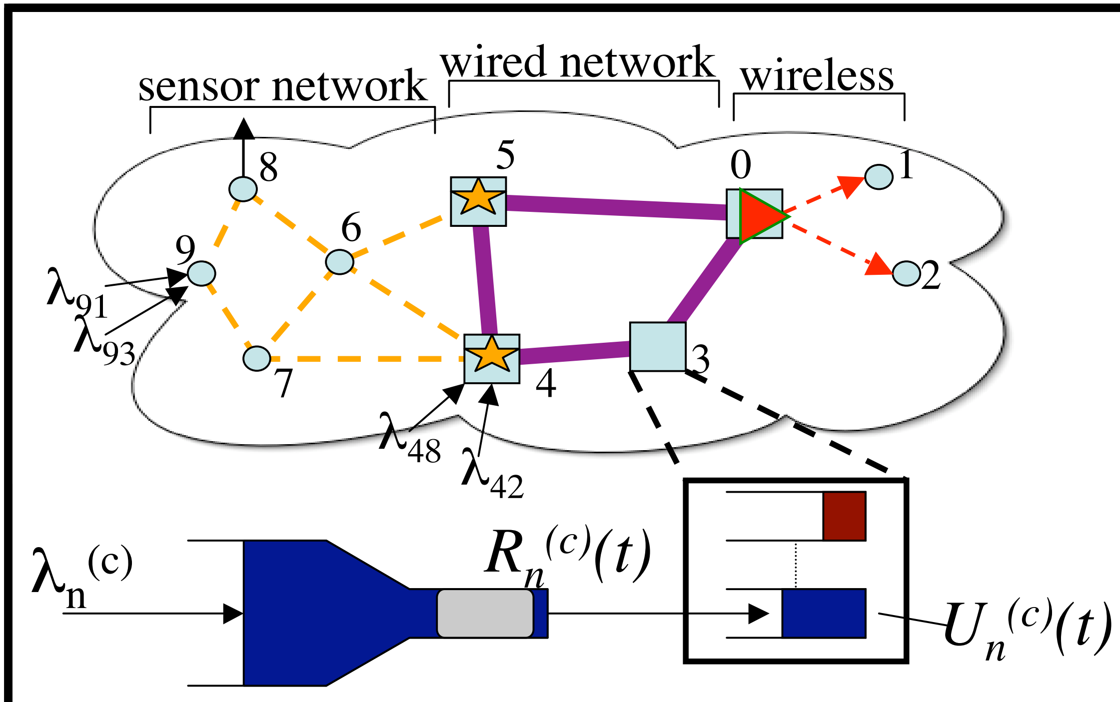
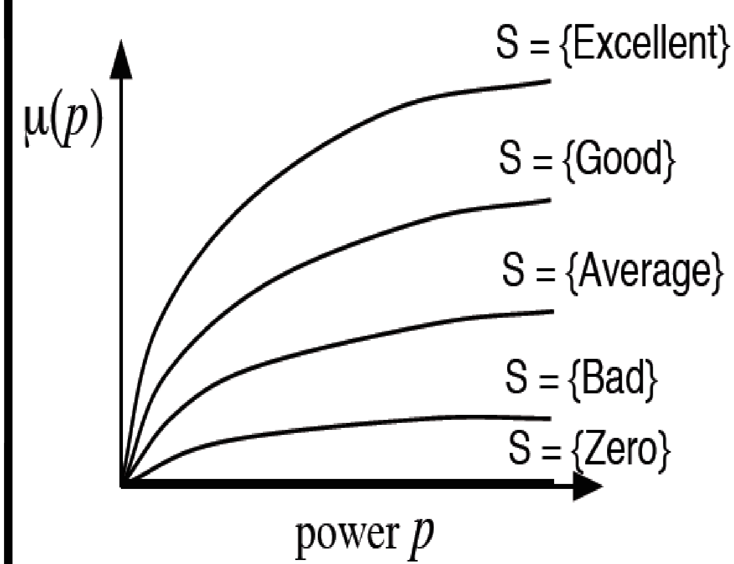
USC Viterbi School of Engineering

Fairness and Optimal Stochastic Control for Heterogeneous Networks

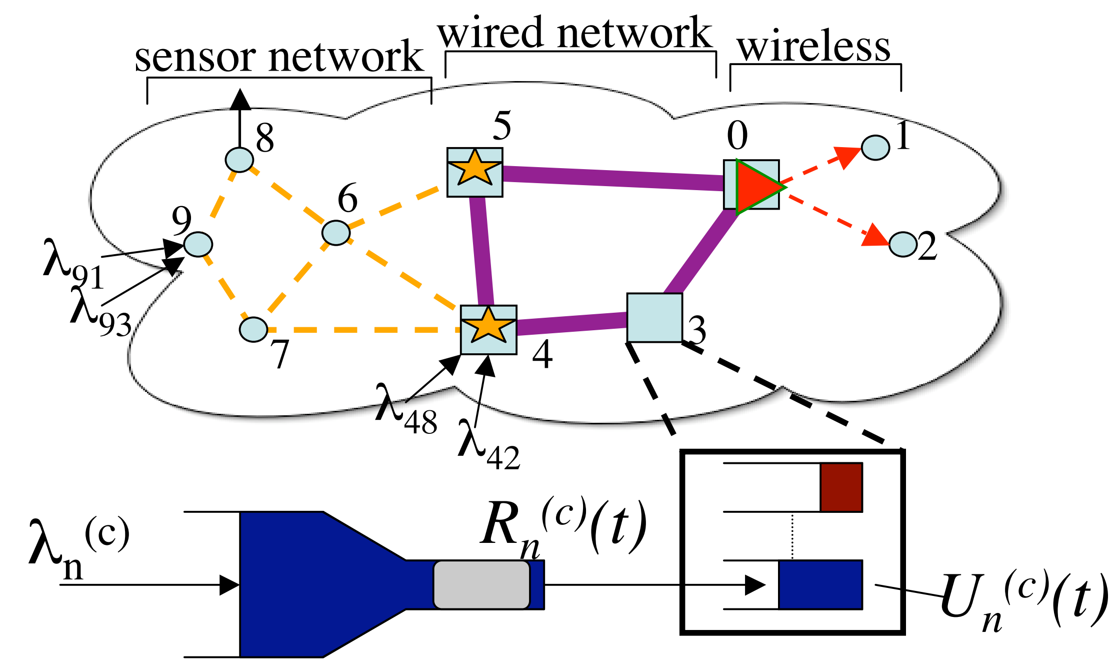




Time-Varying Channels

Michael J. Neely (USC)
Eytan Modiano (MIT)
Chih-Ping Li (USC)

A ***heterogeneous network*** with N nodes and L links:

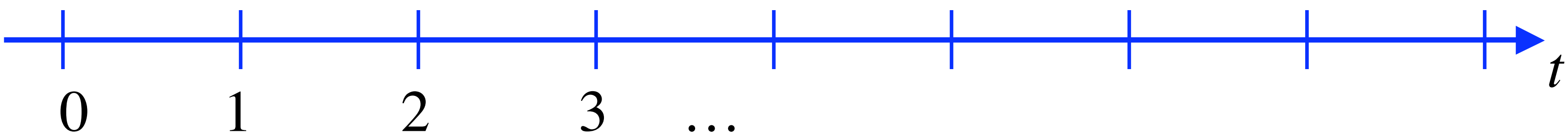


$\Gamma_{\vec{S}}$ = channel dependent set of transmission rate matrices

$$\Gamma_{\vec{S}} = \Gamma^A_{\vec{S}_A} \times \Gamma^B \times \Gamma^C_{\vec{S}_C}$$

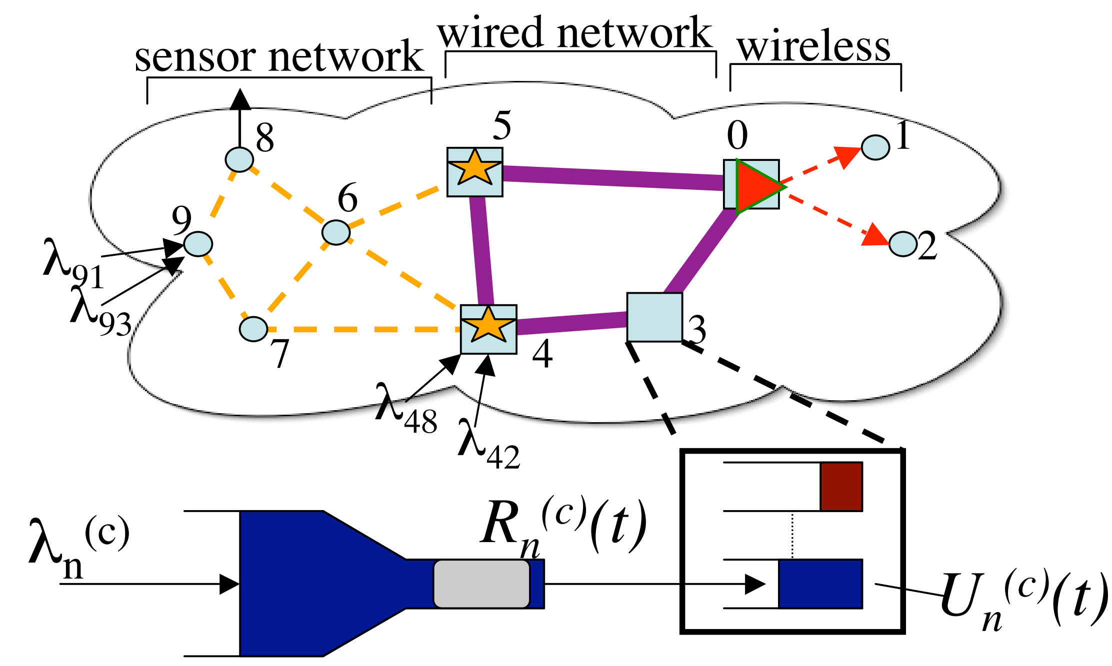
Choose $\vec{\mu}(t) \in \Gamma_{\vec{S}(t)}$

Slotted time $t = 0, 1, 2, \ldots$



Traffic ($A_{ij}(t)$) and channel states $\vec{S}(t)$ i.i.d. over timeslots…

A ***heterogeneous network*** with N nodes and L links:



$\Gamma_{\vec{S}}$ = channel dependent set of transmission rate vectors

$$\Gamma_{\vec{S}} = \Gamma^{A}_{\vec{SA}} \times \Gamma^{B} \times \Gamma^{C}_{\vec{SC}}$$

Choose $\vec{\mu}(t) \in \Gamma_{\vec{S}(t)}$

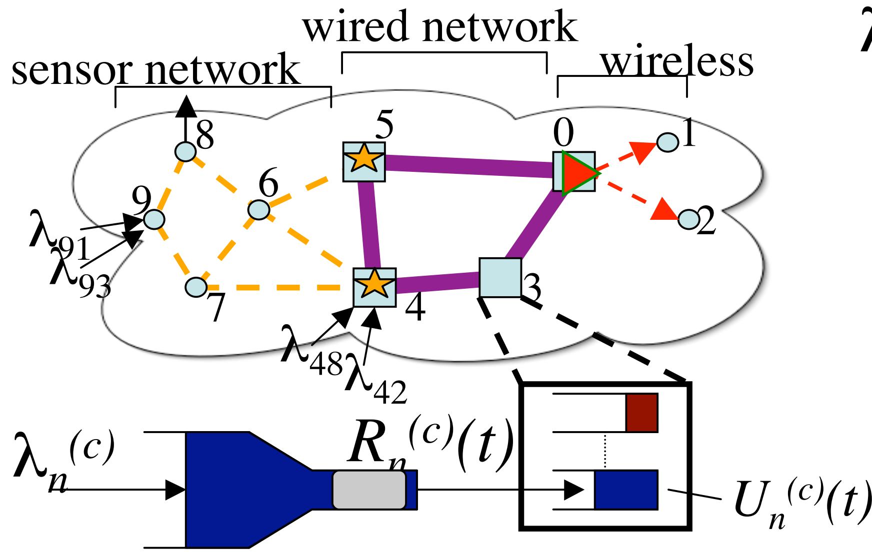
Input rate matrix: (λ_{ij}) (where $E[A_{ij}(t)] = \lambda_{ij}$)

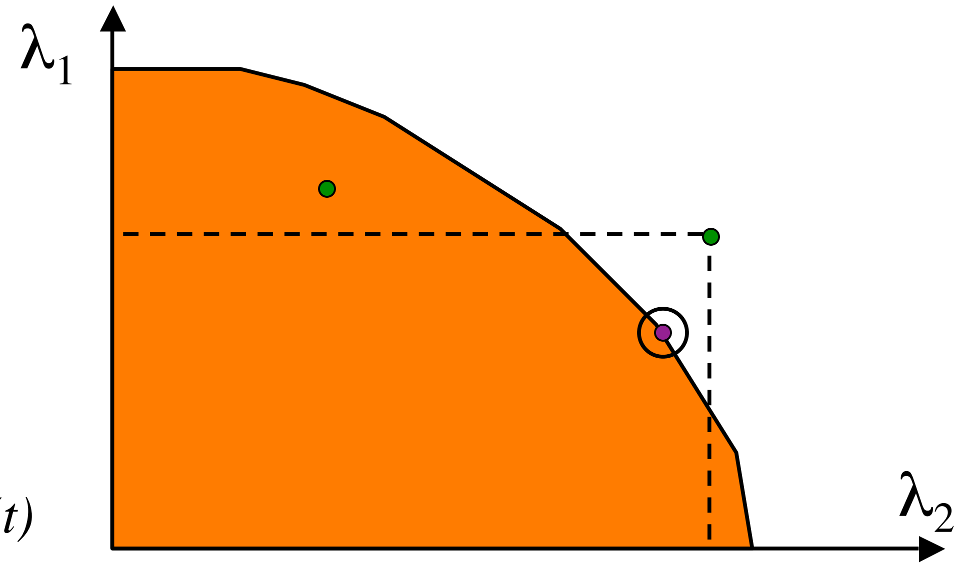
Channel state vector: $\vec{S}(t) = (S_1(t), S_2(t), \ldots, S_L(t))$

Transmission rate vector: $\vec{\mu}(t) = (\mu_1(t), \mu_2(t), \ldots, \mu_L(t))$

Resource allocation: choose $\vec{\mu}(t) \in \Gamma_{\vec{S}(t)}$

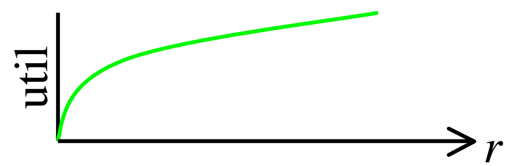
Goal: Develop joint *flow control*, *routing*, *resource allocation*





Λ = Capacity region (considering all routing, resource alloc. policies)

$g_{nc}(r_{nc})$ = concave utility functions



Maximize:	$\sum_{n,c} g_{nc}(r_{nc})$
Subject to:	$(r_{nc}) \in \Lambda$
	$0 \leq (r_{nc}) \leq (\lambda_{nc})$

Some precedents:

Static optimization: (Lagrange multipliers and convex duality)

Kelly, Maulloo, Tan, Oper Res. 1998 [pricing for net. optimization]
Xiao, Johansson, Boyd, Allerton 2001 [network resource opt.]
Julian, Chian, O'Neill, Boyd, Infocom 2002 [static wireless opt]
Lee, Mazumdar, Shroff, Infocom 2002 [static wireless downlink]
Marbach, Infocom 2002 [pricing, fairness static nets]
Krishnamachari, Ordonez, VTC 2003 [static sensor nets]
Low, TON 2003 [internet congestion control]

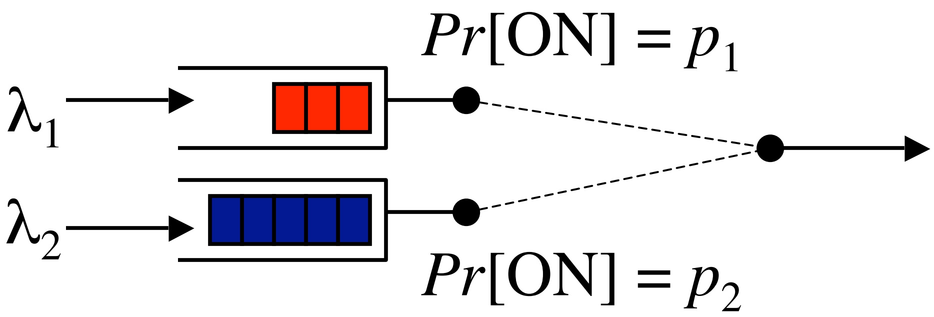
Dynamic control:

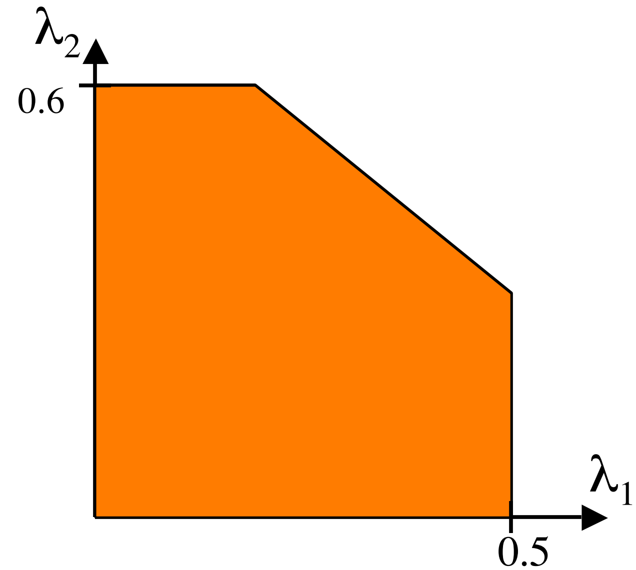
D. Tse, 97, 99 ["proportional fair" algorithm: max U_i/r_i]
Kushner, Whiting, Allerton 2002 ["prop. fair" alg. analysis]
S. Borst, Infocom 2003 [downlink fairness for infinite # users]
Li, Goldsmith, IT 2001 [broadcast downlink]
Tsibonis, Georgiadis, Tassiulas, Infocom 2003 [max thruput outside of capacity region]

Stochastic Stability via Lyapunov Drift:

Tassiulas, Ephremides, AC 1992, IT 1993 [MWM, Diff. backlog]
Andrews et. al., Comm. Mag, 2003 [server selection]
Neely, Modiano, Rohrs, TON 2003, JSAC 2005 [satellite, wireless]
McKeown, Anantharam, Walrand, Infocom 1996 [NxN switch]
Leonardi et. Al., Infocom 2001 [NxN switch]

Example: Server alloc., 2 queue downlink, ON/OFF channels





Capacity region Λ:

$$\boxed{\begin{array}{c} \lambda_1 \leq p_1 \quad , \quad \lambda_2 \leq p_2 \\ \lambda_1 + \lambda_2 \leq p_1 + (1 - p_1)p_2 \end{array}}$$

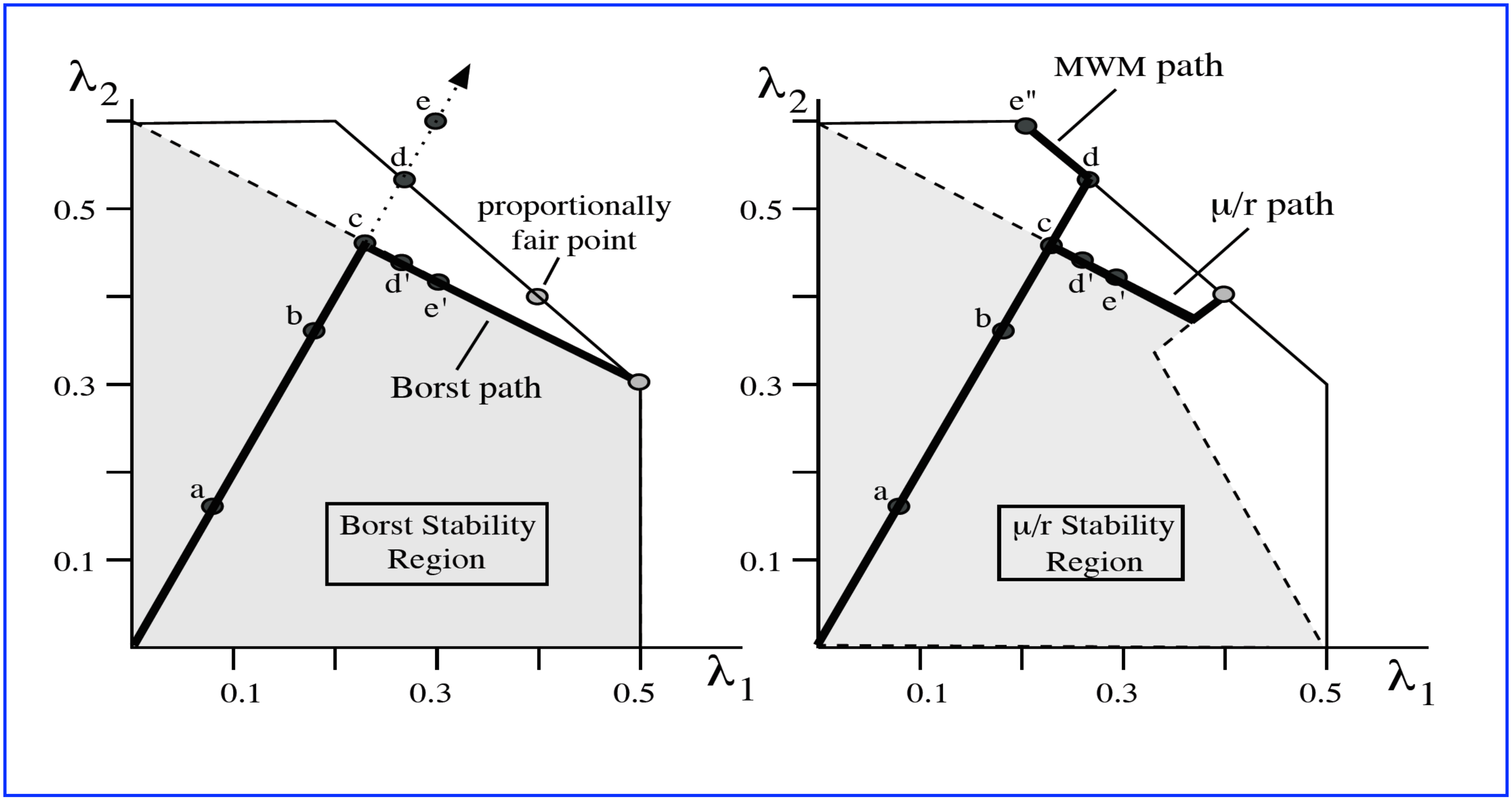
MWM algorithm (choose ON queue with largest backlog)
Stabilizes whenever rates are strictly interior to Λ
[Tassiulas, Ephremides IT 1993]

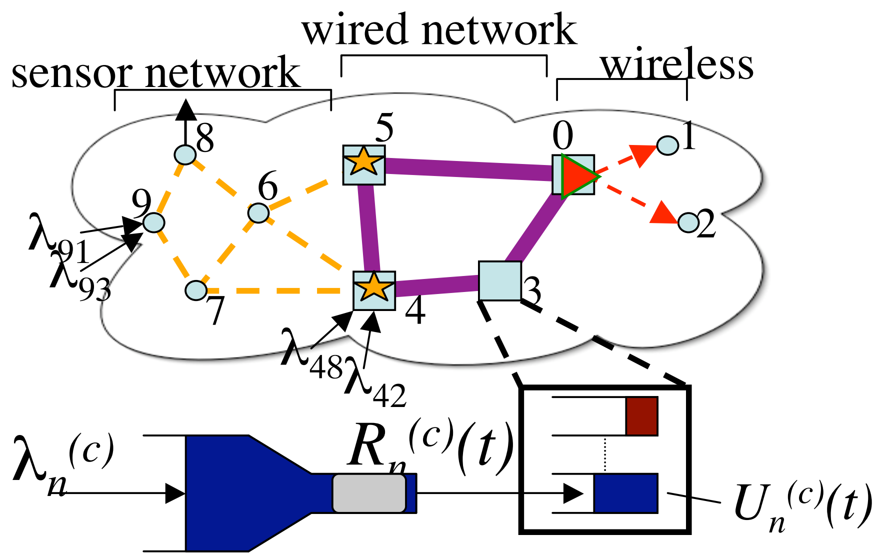
Comparison of previous algorithms:

(1) MWM (max $U_i\mu_i$)

(2) Borst Alg. [Borst Infocom 2003] (max $\mu_i/\overline{\mu_i}$)

(3) Tse Alg. [Tse 97, 99, Kush 2002] (max μ_i/r_i)

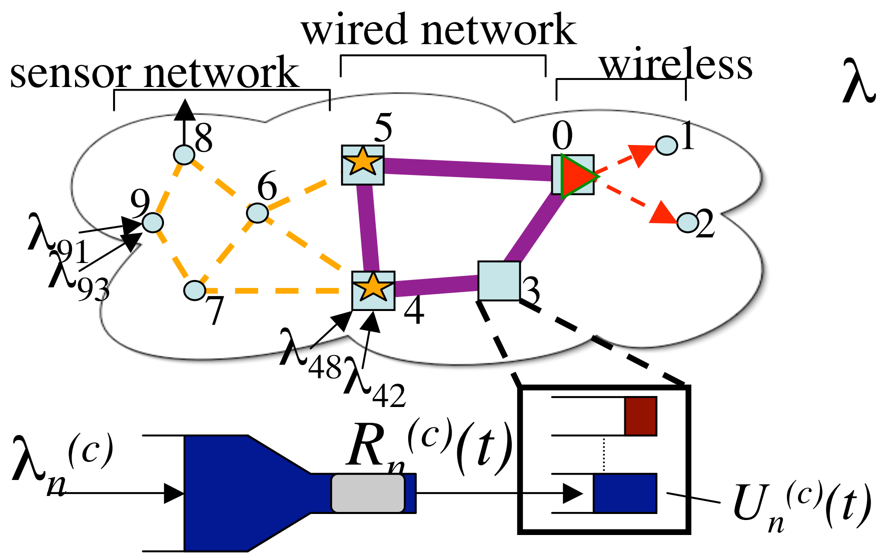


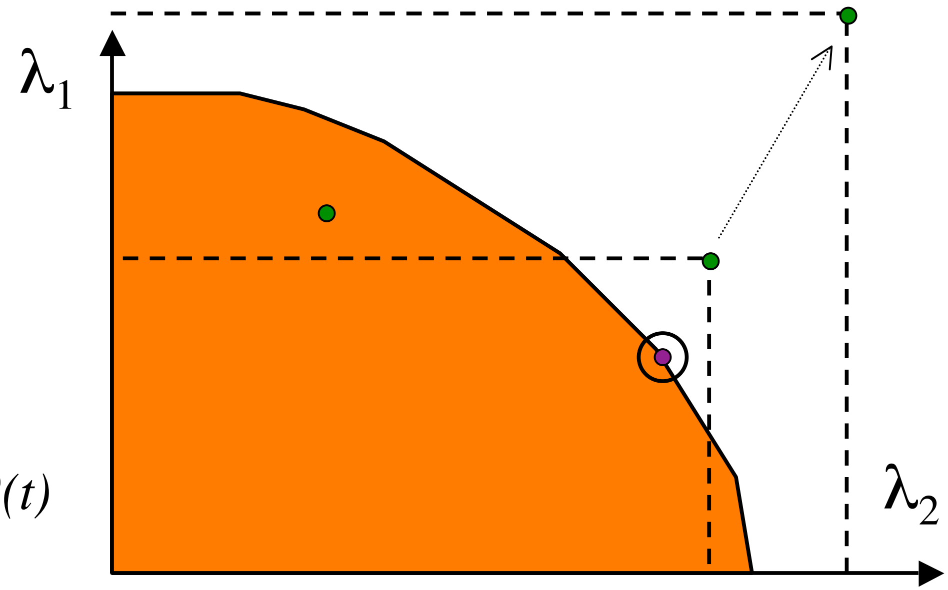


Approach: Put all data in a reservoir before sending into network. Reservoir valve determines $R_n^{(c)}(t)$ (amount delivered to network from reservoir (n,c) at slot t).

Optimize dynamic decisions over all possible *valve control policies*, *network resource allocations*, *routing* to provide optimal fairness.

Part 1: Optimization with <u>infinite demand</u>



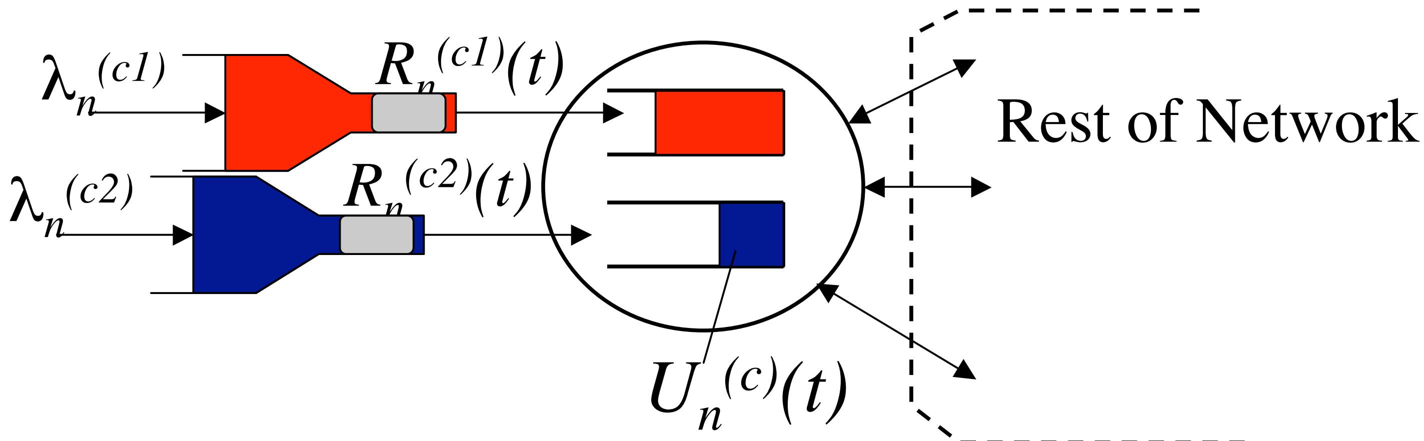


Assume all active sessions infinitely backlogged (general case of arbitrary traffic treated in part 2).

Maximize:	$\sum_{n,c} g_{nc}(r_{nc})$
Subject to:	$(r_{nc}) \in \Lambda$
	$0 \leq (r_{nc}) \leq (\lambda_{nc})$

Cross Layer Control Algorithm (**CLC1**):

(1) <u>Flow Control</u>: At node n, observe queue backlogs $U_n^{(c)}(t)$ for all active sessions c.

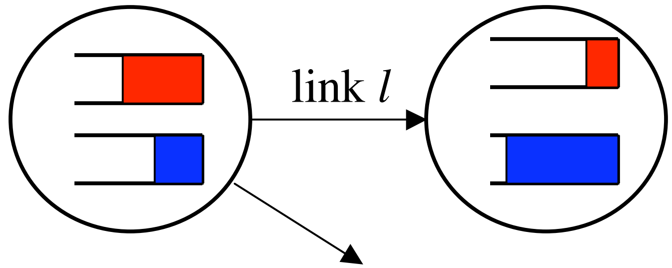


$$\text{Maximize}: \quad \sum_{c=1}^{N} \left[V g_{nc}(r_{nc}) - 2 r_{nc} U_n^{(c)}(t) \right]$$

$$\text{Subject to:} \quad \sum_{c=1}^{N} r_{nc} \leq R_n^{max}$$

(where V is a parameter that affects network delay)

(2) Routing and Scheduling:



$c_l{*}(t) =$

For all links l, find the commodity $c_l^*(t)$ such that:

$$c_l^*(t) = \arg\max_c \left\{ U^c_{tran(l)}(t) - U^c_{rec(l)}(t) \right\}$$

and define:

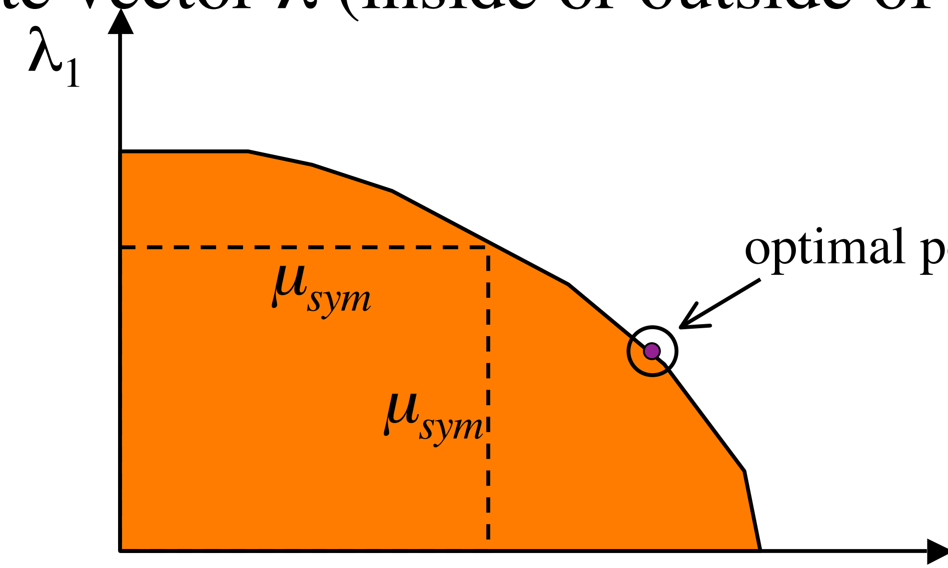
$$W_l^*(t) = \max[U^{c_l^*}_{tran(l)}(t) - U^{c_l^*}_{rec(l)}(t), 0]$$

(similar to the original Tassiulas <u>*differential backlog*</u> routing policy [1992])

(3) Resource Allocation: Observe channel states $\vec{S}(t)$. Allocate resources to yield rates $\vec{\mu}(t)$ such that:

Maximize: $\sum_l W_l^*(t)\mu_l(t)$ Such that: $\vec{\mu}(t) \in \Gamma_{\vec{S}(t)}$

<u>Theorem</u>: If channel states are i.i.d., then for any $V>0$ and any rate vector $\vec{\lambda}$ (inside or outside of Λ),



Avg. delay: $$\overline{\sum_{nc} U_n^{(c)}} \leq \frac{N(B + VG_{max})}{2\mu_{sym}}$$

Fairness: $$\sum_{nc} g_{nc}(\overline{r}_{nc}) \geq \sum_{nc} g_{nc}(r^*_{nc}) - \frac{BN}{V}$$

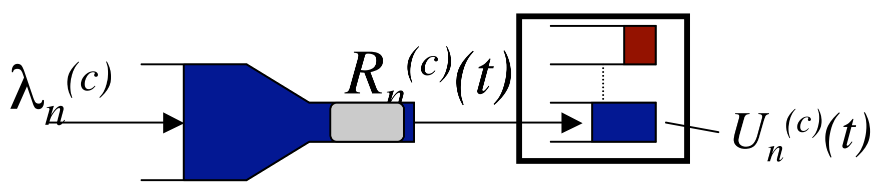
(where $B \triangleq \left(\mu_{max}^{in} + \frac{1}{N}\sum_{n=1}^{N} R_n^{max}\right)^2 + (\mu_{max}^{out})^2$)

Special cases:
(for simplicity, assume only 1 active session per node)

1. Maximum throughput and the threshold rule

Linear utilities: $g_{nc}(r) = \alpha_{nc}\, r$

$$R_{nc_n}(t) = \begin{cases} R_n^{max} & \text{if } U_n^{(c_n)}(t) \leq \frac{V\alpha_{nc_n}}{2} \\ 0 & \text{otherwise} \end{cases}$$

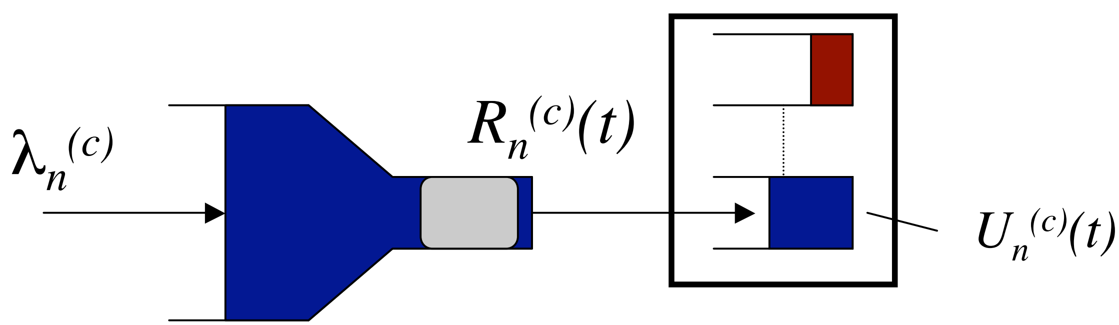


(threshold structure similar to Tsibonis [Infocom 2003] for a downlink with service envelopes)

(2) Proportional Fairness and the 1/*U* rule

logarithmic utilities: $g_{nc}(r) = \log(1 + r_{nc})$

$$R_{nc_n}(t) = \min\left[\max\left[\frac{V}{2U_n^{(c_n)}(t)} - 1, 0\right], R_n^{max}\right]$$

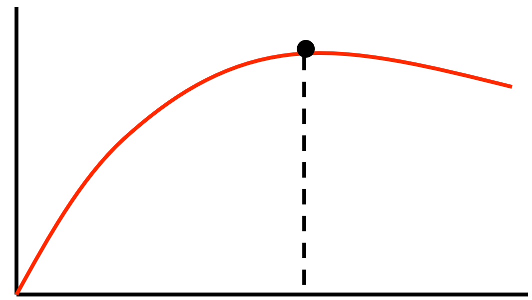


Mechanism Design and Network Pricing:

greedy users…each naturally solves the following:

Maximize: $g_{nc}(r)$ - $PRICE_{nc}(t)r$

Such that : $0 \leq r \leq R_{max}$



This is exactly the same algorithm if we use the following *dynamic pricing strategy*:

$$\boxed{PRICE_{nc}(t) = U_{nc}(t)/V}$$

Analytical technique: ***Lyapunov Drift***

Lyapunov function: $$L(\vec{U}(t)) = \sum_n U_n^2(t)$$

Lyapunov drift: $$\Delta(t) = E[L(\vec{U}(t+1)) - L(\vec{U}(t)) \mid \vec{U}(t)]$$

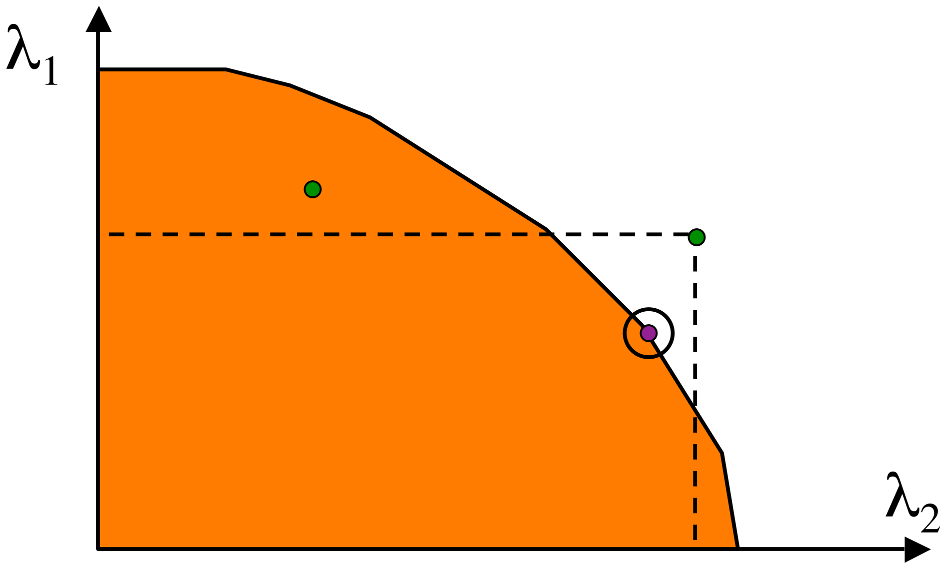
<u>Theorem</u>: (Lyapunov drift with Utility Maximization)

If for all t: $$\Delta(t) \leq C - \varepsilon\sum_n U_n(t) - VE[g(\vec{r}(t))|\vec{U}(t)] - Vg(\vec{r}^*)$$

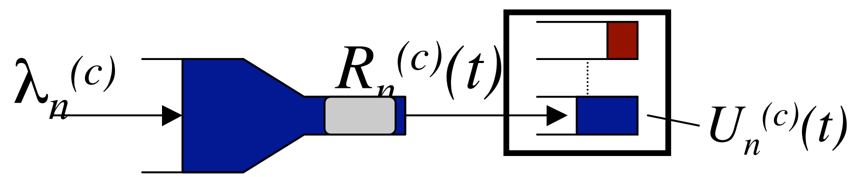
Then: (a) $$\sum_n E[U_n] \leq \frac{C + VNG_{max}}{\varepsilon}$$ (stability and bounded delay)

(b) $$g(\vec{r}_{achieve}) \geq g(\vec{r}^*) + C/V$$ (resulting utility) □

Part 2: Scheduling with ***arbitrary input rates***



$$\begin{array}{ll} \text{Maximize:} & \sum_{n,c} g_{nc}(r_{nc}) \\ \text{Subject to:} & (r_{nc}) \in \Lambda \\ & 0 \leq (r_{nc}) \leq (\lambda_{nc}) \end{array}$$



Novel technique of creating ***flow state variables*** $\boldsymbol{Z_{nc}(t)}$

$$Y_{nc}(t) = R_{max} - R_{nc}(t)$$

$$Z_{nc}(t) = max[Z_{nc}(t) - g_{nc}(t), 0] + Y_{nc}(t)$$

(Reservoir buffer size arbitrary, *possibly zero*)

Cross Layer Control Alg. 2 (CLC2) Every timeslot and for each node n, choose $R_{nc}(t) = r_{nc}$ to solve:

$$\begin{array}{ll} \text{Maximize:} & \sum_c [\frac{Z_{nc}(t)}{N} - U_n^{(c)}(t)] r_{nc} \\ \text{Subject to:} & \sum_c r_{nc} \leq R_n^{max} \\ & r_{nc} \leq L_{nc}(t) + A_{nc}(t) \end{array}$$

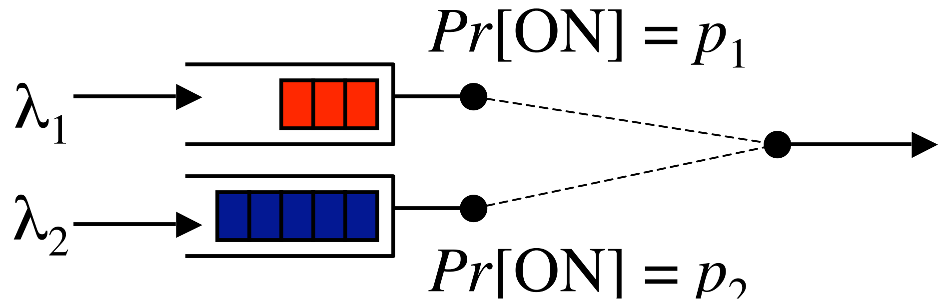
Additionally, the flow controllers at each node n choose $\gamma_{nc}(t)$ for each session (n, c) to solve:

$$\begin{array}{ll} \text{Maximize:} & V g_{nc}(R_n^{max} - \gamma_{nc}) + \frac{2Z_{nc}(t)}{N}\gamma_{nc} \\ \text{Subject to:} & 0 \leq \gamma_{nc} \leq R_n^{max} \end{array}$$

The flow states $Z_{nc}(t)$ are then updated according to the $Z_{nc}(t+1)$ iteration of the previous slide.

Simulation Results for CLC2:

(i) 2 queue downlink



a) $g_1(r)=g_2(r)=$

$\log(1+r)$

b) $g_1(r)=\log(1+r)$
$g_2(r)=1.28\log(1+r)$

(priority service)

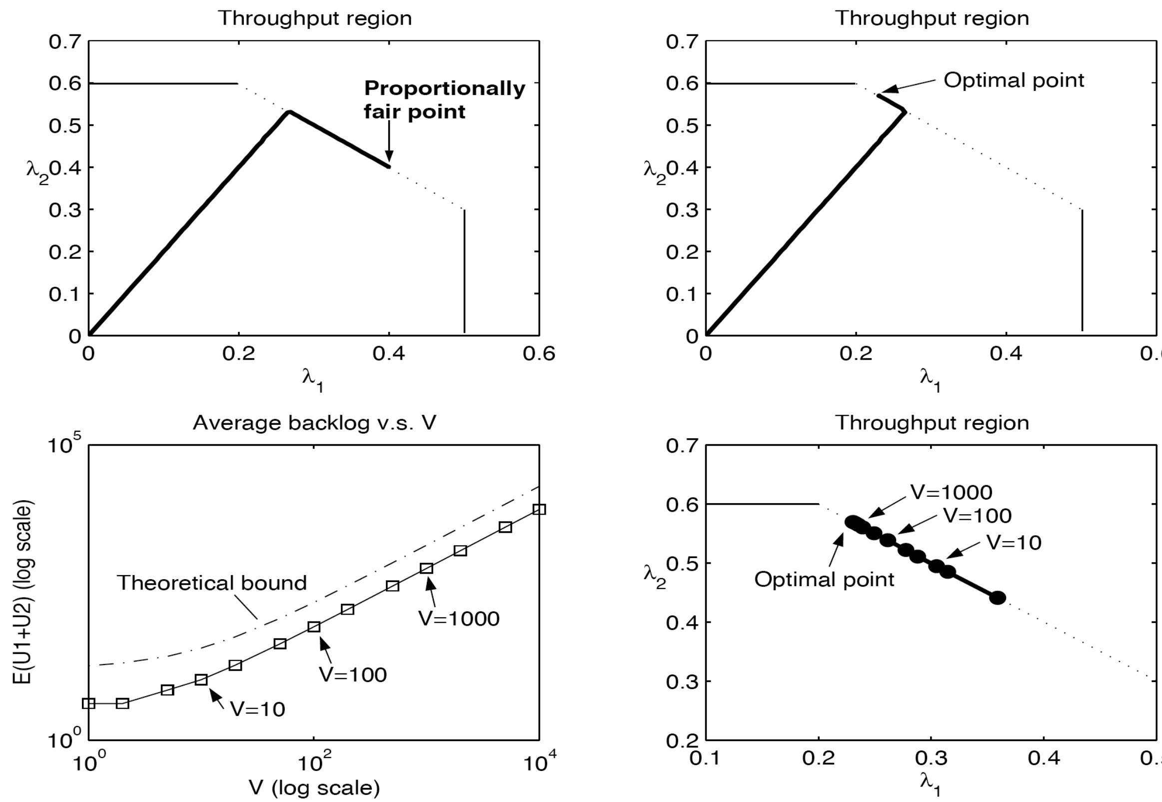


Fig. 3. Simulation of CLC2: (a) Linearly increasing (λ_1, λ_2) to $(0.5, 1.0)$ for $V = 10000$ and $g_1(r) = g_2(r) = \log(1 + r)$. (b) Modifying utility 2 to: $g_2(r) = 1.28 \log(1 + r)$. (c)-(d) Fixing $(\lambda_1, \lambda_2) = (0.5, 1.0)$ and illustrating delay and throughput versus V.

(ii) 3 x 3 packet switch under the crossbar constraint:

Rates (λ_{ij})

.45	.1	.4
.1	.7	.15
.4	.15	.4

Throughput (r_{ij})

.450	.100	.399
.100	.695	.148
.399	.149	.400

Backlog $(\overline{U}_{ij})$

3.3	2.4	3.6
2.4	2.9	2.7
3.6	2.7	3.4

(a) Simulation of a switch with feasible traffic

.6	.1	.3
0	.4	.2
0	.5	0

proportionally fair

Rates (λ_{ij})

.9	.2	.3
0	.4	.2
0	.5	0

Throughput (r_{ij})

.598	.100	.298
0	.399	.200
0	.500	0

Backlog $(\overline{U}_{ij})$

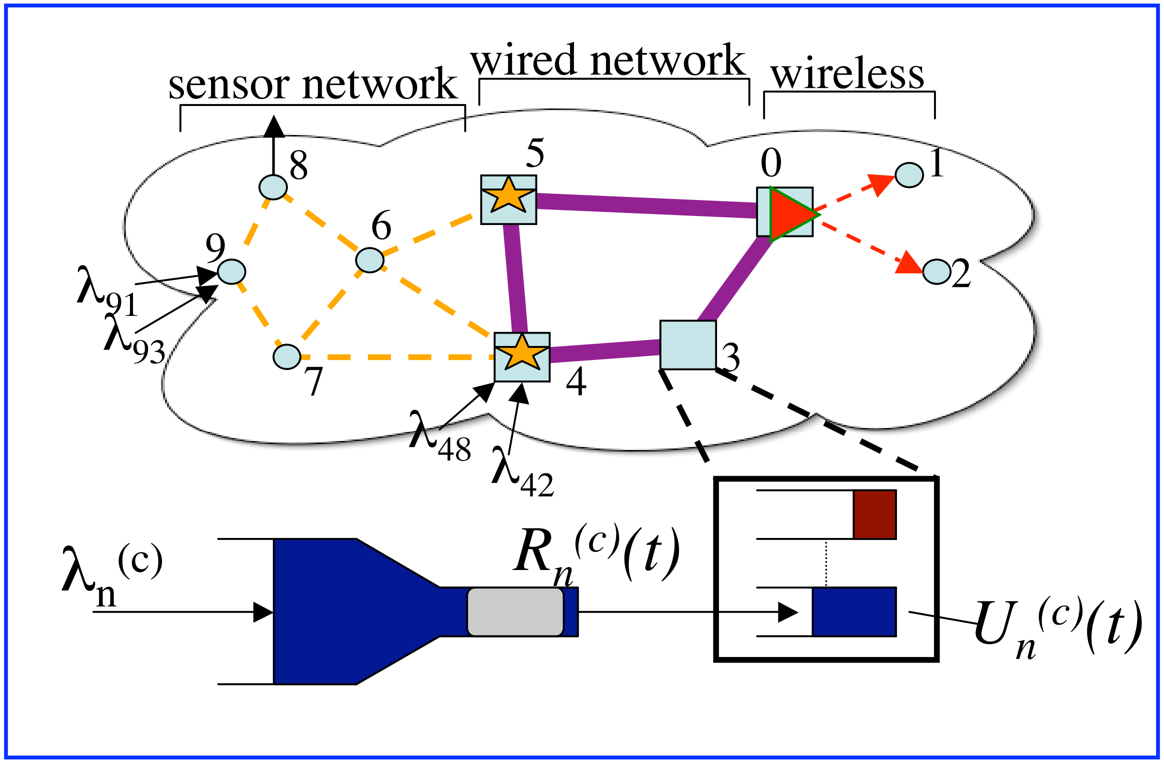
31.6	45.3	32.1
0	14.1	.29
0	14.2	0

(b) Simulation of an overloaded switch

Fig. 4. Simulation results for the CLC2 algorithm with $V = 100$ and zero reservoir buffers. Simulations were run over four million timeslots.

Concluding Slide:

(iii) Multi-hop Heterogeneous Network



$\lambda_{91} = \lambda_{93} = \lambda_{48} = \lambda_{42} = 0.7$ packets/slot (not supportable)

The optimally fair point of this example can be solved in closed form: $r_{91}* = r_{93}* = r_{48}* = 1/6 = 0.1667$, $r_{42} = 0.5$

Use CLC2, V=1000 ------> $\overline{U}_{tot}$ =858.9 packets
$r_{91} = 0.1658$, r_{93}=0.1662, r_{48}=0.1678, r_{42}=0.5000

The end

http://www-rcf.usc.edu/~mjneely/

The end

http://www-rcf.usc.edu/~mjneely/

[3]Strictly speaking, the proportionally fair allocation seeks to maximize $\sum_{nc} \log(r_{nc})$, leading to $\sum_{nc} \frac{\overline{r}_{nc}^{opt} - r_{nc}}{\overline{r}_{nc}^{opt}} \geq 0$ for any other operating point $(r_{nc}) \in \Lambda$. We use non-negative utilities $\log(1 + r)$, and thereby obtain a proportionally fair allocation with respect to the quantity $\overline{r}_{nc}^{opt} + 1$, leading to $\sum_{nc} \frac{\overline{r}_{nc}^{opt} - r_{nc}}{\overline{r}_{nc}^{opt} + 1} \geq 0$.