

МОДЕЛИРАНЕ И СРЕДСТВА ЗА СИМУЛАЦИЯ НА КОМПЮТЪРНИ И КОМУНИКАЦИОННИ МРЕЖИ

1

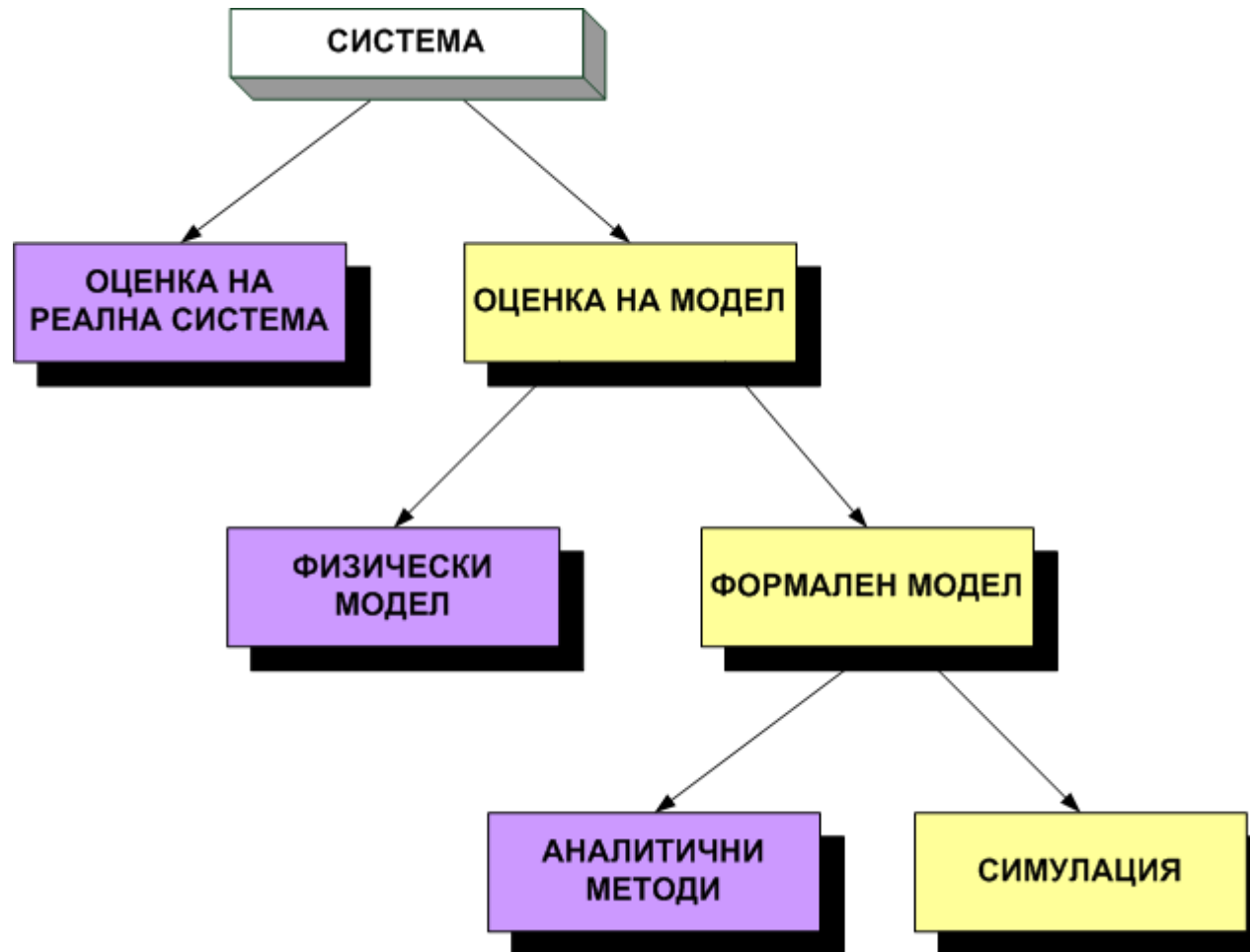
ПРОФ. ПЛАМЕНКА БОРОВСКА
КАТЕДРА
КОМПЮТЪРНИ СИСТЕМИ



СИМУЛАЦИЯ НА КОМПЮТЪРНИ И КОМУНИКАЦИОННИ МРЕЖИ

- Основен проблем при проектирането и имплементацията на комуникационни мрежи (респ., алгоритми, архитектури и протоколи) е оценката на тяхната производителност, както и анализът и визуализацията на микро- и макро поведението на системата и нейните компоненти
- За тази цел се използват три различни методологии:
 1. Експерименти (измервания) в реални системи и прототипи;
 2. Математически анализ;
 3. Симулация.

МЕТОДОЛОГИИ ЗА ОЦЕНКА НА СИСТЕМИ



КЛАСИФИКАЦИЯ НА МЕТОДИТЕ ЗА ОЦЕНКА НА СИСТЕМИТЕ

ОЦЕНКА НА РЕАЛНА СИСТЕМА

- **Изисквания:** съществуваща система, възпроизводимост, достъпност (в пространството и времето), безопасност, ниска цена
- **Примери:** измерване на трафика (телекомуникации, пътен трафик), оценка на прототип, тестване на софтуер

ОЦЕНКА НА МОДЕЛ

- **Изисквания:** знание за системата (кои части на системата са релевантни), валидност на модела, нагледни пояснения, податливост на обработка
- **Измерване на характеристиките:** ниво на абстракция

КЛАСИФИКАЦИЯ НА МЕТОДИТЕ ЗА ОЦЕНКА НА СИСТЕМИТЕ

ОЦЕНКА НА ФИЗИЧЕСКИ МОДЕЛ

- **Физическа имитация на системата:** пълна или частична, пропускане на нерелевантните части на системата, опростявания
- **Примери:** проектиране на автомобили (аеродинамична тръба - wind tunnel, тестове за удари - crash tests), виртуални каюти, оценка на биологични системи, социология (възстановка на ситуации)

ОЦЕНКА НА ФОРМАЛЕН МОДЕЛ

- **Функционално описание:** език за описание на системата, диаграми за преходи на състоянията, мрежи на Петри
- **Модели за масово обслужване:**



КЛАСИФИКАЦИЯ НА МЕТОДИТЕ ЗА ОЦЕНКА НА СИСТЕМИТЕ

АНАЛИТИЧНИ МЕТОДИ

- Точно или приблизително решаване на математически изрази – системи линейни уравнения, диференциални уравнения, и др.
- Критични проблеми: съществуването и стабилността на алгоритмите за решаване, сложността и точността на числените оценки

СИМУЛАЦИЯ

- Изпълнение на релевантните процедури на компютър - обработват се голям брой събития, използват се случайни числа, измерване на проби (samples), статистически оценки
- Критични проблеми: статистическа устойчивост, преходни фази, изчислително време и използвана памет, грешки на имплементацията



КРИТЕРИИ ЗА КЛАСИФИКАЦИЯ НА МЕТОДИТЕ ЗА ОЦЕНКА

- Статистически – функционални
- Динамични – статични
- Дискретни – непрекъснати
- Управлявани от събития (event-driven) –
Управлявани от времето (time-driven)
- Стохастични – детерминистични
- Стационарни – нестационарни
- Последователни – паралелни
- Софтуерно имплементирани – хардуерно
имплементирани

СИМУЛАЦИЯ НА КОМПЮТЪРНИ И КОМУНИКАЦИОННИ МРЕЖИ

- Математическият анализ и симулацията са основните използвани методи при концептуалната фаза (изследване и проектиране) поради финансовите и технически ограничения при създаването на прототипи
- Аналитичните методи показват граничното поведение на системните характеристики или горните и долните граници на изследваните параметри
- Детайлният анализ определя изключително голяма сложност на аналитичните модели

СИМУЛАЦИЯ НА КОМПЮТЪРНИ И КОМУНИКАЦИОННИ МРЕЖИ

- Симулацията е изключително полезна при динамични системи, които трудно подлежат на математически анализ
- Симулацията осигурява управляема среда за изследване, в която могат да бъдат анализирани различни множества параметри и сценарии
- Осигурява мощна и гъвкава методология за анализ и визуализация на поведението и производителността на комуникационни системи и мрежи

СИМУЛАЦИОННИ МОДЕЛИ

- Симулацията се изпълнява върху т.нар. симулационни модели, които трябва да бъдат създадени предварително
- Високото качество на симулационния модел определя качеството на получените резултати от симулацията
- Не съществуват обобщени методи и техники за създаването на адекватни модели, както и унифицирани методологии за симулация на мрежи
- Моделът трябва да отразява достатъчно точно изследвания обект или явление, и в същото време да не съдържа детайли, които няма да се отразят на получените резултати (дозиране на детайлите)



СИМУЛАЦИЯ НА КОМПЮТЪРНИ И КОМУНИКАЦИОННИ МРЕЖИ

- В последните години общността, занимаваща се с проектирането и имплементирането на мрежи, е развила множество подходи за симулация на мрежи, основани на най-добрите практики
- Създаден е широк спектър от средства, поддържащи моделирането, програмирането и изпълнението на симулационен код за оценка на мрежите на всички нива

КОМПЮТЪРНИ СИМУЛАЦИИ

- При компютърните симулации процес или система от реалния свят се “имитират” за определен период от време
- Прилагат се в широк спектър области
- Съществуват различни видове компютърни симулации
 - Симулация по дискретни събития (discrete-event simulation)
 - Непрекъснати симулации (continuous simulations)
 - Симулации по метода Монте Карло
 - Симулация по електронни таблици (spreadsheet simulation)
 - Симулация по следи (trace-driven simulation)



СИМУЛАЦИЯ НА ДИСКРЕТНИ СЪБИТИЯ

DISCRETE-EVENT SIMULATION

- Ключовото свойство на симулацията на дискретни събития е, че състоянието на симулационния модел може да се променя само в дискретни моменти от времето, наричани “събития”
- Широко се използва при симулацията на всички нива на компютърните мрежи
- Важен аспект на компютърните симулации е тяхната повтаряемост – различни проекти могат да бъдат оценявани при едни и същи параметри на средата за целите на сравнителния анализ



СИМУЛАЦИЯ НА ДИСКРЕТНИ СЪБИТИЯ

ТЕРМИНОЛОГИЯ И КОМПОНЕНТИ

- **Обект** (entity, object) – абстракция на специфициран обект, подлежащ на изследване
- Описва се с неговите атрибути – напр., entity “пакет” се описва с атрибутите дължина, адрес на подателя и адрес на дестинацията
- **Система** – дефинира се като множество от обекти и техните релации (взаимоотношения, връзки) – напр., мрежата може да се дефинира като изградена от обектите хостове, рутери и комуникационни връзки
- Всяка система има цел – напр., мрежата има за цел да осигури свързаност от-край-до-край (end-to-end connectivity)

СИМУЛАЦИЯ НА ДИСКРЕТНИ СЪБИТИЯ

ТЕРМИНОЛОГИЯ И КОМПОНЕНТИ

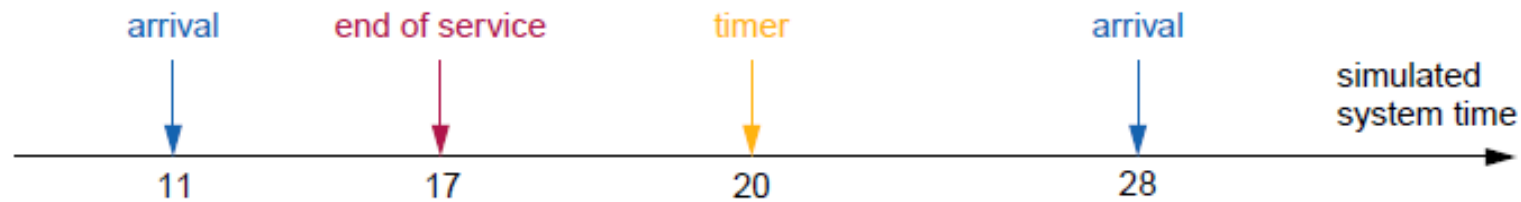
- Всяка система се характеризира в даден момент с нейното състояние, което се дефинира от състоянието на всички нейни обекти
- **Дискретна система** – състоянието на системата (определено от състоянието на всички нейни компоненти) се променя само в дискретни моменти от времето
- Промяната на състоянието на системата се предизвиква (triggered) от възникването на събитие
- Видът на събитието зависи от вида на системата и целта на изследването – напр., изпращането или получаването на пакет

СИМУЛАЦИЯ НА ДИСКРЕТНИ СЪБИТИЯ

СИМУЛАЦИОНЕН МОДЕЛ

- Обикновено системите, които се симулират, са много сложни
- Необходимо е да се изгради модел, на основата на който да се извърши симулацията, и оттам да се оцени производителността
- Симулационният модел представлява абстракция на изследваната система и следователно, обхваща избрани компоненти на системата и избрани взаимоотношения между тях, като селекцията се базира на целта на изследването
- Симулационният модел определя сложността на симулацията и свързаните с нея цена и усилия

ПРИНЦИП НА СИМУЛАЦИЯТА НА ДИСКРЕТНИ СЪБИТИЯ



- Състоянието на системата се променя в дискретни моменти на времето (пристигане на съобщение, край на обслужването, таймаут)
- Промяната в състоянието се представя от събития
- Регистриране на събитията в календар (заедно със маркер за време (timestamp) представящ планираното време за възникването на събитието)
- Последователна обработка на събитията в календара по реда, специфициран от маркерите за време
- Актуализация на симулационното време
- Изпълнение на дейности за осъществяване на преходите на състоянията на системата
- Планиране на повтарящи се събития



ПРИНЦИП НА СИМУЛАЦИЯТА НА ДИСКРЕТНИ СЪБИТИЯ

- Идеята на симулатор, управляван от дискретни събития, е да изпълнява преходи от едно събитие към следващото
- Възникването на събитие може да предизвика промяна в състоянието на системата, както и генерирането на нови, т. нар. *известия за събития (event notices)* в списъка на предстоящите събития (future event list – FEL)
- *Списъкът на предстоящите събития (future event list – FEL)* представлява адекватна и удобна структура данни за управление на всички събития при симулаторите по дискретни събития

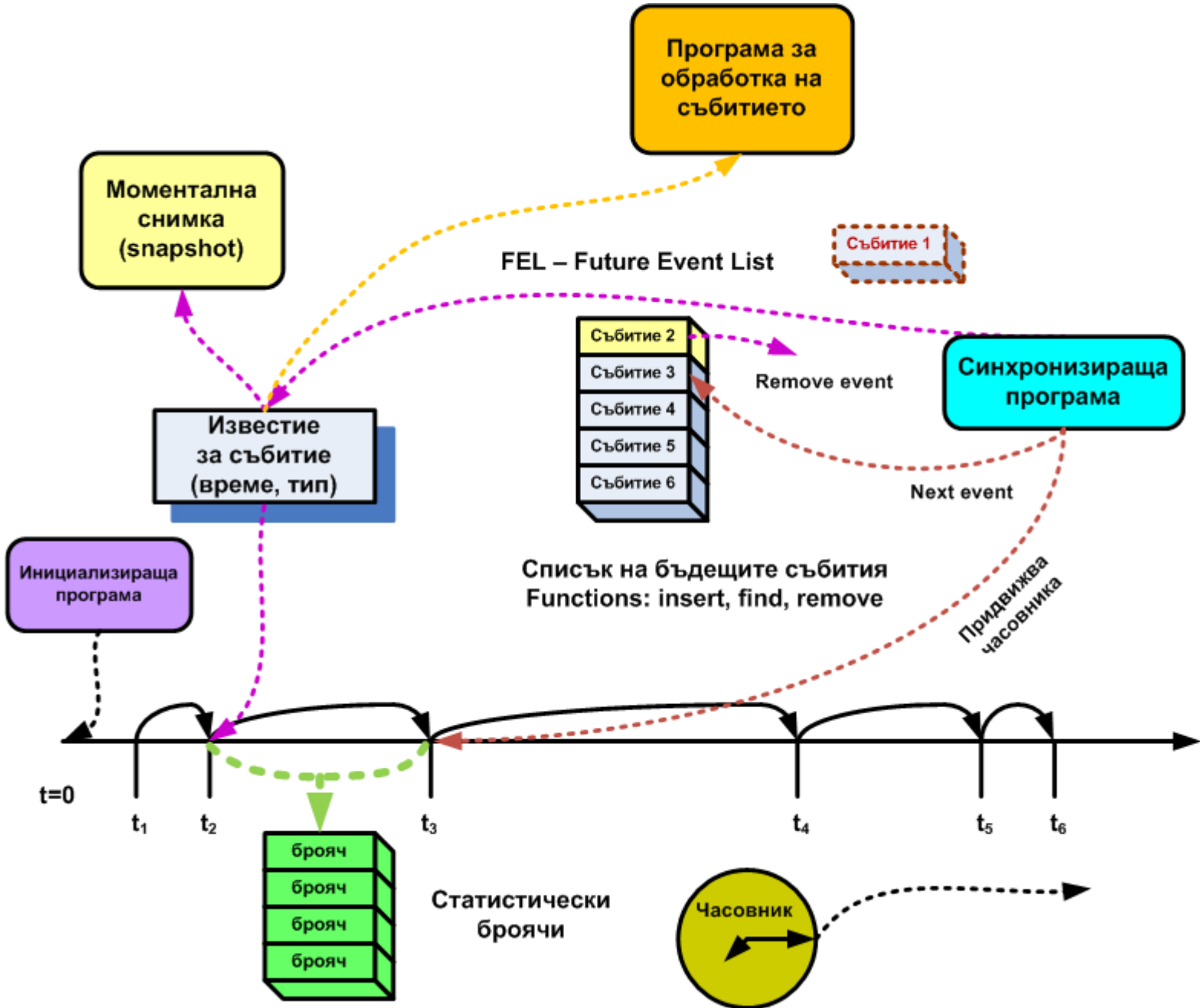
ПРИНЦИП НА СИМУЛАЦИЯТА НА ДИСКРЕТНИ СЪБИТИЯ

- Известието за събитие обхваща поне два типа данни – време на възникване на събитието и тип на събитието *event notice (time, type)*
- Списъкът на предстоящите събития трябва да имплементира ефективни функции за вмъкване, намиране и изтриване на известия за събития
- При възникването на всяко дискретно събитие се създава *моментална снимка (snapshot) на системата* в паметта на компютъра, която съдържа цялата информация, необходима за продължаване на симулацията

КОМПОНЕНТИ НА СИМУЛАТОРИТЕ НА ДИСКРЕТНИ СЪБИТИЯ

- *Състояние на системата* – множество променливи, които го описват
- *Часовник* – индицира текущото време на симулация
- *Списък на предстоящите събития* – структура данни за управление на събитията
- *Статистически броячи* – множество променливи, които съдържат статистическа информация за производителността на системата
- *Инициализираща програма* – инициализира симулационния модел и нулира симулационния часовник
- *Програма за синхронизиране на времето* – извлича следващото събитие от FEL и придвижва часовника към времето за възникване на следващото събитие
- *Програма за обработка на събитията (handler)* – при възникване на събитие се извиква съответната програма за обработка на това събитие – за обработката на всеки тип събитие съществува отделна програма



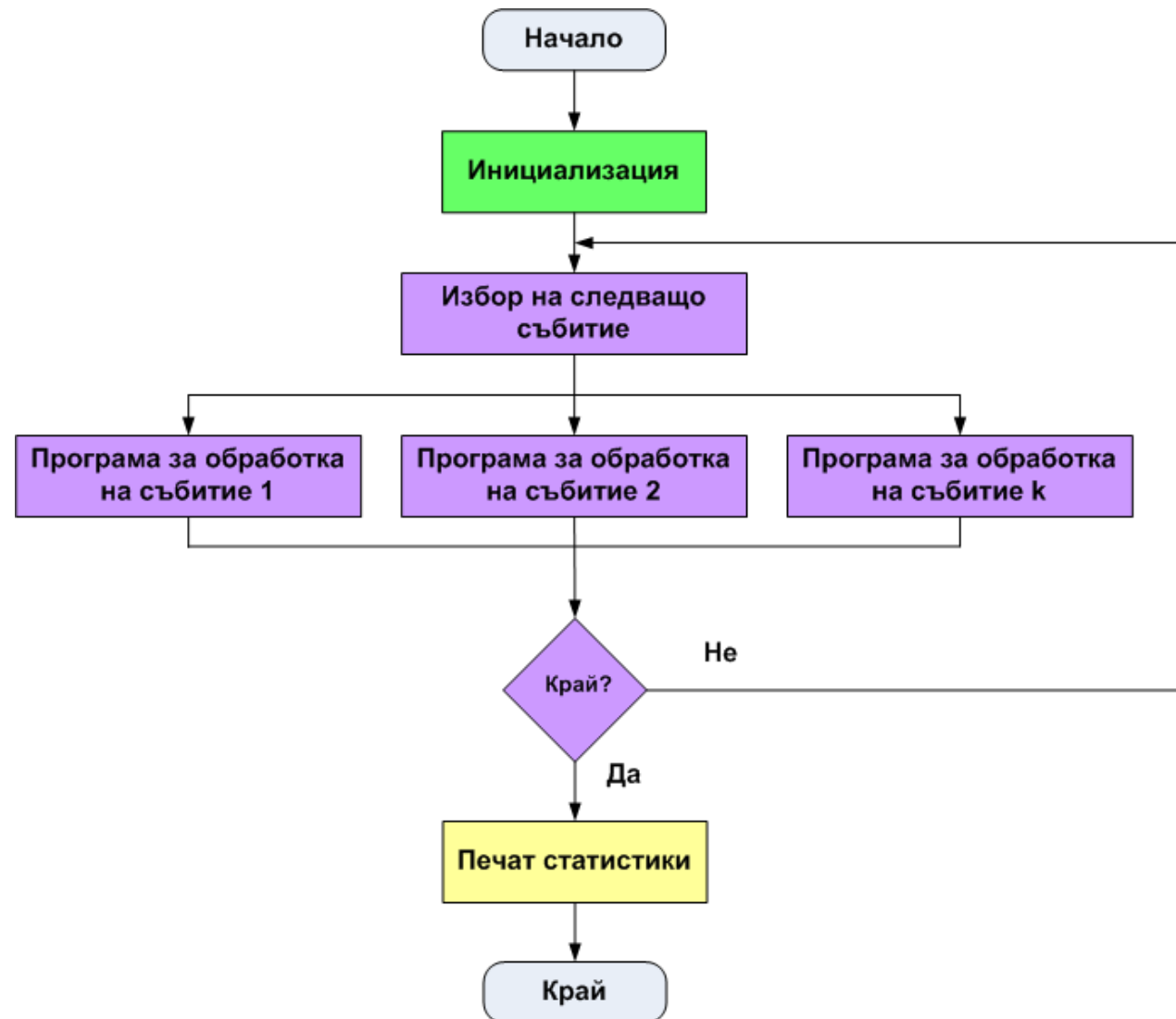


АЛГОРИТЪМ ЗА ПЛАНИРАНЕ НА СЪБИТИЯТА И ПРИДВИЖВАНЕ НА ВРЕМЕТО НА СИМУЛАТОРА

- The Event-scheduling Time-advance Algorithm – ядрото на симулатор на дискретни събития
- По време на симулацията часовникът индицира текущото време на симулация
- Състоянието на системата еволюира във времето
- Списъкът на бъдещите събития съдържа всички известия за събития, подредени по времето за тяхното възникване

$$fel = [t_1, t_2, \dots, t_k] \quad t_1 \leq t_2 \leq \dots \leq t_k$$

THE EVENT-SCHEDULING TIME-ADVANCE ALGORITHM



СТАРТИРАНЕ НА СИМУЛАЦИЯТА

- В началото на симулацията списъкът на бъдещите събития е празен
- По време на инициализацията на симулацията се генерират първоначалните известия за събития и се записват в списъка на бъдещите събития – напр., първият пакет в мрежата

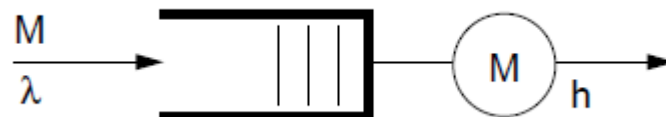


ТЕРМИНИРАНЕ НА СИМУЛАЦИЯТА

- Критерият за терминиране на цикъла за обработката на събитията е от изключителна важност
- Възможни за три подхода за терминиране на симулацията:
 1. Специално събитие за терминиране – напр., закъснение на пакета по-голямо от 500 ms
 2. Предефинирано време на симулацията – напр., 1000 s
 3. До изчерпването на списъка от бъдещи събития

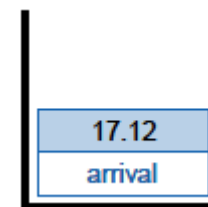
ПРИМЕР – СИМУЛАЦИЯ НА СЪРВЪР

Поток входни
заявки с
експоненциално
разпределение

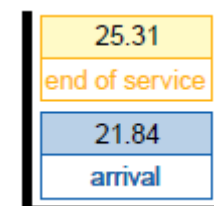


Време за обслужване
с експоненциално
разпределение и
средна стойност h

- Следващо събитие: пристигане на заявка
- Ново системно време: 17.12
- Промяна на състоянието
- Обслужване от сървъра (ако е свободен) или нареждане на опашката
- Актуализация на статистиката
- Повтарящи се събития
 - пристигане на следваща + (евентуално) край на обслужването
 - маркер за време: текущото системно време + случаен интервал според зададеното разпределение
 - регистриране в календара

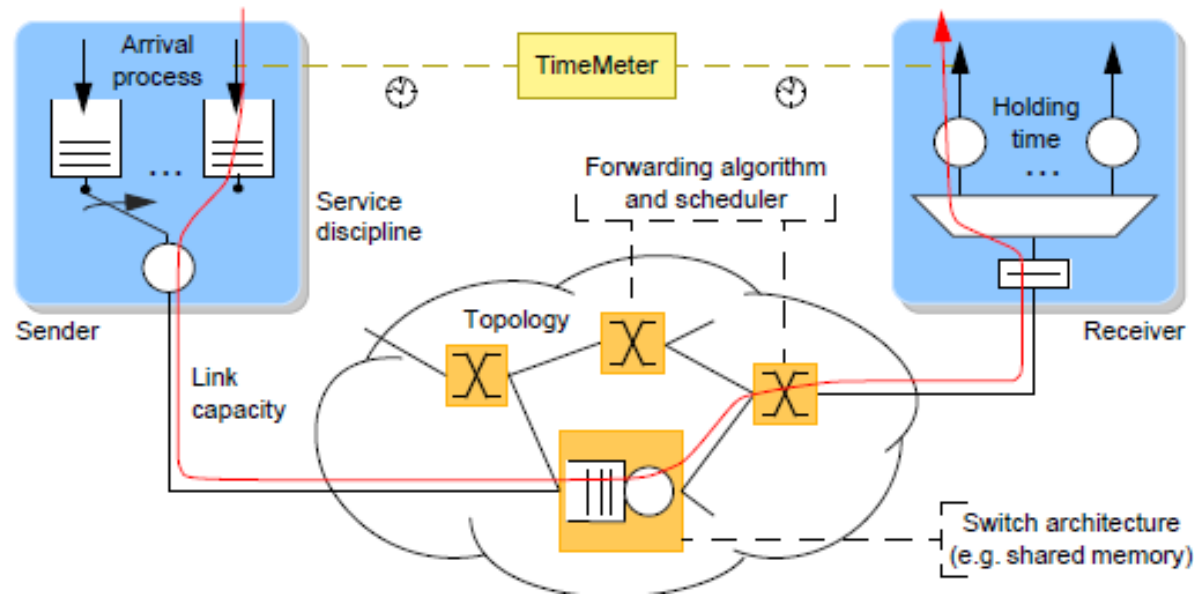


КАЛЕНДАР



КАЛЕНДАР

МОДЕЛИРАНЕ НА ЛОКАЛНА МРЕЖА



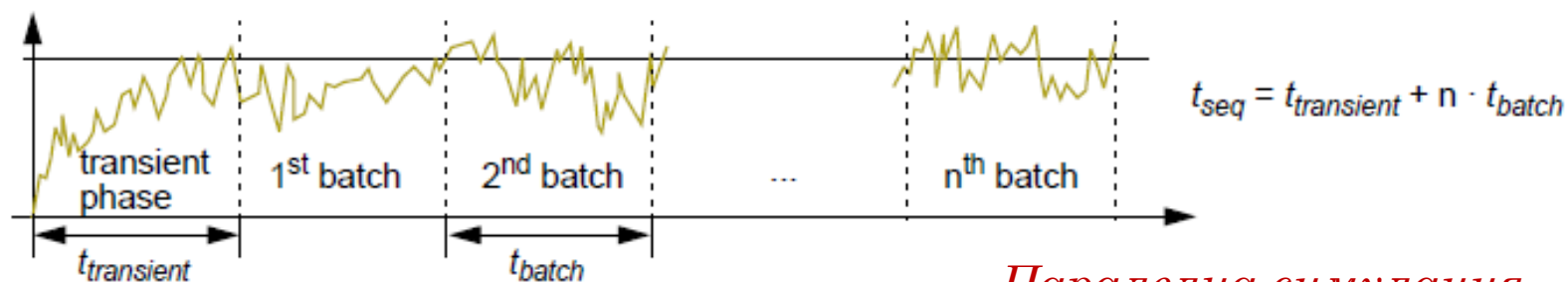
- Оразмеряване на системата (размер на буферите, капацитет на комуникационните връзки)
- Латентност и флуктуации (изисквания по отношение на времето)
- Оценка при различни алгоритми за маршрутизация



ТИПОВЕ СИМУЛАЦИИ

- Класифицират се на *симулации с преходни състояния (transient state)* и *устойчиви (steady state) състояния*
- Изборът на типа на симулацията е важен за анализа на изходните статистики
- *Симулация с преходни (временни, неустойчиви) състояния* – нар. се още *симулация с терминиране (terminating simulation)*, защото се извършва за предефинирано време или се използва терминиращо събитие
- *Симулация с устойчиви състояния* - нар. се още *симулация без терминиране (non-terminating simulation)*
 1. целта е да се изследва дълговременното поведение на системата
 2. началните условия трябва да бъдат зададени така, че да не влияят на изходните резултати
 3. критичен аспект е да се определи необходимото време за симулация

ПОСЛЕДОВАТЕЛНА И ПАРАЛЕЛНА СИМУЛАЦИЯ



Паралелна симулация

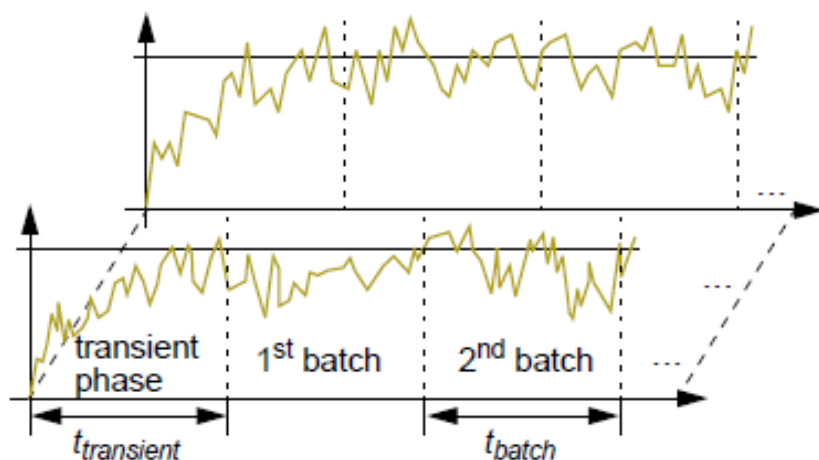
при следните изисквания:

- Голямо симулационно време
- Мощни изчислителни ресурси
- Независими изпълнения на симулациите на различни CPUs и/или ядра
- Времето на паралелната симулация се определя като

$$t_{parallel} = t_{transient} + k \cdot t_{batch}$$

$$\text{при } 1 \leq k \leq n$$

- Недостатък : допълнителни системни разходи поради паралелните преходни фази



МОДЕЛИРАНЕ ЗА КОМПЮТЪРНИ СИМУЛАЦИИ – “4 КРАЙЪГЪЛНИ КАМЪКА”

- Моделът на изследвана система е абстракция – той представя само избрано множество от качества и характеристики на реалната система, абстрахирайки се от голяма част от тях
- Моделът представлява редуцирано представяне на реалната система, базирано на опростявания и допускания
- Моделирането представлява процес на идентифициране и абстракция на релевантните обекти и техните взаимоотношения в изследваната система
- От факта, че всеки модел е абстракция, възникват два основни въпроса:
 1. Какво представлява добрият модел?
 2. Как да бъде създаден добрият модел?

МОДЕЛИРАНЕ ЗА КОМПЮТЪРНИ СИМУЛАЦИИ – “4 КРАЙЪГЪЛНИ КАМЪКА”

- Ако компютърната симулация е избрана като метод за оценка, два нови аспекта трябва да бъдат взети под внимание:
 3. След като бъде избран модел на производителността, той трябва да бъде имплементиран софтуерно
 4. Трябва да бъдат осигурени съответните програмни средства, които да поддържат процедурата за оценяване на производителността

ДОБРИ МОДЕЛИ НА ПРОИЗВОДИТЕЛНОСТ И ДОБРИ СИМУЛАЦИОННИ МОДЕЛИ

- Относно моделите на производителност “Essentially, all models are wrong, but some are useful”
- *Добрите модели на производителност* (за анализ или за симулация) трябва да имат следните характеристики:
 - *Простота* – не означава, че моделът не трябва да отразява детайли или не трябва да взема предвид сложни взаимоотношения, сложността на модела се увеличава само с цел да се повиши прецизността на оценката
 - *Достоверност* – моделът на производителност е достоверен, ако е валидиран и могат да се вземат решения за проектиране
 - *Документиране* – изисква прецизно дефиниране на целта на оценката, задължително се документират опростяванията и допусканията, както и промените (ако са направени такива), изброяват се аспектите, които не са отразени в модела



ДОБРИ МОДЕЛИ НА ПРОИЗВОДИТЕЛНОСТ И ДОБРИ СИМУЛАЦИОННИ МОДЕЛИ

- Освен характеристиките на добър модел на производителност, *добрият симулационен модел* трябва допълнително да се характеризира с:
 - *Ефективност* – трябва да имплементира ефективно модела на производителност, времената за симулация трябва да бъдат приемливи, да осигурява пълна възможност за изследване
 - *Верификация* – проверка на съответствието между модела на производителност и симулационния модел
 - *Качество на кода* – имплементацията трябва да поддържа добър стил на кодиране и да бъде осигурена необходимата документация
 - *Достъпност* – симулационния модел трябва да бъде достъпен, така че други групи да могат да го верифицират и валидират самостоятелно



ДОБРИ ПРАКТИКИ ЗА МОДЕЛИРАНЕ ПРИ КОМПЮТЪРНИТЕ СИМУЛАЦИИ

- Създаването на добър модел изисква опит в моделирането и дълбоко познаване на изследваната система
- Компютърната симулация винаги е “вградена” в рамката на оценяването на производителността, и следователно, важен фактор е да се приложи добра практика за оценка

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ – СТЬПКА 1

1. *Формулиране на проблема и дефиниране на системата/модела*
 - *Прецизно дефиниране на целите на изследването* – компаративен анализ на проекти на система или оптимизиране на параметрите на зададен проект
 - *Изходните резултати не трябва да бъдат прогнозируеми* – обезмисля се симулацията
 - Границите на модела/системата трябва да бъдат определени

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ – СТЬПКА 2

2. *Избор на метрика, фактори и нива*

- Определя начина, по който ще се извършат измерванията
- Факторите представляват параметрите на системата или модела, които се променят при процедурата за оценяване
- Числовите стойности на факторите при оценъчната процедура, се наричат нива

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ – СТЪПКА 3

3. *Събиране на данни и моделиране*

- Преди изграждането на модела на производителност е необходимо да се натрупат знания за изследваната система
- Определяне на релации, на основата на които моделът на производителност може да бъде валидиран в последващ момент
- Трябва да бъдат събрани данни за производителността на изследваната система
- *Създаването на модела на производителност* започва с документиране на допусканията за системата и начина ѝ на функциониране
- Описват се елементите на системата, както и техните качествени и количествени взаимодействия – итеративен процес като постепенно моделът се усложнява
- Създаденият модел на производителност се документира

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ – СТЪПКА 4

4. *Избор на среда за симулация, имплементиране на модела, и верификация*

- След създаването на модела на производителност, той трябва да се имплементира – *решение за работната рамка на симулацията*
- В контекста на мрежите съществуват няколко open-source средства за симулация, които осигуряват специфични моделни библиотеки за различни слоеве от протоколния стек
- Наличието на определени библиотеки като напр., за трафика или за протоколите от ниските слоеве, може да бъде един от определящите фактори за избор
- Други фактори – езика за програмиране, средствата за отстраняване на грешки (debug), вградените функции за статистически анализ, графичния потребителски интерфейс и неговата производителност

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ – СТЬПКА 4

- Различните средства за симулация осигуряват различна поддръжка на модулни и структурирани имплементации на модела на производителност
- *Създаденият модел на производителност се верифицира* – верификацията потвърждава, че симулационния модел е имплементиран правилно
- Съществуват различни методи за верификация

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ – СТЪПКА 5

5. *Валидиране на модела на производителност и анализ на факторите, определящи производителността (sensitivity analysis)*
- Валидирането потвърждава, че моделът на производителност коректно представя изследваната система по отношение на избраната метрика, фактори и слоеве.
 - Валидирането може да се осъществи като се сравнят изходните данни от модела с изхода на реалната система
 - В повечето случаи реалната система е недостъпна – резултатите се сравняват с тези, получени от математически анализ, базиран на същия модел на производителност или публикувани аналогични данни от математически анализ или симулации на други групи
 - След валидирането на модела се прави анализ с цел да се определят факторите, които влияят върху системната производителност (*sensitivity analysis*)



МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ – СТЪПКА 6

6. *Провеждане на експериментите, анализ и представяне на резултатите*
- Оценката на производителността се осъществява на основата на верифициран и валидиран симулационен модел
 - Изпълняват се симулационните експерименти, данните се събират, анализират и накрая се представят графично

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ

- Методът за оценка предопределя итеративен процес – грешки при отделните стъпки, неочаквано поведение на модела, или проблеми със сложността
- Простотата на модела се обуславя от една страна, от стриктната ориентация на модела към целта на изследването
- *Ключова парадигма: моделът зависи от целта на изследването*, а не обратното
- От друга страна, *моделирането трябва да се прави итеративно* – всяка стъпка на итерация задължително трябва да включва верификация и валидиране, след което се прави sensitivity analysis за фактори, кандидати за повишаване на акуратността на модела
- При добавянето на нови елементи този анализ дава количествена оценка на влиянието на новите елементи върху системната производителност

МЕТОДОЛОГИЯ ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА ОСНОВАТА НА СИМУЛАЦИИ

- Сложността на модела са повишава, само ако това усложняване допринася за постигането на целта на изследването
- Достоверността на модела се поддържа от методологията поради факта, че симулационните модели задължително се верифицират и валидират
- В идеалния случай, валидирането се извършва като се използват външни данни или различни техники за оценяване (анализ или експерименти)
- Методологията осигурява необходимата и достатъчна документация
- В явен вид методологията не поддържа ефективността и качеството на кода на симулационния модел – зависи от опитността на програмиста





Методология за изследване и оценка на производителността на основата на симулации

ПОПУЛЯРНА ПРАКТИКА ЗА МОДЕЛИРАНЕ

- Разликата се състои в това, че имплементираният симулационен модел представлява по същество модел на производителността т.е. изследването на производителността не се базира на релевантен модел на производителността, който впоследствие се имплементира
- В случая не е необходима верификация на модела, а само валидиране
- Тази практика спестява време
- Валидирането е по-трудно поради несъответствия на модела с поведението на реалната система, обикновено дължащи се на грешки при моделирането
- Сложността на съвременните симулационни модели обуславя тенденцията да не се изгражда в явен вид модел на производителността

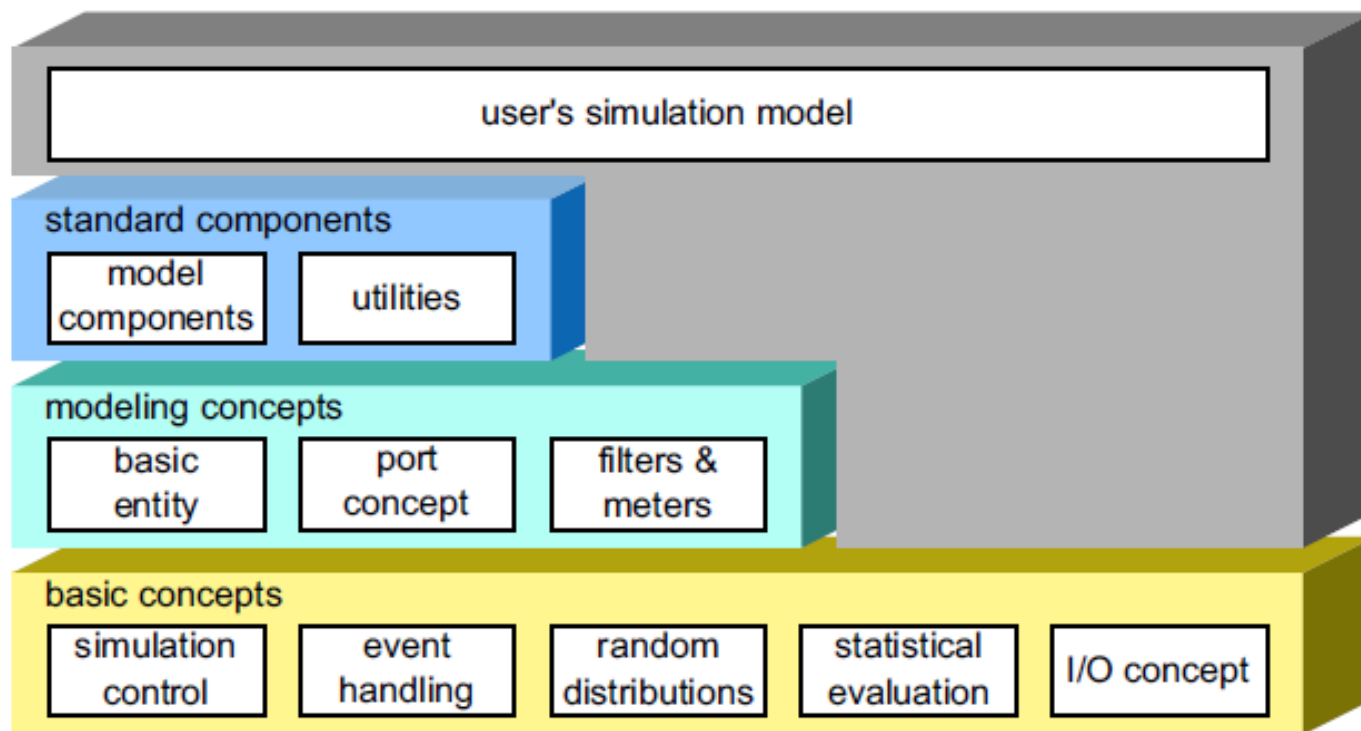
ПОПУЛЯРНА ПРАКТИКА ЗА МОДЕЛИРАНЕ

- В много случаи, особено за мрежи във фазата на проектиране, не е възможно да се валидира модела, тъй като мрежата още не съществува
- Използване на съществуващи симулационни модели – опасност от необосновани допускания и функционални зависимости, които са потенциални източници на грешки при оценяването на производителността
- Симулационните модели не са придружени с модели на производителност и следователно, трябва да се анализира source кода за установяване на реалния обхват на симулационния модел

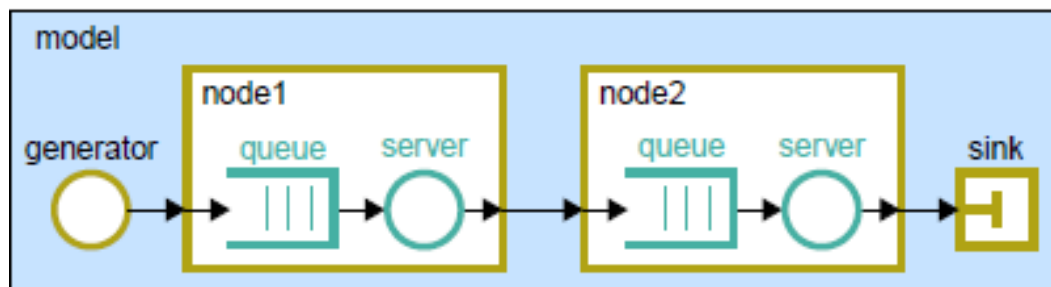
МРЕЖОВИ СИМУЛАТОРИ, ЕМУЛАТОРИ И ИНТЕГРАЦИОННИ РАБОТНИ РАМКИ

- **IKR Simulation library (SimLib)** - Institute of Communication Networks and Computer Engineering (IKR)

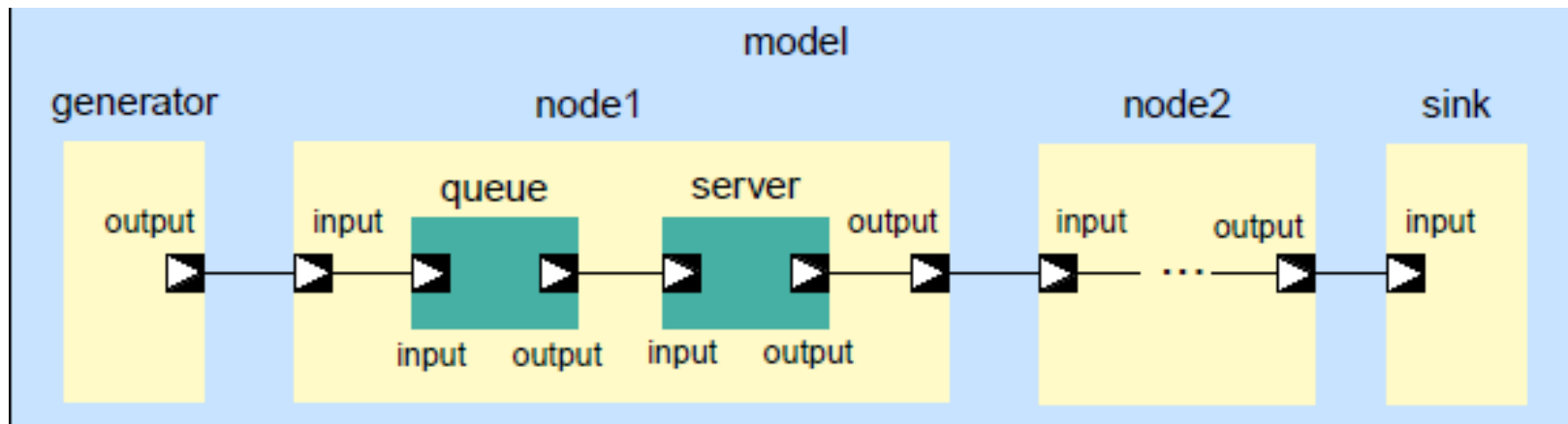
<http://www.ikr.uni-stuttgart.de/INDSimLib/>



IKR SIMULATION LIBRARY (SIMLIB)



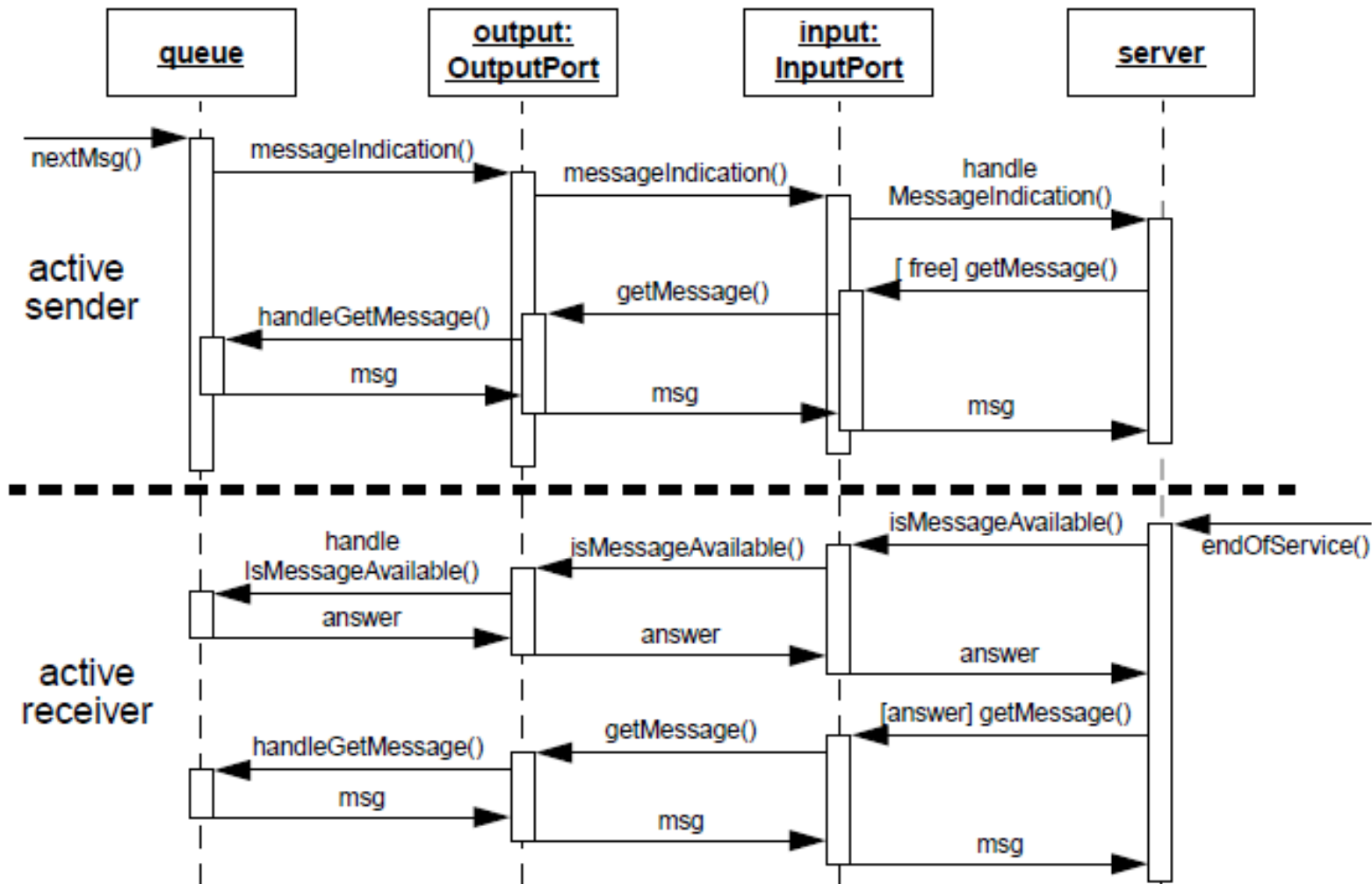
ИЕРАРХИЯ НА ОБЕКТИТЕ



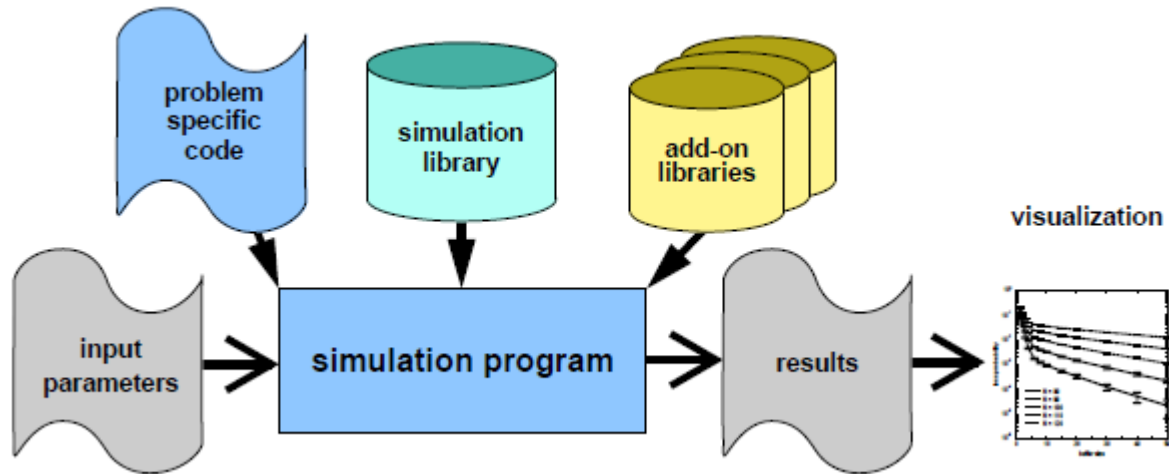
ОБМЕН НА СЪОБЩЕНИЯ ПРЕЗ ПОРТОВЕ

IKR SIMULATION LIBRARY (SIMLIB)

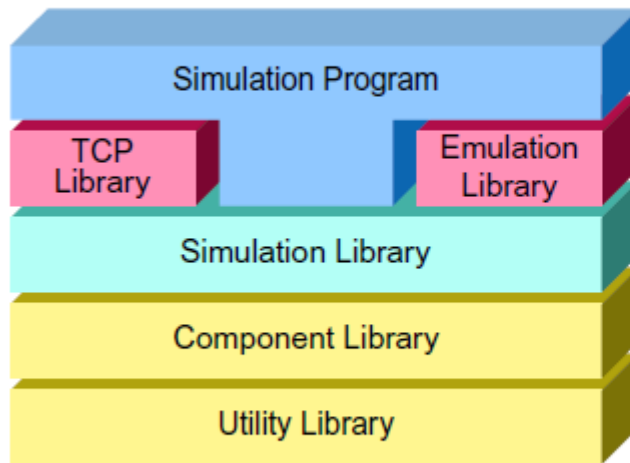
ОБМЕН НА СЪОБЩЕНИЯ



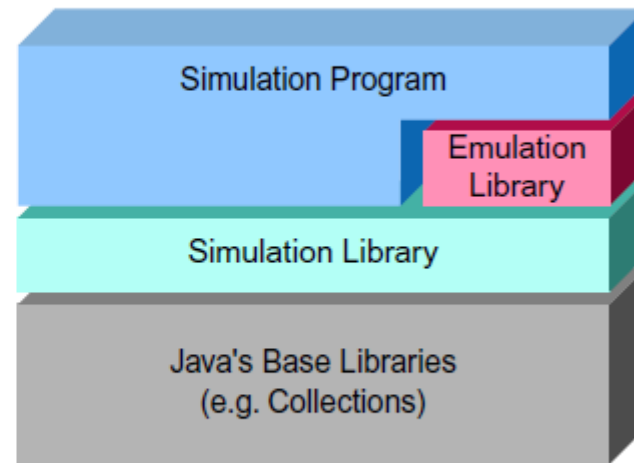
IKR SIMULATION LIBRARY (SIMLIB)



C++ Edition

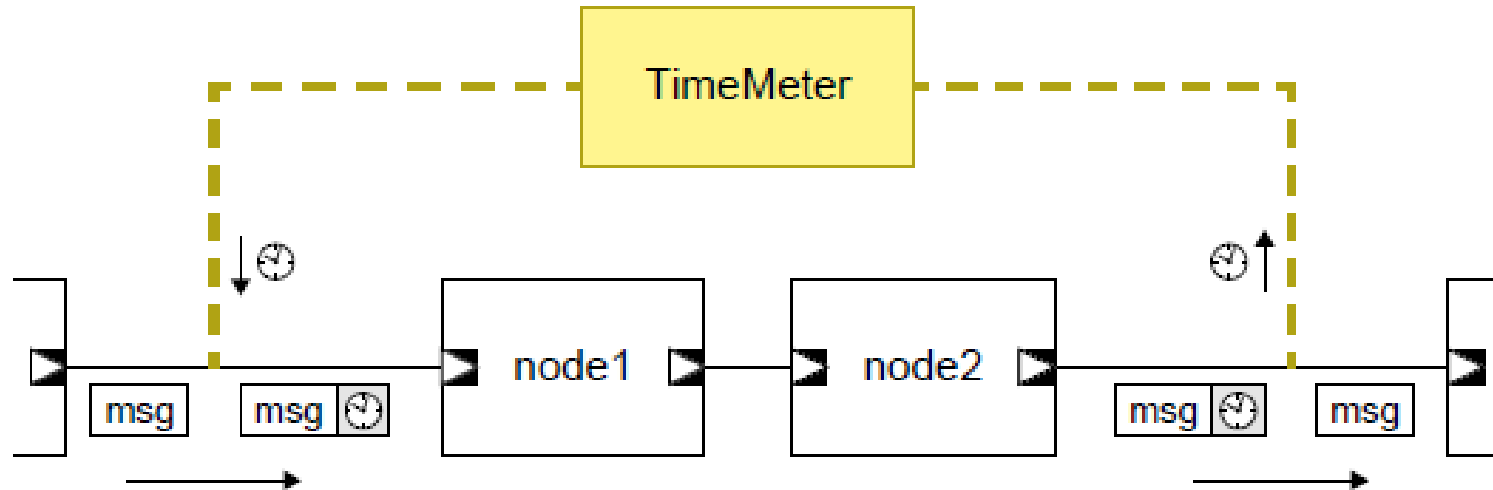


Java Edition



libraries with more than 400 classes and more than 70,000 lines of codes

IKR SIMULATION LIBRARY - ИЗМЕРВАНИЯ



- Измерването се осъществява при портовете посредством филтри на съобщенията
- Използва се за:
 - преброяване на съобщенията/измерване на скорост
 - измерване на времето за трансфер
- Вътрешна оценка на измерените стойности с използване на статистически класове

МРЕЖОВИ СИМУЛАТОРИ, ЕМУЛАТОРИ И ИНТЕГРАЦИОННИ РАБОТНИ РАМКИ

- **openWNS (Open Source Wireless Network Simulator)**

Симулационна платформа за безжични и многоклетъчни (multi-cellular) мобилни комуникационни системи (**Aachen University**)

- **ns-2 - NS (Network Simulator)** - симулатор по дискретни събития. Поддържа симулация на TCP, маршрутизация, multicast протоколи по жични и безжични (локални и сателитни) мрежи.

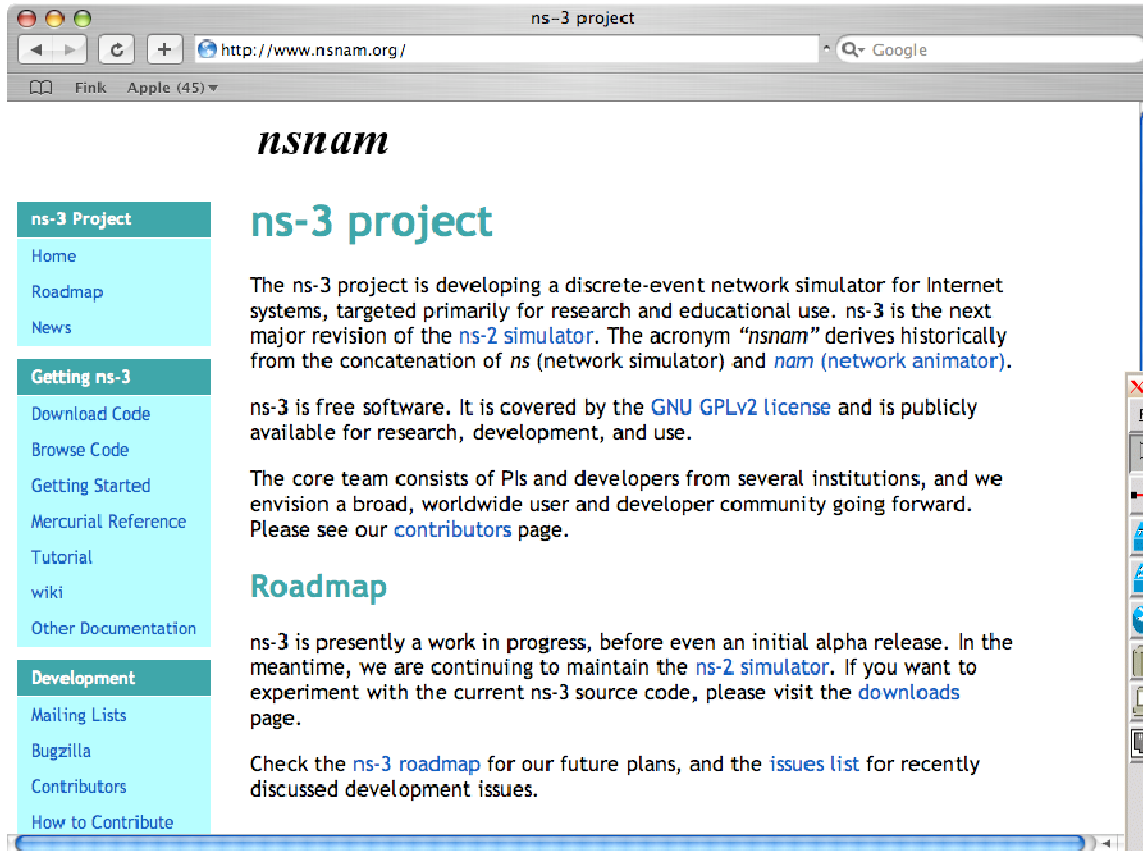
Information Sciences Institute – USC (University of Southern California)

<http://www.isi.edu/nsnam/ns/ns-build.html>

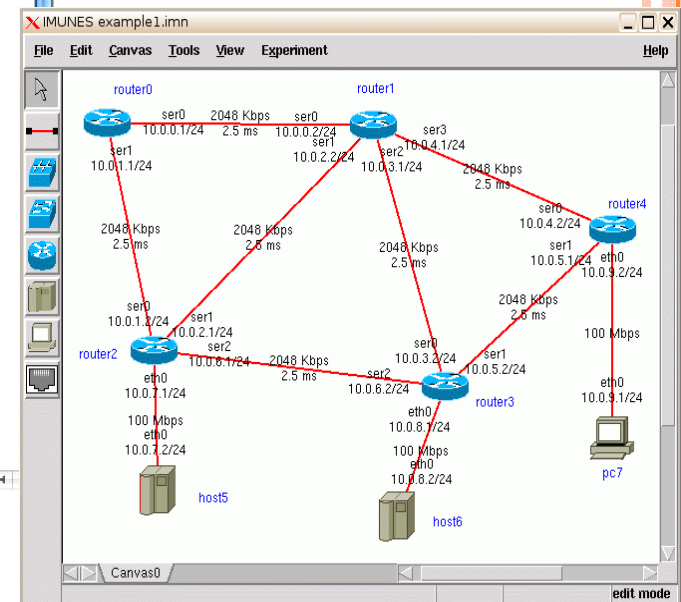
<http://www.cse.msu.edu/~wangbo1/ns2/>



NS-3 NETWORK SIMULATOR (C++, DISCRETE-EVENT)



The screenshot shows a web browser window with the URL <http://www.nsnam.org/>. The page title is "ns-3 project". The main heading is "nsnam" in a stylized font. Below it, the "ns-3 project" section describes the simulator as a discrete-event network simulator for Internet systems, targeted for research and educational use. It mentions that ns-3 is the next major revision of the ns-2 simulator and that the acronym "nsnam" derives from "ns" (network simulator) and "nam" (network animator). The text states that ns-3 is free software under the GNU GPLv2 license and is publicly available. It also notes that the core team consists of PIs and developers from several institutions. A "Roadmap" section indicates that ns-3 is currently a work in progress and that the ns-2 simulator is still maintained. A sidebar on the left contains navigation links for Home, Roadmap, News, Getting ns-3 (Download Code, Browse Code, Getting Started, Mercurial Reference, Tutorial, wiki, Other Documentation), and Development (Mailing Lists, Bugzilla, Contributors, How to Contribute).





DSIM: A DISTRIBUTED OPTIMISTIC PARALLEL DISCRETE EVENT SIMULATOR

- DSIM е предназначен за изпълнение на компютърни клъстери
- Center for Pervasive Computing and Networking, Rensselaer Polytechnic Institute, NY
- <http://www.rpi.edu/cpcn/>
- <http://www.cs.rpi.edu/~cheng3/dsim/>
- Агрегатна изчислителна скорост 218 милиона събития за секунда за 1033 процесора (симулации в Pittsburgh Supercomputing Center), ускорение 296
- **Genesis: Many Processors Sharing the Work**



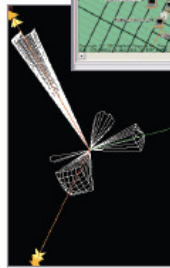
OPNET

o http://www.opnet.com/solutions/network_rd/mode_ler.html#



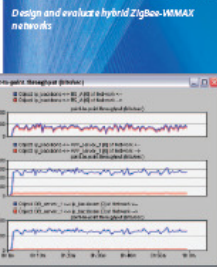
Design and simulation of advanced network scenarios. Studies include, for example, designing next-generation applications and subscriber behavior.

Design and simulation of advanced network scenarios. Studies include, for example, designing next-generation applications and subscriber behavior.



View antenna lobes in three dimensions

Accelerates numerous features to accelerate simulation: a 64-bit kernel, general parallel simulation, event parallel discrete event simulation kernel, and processor machines to accelerate simulation runtime. Can distribute a series of simulations to multiple processors.

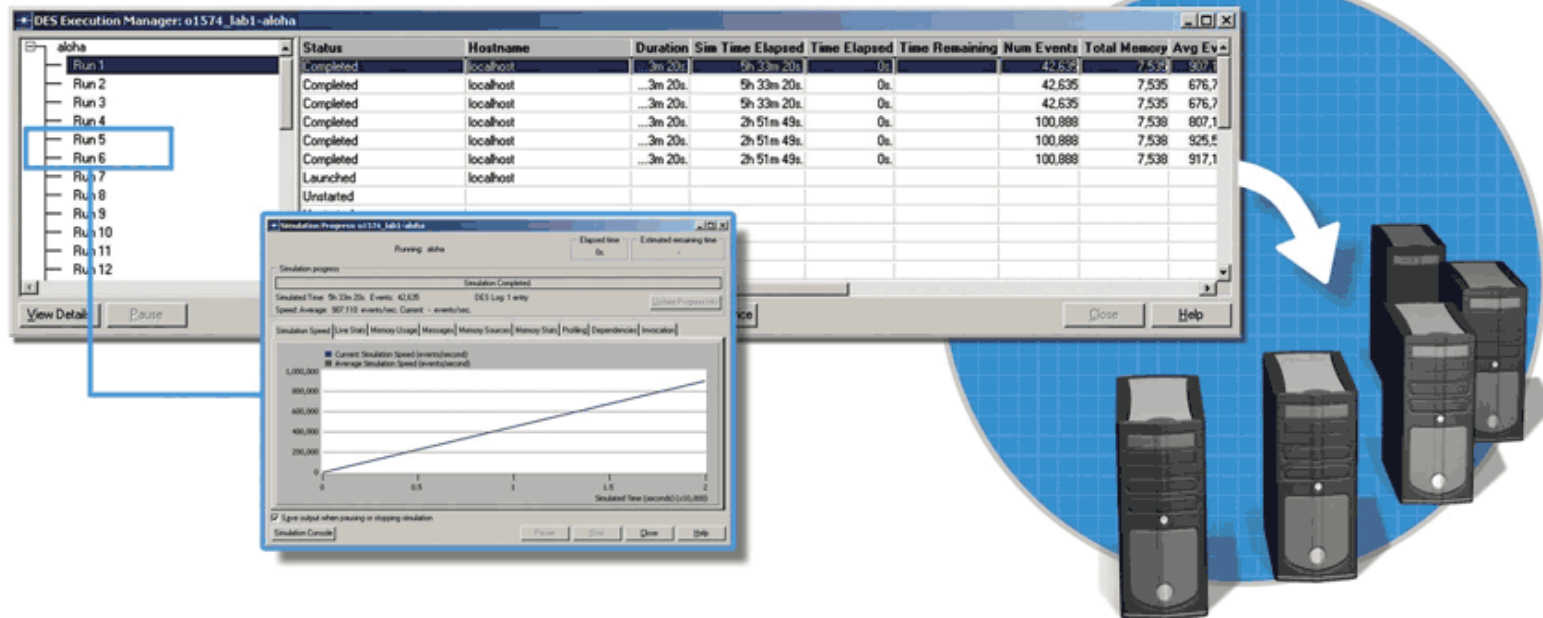
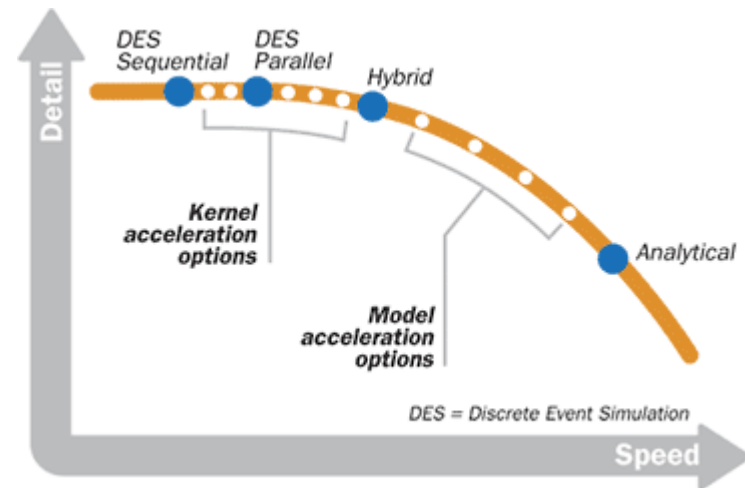


Visualize simulation results in a rich, integrated environment

Open Model Source Code

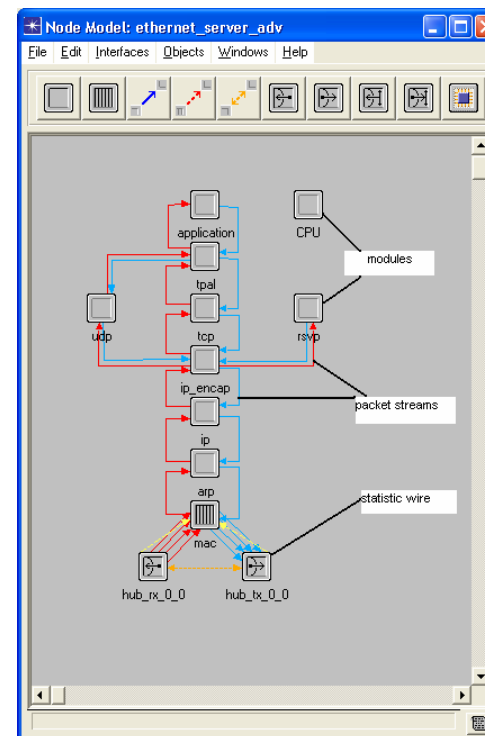
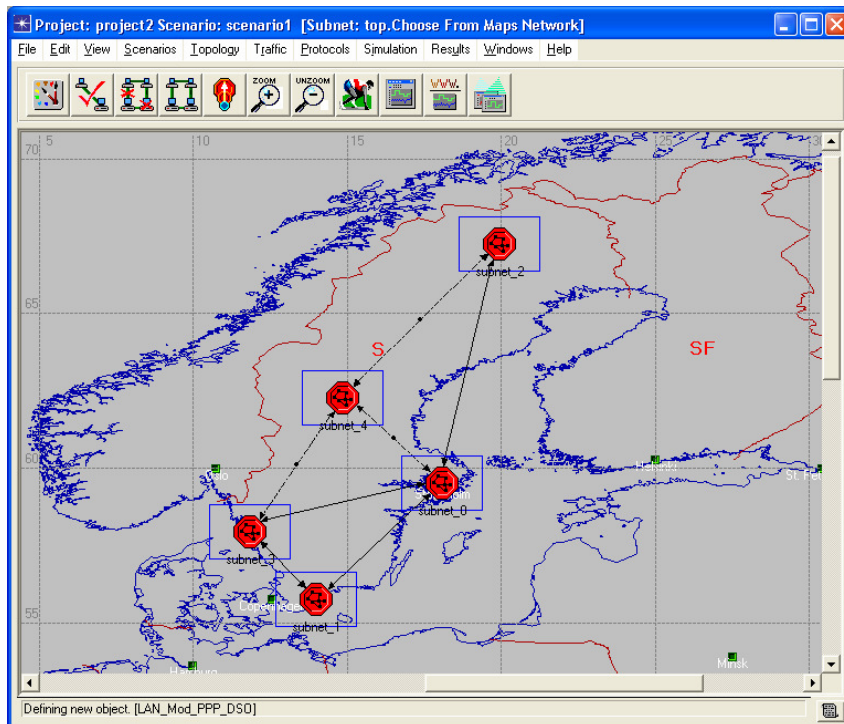
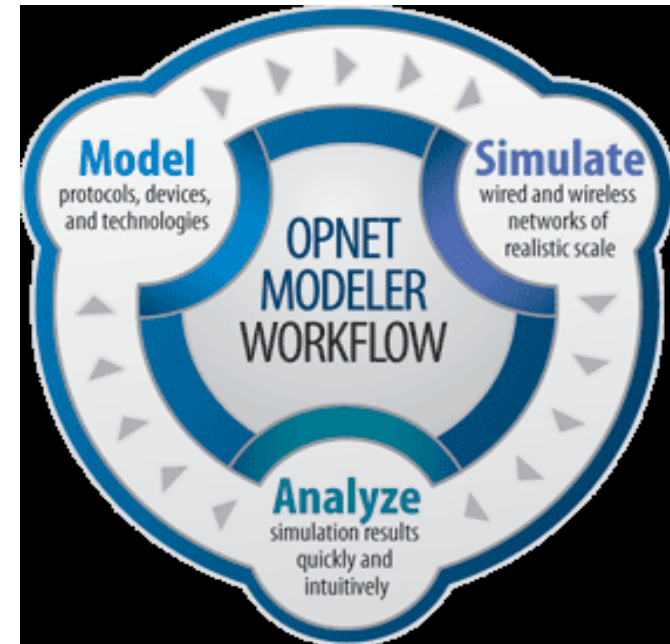
OPNET

- A 64-bit parallel discrete event simulation kernel
- General parallel simulation
- Grid computing support



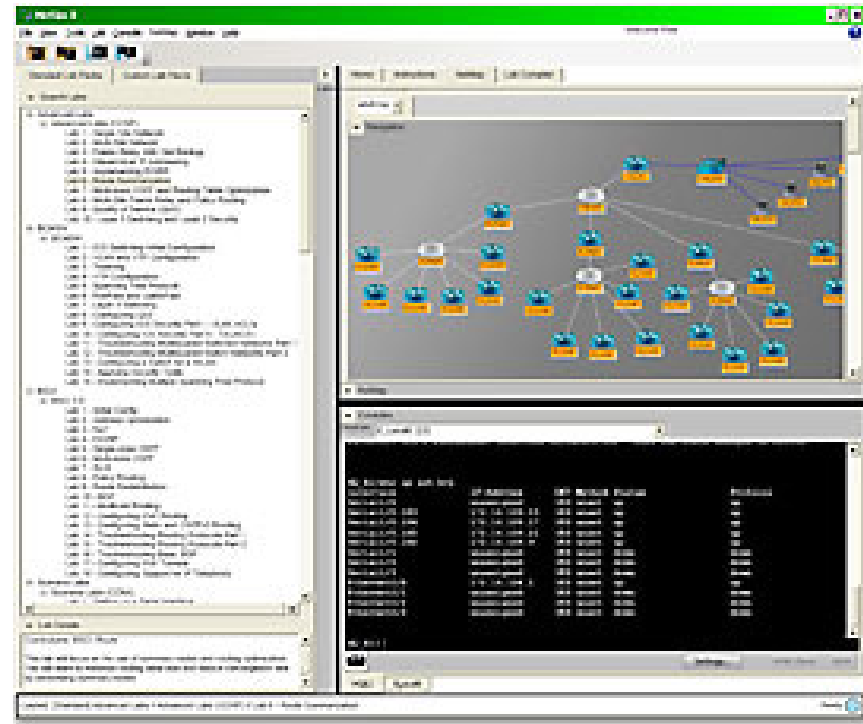
OPNET OPNET TECHNOLOGIES, INC.

- <http://www.opnet.com/>
- OPNET modeler
- Discrete event simulation library



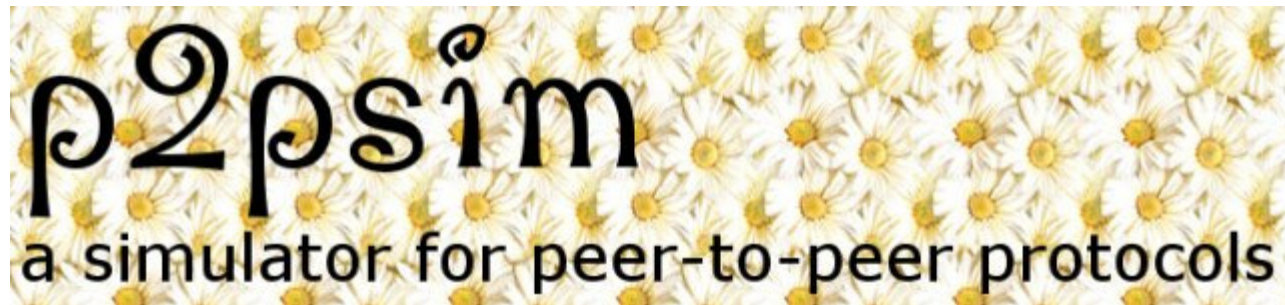
NETSIM (NETWORK SIMULATOR & ROUTER SIMULATOR) CISCO

- Мрежовията симулатор NetSim на Boson е предназначен за симулация на мрежов хардуер и софтуер на Cisco Systems
- За създаването на индивидуалните пакети NetSim използва Boson's Network Simulator, софтуерните технологии на Router Simulator® и EROUTER®, заедно с Boson Virtual Packet Technology® engine
- Тези пакети се маршрутизират и комутират през симулираната мрежа, осигурявайки възможност на NetSim да изгражда съответните виртуални таблици за маршрутизация и да симулира динамичното поведение на мрежата.





P2PSIM



- Massachusetts Institute of Technology
- <http://pdos.csail.mit.edu/p2psim/index.html>
- Част от проекта IRIS: Infrastructure for Resilient Internet Systems <http://iris.csail.mit.edu/>
- p2psim е безплатен, многонишков, симулатор по дискретни събития – използва се за оценка, изследване, и проучване на peer-to-peer (p2p) протоколи
- p2psim се изпълнява в няколко UNIX-like операционни системи
- **Цели:**
 - Лесно разбиране на source code на peer-to-peer протоколите, сравнителен анализ на различни протоколи, и получаване на добра производителност
 - p2psim максимизира степента на паралелизмите за повишаване на производителността, минимизирайки необходимостта от синхронизация, и избягване на мъртви блокирания.

SIMSCRIPT

- <http://www.simscrip.com/>
- SIMSCRIPT III
- Обектно-ориентиран, модулен, интегриран инструмент за развитие на софтуер
- SIMSCRIPT е предназначен за изграждане на симулационни модели in decision support systems за широк спектър от приложения :
 - Военни приложения
 - Комуникационни мрежи, анализ на производителността и оптимизация
 - Транспорт и производство
 - Управление на инвентара и логистично планиране
 - Финансови пазари
- SIMSCRIPT III е също универсален О-О език с добри графични интерфейси и интерфейси за база данни и може да се използва вместо С++ или Java



ДРУГИ СИМУЛАТОРИ

- NCTUns
- QualNet
- SSFNet
- Yans
- GTNetS
- GloMoSim
- OSA
- JiST/SWANS
- Cnet
- Traffic
- Shunra VE
- Extend
- INES
- J-Sim
- HEGONS
- Narses
- 3LS
- NeuroGrid
- PeerSim
- ONE

